PYRAMID Technical Standard Version 1.0

Ministry
of Defence

This document sets out a generic approach to the implementation of the PYRAMID Reference Architecture. The PYRAMID Reference Architecture has not been created for any specific system. It is the user's responsibility to ensure that any article created using this document meets any required operational, functional and safety needs. The Author accepts no liabilities for any damages arising due to a failure of the user to verify the safety of any product produced using this document, nor for any damages caused by the user failing to meet any technical specification.

For further information regarding how you can exploit PYRAMID on your project, provide feedback, or have a technical query that you would like answering, please contact the PYRAMID Team using the following email address: PYRAMID@mod.gov.uk

This publication is available at https://www.gov.uk/government/publications/pyramid/pyramid-programme

CHANGE HISTORY

Date	Version	Description of Changes
February 2025	1.0	First Issue

List of Effective Pages

1513 pages in total

TABLE OF CONTENTS

TERMS AND ABBREVIATIONS USED IN THIS DOCUMENT

Definitions of terms, the meaning of acronyms and the meaning of abbreviations used in this document can be found in Appendix A: Glossary.

References to content defined in the PYRAMID Technical Standard Guidance document, Ref. [2], are shown as (non-hyperlinked) green text.

REFERENCES

The reference numbers are consistent across the PYRAMID Technical Standard and PYRAMID Technical Standard Guidance documents. Only the subset of references used in this document are listed in the reference list.

PYRAMID Document References:

Reference Author/Organisation, Date, Title, Document Number & Issue

[2] Ministry of Defence, (February 2025). PYRAMID Technical Standard Guidance, PYD/TechStanGuide/V1.0.

[3] Ministry of Defence, (February 2025). PYRAMID Model, PYD/TechStanModel/V1.0.

[4] Ministry of Defence, (September 2023). PYRAMID Exploiter's Pack Version 4.1, RCO_FUT_23_004.

Note: PYRAMID documentation is available under the Open Government Licence v3.0.

Other References:

Reference Author/Organisation, Date, Title, Document Number & Issue

[11] Ministry of Defence, (March 2019). Design and Airworthiness Requirements for Service Aircraft, Defence Standard 00-970, Issue 21.

[13] International Organization for Standardization, (2003-2015). Condition monitoring and diagnostics of machines, ISO 13374.

[14] Airlines Electronic Engineering Committee, (2016). Cockpit Display System Interfaces to User Systems, ARINC Specification 661-6.

[16] European Parliament and Council of the European Union (2016). EU General Data Protection Regulation, Regulation (EU) 2016/679.

[27] SAE International / EUROCAE, (2010). Guidelines for Development of Civil Aircraft and Systems, ARP-4754A.

[30] ISO, (2018). Information technology - Security techniques - Information security management systems - Overview and vocabulary, BS EN ISO/IEC 27000:2018.

[31] M. Endsley, (1988). Design & Evaluation for Situation Awareness, Proceedings of the Human Factors Society 32rd Annual /Meeting (pp. 97-110), Santa Monica, CA: Human Factors Society.

[32] Ministry of Defence, (February 2017). Safety Management Requirements for Defence Systems, Defence Standard 00-56 Part 2, Issue 5.

[47] United States Department of Defense, (2024). DoD Interface Standard Tactical Data Link (TDL) 16 Message Standard, MIL-STD-6016H.

[48] NATO, (2017). STANAG 4586 Standard Interfaces of UA Control System (UCS) for NATO UA Interoperability, NSO/0471(2017)JCGUAS/4586.

[49] Object Management Group, (2015). OMG Data Distribution Service (DDS). [Online]. Available: https://www.omg.org/spec/DDS/1.4. [Accessed 2024].

1 Introduction

1.1 PYRAMID and PYRAMID Reference Architecture

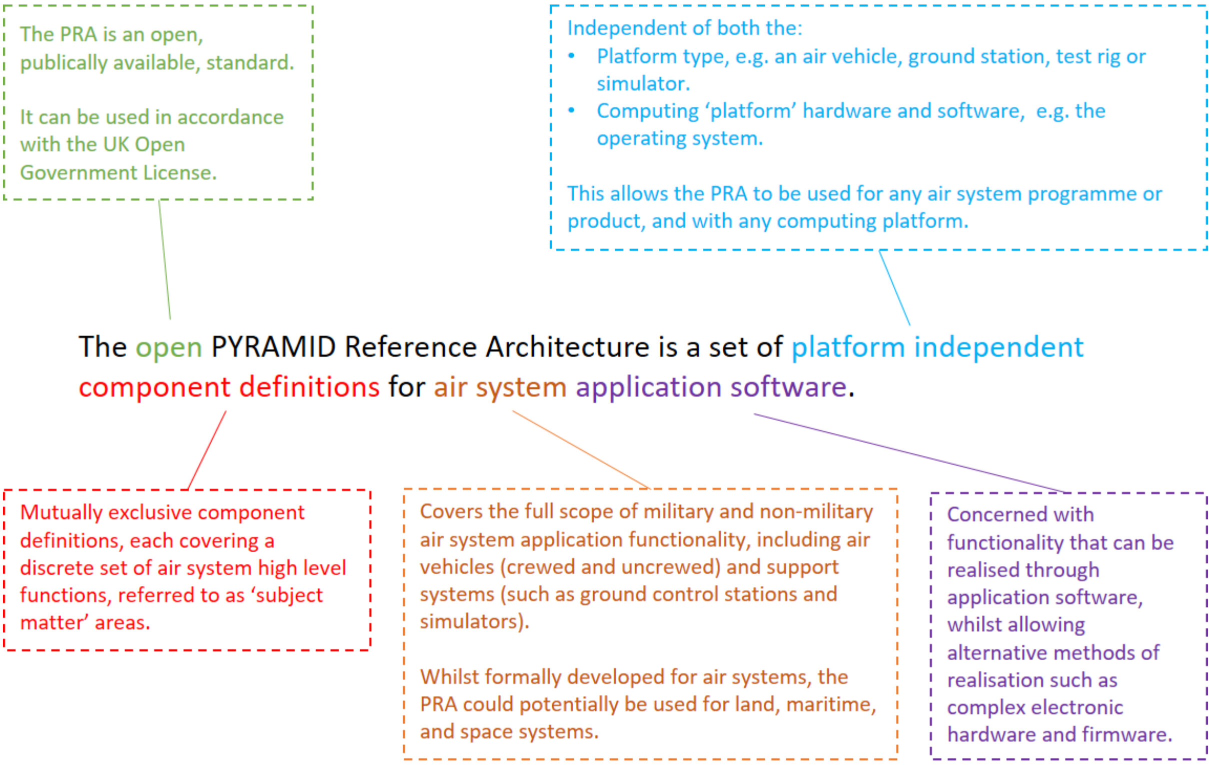
Military aircraft effectiveness is critically dependent on software, especially mission systems software, and fundamental to this effectiveness is the ability to provide new capability where and when it is required. Further to this, effective partnering, capability exchange, and interoperability between allies is essential for operational success.

Traditional software design has been such that relatively small changes can have wide reaching consequences across the aircraft, and the scope for reuse across air platforms (including support systems) and programmes has been limited. This problem has become even more significant with the rapid growth in the complexity of military air system software to meet capability needs. In response, the PYRAMID programme was established to enable technology advantage though systematic software reuse and rapid adaptability.

Modularity and open architectures have been identified as key enablers, but their consistent application across air platforms, and ensuring compatibility with other standards, is essential if the benefits are to be fully realised. In response, a number of open architecture standards have been developed to address areas such as hardware design, data architectures, and software architectures including middleware; but a gap was identified for application software.

This PYRAMID Technical Standard has been developed to provide a consistent approach to modularising air system application software though the PYRAMID Reference Architecture (PRA), whilst ensuring that fundamental requirements, including airworthiness certification and security accreditation, can also be achieved.

An accompanying document, the PYRAMID Technical Standard Guidance document, Ref. [2], has also been produced to provide guidance and supporting information to aid understanding and application of the PYRAMID Technical Standard, enabling the development of PYRAMID compliant systems.



The PRA is an open, publically available, standard.

It can be used in accordance with the UK Open Government License.

Independent of both the:

- Platform type, e.g. an air vehicle, ground station, test rig or simulator.
- Computing ‘platform’ hardware and software, e.g. the operating system.

This allows the PRA to be used for any air system programme or product, and with any computing platform.

The open PYRAMID Reference Architecture is a set of platform independent component definitions for air system application software.

Mutually exclusive component definitions, each covering a discrete set of air system high level functions, referred to as ‘subject matter’ areas.

Covers the full scope of military and non-military air system application functionality, including air vehicles (crewed and uncrewed) and support systems (such as ground control stations and simulators).

Whilst formally developed for air systems, the PRA could potentially be used for land, maritime, and space systems.

Concerned with functionality that can be realised through application software, whilst allowing alternative methods of realisation such as complex electronic hardware and firmware.

Figure 1: What is the PYRAMID Reference Architecture?

The PRA defines an open modular architecture for the software aspects of air system functionality. At its core is the decomposition of these software aspects into discrete areas of functionality called PRA components. The PRA component definitions provide the basis from which application software components, with well-defined boundaries, can be developed and integrated into an air system, whether that be an air vehicle or supporting systems. The cohesive and loosely coupled nature of the PRA component decomposition is an enabler to rapid adaptability and reuse across different programmes, air systems, and computing hardware.

The development of the PRA has followed a set of design principles that support the following key goals, with the associated benefits to Exploiting Programmes:

- Exploitability - application across different programmes, air systems, and computing platforms.
- Scalability - the ability to use varying numbers of components, and component variants, to produce air systems and subsystems.
- Utility across a range of mission requirements - the ability to create an air system for various mission scenarios and organisational structures.
- Configurability - the ability to change component behaviour to support the needs of different air systems or missions/operations.
- Flight Certification - a structure that supports the process of certification (and re-certification after system change) in a structured and straightforward manner.
- Security Accreditation - a structure that supports the process of accreditation in a structured and straightforward manner.

- Resilient to Obsolescence - the ability to port system software on to different computing hardware and operating systems with minimal rework.
- Potential for Future Growth - the flexibility for developed systems to adapt to change.
- Supportability - the ability to create reliable and maintainable systems.

1.2 Scope and Purpose

The purpose of this document is to define:

- The PYRAMID Reference Architecture.
- The rules for achieving compliance with the PYRAMID Technical Standard.

This document supersedes earlier versions of the PRA, issued as part of the PYRAMID Exploiter’s Pack, Ref. [4].

2 Document Structure

Figure 2: PYRAMID Technical Standard Document Structure summarises the structure and content of this document, and highlights the content that is generated from a maintained set of UML models, PYRAMID Model, Ref. [3].

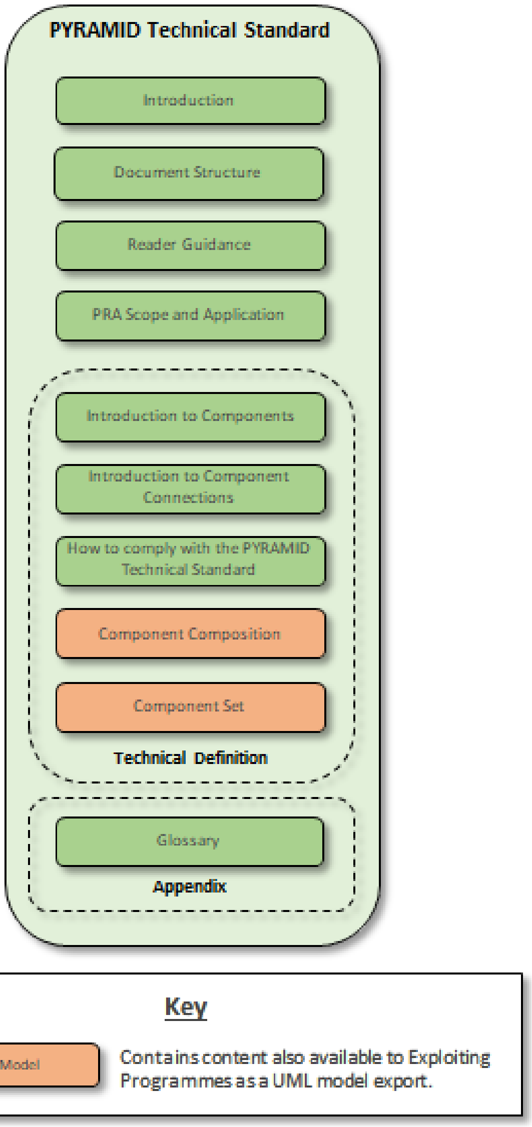

PYRAMID Technical Standard
Introduction
Document Structure
Reader Guidance
PRA Scope and Application
Introduction to Components
Introduction to Component Connections
How to comply with the PYRAMID Technical Standard
Component Composition
Component Set
Technical Definition
Glossary
Appendix
Key
PYRAMID Model
Contains content also available to Exploiting Programmes as a UML model export.

Figure 2: PYRAMID Technical Standard Document Structure

This document contains the following sections:

- Reader Guidance: Provides guidance on how to read the PYRAMID Technical Standard and introduces some key terms to the reader to aid understanding.
- PRA Scope and Application: Describes the scope of the PRA and discusses considerations for Exploiting Programmes in applying the PRA.
- Introduction to Components: Provides an overview of the PRA component set and the patterns and structure of the PRA component definitions.

- Introduction to Component Connections: Defines the principles for connecting PYRAMID components to work as a system; it introduces the concept of bridges and outlines the functions they provide.
- How to Comply with the PYRAMID Technical Standard: Describes three aspects of compliance: component compliance, component connection compliance, and deployment compliance, and defines the rules for achieving compliance with the standard.
- Component Composition: Describes the generic pattern for a PRA component definition. It defines a generic set of responsibilities and services and identifies where these have been specialised within the PRA component set.
- Component Set: Provides the PRA component definitions, including the detailed breakdown of the role, responsibilities, entities and services for each subject matter. Each PRA component definition includes specialised content based on the component composition.
- Appendix A: Glossary: Provides definitions of the terms and abbreviations used within the PYRAMID Technical Standard and PYRAMID Technical Standard Guidance document.

3 Reader Guidance

This section introduces some key terms to the reader to aid understanding and provides guidance on how to read the PYRAMID Technical Standard.

3.1 Key Terms

Although a full set of glossary terms is provided in Appendix A: Glossary, an appreciation of the following key terms is essential to understanding the PYRAMID Technical Standard:

- Deployment: A set of hardware and software elements forming a system (or part thereof) that satisfy the overall system requirements.
- Execution Platform: The infrastructure supporting the execution, communication, etc. of application functionality, e.g. ECOA, ARINC 653, Linux, Windows, and the computing hardware.
- Exploiter: An organisation involved in the design and development of PYRAMID components or the design of PYRAMID deployments.
- Exploiting Platform: A product (e.g. an air vehicle, ground station, or a test rig) that incorporates a PYRAMID deployment.
- Exploiting Programme: A programme developing or incorporating PYRAMID components or a PYRAMID deployment.
- PRA component: A PYRAMID reference artefact, defined by a role, a distinct set of responsibilities, entities and services, for a specific, discrete area of subject matter.
- PYRAMID component: A component that is intended to comply with a PRA component definition.

The term ‘component’ refers to a PRA component or PYRAMID component. Where not explicitly stated, the surrounding text will provide context to identify usage.

The term ‘system’ is used in a variety of ways depending on the context. The system of focus may range in size and scope; from a system of systems (such as multiple air vehicles and supporting assets) to a small sub-system or equipment within a larger system (such as a sensing sub-system or a tactical sensor). It may comprise a variety of hardware (e.g. computing, structural, mechanical and electrical hardware) and software (e.g. PYRAMID application software, other application software, middleware and an operating system).

Since the PRA is designed to be scalable, it is rarely possible to be specific about the size or scope of the system in question. However, whilst not true for all cases, within this document it is helpful to think of the system as a full military air system, e.g. an air vehicle and, if relevant, an associated ground station.

When referring to a PYRAMID system, or a similar terminology, this refers to a system containing PYRAMID components, typically with the assumption that most or all of the application software comprises PYRAMID components.

In summary, while the term ‘system’ has a typical usage in some areas of the documentation, as described above, the appropriate usage should be determined from the context within which it is written.

3.2 How to read the PYRAMID Technical Standard

Figure 3: Recommended PYRAMID Technical Standard Reading Order provides a detailed breakdown of the sections that comprise the PYRAMID Technical Standard. It indicates which sections are introductory material, which are reference material, and which are essential PRA artefacts. It also provides a recommended order in which to read them.

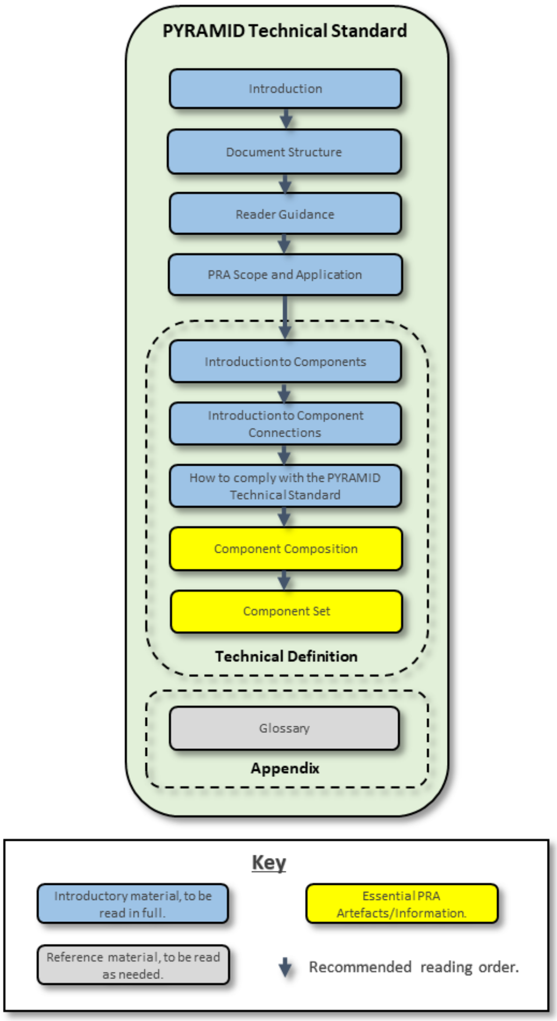

PYRAMID Technical Standard
Introduction
Document Structure
Reader Guidance
PRA Scope and Application
Introduction to Components
Introduction to Component Connections
How to comply with the PYRAMID Technical Standard
Component Composition
Component Set
Technical Definition
Glossary
Appendix
Key
Introductory material, to be read in full.
Essential PRA Artefacts/Information.
Reference material, to be read as needed.
Recommended reading order.

Figure 3: Recommended PYRAMID Technical Standard Reading Order

An accompanying document, the PYRAMID Technical Standard Guidance document, Ref. [2], has also been produced to provide further guidance and supporting information to aid understanding and application of the PYRAMID Technical Standard.

3.3 UML Notation

The PYRAMID Technical Standard includes content which is defined and maintained within the PYRAMID Model, Ref. [3], as illustrated in section Document Structure, Figure 2: PYRAMID Technical Standard Document Structure. This content is also available to Exploiting Programmes in a UML model export format. The PYRAMID model uses UML but it is not a UML software design. It therefore does not adhere rigidly to all UML rules or common conventions. Furthermore, some UML artefacts are used to represent something different to what they would normally be used to represent within a UML software design. It must be emphasised that these deviations are deliberate in order to articulate information in a clear way.

4 PRA Scope and Application

This section describes the scope of the PRA, identifying aspects that are both in and out of scope. It also discusses considerations for Exploiting Programmes in applying the PRA in respect of: defining the applicable scope of the PRA, the implementation of PYRAMID components and the development lifecycle.

4.1 PRA Functional Scope

The PRA is intended to be used as the reference architecture from which PYRAMID components can be developed and integrated as part of an air system.

The PRA covers the full scope of air system application software functionality, including:

- Full range of military and non-military air vehicles, including crewed and uncrewed
- Operational support systems, including:
    - Mission planning and briefing systems
    - Development and training simulators
    - Ground control stations
    - Debriefing and post mission analysis systems
    - Maintenance support systems
- Full range of mission types and all operational/mission phases, including:
    - Planning, rehearsal and briefing
    - Execution (including simulation execution)
    - Debriefing, post mission data analysis and replay
    - Maintenance and support

Whilst formally developed for air systems, the PRA could potentially be used for land, maritime, and space systems.

The PRA provides a set of mutually exclusive component definitions, each covering a discrete area of air system functionality, referred to as subject matter areas. The functional scope of any individual PRA component is defined, and bounded by, a set of component responsibilities. These responsibilities provide the normative element of the component definition for the purpose of assessing the compliance of a PYRAMID component (see section How to Comply with the PYRAMID Technical Standard). The overall functional scope of the PRA is defined by the collective scope of the set of PRA components.

Figure 4: PRA Component Set shows the PRA components. Purely for illustrative purposes, the figure groups the components, to emphasise the scope of the PRA and to aid the reader in quickly identifying components potentially relevant to an Exploiting Programme. The groupings are not intended to provide any basis for system partitioning or component interactions, nor do the groupings imply any intent or restriction on how components may be used.

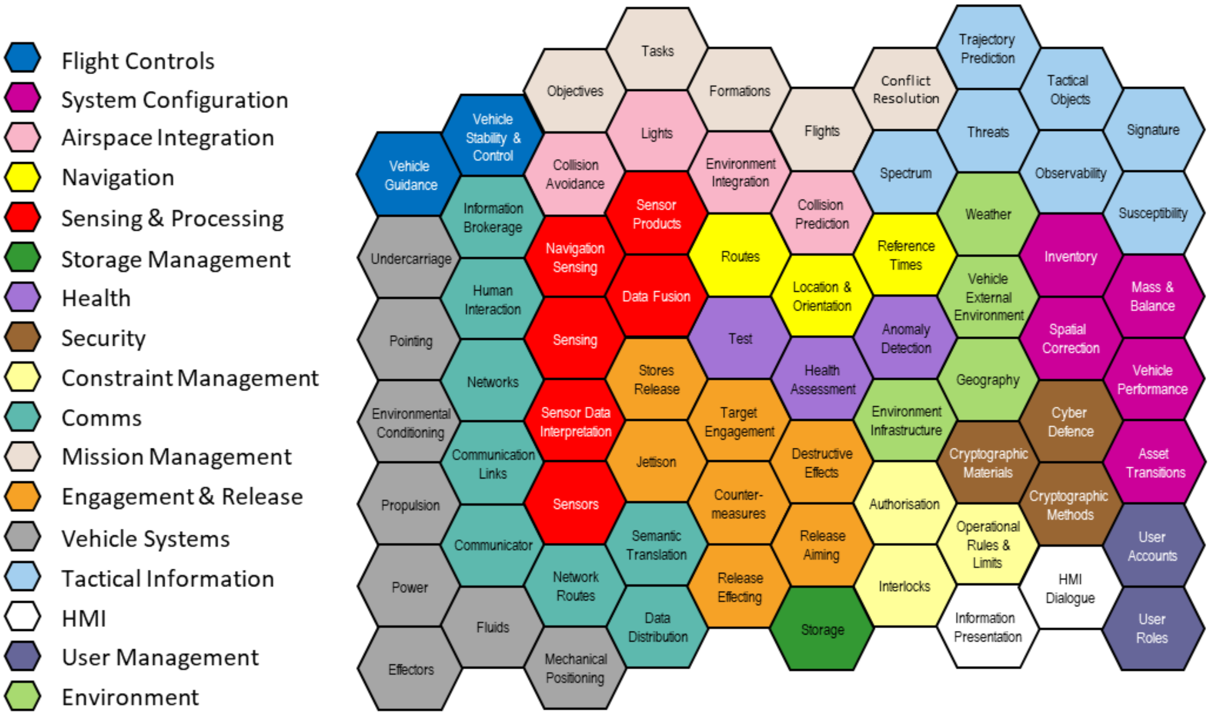


Figure 4: PRA Component Set

The PRA is designed to be independent of the platform type and the computing platform allowing the PRA to be used for any air system programme or product and with any computing platform. The following are therefore out of scope of the PRA:

- **Computing Infrastructure:** The choice of computing hardware, operating system and middleware are all platform specific decisions and as such the PRA makes no assumptions about this infrastructure. Note that, as shown in Figure 4: PRA Component Set, the PRA scope does encompass the use of communications aspects of which might under some circumstances be classified as middleware (for more information, see PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, sections Data Exchange PYRAMID Concept and Use of Communications PYRAMID Concept). The management of computing infrastructure, including the management of errors and faults, and scheduling and latency management, is also out of scope of the PRA. Similarly, storage infrastructure and media are also platform specific and thus outside the scope of the PRA. However, the PRA does make provision for the management of storage infrastructure (see section Storage component definition and PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Storage PYRAMID Concept).
- **Security:** The PRA makes no assumption about the security environment, security risks and security targets that might be applicable to Exploiting Programmes. The PRA does define a number of components whose role relates to security, but does not formally assign component responsibilities for security enforcing functions (SEFs) or security related functions (SRFs). It does not define the security classification of components or component partitioning to meet security requirements, nor does it encompass the management of security partitions. Each PRA component definition does however include an indicative classification and

identifies the nature of any SEFs and SRFs that the component might potentially include given its subject matter.

- **Safety:** The PRA makes no assumption about the operating context, applicable safety standards and safety targets that might be applicable to Exploiting Programmes. Similarly, the PRA does not assign any specific component responsibilities in relation to safety. It does not define component development assurance levels (DALs) or component partitioning to meet safety requirements, nor does it encompass the management of safety partitions. Each PRA component definition does however include an indicative DAL and identifies safety related considerations that might be relevant to the component given its subject matter.

4.2 PRA Application

While the PRA provides well-bounded component subject matters definitions, as described above, the following aspects of how the PRA is applied are left as a matter for Exploiting Programmes:

PYRAMID Deployment Scope Boundary Definition: It is for an Exploiting Programme to determine the elements of a deployment that are intended to comply with the PYRAMID Technical Standard and thus the applicable scope of the PRA. For any Exploiting Programme there will be a boundary beyond which the PRA does not apply. It is for the designers to determine exactly where the boundary lies, including for example whether any equipment control software is within the PRA boundary or is part of the installed equipment. For example, the software for an Inertial Navigation System (INS) could be built from PYRAMID components, or an off-the-shelf INS solution could be used. For more information, see PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Interaction with Equipment PYRAMID Concept, and Appendix D: Deployment Guide, section Consider Interaction with Equipment.

PYRAMID Component Implementation: The PRA defines the functional scope and bounds of each component, as described in section PRA Functional Scope above, but does not mandate any particular implementation approach within those bounds. The functionality of a PYRAMID component must therefore be consistent with the scope of the responsibilities of the PRA component on which it is based, as described in section How to Comply with the PYRAMID Technical Standard. However, the PRA is not prescriptive on the following implementation choices, which are a matter for individual Exploiting Programmes:

- **Component Variants:** Any given variant may implement all, or a subset, of the responsibilities defined within the PRA for that component. For any given PRA component, one or more PYRAMID component variants may be developed. The determination of the functional scope of any given variant will be driven by the system requirements for the Exploiting Programme including safety, security and performance requirements.
- **Component Instances:** The number of deployed instances of any given PYRAMID component variant is also an Exploiting Programme and Exploiting Platform specific consideration. This may also be driven by safety, security and performance requirements.
- **Component Structure:** The PRA component definitions define the functional scope of any given component in the form of a set of component responsibilities. The PRA makes no prescription about the internal structure of any given PYRAMID component development. This

includes the development of different variants as described above, but also the use of component extensions and data driving. The use of such techniques supports the future growth and adaptability of components as described in PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, sections Data Driving PYRAMID Concept and Component Extensions PYRAMID Concept. Where appropriate, the PRA component definitions identify aspects of the component subject matter that may benefit from the application of component extensions and data driving but this is not prescriptive.

- **Software Realisation:** While the PRA is concerned with air system functionality that can be realised through application software, it does not preclude alternative methods of their realisation such as complex electronic hardware and firmware. For example, functionality could be implemented through a single purpose field programmable gate array (FPGA). Furthermore, whilst operating systems are outside the PRA scope, it should be recognised that application software may need to incorporate some operating system functions when used on computing hardware that does not use an operating system.

4.3 Development Lifecycle

PYRAMID Technical Standard does not mandate a particular development lifecycle nor any aspect of the system and software development process. Guidance is provided however on how the system and software development strategies used to develop air system software might be adopted when using the PRA, see PYRAMID Technical Standard Guidance document, Ref. [2], Appendix D: Deployment Guide. As a reference architecture the PRA has particular applicability at the following points in the development lifecycle:

- **Architecture Definition:** The PRA provides a starting point for system architecture development, including component identification and scoping. PRA component definitions provide the basis of PYRAMID component development, while existing PYRAMID compliant developments provide the opportunity for reuse.
- **Compliance assessment:** An assessment of compliance against the PRA may be made throughout the development lifecycle. The rules for achieving compliance with the PYRAMID Technical Standard are defined in section How to Comply with the PYRAMID Technical Standard with supporting guidance provided in PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide. The assessment of compliance is the responsibility of the Exploiting Programme (including its suppliers and customers).

The PRA does not define:

- The approach to achieving safety and security requirements: As noted above, safety and security analysis are specific to each Exploiting Programme and will be expected to take account of the particular safety and security targets, system type and operating scenarios of that programme. The PRA only provides indicative safety and security analysis. Therefore, the Exploiting Programme will be entirely responsible for demonstrating that the Exploiting Platform meets the safety and security targets applicable to the Exploiting Platform. Within the PRA, safety and security have been considered and observations have been recorded, however:

    - The PRA does not place safety and security requirements on an Exploiting Programme.
    - The safety and security considerations in the PRA are not expected to directly contribute to the Exploiting Programmes safety or security case.

  See PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, sections Safety Analysis PYRAMID Concept and Security Approach PYRAMID Concept, for further information.

- The approach and methodology for system qualification, verification or certification (validation).

5 Technical Definition

This section comprises the following:

- An overview of the PRA component definitions supported by the generic component definition defined in the component composition.
- An overview of component connection considerations that allow PYRAMID components to work together as a system.
- How to comply with the standard (i.e. the rules).
- The component definitions that form the PRA.

The PYRAMID Technical Standard definition includes both normative and informative content.

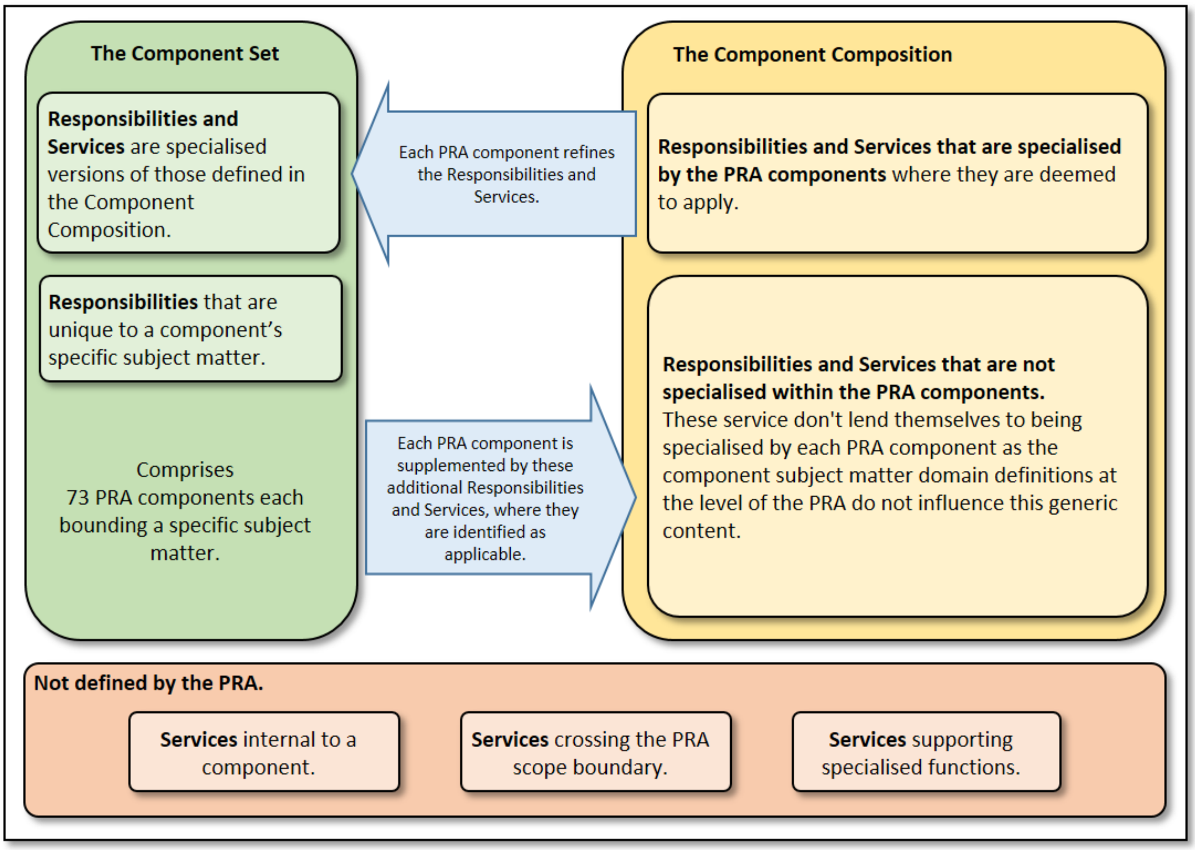
5.1 Introduction to Components

The Component Set section defines 73 PRA components each bounding a specific subject matter in terms of what each component knows about and what each component does.

The defined set of PRA components have been developed from consideration of a broad cross-section of themes that affect air systems, e.g. control architectural considerations, capability management, autonomy, and distinguishing tactical information, to mention a few of the topics. A discussion on each of these themes, including how the PRA has been shaped to address each topic can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], section Introduction to PYRAMID Concepts and Appendix A: PYRAMID Concepts.

These considerations help exploiters understand how to utilise the responsibilities and services that form part of each component definition.

The PRA components, within the Component Set, have been produced using a pattern for a generalised PRA component, referred to as the Component Composition. Each PRA component definition should be read alongside the Component Composition content.



The Component Set

Responsibilities and Services are specialised versions of those defined in the Component Composition.

Responsibilities that are unique to a component's specific subject matter.

Comprises 73 PRA components each bounding a specific subject matter.

Each PRA component refines the Responsibilities and Services.

Each PRA component is supplemented by these additional Responsibilities and Services, where they are identified as applicable.

The Component Composition

Responsibilities and Services that are specialised by the PRA components where they are deemed to apply.

Responsibilities and Services that are not specialised within the PRA components. These service don't lend themselves to being specialised by each PRA component as the component subject matter domain definitions at the level of the PRA do not influence this generic content.

Not defined by the PRA.

Services internal to a component.

Services crossing the PRA scope boundary.

Services supporting specialised functions.

Figure 5: PRA Component Responsibilities and Services

The inclusion of responsibilities and services in the PRA component definitions is shown in Figure 5: PRA Component Responsibilities and Services. The scope of the PRA component definitions therefore include:

- The responsibilities and services defined for a particular component in the Component Set, plus
- The responsibilities and services defined in the Component Composition that are identified as applicable to that component but have not been specialised within the Component Set definition.

The responsibilities define the normative element of the PRA Component definition and are referenced within the rules that specify how to comply with the standard, as defined in the section How to Comply with the PYRAMID Technical Standard**.** The component services are not normative, but aid the reader by providing detailed information on the nature of the expected component interactions with the wider system, both in terms of what is provided and consumed, and an abstract view of the data needed to support these interactions.

5.1.1 Component Composition Overview

The component composition both supplements the PRA component definitions and helps Exploiters to understand concepts that are applicable to most or all PRA components, including how components can interact.

The component composition uses a format and notation similar to that used in the PRA component definitions for responsibilities and services. These are supported by an equivalent to the subject matter semantics, used in the PRA component definitions, only it does not contain any subject matter. Instead, it contains generalised concepts, such as requirements and solutions to requirements. The modelling notation used to define the component composition is further explained by the Component Definition Content and Notation section content.

The other aspects of the PRA component definitions (the role, overview, and design rationale) do not have a useful generalised equivalent, and so are not present within the component composition.

The component composition has several purposes:

- It defines a set of responsibilities and services that are not specialised on the PRA components. These responsibilities and services are potentially applicable to most or all of the PRA components; they therefore supplement the responsibilities and services defined on the PRA components. These responsibilities and services are not included on the PRA components because, at the level of abstraction at which the PRA defines PRA component responsibilities and services, if they were included these would be defined identically within each PRA component. These responsibilities and services are contained within sections Responsibilities that are Not Specialised in PRA Components and Service Definitions that are Not Specialised in PRA Components respectively.
- It defines a set of responsibilities and services that are specialised on the PRA components where they are applicable, tailoring the responsibilities and services from the component composition to the specific subject matter of the PRA component. These responsibilities and services are contained within sections Responsibilities that are Specialised in PRA Components and Service Definitions that are Specialised in PRA Components respectively. This approach allows:
    - A framework for a consistent approach to responsibility and service definitions across the PRA component set.
    - The component composition to provide additional, non subject matter specific, detail about the services, over and above the detail shown within the PRA components. This allows the PRA component definition to focus on subject matter specific detail, whilst the component composition focuses on details that are applicable to most or all PRA components. This additional detail, within the component composition, includes: explanations on how services may not always be triggered by explicit instructions and can be data-driven; more detailed activity decompositions; and more detailed service dependency diagrams, showing the relationships between activities, within the same service and across services.
- It supports use case and PYRAMID concept guidance material, described within the PYRAMID Technical Standard Guidance document, Ref. [2], sections Appendix B: Use Cases and Appendix A: PYRAMID Concepts, which both show patterns of use that can be applied in a deployment:

- The use cases, with their supporting sequence diagrams, show examples of how services can be used by an individual component, to achieve different goals, such as planning how to carry out a solution in response to a requirement.
- The PYRAMID concepts use the component composition services to show how different components can collectively achieve the functionality described within the PYRAMID concepts.

5.1.2 Component Set Overview

The PRA is composed of a set of platform independent component definitions whose scope covers a discrete functional area related to air systems. These PRA components are defined in the Component Set section.

Each PRA component definition contains the following information:

- A Name, which reflects the subject matter of the component.
- A Role, which is its purpose.
- An Overview, which defines the standard pattern of use and examples of use.
- A Service Summary Diagram, which shows the services and interfaces that are defined for the component.
- A set of Responsibilities, which describe the behaviour the component may fulfil within the system.
- Subject Matter Semantics, which defines the scope of the component by showing its entities (artefacts that model concepts that may or may not exist in the real world). This information also includes a semantics diagram, showing the relationships between entities.
- A Design Rationale containing a set of assumptions, design considerations, exploitation considerations, safety considerations and security considerations.
- A set of services, which provide a more detailed view of the scope of a component beyond that of the responsibilities. Each PRA component definition comprise:
  - A Service Definition for each service, which is a detailed description of the service, interfaces and attributes.
  - A Service Dependencies Diagram, which shows how each service depends on other services.

The different parts of the component definition provide different views on the component's subject matter and guidance on its use within a deployment. **The responsibilities define the normative element of the PRA component definition and are referenced within the rules that specify how to comply with the standard, as defined in section** How to Comply with the PYRAMID Technical Standard**.**

5.1.3 Component Definition Content and Notation

The following sub-sections elaborate on the content of the Component Definitions section introduced by the Component Composition Overview and Component Set Overview sections and explain the notation used.

5.1.3.1 Responsibilities

Each PRA component is defined by a set of responsibilities specific to its discrete area of subject matter. A responsibility describes a behaviour a component may fulfil within the system. Unless explicitly restricted, a component’s responsibilities apply to the whole mission lifecycle, including planning, execution and post-mission analysis, and simulation. Responsibilities often reference entities described in the subject matter semantics diagram for a PRA component. The entity definitions provide precise definitions that therefore help to scope the component responsibilities.

Each component has a unique set of responsibilities. In some cases a responsibility may be derived from the generalised pattern, or be expressed in a generic form. It is important to recognise, however, that whether uniquely or generically expressed, the responsibility should be understood in the context of the subject matter of the component and is therefore unique to the component.

The responsibilities define the normative element of the PRA component definition and are referenced within the rules that specify how to comply with the standard, as defined in section How to Comply with the PYRAMID Technical Standard**.**

An example responsibility is (from the PRA component, Authorisation):

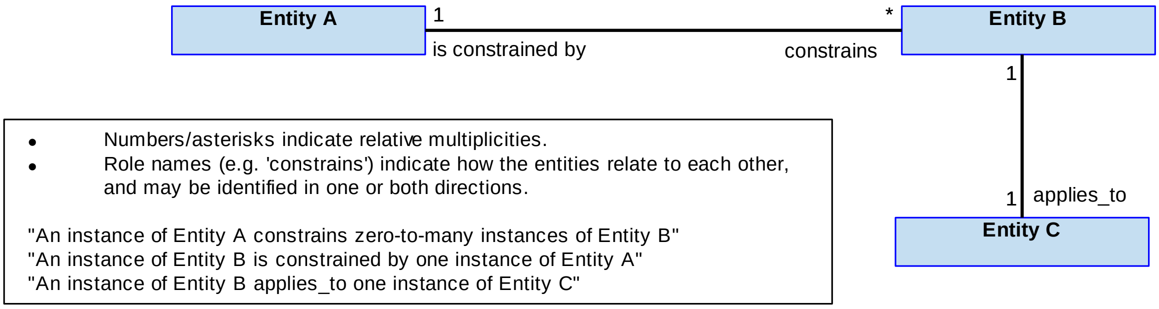
determine_authorisation_solution

- To determine the Steps required to obtain an Authorisation that meets the given Authorisation_Requirements, using available Authorisers in accordance with the applicable Authorisation_Policy.

As indicated in the above responsibility, this responsibility links to entities described in the subject matter semantics diagram, for that component, that are relevant to the responsibility.

5.1.3.2 Subject Matter Semantics Diagram

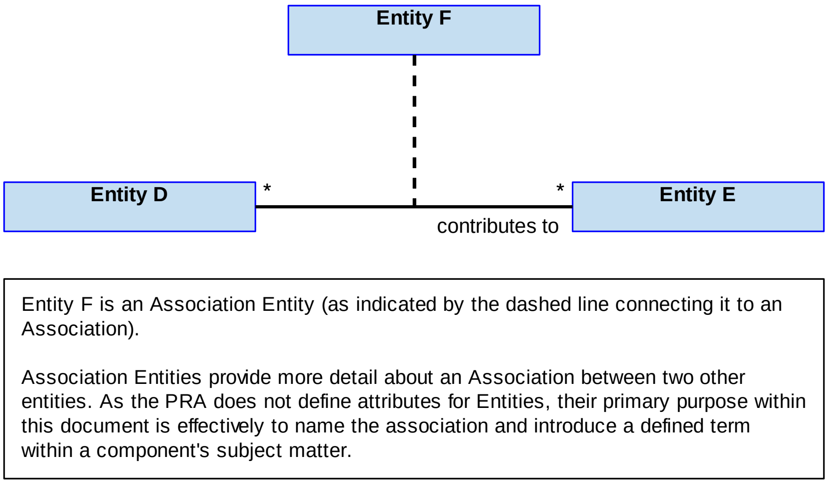
The subject matter semantics diagrams identify entities, representing the types of information that a component knows and reasons about, and the relationships between the entities. The entities are expressed in terms that scope the subject matter of the component. The relationships between entities on the diagram are indicated by solid lines between the entities, an “association”, see Figure 6: Association notation. Figure 7: Association Entity notation provides a notation for an “association entity” which is connected via a dashed line to an association between two entities.



- Numbers/asterisks indicate relative multiplicities.
- Role names (e.g. 'constrains') indicate how the entities relate to each other, and may be identified in one or both directions.

"An instance of Entity A constrains zero-to-many instances of Entity B"
"An instance of Entity B is constrained by one instance of Entity A"
"An instance of Entity B applies_to one instance of Entity C"

Figure 6: Association notation



Entity F is an Association Entity (as indicated by the dashed line connecting it to an Association).

Association Entities provide more detail about an Association between two other entities. As the PRA does not define attributes for Entities, their primary purpose within this document is effectively to name the association and introduce a defined term within a component's subject matter.

Figure 7: Association Entity notation

5.1.3.3 Design Rationale

The Design Rationale contains a set of assumptions, design considerations, exploitation considerations, safety considerations and security considerations, to provide guidance to support PYRAMID component development.

The Design Rationale sections include the following:

- Design considerations which provide Exploiters with guidance on which PYRAMID concepts, those described within the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, are particularly relevant to a component, and other factors which should be taken into account (e.g. applicable standards).
- Exploitation considerations which highlight specific design choices that may need to be made for a deployment.
- Safety considerations provide a statement of, and rationale for, the component's indicative Item Development Assurance Level (IDAL). However the responsibility for safety analysis is the responsibility of the Exploiting Programme.
- Security considerations provide a statement of, and rationale for, the component's indicative security classification. However the responsibility for security analysis is the responsibility of the Exploiting Programme.

5.1.3.4 Services

The PRA uses a service modelling approach which allows the component services to be defined without reference to the internals of a component. It should be noted that the service modelling approach is only one method that an Exploiting Programme could use to define the interfaces at a component's boundary and activities associated with those interfaces.

Services in the PRA are the means by which a component is asked to do something, or by which a component gets something done for it. Component services are modelled as classes. These services are defined by both interfaces and activities. Interfaces describe the conceptual data of the service by specifying their attributes. Activities describe the behaviour that the service will need to fulfil.

Services can either be provided or consumed. A provided service supports the wider system by doing work or providing a definition of the work that it can do. Its activities describe the work to be done. A consumed service requires or uses work done outside of the component, or uses the definition of work that can be done outside of the component to understand the work that it can rely upon being done. Its activities identify the work that the component needs to have done and to assess the response to decide whether any further action needs to be taken.

Both provided and consumed services are defined in terms of the semantics of the component, shown in the PRA component's semantics diagram.

5.1.3.4.1 Interfaces

Interfaces are containers of information which have no directionality. Information can be represented in both the interface description and its associated attributes. Interfaces on the service summary diagrams are represented by balls and cups. These representations are purely based on whether an interface is part of a provided service (ball) or consumed service (cup). In the PRA the interfaces do not dictate the flow of information, as it can be in either direction or bidirectional. Where appropriate the PRA may imply the expected direction, through the interface description or associated attributes.

Attributes of interfaces are expected to be made specific to an Exploiting Programme, e.g. temporal_information may become start_time and end_time. The interfaces of a service can also be made specific to an Exploiting Programme, e.g. either the name or description made specific to the programme while remaining within the PRA component’s subject matter.

Interfaces only define information, not operations since the PRA does not mandate the implementation of whether, when, or how a service is provided. For example, the service interface does not indicate whether the information should be broadcasted continuously or provided on demand. Implementation details, such as operations, are specific to a deployment and so should be added by the Exploiting Programme during the system design phase.

5.1.3.4.2 Service Summary Diagram

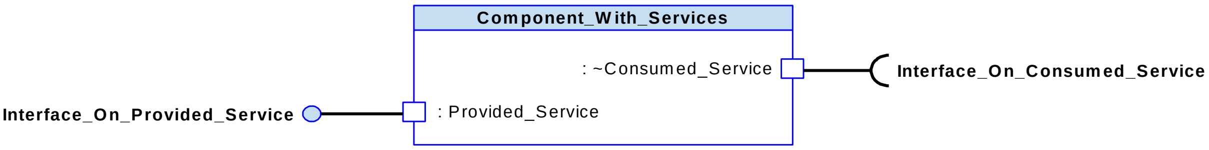


Figure 8: Service Summary Diagram

The service summary diagram is an overview of the services and their respective interfaces on a component. As can be seen on Figure 8: Service Summary Diagram, provided services are shown on the left hand side, with the interface represented by a ball, and consumed services are on the right hand side, with the interface represented by a cup.

5.1.3.4.3 Service Diagrams

Each service definition of a component includes a Service Definition Diagram and a Service Policy Diagram, as described in the following sub-sections.

5.1.3.4.3.1 Service Definition Diagram

A service definition diagram defines a service on a component; it shows the interface(s) which comprise the service and the aspects of the subject matter to which it relates. A component specific interface can inherit the attributes of a generic interface. Services are linked to the entity or entities that represent the aspect of the subject matter that is covered by that particular service.

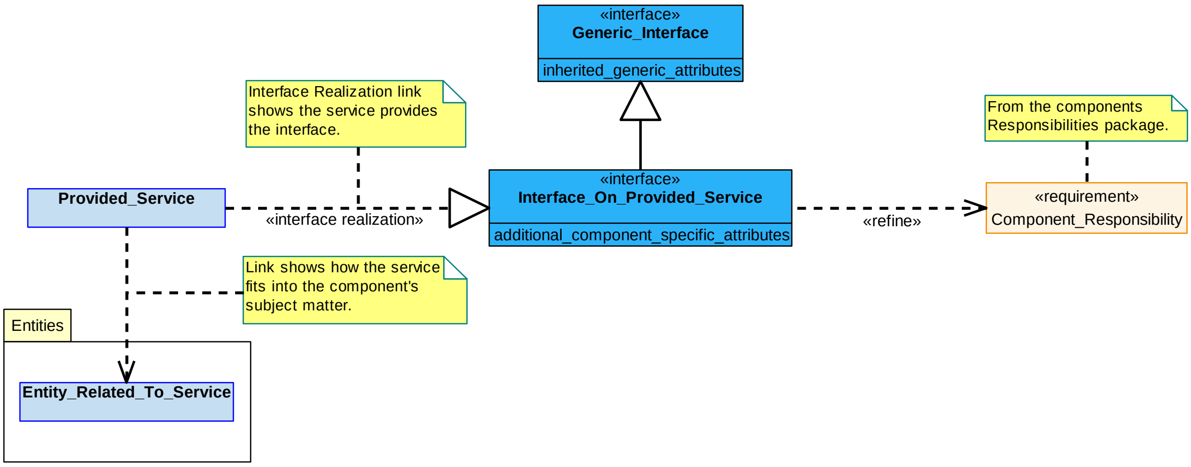

«interface»
Generic_Interface
inherited_generic_attributes
Interface Realization link shows the service provides the interface.
From the components Responsibilities package.
Provided_Service
«interface realization»
«interface»
Interface_On_Provided_Service
additional_component_specific_attributes
«refine»
«requirement»
Component_Responsibility
Link shows how the service fits into the component's subject matter.
Entities
Entity_Related_To_Service

Figure 9: Provided Service Definition Diagram

In Figure 9: Provided Service Definition Diagram the “interface_realization” link shows that this is a provided service. A provided service helps to satisfy one or more component responsibilities as shown by the “refine” link.

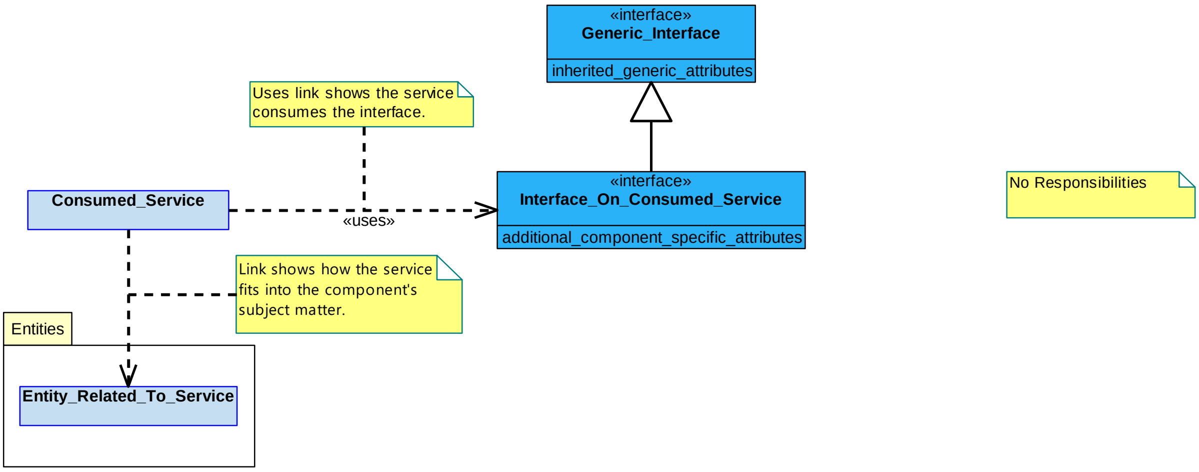

«interface»
Generic_Interface
inherited_generic_attributes
Uses link shows the service consumes the interface.
Consumed_Service
«uses»
«interface»
Interface_On_Consumed_Service
additional_component_specific_attributes
No Responsibilities
Link shows how the service fits into the component's subject matter.
Entities
Entity_Related_To_Service

Figure 10: Consumed Service Definition Diagram

In Figure 10: Consumed Service Definition Diagram, the “uses” link shows that this is a consumed service.

Note that the PRA only shows tracing between the provided services and responsibilities. The tracing between consumed services and responsibilities is not shown in the PRA as this is an indirect mapping, i.e. consumed service support the fulfilment of provided service. When analysing a component it is important to understand that a responsibility could map to provided services and consumed services concurrently.

5.1.3.4.3.2 Service Policy Diagram

A service policy diagram gives an alternative view of a service, focussing on the textual descriptions of the interfaces, activities and the service itself. In the case of provided services, the related responsibilities and their descriptions are also shown. See Figure 11: Provided Service Policy Diagram and Figure 12: Consumed Service Policy Diagram.

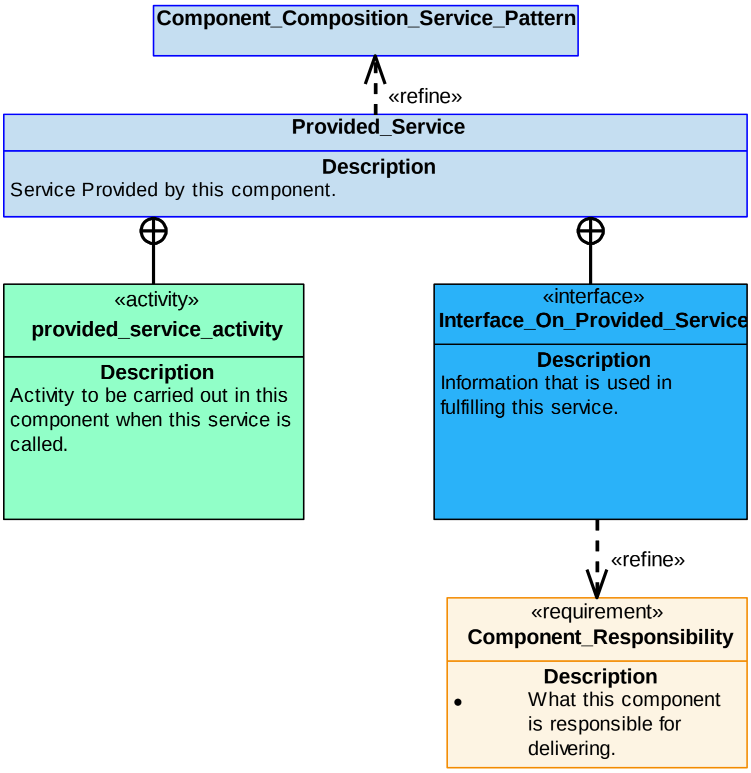

Component_Composition_Service_Pattern
«refine»
Provided_Service
Description
Service Provided by this component.
«activity»
provided_service_activity
Description
Activity to be carried out in this component when this service is called.
«interface»
Interface_On_Provided_Service
Description
Information that is used in fulfilling this service.
«refine»
«requirement»
Component_Responsibility
Description
What this component is responsible for delivering.

Figure 11: Provided Service Policy Diagram

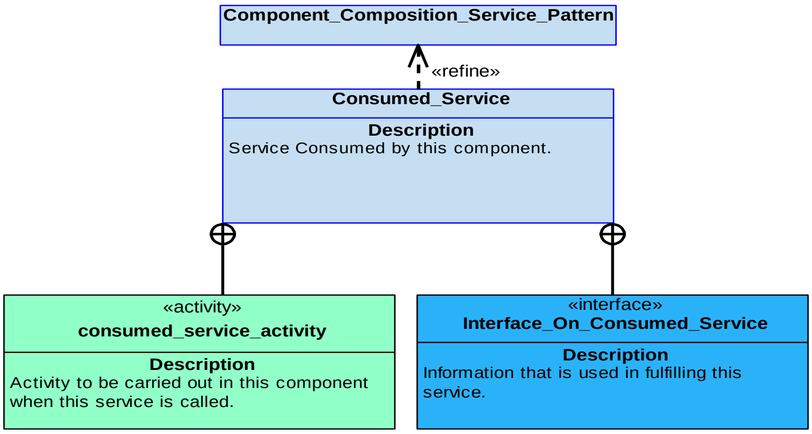

Component_Composition_Service_Pattern
«refine»
Consumed_Service
Description
Service Consumed by this component.
«activity»
consumed_service_activity
Description
Activity to be carried out in this component when this service is called.
«interface»
Interface_On_Consumed_Service
Description
Information that is used in fulfilling this service.

Figure 12: Consumed Service Policy Diagram

5.1.3.4.4 Service Dependency Diagrams

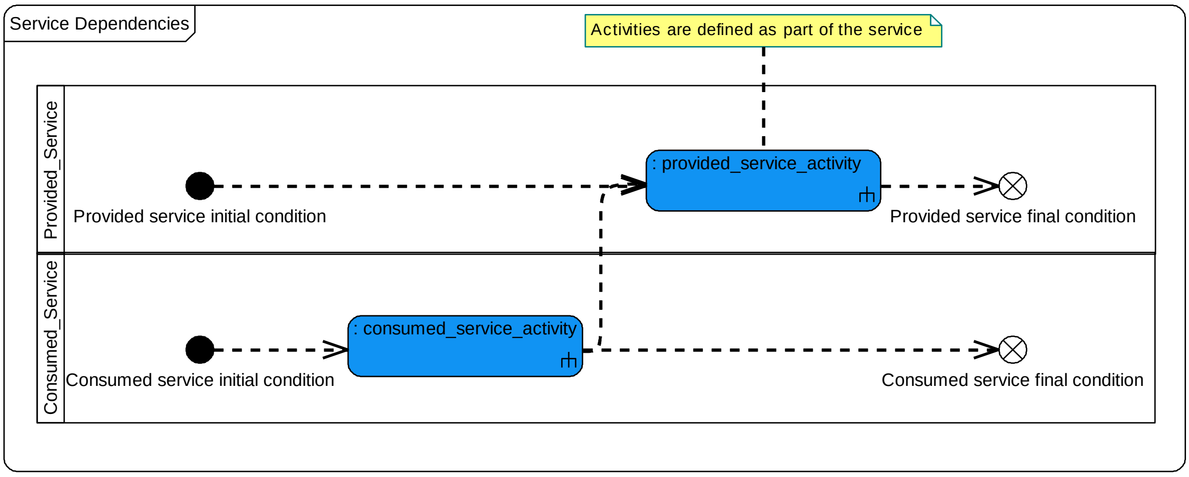

Service Dependencies
Activities are defined as part of the service
Provided_Service
Provided service initial condition
: provided_service_activity
Provided service final condition
Consumed_Service
Consumed service initial condition
: consumed_service_activity
Consumed service final condition

Figure 13: Service Dependencies Diagram

A service dependencies diagram (see Figure 13: Service Dependencies Diagram for an example) shows how a component's services depend on one another. Each service is represented by a swimlane showing the activities involved with that service. Control flows between the swimlanes show the service dependencies.

5.1.4 Component Services Not Defined by the PRA

To ensure the PRA remains platform independent, it does not define services that will be unique to an Exploiting Programme, even though they may be required by a deployment. These services can typically be grouped under one of three categories, described in the three following sub-sections.

5.1.4.1 Services Internal to the Scope of a PRA Component

The services to support interactions between two or more PYRAMID components that are based on the same PRA component are not included in the PRA, because these interactions are considered to be internal to the PRA component. This is irrespective of where these PYRAMID components are located, whether within a single deployment or across separate deployments.

Examples of such interactions are:

- **Instanced Components:** Services between different instances of the same PYRAMID components.
- **Component Extensions:** Services between a parent component and its extension components (see PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, PRA Exploitation Principles PYRAMID Concepts, Component Extensions).
- **Component Variants:** Services between different PYRAMID component variants that are all based on the same PRA component.

Where the capability corresponding to a PRA component is partly provided by non-PYRAMID software, this is equivalent to a variant or another instance of a PYRAMID components. As such, these interactions fall under the ‘Instanced Components’ or 'Component Variants' groups above, and are therefore not included in the PRA.

5.1.4.2 Services Crossing the PRA Scope Boundary

In order to achieve their objectives, PYRAMID components will need to interact with systems outside the scope of the PRA, most often a specific resource or the host computing infrastructure. The services to interact across this scope boundary are not included in the PRA.

Examples of such interactions are:

- **Resource Access:** Services supporting the interaction between any resource and the resource manager.
- **Computing Infrastructure:** Use of the computing infrastructure to allow access to:
  - **Data Storage** - Allowing data to be written to and read from a storage device.
  - **Operating System Layer** - Use of common system functions, e.g. timers or interrupt handlers.
  - **Device Drivers** - Interfaces to the drivers that support peripheral devices.
  - **Library Functions** - Use of software libraries, e.g. mathematical or graphical libraries.

5.1.4.3 Deployment Specific Services

Components may need specific services to support activities such as data loading, data extraction and component configuration. Services to support these extended functions are not defined in the PRA.

Examples of such interactions are:

- **Data Driving Support:** Services needed to load data sets allowing data-driven design.
- **Injection and Override:** Services allowing a user or external system to adjust data inside a component, or to issue commands to affect its operation. This will support activities like mission data loading, user override, and interactions in support of test execution.
- **Data Visibility:** Providing access to the data held internally to a component. This may be for display to a user, fault logging, monitoring by an external system, or report generation.

5.2 Introduction to Component Connections

Component connections define relationships between components that allow them to work together as a system. The PRA has been designed around a key set of design principles (see the PYRAMID Technical Standard Guidance document, Ref. [2], section PRA Design Principles) resulting in a set of component definitions that each describe a discrete area of functionality related to air systems, referred to as ‘subject matter’ areas. Components in the PRA are deliberately scoped to have no knowledge of other components within a system. This approach creates a set of well-bounded and loosely coupled components that can be developed in isolation from one another.

An Exploiting Programme will combine components to provide system functionality through the use of bridges to connect components together. Since each PRA component represents a distinct subject matter area that is defined in its own language, there is no shared interface definition between PRA components. Instead, a deployment will use bridges to close the semantic gap, aligning the interfaces between components so that they are able to share information. A bridge is designed to achieve the component connections that are required by the system’s needs.

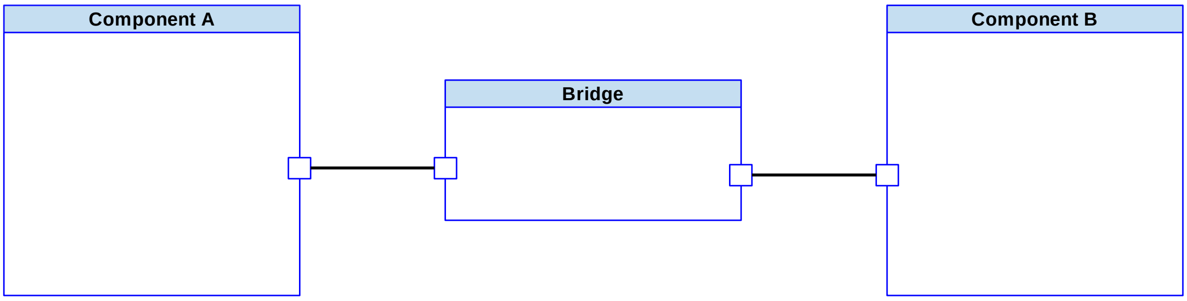


Figure 14: Basic Bridge Diagram

The Figure 14: Basic Bridge Diagram shows a single bridge facilitating connection between two components, however a bridge may also be used in a deployment to facilitate one to many connections and even many to many connections.

In general, a bridge defines which component services and information are connected to which other component services and information. To achieve this a bridge will provide the following functions:

- Data type conversion (translating between the different formats and measurements used by different components).
- Data element mapping (translating the meaning of data in order to bridge the semantic gap between different component subject matter understandings).
- Triggering activities in a component, driven by event(s) generated in other components.

A bridge should be limited to performing these functions and should not include functionality that is the responsibility of a PRA component. The PRA defines compliance rules for component connections to ensure that the intent of the PRA is met, these compliance rules are defined within section How to Comply with the PYRAMID Technical Standard.

Further guidance on component connections and use of bridges is provided in the PYRAMID Technical Standard Guidance document, Ref.[2], Appendix A: PYRAMID Concepts, section

Component Connections. This reference material also provides guidance on counterparting and component interactions, along with examples of their use.

5.3 How to Comply with the PYRAMID Technical Standard

This section defines the rules for achieving compliance with the PYRAMID Technical Standard.

The goal of PYRAMID compliance is to help ensure that an Exploiting Programme realises the benefits provided by the PRA. The subsections below cover the following aspects of PYRAMID compliance:

- **Component Compliance**: This seeks to preserve the PRA defined subject matter separations and ensure that components remain highly cohesive and loosely coupled.
- **Component Connections Compliance**: This seeks to ensure that bridges, used to connect PYRAMID components, do not fulfil the role of PRA components.
- **Deployment Compliance**: This provides an overall measure of compliance for the PYRAMID designated elements of a deployment.

Compliance can be assessed against the PRA at any phase of an Exploiting Programme's implementation lifecycle, whether that be at the stage of specification, design, or qualification. Supporting information on all aspects of compliance with the PYRAMID Technical Standard can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide.

The following key terms are used in this section:

- **PRA Component**: A PYRAMID reference artefact, defined by a role, a distinct set of responsibilities, entities and services, for a specific, discrete area of subject matter.
- **PYRAMID Component**: A component that is intended to comply with a PRA Component definition.
- **Target PRA Component**: A PRA component against which the compliance of a PYRAMID component is being determined.
- **PYRAMID Deployment Scope**: The elements of a deployment that are intended to comply with the PYRAMID Technical Standard.

5.3.1 PYRAMID Component Compliance Rules

This section defines the rules used to determine if a PYRAMID component is compliant with the PYRAMID Technical Standard.

The basis of component compliance is conformance with the PRA defined subject matters, and thus, the absence of pollution.

Each PRA component represents a discrete area of subject matter. Subject matter pollution occurs when a PYRAMID component includes any subject matter of a PRA component other than the target PRA component. The PRA defines a set of responsibilities for each component that bound the functional scope of the component. The responsibilities for each component are defined in section Component Definitions and provide the normative aspect of the component definition for the purpose of component compliance assessment. Component compliance is achieved by demonstrating that the functionality of a PYRAMID component is consistent with the scope of the responsibilities of the target PRA component.

The rule for component compliance is defined in Table 1: PYRAMID Component Compliance Rule. The rule is applicable to all PYRAMID component variants, for example a PYRAMID component that implements only a subset of the PRA components responsibilities. Where a component is realised using extensions, the rule is applicable to the parent and extension components.

Rule Number	**PYRAMID Component Compliance Rule**
Component_rule_1	A PYRAMID component's content shall be consistent with the responsibilities of the target PRA component.

Table 1: PYRAMID Component Compliance Rule

Ensuring that a PYRAMID component is consistent with the responsibilities of the target PRA component requires checking not only that its functionality is required to fulfil one or more of the target PRA component responsibilities but also that the component does not incorporate functionality that should be provided by another PRA component. This is discussed further in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide.

It is also intended that PYRAMID components are agnostic of their environment in as much as they are defined in a way that makes minimal assumptions about how they will be used or how they will be connected in any specific deployment. The use of bridges supports this goal enabling the creation of a set of well bounded and loosely coupled components, maximising the opportunity for reuse and minimising the impact of changes, as described in the PYRAMID Technical Standard Guidance document, Ref. [2], section Component Connections. While not expressed as a compliance rule, it is recommended that PYRAMID components do not therefore implement the functions of bridges except where this is consistent with the responsibilities of the target PRA component. Further guidance on this is provided in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide.

Supporting guidance material in relation to the component compliance rule and the assessment of PYRAMID component compliance can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide.

5.3.2 PYRAMID Component Connection Compliance Rules

This section defines the rules used to determine if a component connection is compliant with the PYRAMID Technical Standard.

The basis of component connection compliance is the correct use of bridges to connect components together.

In addition to the rule for PYRAMID components, described above, it is also important that component connections are implemented appropriately. Bridges provide a mechanism for connecting a component to other system elements and are used to perform the translations necessary to enable components to remain independent of the structure and semantics of other system elements. However, it is important that bridge implementations do not inappropriately fulfil (or partially fulfil) the responsibilities defined for a PRA component. Rather, it should be recognised that the required capability falls within the scope of the responsibilities of a defined PRA component and the appropriate PYRAMID component developed (based on this PRA component definition). Compliance is therefore achieved by demonstrating that bridges include only the necessary functionality to enable components to remain independent of knowledge of the structure and semantics of other system elements, without incorrectly incorporating functionality that is the responsibility of a PRA component.

Since the subject matter of some PRA components encompass some aspects of what are considered bridging functions, there are some exceptions that need to be taken in to consideration. These exceptions are discussed further in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide.

The rule for component connections compliance is defined in Table 2: PYRAMID Component Connection Compliance Rule. The rule for component connections applies to all connections between PYRAMID components. Note that bridges are not necessarily required between different instances or variants of PYRAMID components conforming to the same PRA component, or between a parent component and an extension of that component, since such interactions are internal to a single PRA component.

Rule Number	Component Connection Compliance Rule
Connections_rule_1	A bridge shall not fulfil a responsibility of a PRA component.

Table 2: PYRAMID Component Connection Compliance Rule

Further information in relation to component connections can be found in section Introduction to Component Connections and the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts section Component Connections. Supporting guidance material in relation to the component connection rule and the assessment of component connections compliance can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide.

5.3.3 PYRAMID Deployment Compliance Rules

This section defines the rules used to determine if a PYRAMID deployment is compliant with the PYRAMID Technical Standard.

The basis of deployment compliance is the achievement of component compliance for all components within the PYRAMID deployment scope and a compliant means of connecting those components.

The rules for deployment compliance are defined in Table 3: PYRAMID Deployment Compliance Rules. The rules for deployment compliance apply to all components and component connections defined as being within the PYRAMID deployment scope.

Rule Number	PYRAMID Deployment Compliance Rule
Deployment_rule_1	All the components within the PYRAMID deployment scope shall satisfy the rules for PYRAMID component compliance.
Deployment_rule_2	All the component connections within the PYRAMID deployment scope shall satisfy the rules for PYRAMID component connection compliance.

Table 3: PYRAMID Deployment Compliance Rules

Supporting guidance material in relation to the deployment compliance rules and the assessment of PYRAMID deployment compliance can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix E: Compliance Guide.

5.4 Component Definitions

This section defines the Component Composition and PRA Component Set.

5.4.1 Component Composition

5.4.1.1 Subject Matter Semantics

The subject matter semantics of the Component Composition is equivalent to the subject matter semantics used in the PRA component definitions, only it does not contain any subject matter; instead, it contains generalised concepts, such as requirements and solutions to requirements. PRA components include entities that correspond to some or all of the entities shown here, but they are expressed in terms suitable to the subject matter of the PRA component.

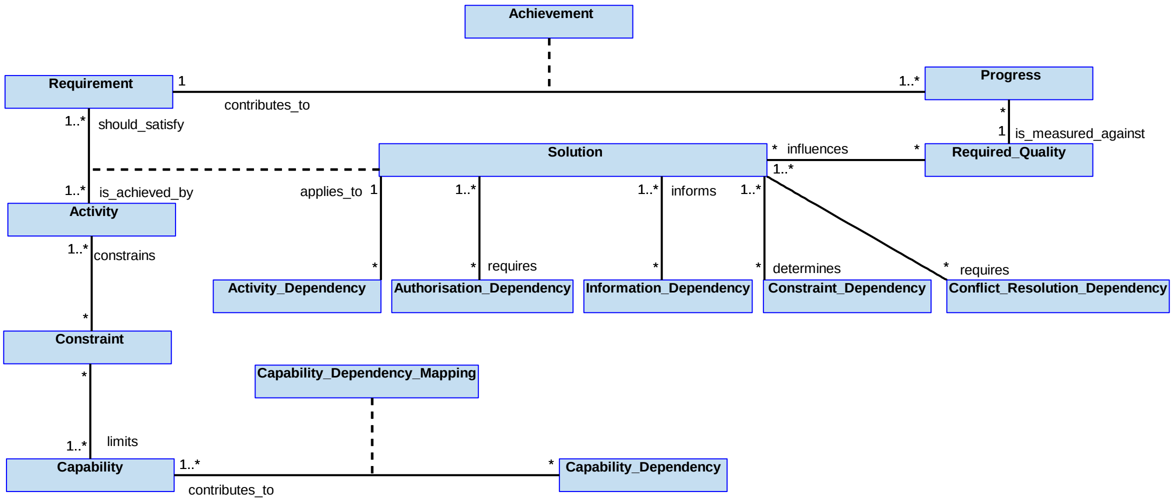


Figure 15: Semantics

5.4.1.1.1 Entities

Constraint

A limitation on the behaviour of the component.

Capability_Dependency

A capability provided by the rest of the system, that the component relies on in order to provide one or more of its capabilities.

Capability_Dependency_Mapping

An identification of a dependency on a capability from the rest of the system in order for a capability to be enacted.

Solution

An approach to fulfilling a Requirement, which may involve performing work within the component and using resources managed by the component, as well as placing dependencies on the rest of the system.

Capability

The ability for the component to perform a particular function. A component's capabilities are derived from the capabilities provided by the rest of the system, along with resources it has available for use.

Activity

Something the component may do as part of satisfying a particular Requirement or requirements.

Requirement

Something that the component is required to do.

Progress

A quantification of what steps have been made towards achieving a Requirement.

Achievement

A measure of whether a Requirement placed onto a component has been met.

Activity_Dependency

An activity that the component needs the rest of the system to perform in order to enact a Solution.

Information_Dependency

Information that the component needs the rest of the system to provide in order to execute a Solution.

Required_Quality

A measure of the properties that a Solution needs to meet in order to fulfil a Requirement.

Constraint_Dependency

A constraint that the component needs the rest of the system to comply with as part of executing the required Solution.

Authorisation_Dependency

An authorisation that the component needs the rest of the system to provide in order to execute the required Solution.

Conflict_Resolution_Dependency

A conflict that the component needs resolving in order to have a Solution that is part of a coherent set of Solutions, that can be used by the component to fulfil all Requirements that may need to be fulfilled collectively.

5.4.1.2 Responsibilities

This section provides details of generalised component responsibilities, including those that are:

- Specialised within the Component Set.
- Not specialised within the Component Set, since their generalised definition is applicable to most of if not all PRA components.

These responsibilities align with the interaction patterns shown in the use cases within the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix B: Use Cases, which demonstrate how services related to the responsibilities can be used. The PYRAMID concepts, within the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, explain the significance of many of the responsibilities.

A PRA component may have other responsibilities, which are not generalised within the component composition, that are specific to its subject matter.

5.4.1.2.1 Responsibilities that are Specialised in PRA Components

This section lists generic responsibilities, which are specialised within the PRA components. However, it should be noted that not all of these responsibilities apply to every PRA component; where they do apply, they are specialised to the subject matter of the PRA component.

capture_requirements

- To capture provided Requirements, including relationships between a source Requirement and derived requirements.

capture_measurement_criteria

- To capture given measurement criteria.

capture_constraints

- To capture provided Constraints.

identify_whether_requirement_is_achievable

- To identify whether a Requirement is still achievable given current or predicted Capability and conditions.

determine_solution

- To determine a solution, within the demanded Constraints, that either meets a Requirement, ensures that a Constraint is complied with, or recovers a Capability.

determine_solution_dependencies

- To determine dependencies required to support the Solution or a step of the Solution.

determine_if_solution_remains_feasible

- To determine the feasibility of a planned or on-going Solution.

coordinate_dependencies

- To coordinate the dependencies to execute a Solution.

identify_progress

- To identify progress against a Requirement.

determine_quality_of_deliverables

- To determine the quality of provided deliverables against given measurement criteria.

assess_capability

- To assess the Capability of the component taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage, or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the component's Capability over time and with use.

5.4.1.2.2 Responsibilities that are Not Specialised in PRA Components

This section lists generic responsibilities, which are applicable to most of if not all PRA components. The responsibilities do not lend themselves to being specialised by each PRA component as the component subject matter specific definitions at the level of the PRA do not influence their generic definitions. An Exploiter should consider these generic responsibilities as supplementing the list of responsibilities in any given PRA component definition in order to complete the definition of the PRA component.

determine_authorisation_dependencies

- To determine authorisation dependencies required to support the Solution or a step of the Solution.

  Applicability: This responsibility is applicable to all PRA components.

identify_conflict

- To identify a Requirement or Constraint placed on the component that is unable to be satisfied as a result of other Requirements or Constraints.

  Applicability: This responsibility is applicable to all PRA components with the exception of the Authorisation component.

determine_refinement_goal

- To determine the refinement goal needed for a derived demand (e.g. a derived requirement or constraint) involved in a conflict.

  Applicability: This responsibility is applicable to all PRA components with the exception of the Conflict Resolution component.

address_capability_issue

- To address shortfalls within the component when a necessary capability that can be established is unavailable or degraded.

  Applicability: This responsibility is applicable to all PRA components.

determine_retention_requirements

- To determine own retention needs for the recording and logging of data.

  Applicability: This responsibility is applicable to all PRA components.

manage_data_retention_and_storage

- To save, retrieve and delete own data in accordance with data retention and storage requirements.

  Applicability: This responsibility is applicable to all PRA components.

coordinate_retention_activities

- To coordinate recording and logging activities in accordance with data retention requirements.

  Applicability: This responsibility is applicable to all PRA components with the exception of the Storage component.

determine_storage_requirements

- To determine own storage needs for the retention (i.e. recording and logging) and processing of data.

  Applicability: This responsibility is applicable to all PRA components with the exception of the Storage component.

data_validation

- To perform the validation of received data before use. This validation may range from simple data formatting to checking that the data is valid and appropriate for use.

  Applicability: This responsibility is applicable to all PRA components.

capture_autonomy_remit

- To capture own remit to perform activities autonomously and the conditions under which authorisation is required.

  Applicability: This responsibility is applicable to all PRA components.

5.4.1.3 Services

This section provides the services defined within the Component Composition, which are divided into sections for services that are further specialised with the definitions for individual PRA components and those that are not.

Many of the services are supported by a set of use cases, with supporting sequence diagrams, in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix B: Use Cases. These show how, from the perspective of a single component, the services can be used to interact with other system elements, including other components, or system users. (These are not provided for the use of the Retention_Requirement, Storage_Dependency and Retention_Coordination_Dependency services).

Not all of the services within the component composition apply to every PRA component. Where they apply from section Service Definitions that are Specialised in PRA Components, they are specialised to the subject matter of the component, and contain a reference to the associated component composition generalised service.

5.4.1.3.1 Service Summary

The following details a summary of services, categorised and shown separately as follows:

- Specialised within the Component Set.
- Not specialised within the Component Set, since their generalised definition is applicable to most PRA components.

5.4.1.3.1.1 Summary of Services that are Specialised in PRA Components

Figure 16: Summary of Services that are Specialised in PRA Components shows generic services that are specialised within the PRA components to their subject matter.

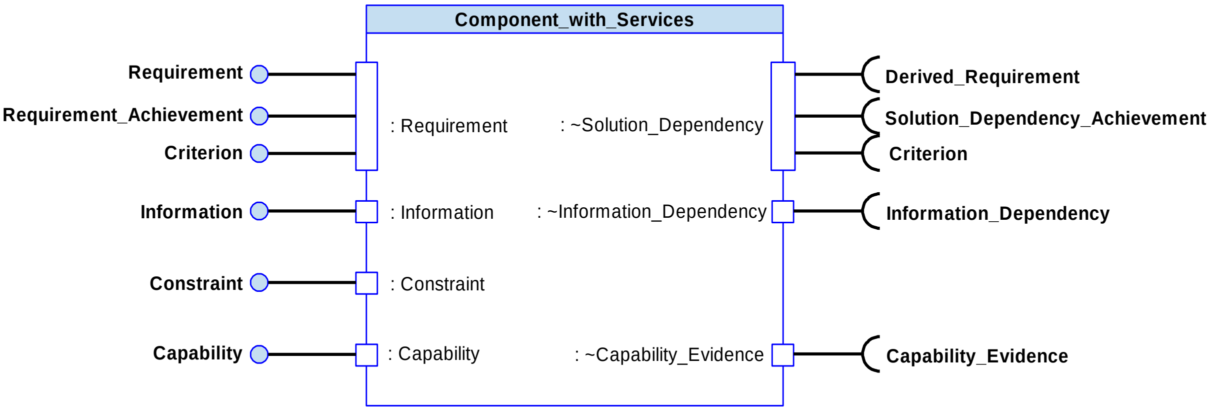


Figure 16: Summary of Services that are Specialised in PRA Components

5.4.1.3.1.2 Summary of Services that are Not Specialised in PRA Components

Figure 17: Summary of Services that are Not Specialised in PRA Components shows services that are not specialised within the PRA components.

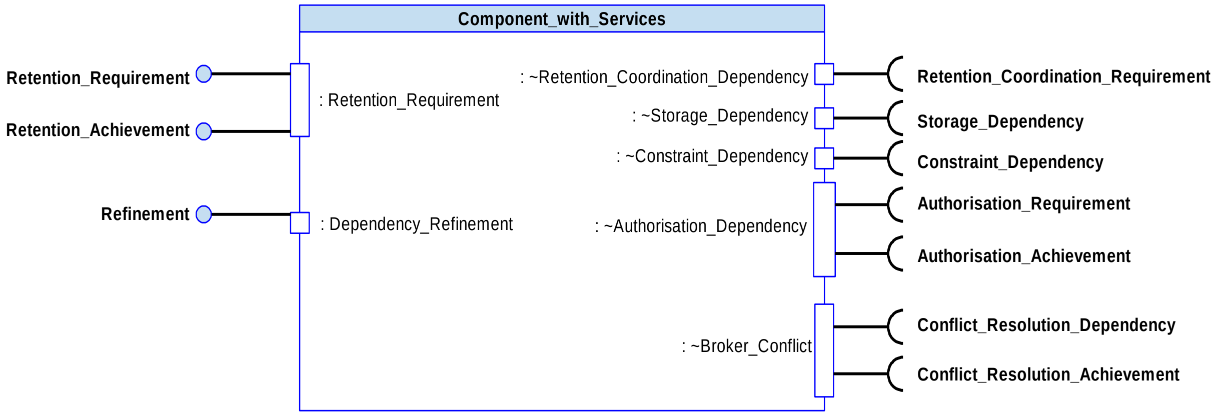


Figure 17: Summary of Services that are Not Specialised in PRA Components

5.4.1.3.2 Service Definitions

This section defines the component composition services, which are categorised and shown separately as follows:

- Specialised within the Component Set.
- Not specialised within the Component Set, since their generalised definition is applicable to most PRA components.

Each category is sub categorised as being either a provided service or a consumed service.

5.4.1.3.2.1 Service Definitions that are specialised in PRA Components

This section defines the provided and consumed services that are specialised by PRA components within the Component Set. The PRA components tailor the services within this part of the component composition to the specific subject matter of the PRA component.

5.4.1.3.2.1.1 Provided Services

5.4.1.3.2.1.1.1 Requirement

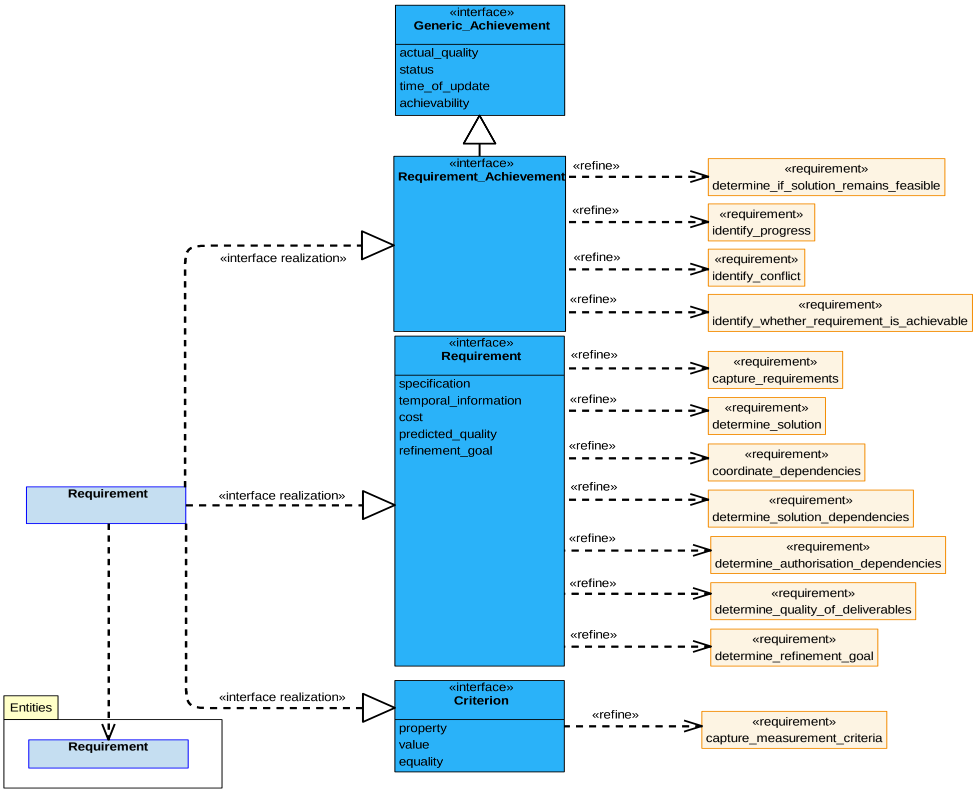
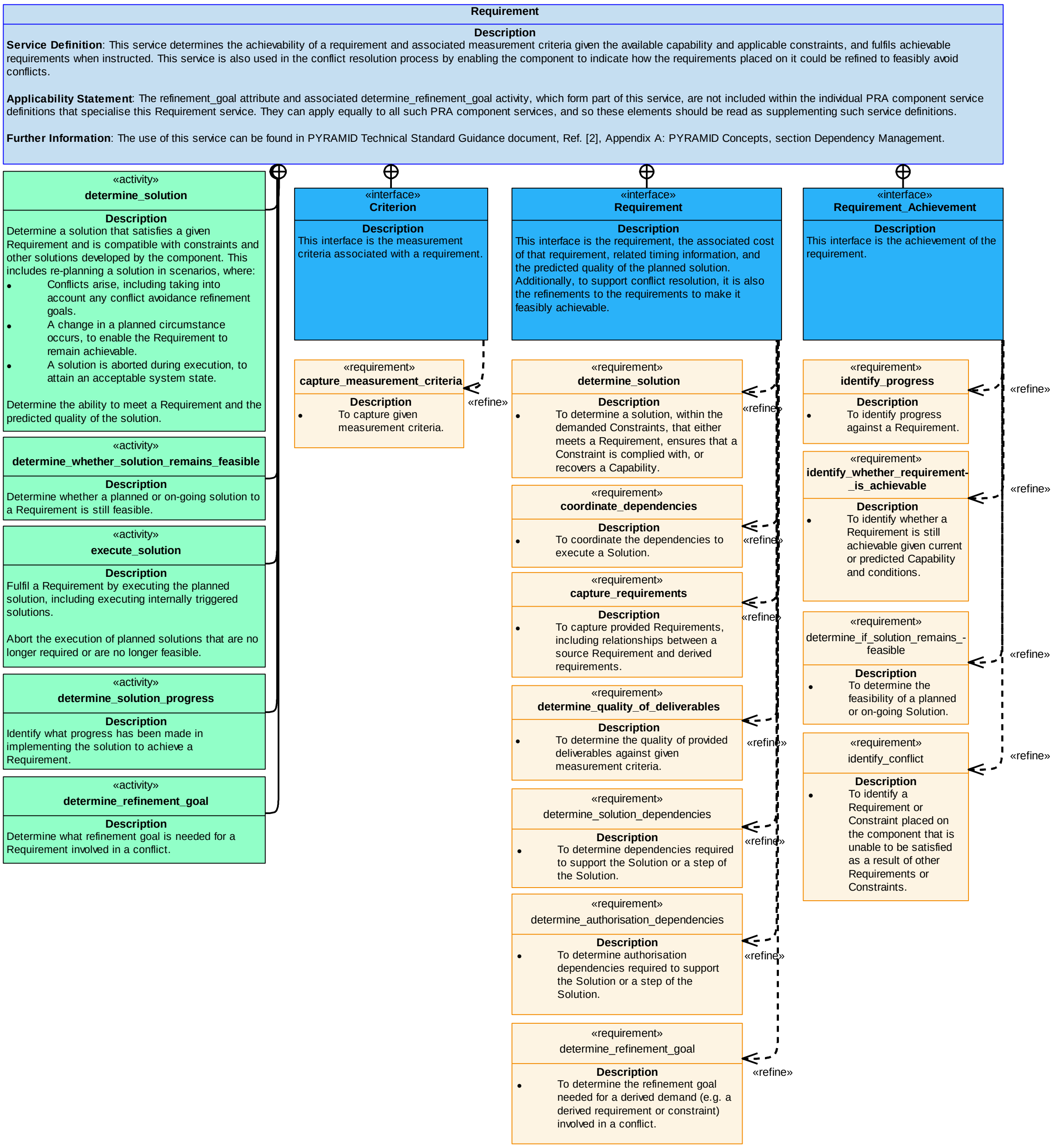


Figure 18: Requirement Service Definition



Requirement

Description

Service Definition: This service determines the achievability of a requirement and associated measurement criteria given the available capability and applicable constraints, and fulfils achievable requirements when instructed. This service is also used in the conflict resolution process by enabling the component to indicate how the requirements placed on it could be refined to feasibly avoid conflicts.

Applicability Statement: The refinement_goal attribute and associated determine_refinement_goal activity, which form part of this service, are not included within the individual PRA component service definitions that specialise this Requirement service. They can apply equally to all such PRA component services, and so these elements should be read as supplementing such service definitions.

Further Information: The use of this service can be found in PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

«activity» **determine_solution**

Description

Determine a solution that satisfies a given Requirement and is compatible with constraints and other solutions developed by the component. This includes re-planning a solution in scenarios, where:

- Conflicts arise, including taking into account any conflict avoidance refinement goals.
- A change in a planned circumstance occurs, to enable the Requirement to remain achievable.
- A solution is aborted during execution, to attain an acceptable system state.

Determine the ability to meet a Requirement and the predicted quality of the solution.

«activity» **determine_whether_solution_remains_feasible**

Description

Determine whether a planned or on-going solution to a Requirement is still feasible.

«activity» **execute_solution**

Description

Fulfil a Requirement by executing the planned solution, including executing internally triggered solutions.

Abort the execution of planned solutions that are no longer required or are no longer feasible.

«activity» **determine_solution_progress**

Description

Identify what progress has been made in implementing the solution to achieve a Requirement.

«activity» **determine_refinement_goal**

Description

Determine what refinement goal is needed for a Requirement involved in a conflict.

«interface» **Criterion**

Description

This interface is the measurement criteria associated with a requirement.

«interface» **Requirement**

Description

This interface is the requirement, the associated cost of that requirement, related timing information, and the predicted quality of the planned solution. Additionally, to support conflict resolution, it is also the refinements to the requirements to make it feasibly achievable.

«interface» **Requirement_Achievement**

Description

This interface is the achievement of the requirement.

«requirement» **capture_measurement_criteria**

Description

- To capture given measurement criteria.

«requirement» **determine_solution**

Description

- To determine a solution, within the demanded Constraints, that either meets a Requirement, ensures that a Constraint is complied with, or recovers a Capability.

«requirement» **coordinate_dependencies**

Description

- To coordinate the dependencies to execute a Solution.

«requirement» **capture_requirements**

Description

- To capture provided Requirements, including relationships between a source Requirement and derived requirements.

«requirement» **determine_quality_of_deliverables**

Description

- To determine the quality of provided deliverables against given measurement criteria.

«requirement» determine_solution_dependencies

Description

- To determine dependencies required to support the Solution or a step of the Solution.

«requirement» determine_authorisation_dependencies

Description

- To determine authorisation dependencies required to support the Solution or a step of the Solution.

«requirement» determine_refinement_goal

Description

- To determine the refinement goal needed for a derived demand (e.g. a derived requirement or constraint) involved in a conflict.

«requirement» **identify_progress**

Description

- To identify progress against a Requirement.

«requirement» **identify_whether_requirement-_is_achievable**

Description

- To identify whether a Requirement is still achievable given current or predicted Capability and conditions.

«requirement» determine_if_solution_remains_-feasible

Description

- To determine the feasibility of a planned or on-going Solution.

«requirement» identify_conflict

Description

- To identify a Requirement or Constraint placed on the component that is unable to be satisfied as a result of other Requirements or Constraints.

Figure 19: Requirement Service Policy

Requirement

Service Definition: This service determines the achievability of a requirement and associated measurement criteria given the available capability and applicable constraints, and fulfils achievable requirements when instructed. This service is also used in the conflict resolution process by enabling the component to indicate how the requirements placed on it could be refined to feasibly avoid conflicts.

Applicability Statement: The refinement_goal attribute and associated determine_refinement_goal activity, which form part of this service, are not included within the individual PRA component service definitions that specialise this Requirement service. They can apply equally to all such PRA component services, and so these elements should be read as supplementing such service definitions.

Further Information: The use of this service can be found in PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Interfaces

Requirement

This interface is the requirement, the associated cost of that requirement, related timing information, and the predicted quality of the planned solution. Additionally, to support conflict resolution, it is also the refinements to the requirements to make it feasibly achievable.

Attributes

specification	The definition of the requirement.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used, time taken.
predicted_quality	How well the planned solution is predicted to satisfy the requirement.
refinement_goal	The specific aspects of the requirement that need to be modified and how they need to be modified.

Criterion

This interface is the measurement criteria associated with a requirement.

Attributes

property	The property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Requirement_Achievement

This interface is the achievement of the requirement.

Activities

determine_solution

Determine a solution that satisfies a given Requirement and is compatible with constraints and other solutions developed by the component. This includes re-planning a solution in scenarios, where:

- Conflicts arise, including taking into account any conflict avoidance refinement goals.
- A change in a planned circumstance occurs, to enable the Requirement to remain achievable.
- A solution is aborted during execution, to attain an acceptable system state.

Determine the ability to meet a Requirement and the predicted quality of the solution.

determine_whether_solution_remains_feasible

Determine whether a planned or on-going solution to a Requirement is still feasible.

execute_solution

Fulfil a Requirement by executing the planned solution, including executing internally triggered solutions.

Abort the execution of planned solutions that are no longer required or are no longer feasible.

determine_solution_progress

Identify what progress has been made in implementing the solution to achieve a Requirement.

determine_refinement_goal

Determine what refinement goal is needed for a Requirement involved in a conflict.

5.4.1.3.2.1.1.2 Information

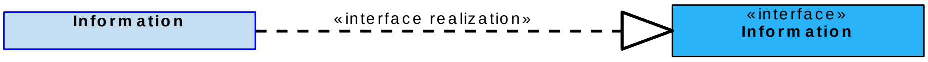


Figure 20: Information Service Definition

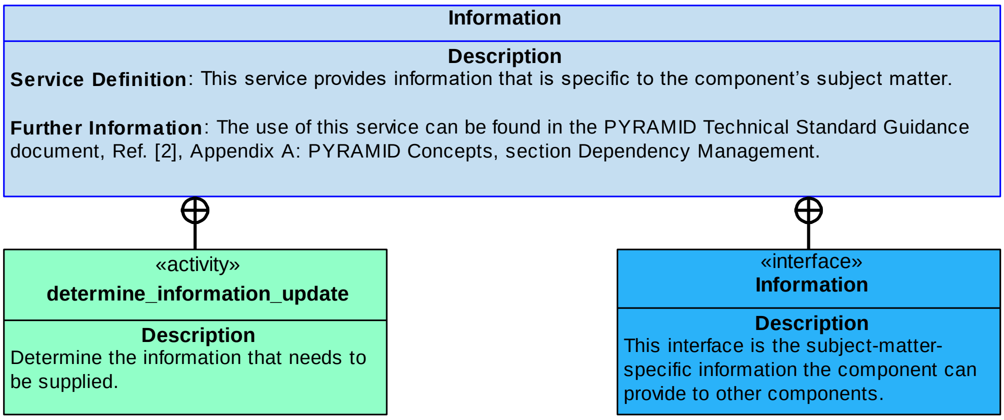

Information
Description
Service Definition: This service provides information that is specific to the component's subject matter.
Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.
«activity»
determine_information_update
Description
Determine the information that needs to be supplied.
«interface»
Information
Description
This interface is the subject-matter-specific information the component can provide to other components.

Figure 21: Information Service Policy

Information

Service Definition: This service provides information that is specific to the component's subject matter.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Interface

Information

This interface is the subject-matter-specific information the component can provide to other components.

Activity

determine_information_update

Determine the information that needs to be supplied.

5.4.1.3.2.1.1.3 Constraint

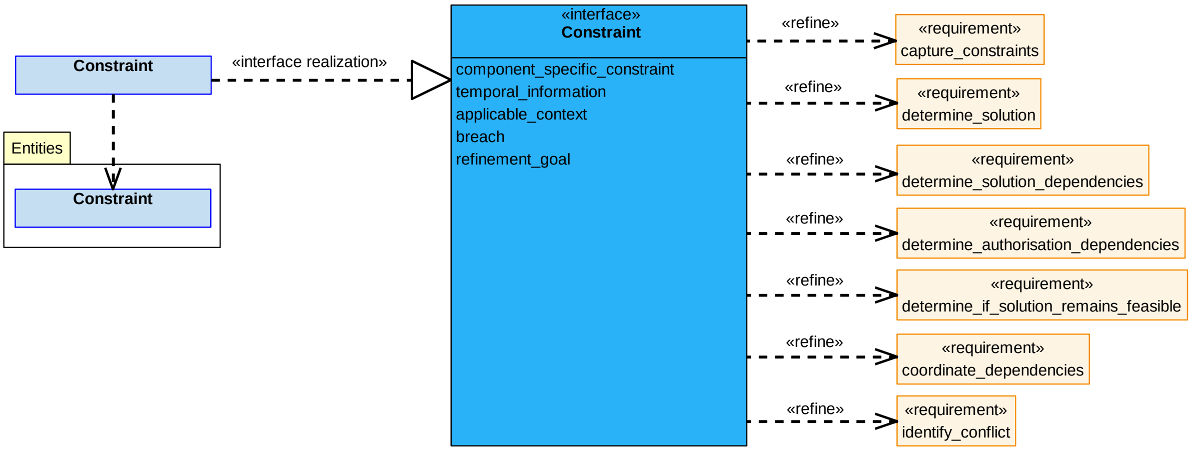
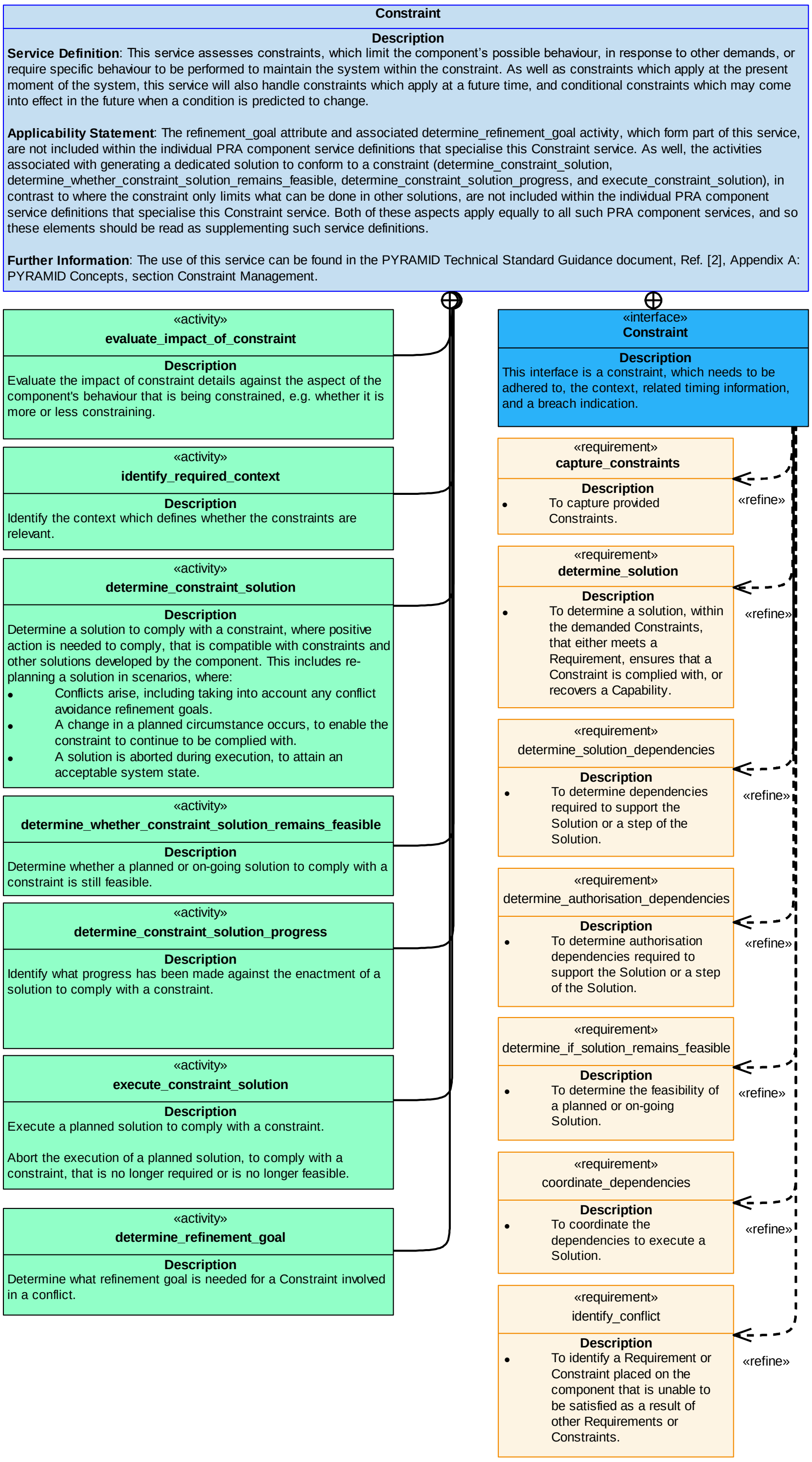


Figure 22: Constraint Service Definition



Constraint

Description

Service Definition: This service assesses constraints, which limit the component's possible behaviour, in response to other demands, or require specific behaviour to be performed to maintain the system within the constraint. As well as constraints which apply at the present moment of the system, this service will also handle constraints which apply at a future time, and conditional constraints which may come into effect in the future when a condition is predicted to change.

Applicability Statement: The refinement_goal attribute and associated determine_refinement_goal activity, which form part of this service, are not included within the individual PRA component service definitions that specialise this Constraint service. As well, the activities associated with generating a dedicated solution to conform to a constraint (determine_constraint_solution, determine_whether_constraint_solution_remains_feasible, determine_constraint_solution_progress, and execute_constraint_solution), in contrast to where the constraint only limits what can be done in other solutions, are not included within the individual PRA component service definitions that specialise this Constraint service. Both of these aspects apply equally to all such PRA component services, and so these elements should be read as supplementing such service definitions.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Constraint Management.

«activity»
evaluate_impact_of_constraint

Description

Evaluate the impact of constraint details against the aspect of the component's behaviour that is being constrained, e.g. whether it is more or less constraining.

«activity»
identify_required_context

Description

Identify the context which defines whether the constraints are relevant.

«activity»
determine_constraint_solution

Description

Determine a solution to comply with a constraint, where positive action is needed to comply, that is compatible with constraints and other solutions developed by the component. This includes re-planning a solution in scenarios, where:

- Conflicts arise, including taking into account any conflict avoidance refinement goals.
- A change in a planned circumstance occurs, to enable the constraint to continue to be complied with.
- A solution is aborted during execution, to attain an acceptable system state.

«activity»
determine_whether_constraint_solution_remains_feasible

Description

Determine whether a planned or on-going solution to comply with a constraint is still feasible.

«activity»
determine_constraint_solution_progress

Description

Identify what progress has been made against the enactment of a solution to comply with a constraint.

«activity»
execute_constraint_solution

Description

Execute a planned solution to comply with a constraint.

Abort the execution of a planned solution, to comply with a constraint, that is no longer required or is no longer feasible.

«activity»
determine_refinement_goal

Description

Determine what refinement goal is needed for a Constraint involved in a conflict.

«interface»
Constraint

Description

This interface is a constraint, which needs to be adhered to, the context, related timing information, and a breach indication.

«requirement»
capture_constraints

Description

- To capture provided Constraints.

«requirement»
determine_solution

Description

- To determine a solution, within the demanded Constraints, that either meets a Requirement, ensures that a Constraint is complied with, or recovers a Capability.

«requirement»
determine_solution_dependencies

Description

- To determine dependencies required to support the Solution or a step of the Solution.

«requirement»
determine_authorisation_dependencies

Description

- To determine authorisation dependencies required to support the Solution or a step of the Solution.

«requirement»
determine_if_solution_remains_feasible

Description

- To determine the feasibility of a planned or on-going Solution.

«requirement»
coordinate_dependencies

Description

- To coordinate the dependencies to execute a Solution.

«requirement»
identify_conflict

Description

- To identify a Requirement or Constraint placed on the component that is unable to be satisfied as a result of other Requirements or Constraints.

Figure 23: Constraint Service Policy

Constraint

Service Definition: This service assesses constraints, which limit the component's possible behaviour, in response to other demands, or require specific behaviour to be performed to maintain the system within the constraint. As well as constraints which apply at the present moment of the system, this service will also handle constraints which apply at a future time, and conditional constraints which may come into effect in the future when a condition is predicted to change.

Applicability Statement: The refinement_goal attribute and associated determine_refinement_goal activity, which form part of this service, are not included within the individual PRA component service definitions that specialise this Constraint service. As well, the activities associated with generating a dedicated solution to conform to a constraint (determine_constraint_solution, determine_whether_constraint_solution_remains_feasible, determine_constraint_solution_progress, and execute_constraint_solution), in contrast to where the constraint only limits what can be done in other solutions, are not included within the individual PRA component service definitions that specialise this Constraint service. Both of these aspects apply equally to all such PRA component services, and so these elements should be read as supplementing such service definitions.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Constraint Management.

Interface

Constraint

This interface is a constraint, which needs to be adhered to, the context, related timing information, and a breach indication.

Attributes

component_specific_constraint	A constraint that impacts the component's behaviour. This will be something that is of its subject matter and which the component is inherently aware of regardless of solutions or rules, e.g. transmission restrictions. This constraint could come from either a rule or a solution of another component, although from the component's perspective it treats both rule-based and solution-based constraints in the same manner.
temporal_information	Timing information pertaining to the periods of time when the constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the constraint is applicable, e.g. spatial zones in which the constraint applies.
breach	A statement that the constraint has been breached, or is likely to be breached if enforced.
refinement_goal	The specific aspects of the constraint that need to be modified and how they need to be modified.

Activities

evaluate_impact_of_constraint

Evaluate the impact of constraint details against the aspect of the component's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the constraints are relevant.

determine_constraint_solution

Determine a solution to comply with a constraint, where positive action is needed to comply, that is compatible with constraints and other solutions developed by the component. This includes re-planning a solution in scenarios, where:

- Conflicts arise, including taking into account any conflict avoidance refinement goals.
- A change in a planned circumstance occurs, to enable the constraint to continue to be complied with.
- A solution is aborted during execution, to attain an acceptable system state.

determine_whether_constraint_solution_remains_feasible

Determine whether a planned or on-going solution to comply with a constraint is still feasible.

determine_constraint_solution_progress

Identify what progress has been made against the enactment of a solution to comply with a constraint.

execute_constraint_solution

Execute a planned solution to comply with a constraint.

Abort the execution of a planned solution, to comply with a constraint, that is no longer required or is no longer feasible.

determine_refinement_goal

Determine what refinement goal is needed for a Constraint involved in a conflict.

5.4.1.3.2.1.1.4 Capability

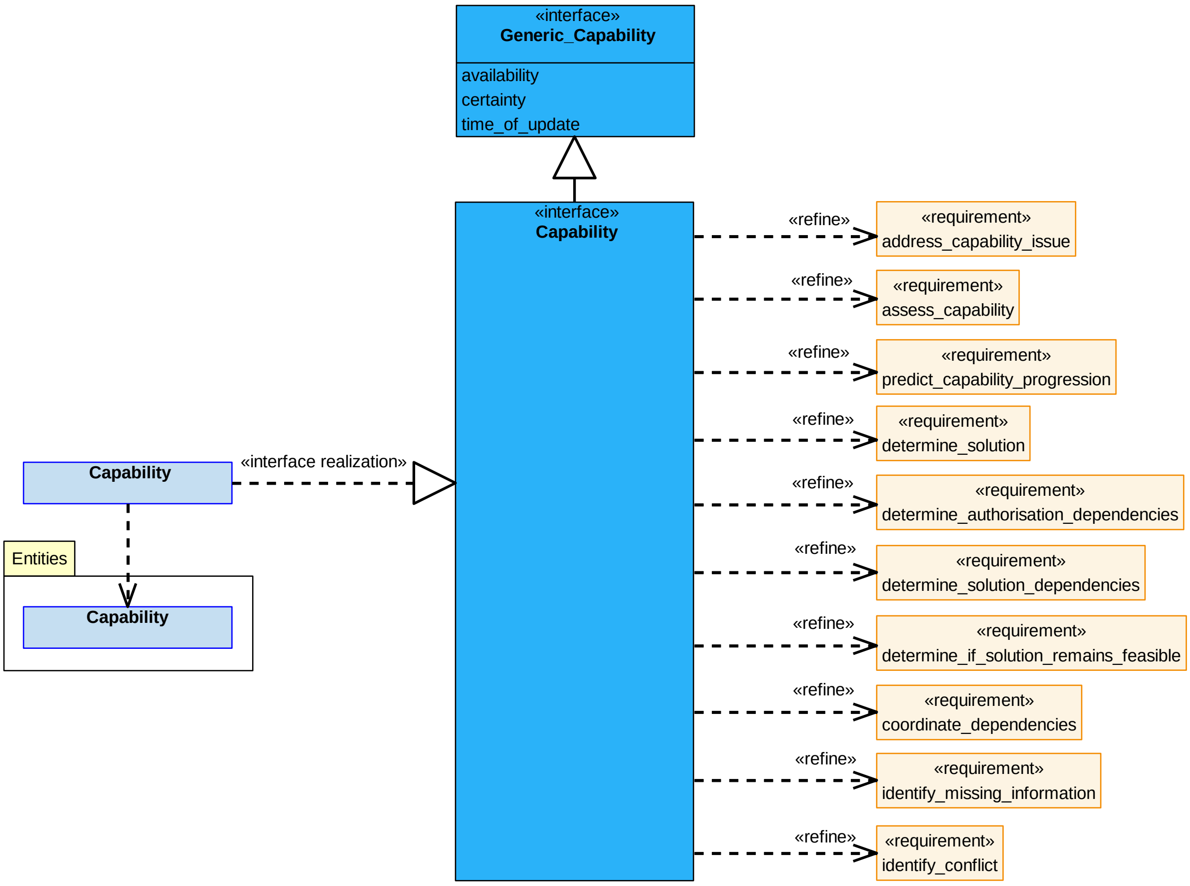
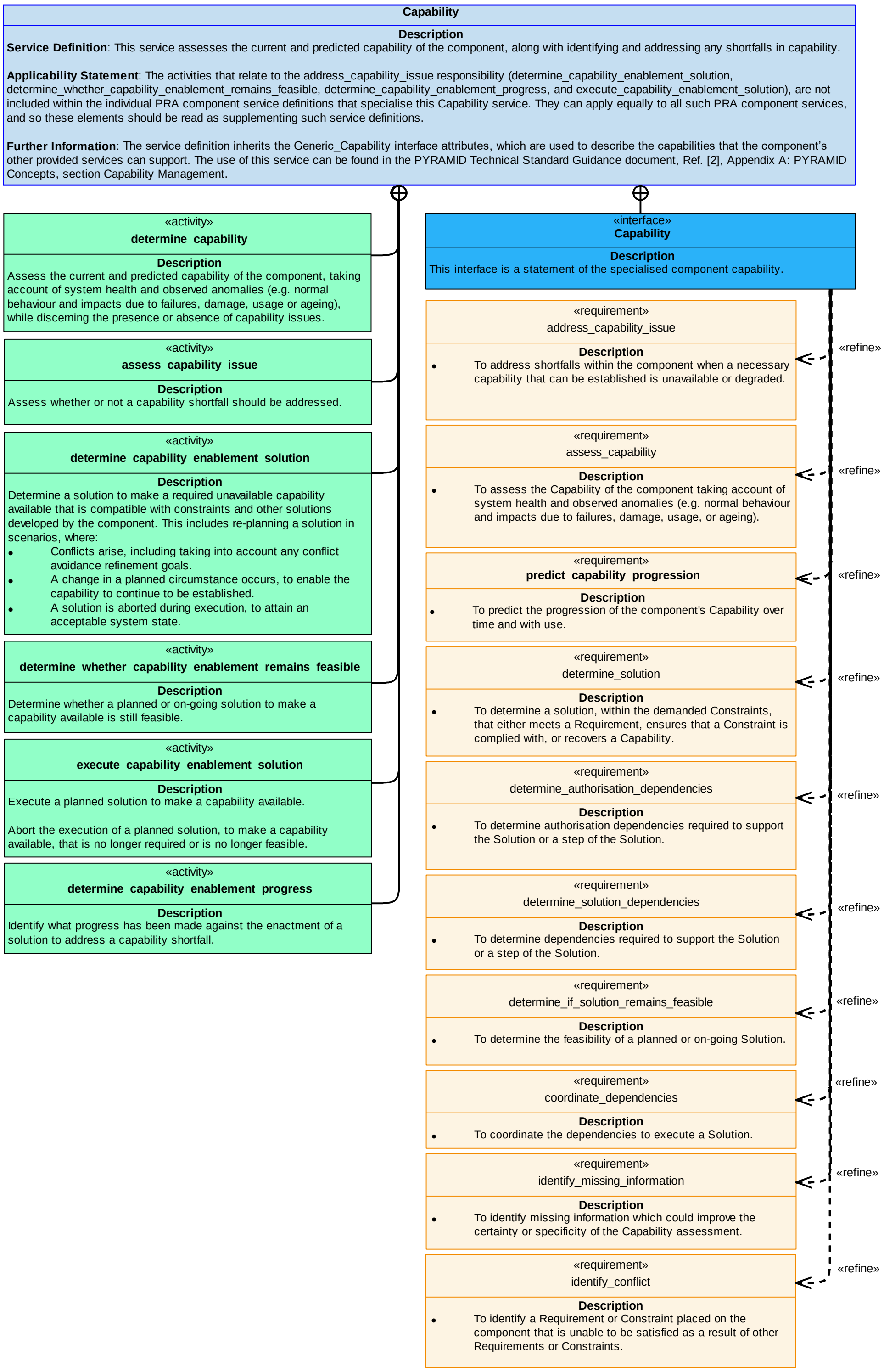


Figure 24: Capability Service Definition



Capability

Description

Service Definition: This service assesses the current and predicted capability of the component, along with identifying and addressing any shortfalls in capability.

Applicability Statement: The activities that relate to the address_capability_issue responsibility (determine_capability_enablement_solution, determine_whether_capability_enablement_remains_feasible, determine_capability_enablement_progress, and execute_capability_enablement_solution), are not included within the individual PRA component service definitions that specialise this Capability service. They can apply equally to all such PRA component services, and so these elements should be read as supplementing such service definitions.

Further Information: The service definition inherits the Generic_Capability interface attributes, which are used to describe the capabilities that the component's other provided services can support. The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Capability Management.

«activity» determine_capability

Description

Assess the current and predicted capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing), while discerning the presence or absence of capability issues.

«activity» assess_capability_issue

Description

Assess whether or not a capability shortfall should be addressed.

«activity» determine_capability_enablement_solution

Description

Determine a solution to make a required unavailable capability available that is compatible with constraints and other solutions developed by the component. This includes re-planning a solution in scenarios, where:

- Conflicts arise, including taking into account any conflict avoidance refinement goals.
- A change in a planned circumstance occurs, to enable the capability to continue to be established.
- A solution is aborted during execution, to attain an acceptable system state.

«activity» determine_whether_capability_enablement_remains_feasible

Description

Determine whether a planned or on-going solution to make a capability available is still feasible.

«activity» execute_capability_enablement_solution

Description

Execute a planned solution to make a capability available.

Abort the execution of a planned solution, to make a capability available, that is no longer required or is no longer feasible.

«activity» determine_capability_enablement_progress

Description

Identify what progress has been made against the enactment of a solution to address a capability shortfall.

«interface» Capability

Description

This interface is a statement of the specialised component capability.

«requirement» address_capability_issue

Description

- To address shortfalls within the component when a necessary capability that can be established is unavailable or degraded.

«requirement» assess_capability

Description

- To assess the Capability of the component taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage, or ageing).

«requirement» predict_capability_progression

Description

- To predict the progression of the component's Capability over time and with use.

«requirement» determine_solution

Description

- To determine a solution, within the demanded Constraints, that either meets a Requirement, ensures that a Constraint is complied with, or recovers a Capability.

«requirement» determine_authorisation_dependencies

Description

- To determine authorisation dependencies required to support the Solution or a step of the Solution.

«requirement» determine_solution_dependencies

Description

- To determine dependencies required to support the Solution or a step of the Solution.

«requirement» determine_if_solution_remains_feasible

Description

- To determine the feasibility of a planned or on-going Solution.

«requirement» coordinate_dependencies

Description

- To coordinate the dependencies to execute a Solution.

«requirement» identify_missing_information

Description

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

«requirement» identify_conflict

Description

- To identify a Requirement or Constraint placed on the component that is unable to be satisfied as a result of other Requirements or Constraints.

Figure 25: Capability Service Policy

Capability

Service Definition: This service assesses the current and predicted capability of the component, along with identifying and addressing any shortfalls in capability.

Applicability Statement: The activities that relate to the address_capability_issue responsibility (determine_capability_enablement_solution, determine_whether_capability_enablement_remains_feasible, determine_capability_enablement_progress, and execute_capability_enablement_solution), are not included within the individual PRA component service definitions that specialise this Capability service. They can apply equally to all such PRA component services, and so these elements should be read as supplementing such service definitions.

Further Information: The service definition inherits the Generic_Capability interface attributes, which are used to describe the capabilities that the component's other provided services can support. The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Capability Management.

Interface

Capability

This interface is a statement of the specialised component capability.

Activities

determine_capability

Assess the current and predicted capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing), while discerning the presence or absence of capability issues.

assess_capability_issue

Assess whether or not a capability shortfall should be addressed.

determine_capability_enablement_solution

Determine a solution to make a required unavailable capability available that is compatible with constraints and other solutions developed by the component. This includes re-planning a solution in scenarios, where:

- Conflicts arise, including taking into account any conflict avoidance refinement goals.
- A change in a planned circumstance occurs, to enable the capability to continue to be established.
- A solution is aborted during execution, to attain an acceptable system state.

determine_whether_capability_enablement_remains_feasible

Determine whether a planned or on-going solution to make a capability available is still feasible.

execute_capability_enablement_solution

Execute a planned solution to make a capability available.

Abort the execution of a planned solution, to make a capability available, that is no longer required or is no longer feasible.

determine_capability_enablement_progress

Identify what progress has been made against the enactment of a solution to address a capability shortfall.

5.4.1.3.2.1.2 Consumed Services

5.4.1.3.2.1.2.1 Solution_Dependency

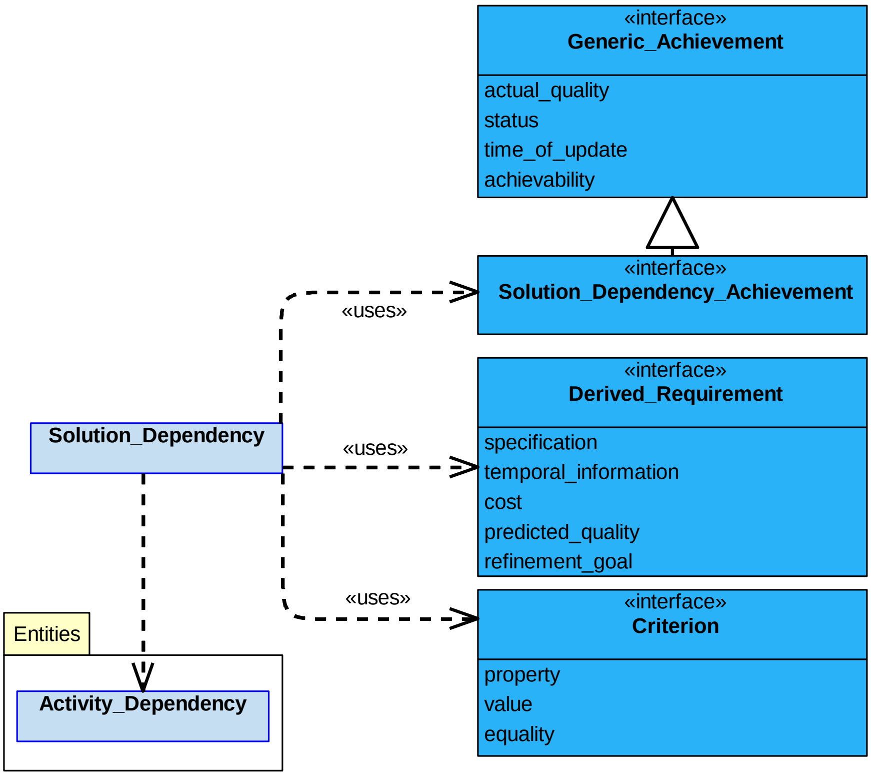
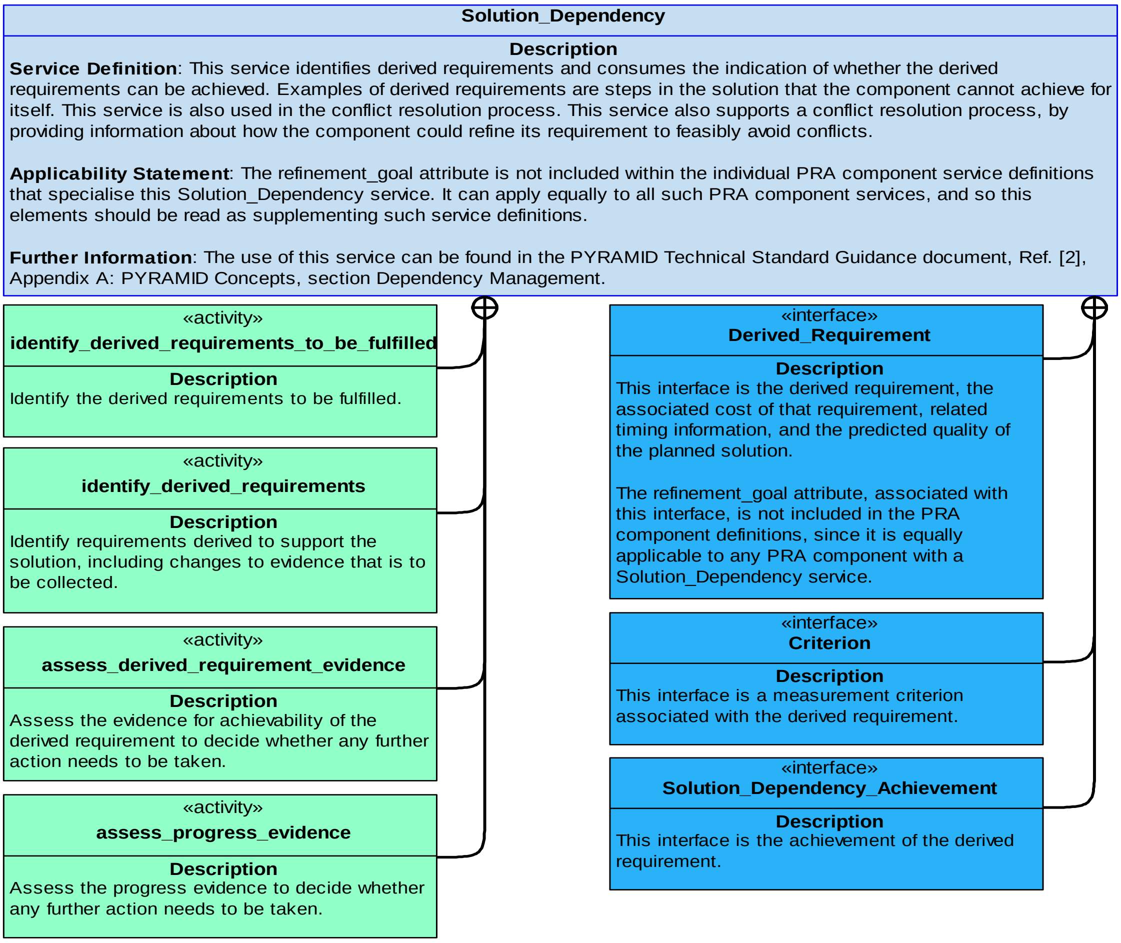


Figure 26: Solution_Dependency Service Definition



Service Definition: This service identifies derived requirements and consumes the indication of whether the derived requirements can be achieved. Examples of derived requirements are steps in the solution that the component cannot achieve for itself. This service is also used in the conflict resolution process. This service also supports a conflict resolution process, by providing information about how the component could refine its requirement to feasibly avoid conflicts.

Applicability Statement: The refinement_goal attribute is not included within the individual PRA component service definitions that specialise this Solution_Dependency service. It can apply equally to all such PRA component services, and so this elements should be read as supplementing such service definitions.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Identify the derived requirements to be fulfilled.

Identify requirements derived to support the solution, including changes to evidence that is to be collected.

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

Assess the progress evidence to decide whether any further action needs to be taken.

This interface is the derived requirement, the associated cost of that requirement, related timing information, and the predicted quality of the planned solution.

The refinement_goal attribute, associated with this interface, is not included in the PRA component definitions, since it is equally applicable to any PRA component with a Solution_Dependency service.

This interface is a measurement criterion associated with the derived requirement.

This interface is the achievement of the derived requirement.

Figure 27: Solution_Dependency Service Policy

Solution_Dependency

Service Definition: This service identifies derived requirements and consumes the indication of whether the derived requirements can be achieved. Examples of derived requirements are steps in the solution that the component cannot achieve for itself. This service is also used in the conflict resolution process. This service also supports a conflict resolution process, by providing information about how the component could refine its requirement to feasibly avoid conflicts.

Applicability Statement: The refinement_goal attribute is not included within the individual PRA component service definitions that specialise this Solution_Dependency service. It can apply equally to all such PRA component services, and so this elements should be read as supplementing such service definitions.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Interfaces

Derived_Requirement

This interface is the derived requirement, the associated cost of that requirement, related timing information, and the predicted quality of the planned solution.

The refinement_goal attribute, associated with this interface, is not included in the PRA component definitions, since it is equally applicable to any PRA component with a Solution_Dependency service.

Attributes

specification	The definition of the derived requirement.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used, time taken.
predicted_quality	How well the planned solution is predicted to satisfy the requirement.
refinement_goal	The specific aspects of the requirement that need to be modified and how they need to be modified.

Criterion

This interface is a measurement criterion associated with the derived requirement.

Attributes

property	The property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Solution_Dependency_Achievement

This interface is the achievement of the derived requirement.

Activities

identify_derived_requirements_to_be_fulfilled

Identify the derived requirements to be fulfilled.

identify_derived_requirements

Identify requirements derived to support the solution, including changes to evidence that is to be collected.

assess_derived_requirement_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

5.4.1.3.2.1.2.2 Information_Dependency

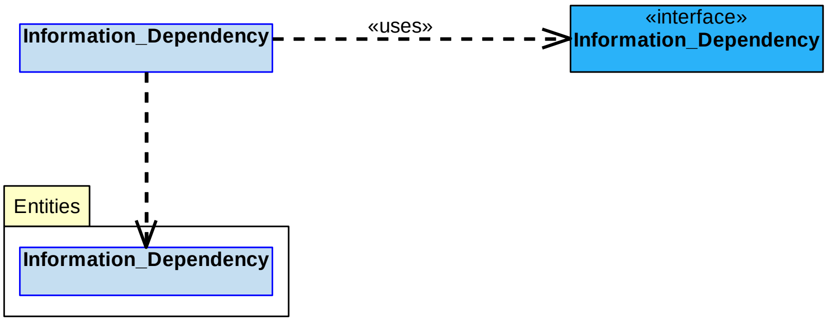


Figure 28: Information_Dependency Service Definition

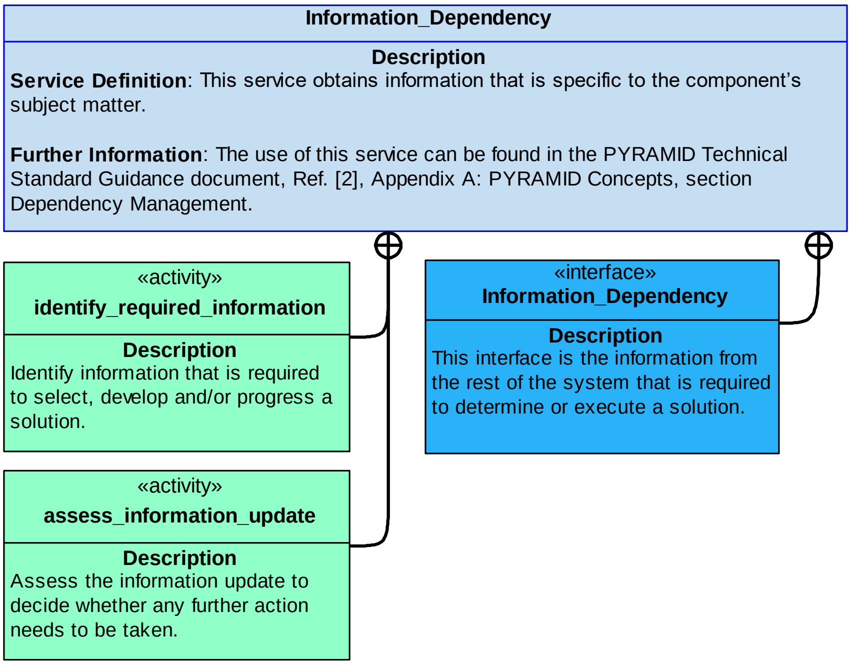

Information_Dependency
Description
Service Definition: This service obtains information that is specific to the component's subject matter.
Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.
«activity»
identify_required_information
Description
Identify information that is required to select, develop and/or progress a solution.
«interface»
Information_Dependency
Description
This interface is the information from the rest of the system that is required to determine or execute a solution.
«activity»
assess_information_update
Description
Assess the information update to decide whether any further action needs to be taken.

Figure 29: Information_Dependency Service Policy

Information_Dependency

Service Definition: This service obtains information that is specific to the component's subject matter.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Interface

Information_Dependency

This interface is the information from the rest of the system that is required to determine or execute a solution.

Activities

identify_required_information

Identify information that is required to select, develop and/or progress a solution.

assess_information_update

Assess the information update to decide whether any further action needs to be taken.

5.4.1.3.2.1.2.3 Capability_Evidence

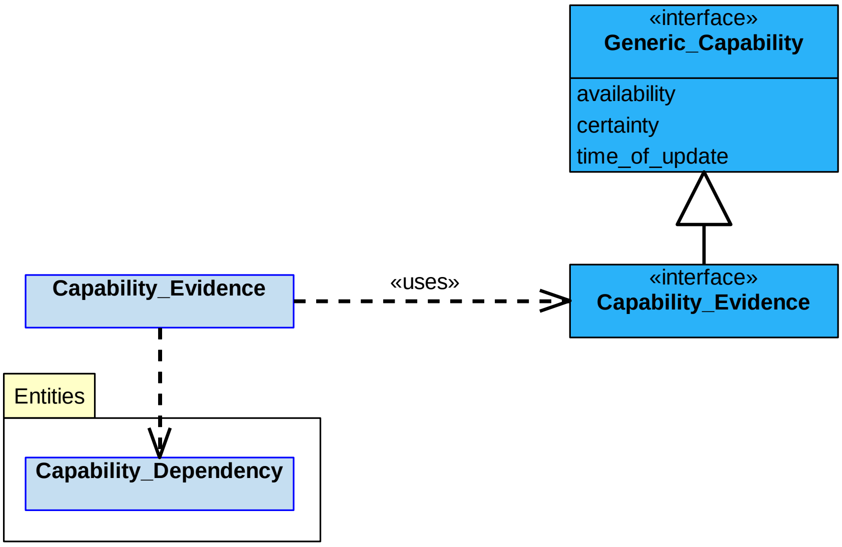
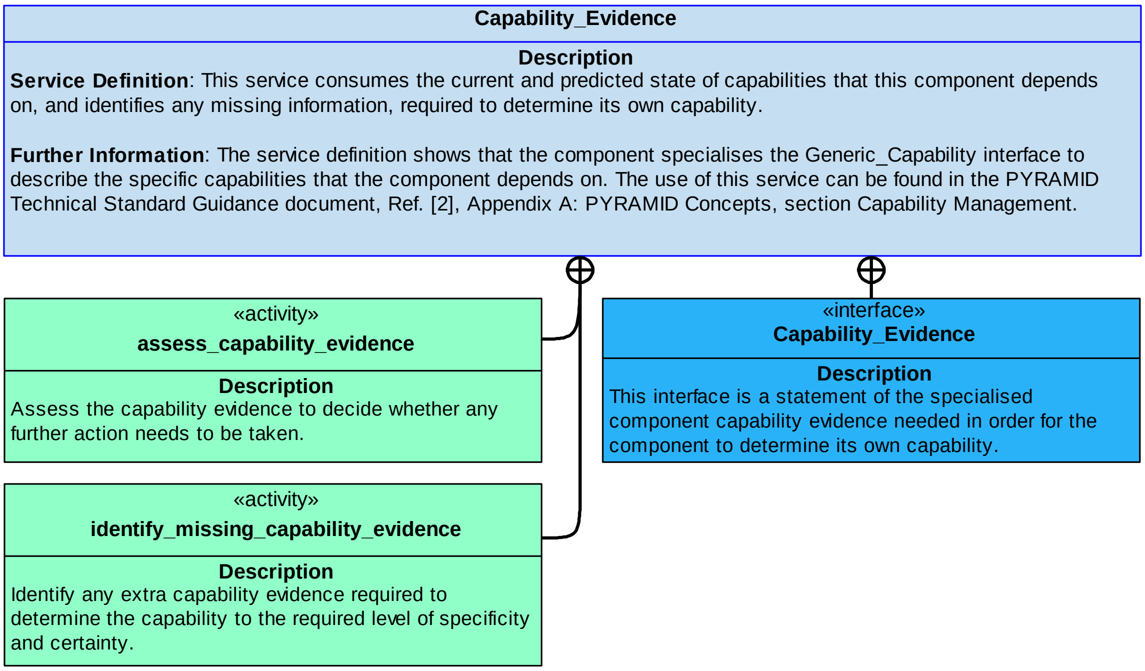


Figure 30: Capability_Evidence Service Definition



Service Definition: This service consumes the current and predicted state of capabilities that this component depends on, and identifies any missing information, required to determine its own capability.

Further Information: The service definition shows that the component specialises the Generic_Capability interface to describe the specific capabilities that the component depends on. The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Capability Management.

Description
Assess the capability evidence to decide whether any further action needs to be taken.

Description
This interface is a statement of the specialised component capability evidence needed in order for the component to determine its own capability.

Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

Figure 31: Capability_Evidence Service Policy

Capability_Evidence

Service Definition: This service consumes the current and predicted state of capabilities that this component depends on, and identifies any missing information, required to determine its own capability.

Further Information: The service definition shows that the component specialises the Generic_Capability interface to describe the specific capabilities that the component depends on. The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Capability Management.

Interface

Capability_Evidence

This interface is a statement of the specialised component capability evidence needed in order for the component to determine its own capability.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.1.3.2.1.3 Generic Interfaces

Each component in the PRA represents a discrete area of subject matter and it models the associated data in terms appropriate to the subject matter. Components do not share a common understanding of their data and so the PRA does not include a shared data model. Bridges are used (see PYRAMID Technical Standard Guidance document, Ref. [2], section Component Connections) to close the semantic gap and to map and translate between component services.

However, in some limited areas it is helpful to derive data from a generic interface so that components can communicate some categories of data in a uniform way. The following interfaces are generic and the attributes are inherited by specialised versions of the achievement, Capability, and Capability_Evidence interfaces. These interfaces are inherited both within the Component Composition and within the Component Set.

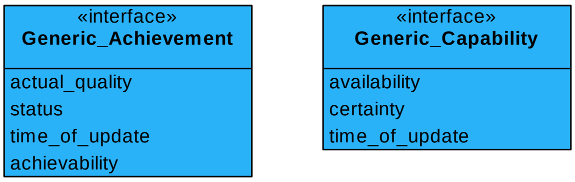


Figure 32: Generic Interfaces

Generic_Achievement

This interface is the statement of achievement, or achievability against a requirement.

Attributes

actual_quality	How well the deliverables are satisfying the requirement.
status	A high-level representation of achievement in relation to the requirement (e.g. not started, in progress, or complete).
time_of_update	The time at which an achievement update occurred.
achievability	Whether the requirement is expected to be able to be met.

Generic_Capability

This interface is the generic statement of capability.

Attributes

availability	Whether the capability is available.
certainty	The level of certainty of the operational status.
time_of_update	The time at which the availability was updated.

5.4.1.3.2.2 Service Definitions that are Not specialised in PRA Components

This section defines generic services, which apply to most PRA components to complete their component definitions. These services do not lend themselves to being specialised by each PRA component as the component subject matter specific definitions at the level of the PRA do not influence their generic definitions, with the exception of the Constraint_Dependency service in the specific cases detailed in its service description. An Exploiter should consider these generic services as supplementing the list of services in any given PRA component definition in order to complete the definition of the PRA component.

5.4.1.3.2.2.1 Provided Services

5.4.1.3.2.2.1.1 Retention_Requirement

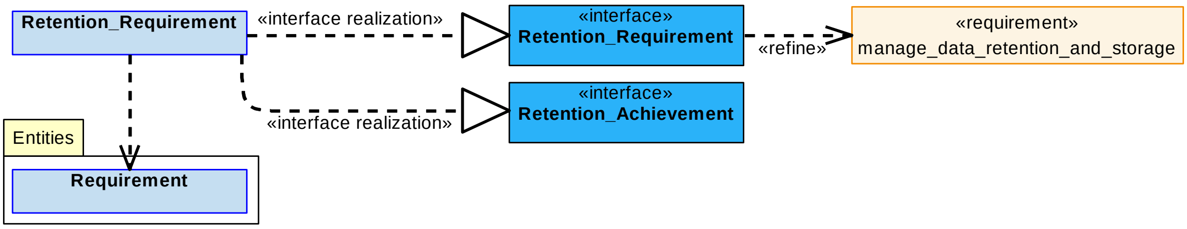
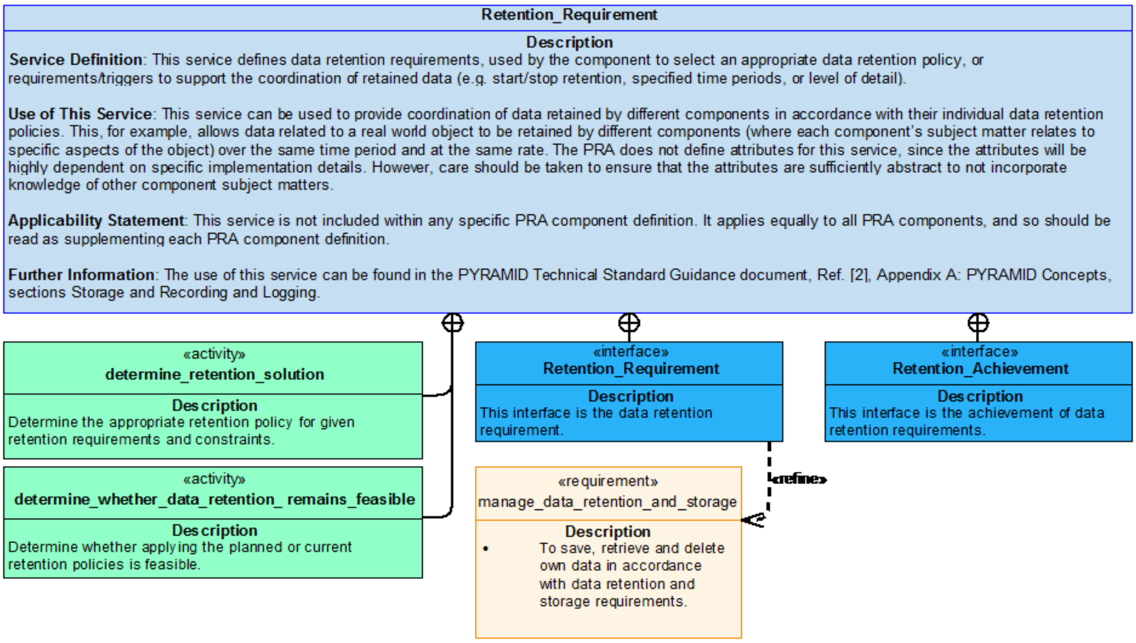


Figure 33: Retention Requirement Service Definition



Service Definition: This service defines data retention requirements, used by the component to select an appropriate data retention policy, or requirements/triggers to support the coordination of retained data (e.g. start/stop retention, specified time periods, or level of detail).

Use of This Service: This service can be used to provide coordination of data retained by different components in accordance with their individual data retention policies. This, for example, allows data related to a real world object to be retained by different components (where each component's subject matter relates to specific aspects of the object) over the same time period and at the same rate. The PRA does not define attributes for this service, since the attributes will be highly dependent on specific implementation details. However, care should be taken to ensure that the attributes are sufficiently abstract to not incorporate knowledge of other component subject matters.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, sections Storage and Recording and Logging.

Determine the appropriate retention policy for given retention requirements and constraints.

Determine whether applying the planned or current retention policies is feasible.

This interface is the data retention requirement.

This interface is the achievement of data retention requirements.

- To save, retrieve and delete own data in accordance with data retention and storage requirements.

Figure 34: Retention_Requirement Service Policy

Retention_Requirement

Service Definition: This service defines data retention requirements, used by the component to select an appropriate data retention policy, or requirements/triggers to support the coordination of retained data (e.g. start/stop retention, specified time periods, or level of detail).

Use of This Service: This service can be used to provide coordination of data retained by different components in accordance with their individual data retention policies. This, for example, allows data related to a real world object to be retained by different components (where each component's subject matter relates to specific aspects of the object) over the same time period and at the same rate.

The PRA does not define attributes for this service, since the attributes will be highly dependent on specific implementation details. However, care should be taken to ensure that the attributes are sufficiently abstract to not incorporate knowledge of other component subject matters.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, sections Storage and Recording and Logging.

Interfaces

Retention_Requirement

This interface is the data retention requirement.

Retention_Achievement

This interface is the achievement of data retention requirements.

Activities

determine_retention_solution

Determine the appropriate retention policy for given retention requirements and constraints.

determine_whether_data_retention_ remains_feasible

Determine whether applying the planned or current retention policies is feasible.

5.4.1.3.2.2.1.2 Dependency_Refinement

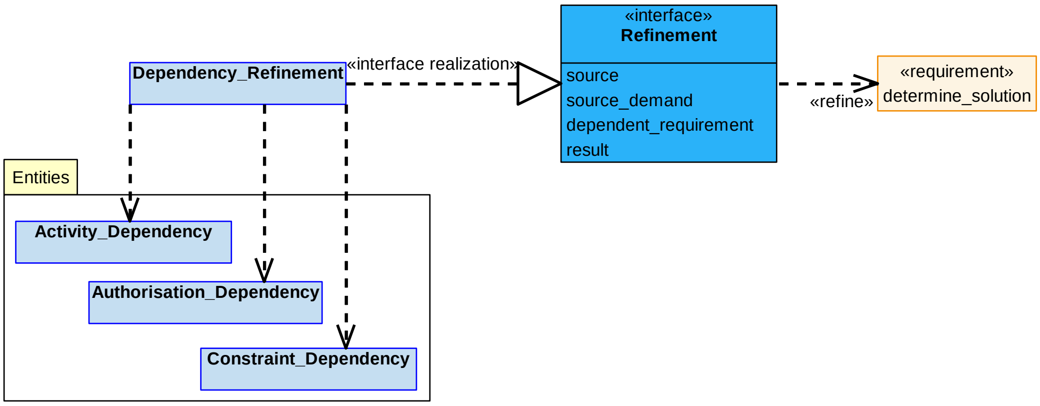
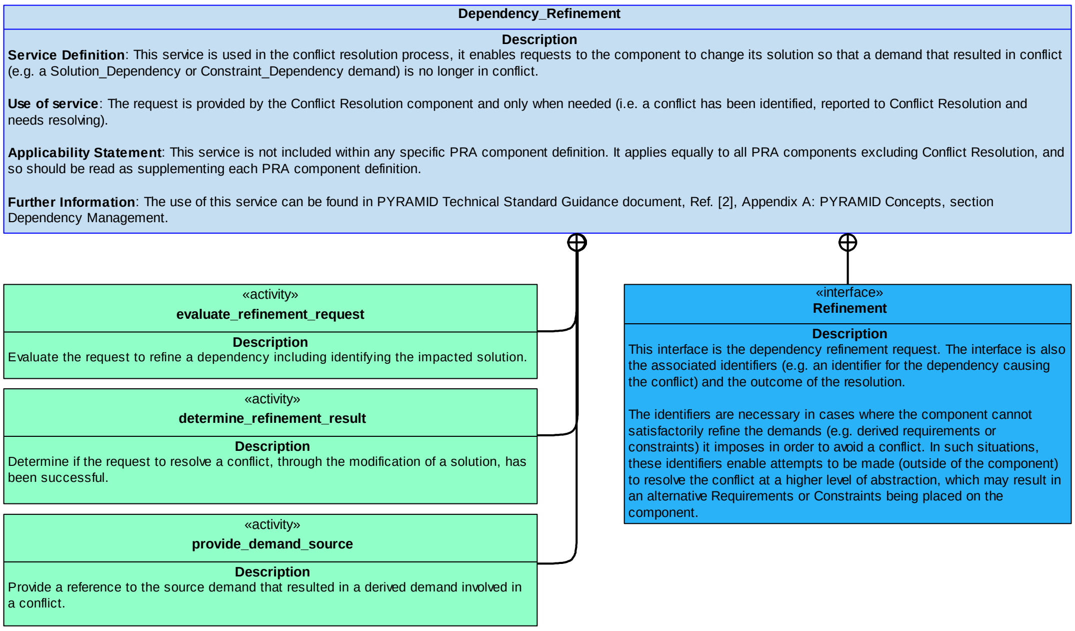


Figure 35: Dependency_Refinement Service Definition



Service Definition: This service is used in the conflict resolution process, it enables requests to the component to change its solution so that a demand that resulted in conflict (e.g. a Solution_Dependency or Constraint_Dependency demand) is no longer in conflict.

Use of service: The request is provided by the Conflict Resolution component and only when needed (i.e. a conflict has been identified, reported to Conflict Resolution and needs resolving).

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components excluding Conflict Resolution, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Evaluate the request to refine a dependency including identifying the impacted solution.

Determine if the request to resolve a conflict, through the modification of a solution, has been successful.

Provide a reference to the source demand that resulted in a derived demand involved in a conflict.

This interface is the dependency refinement request. The interface is also the associated identifiers (e.g. an identifier for the dependency causing the conflict) and the outcome of the resolution.

The identifiers are necessary in cases where the component cannot satisfactorily refine the demands (e.g. derived requirements or constraints) it imposes in order to avoid a conflict. In such situations, these identifiers enable attempts to be made (outside of the component) to resolve the conflict at a higher level of abstraction, which may result in an alternative Requirements or Constraints being placed on the component.

Figure 36: Dependency_Refinement Service Policy

Dependency_Refinement

Service Definition: This service is used in the conflict resolution process, it enables requests to the component to change its solution so that a demand that resulted in conflict (e.g. a Solution_Dependency or Constraint_Dependency demand) is no longer in conflict.

Use of service: The request is provided by the Conflict Resolution component and only when needed (i.e. a conflict has been identified, reported to Conflict Resolution and needs resolving).

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components excluding Conflict Resolution, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Interface

Refinement

This interface is the dependency refinement request. The interface is also the associated identifiers (e.g. an identifier for the dependency causing the conflict) and the outcome of the resolution.

The identifiers are necessary in cases where the component cannot satisfactorily refine the demands (e.g. derived requirements or constraints) it imposes in order to avoid a conflict. In such situations, these identifiers enable attempts to be made (outside of the component) to resolve the conflict at a higher level of abstraction, which may result in an alternative Requirements or Constraints being placed on the component.

Attributes

source	The source of the demand, placed on this component, relating to the conflict that needs to be resolved. (This is provided to the Conflict Resolution component). Note: Depending on how traceability is managed within a deployment this attribute may not be needed.
source_demand	The actual demand, placed on this component, relating to the conflict that needs to be resolved. (This is provided to the Conflict Resolution component). Note: Depending on how traceability is managed within a deployment this attribute may not be needed.
dependent_requirement	The identifier for the Solution_Dependency or Constraint_Dependency demand, placed by this component, that resulted in a conflict. (This is provided by the Conflict Resolution component).
result	The result of the dependency refinement, e.g. refinement goal cannot be met. (This is received by the Conflict Resolution component).

Activities

evaluate_refinement_request

Evaluate the request to refine a dependency including identifying the impacted solution.

determine_refinement_result

Determine if the request to resolve a conflict, through the modification of a solution, has been successful.

provide_demand_source

Provide a reference to the source demand that resulted in a derived demand involved in a conflict.

5.4.1.3.2.2.2 Consumed Services

5.4.1.3.2.2.2.1 Constraint_Dependency

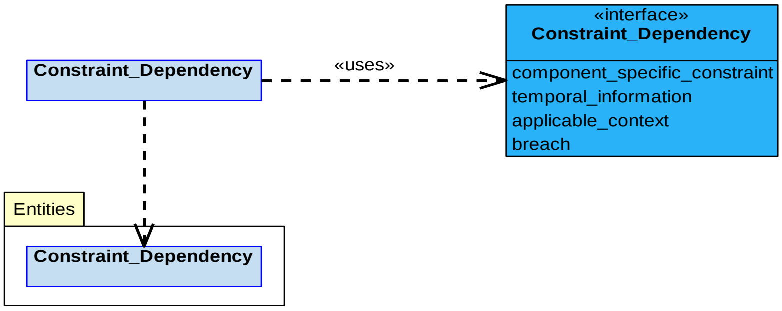
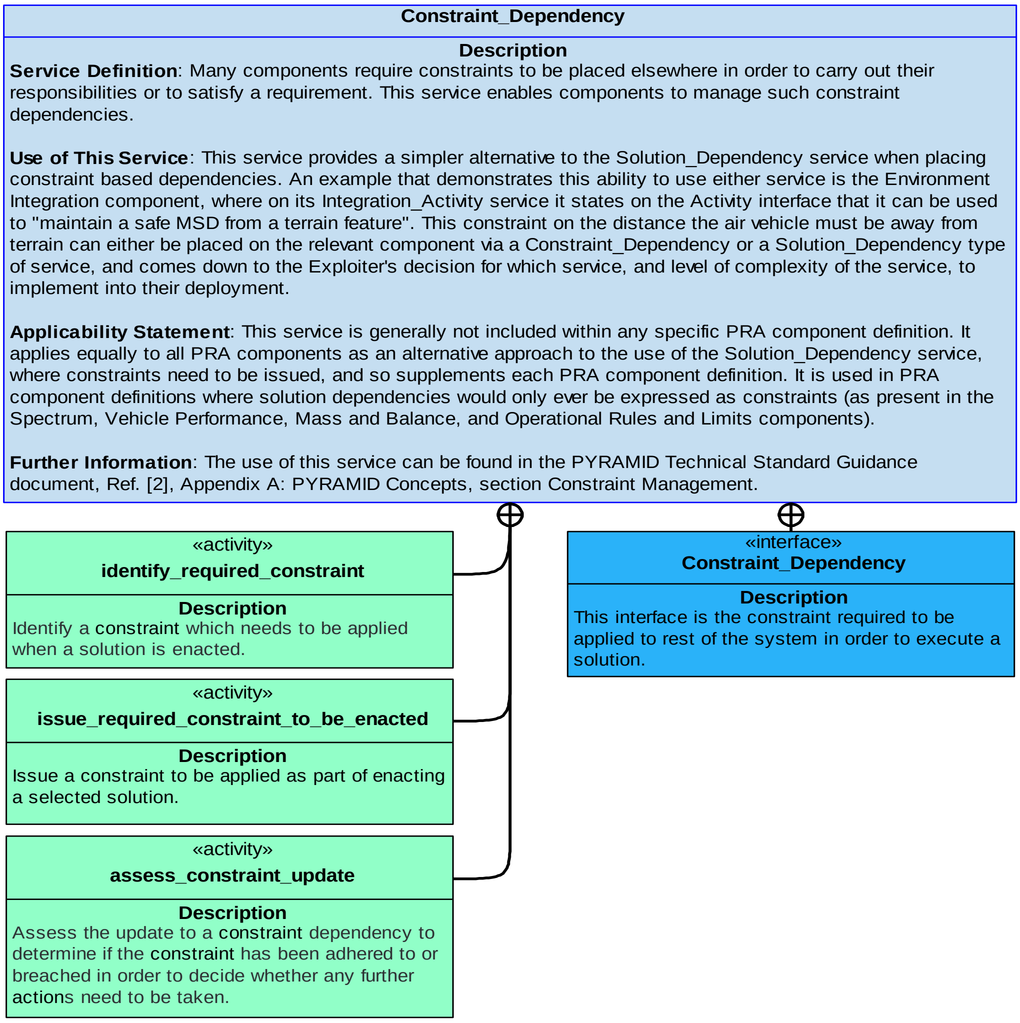


Figure 37: Constraint_Dependency Service Definition



Service Definition: Many components require constraints to be placed elsewhere in order to carry out their responsibilities or to satisfy a requirement. This service enables components to manage such constraint dependencies.

Use of This Service: This service provides a simpler alternative to the Solution_Dependency service when placing constraint based dependencies. An example that demonstrates this ability to use either service is the Environment Integration component, where on its Integration_Activity service it states on the Activity interface that it can be used to "maintain a safe MSD from a terrain feature". This constraint on the distance the air vehicle must be away from terrain can either be placed on the relevant component via a Constraint_Dependency or a Solution_Dependency type of service, and comes down to the Exploiter's decision for which service, and level of complexity of the service, to implement into their deployment.

Applicability Statement: This service is generally not included within any specific PRA component definition. It applies equally to all PRA components as an alternative approach to the use of the Solution_Dependency service, where constraints need to be issued, and so supplements each PRA component definition. It is used in PRA component definitions where solution dependencies would only ever be expressed as constraints (as present in the Spectrum, Vehicle Performance, Mass and Balance, and Operational Rules and Limits components).

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Constraint Management.

identify_required_constraint — Description: Identify a constraint which needs to be applied when a solution is enacted.

issue_required_constraint_to_be_enacted — Description: Issue a constraint to be applied as part of enacting a selected solution.

assess_constraint_update — Description: Assess the update to a constraint dependency to determine if the constraint has been adhered to or breached in order to decide whether any further actions need to be taken.

Constraint_Dependency — Description: This interface is the constraint required to be applied to rest of the system in order to execute a solution.

Figure 38: Constraint_Dependency Service Policy

Constraint_Dependency

Service Definition: Many components require constraints to be placed elsewhere in order to carry out their responsibilities or to satisfy a requirement. This service enables components to manage such constraint dependencies.

Use of This Service: This service provides a simpler alternative to the Solution_Dependency service when placing constraint based dependencies. An example that demonstrates this ability to use either service is the Environment Integration component, where on its Integration_Activity service it states on the Activity interface that it can be used to "maintain a safe MSD from a terrain feature". This constraint on the distance the air vehicle must be away from terrain can either be placed on the relevant component via a Constraint_Dependency or a Solution_Dependency type of service, and comes down to the Exploiter's decision for which service, and level of complexity of the service, to implement into their deployment.

Applicability Statement: This service is generally not included within any specific PRA component definition. It applies equally to all PRA components as an alternative approach to the use of the Solution_Dependency service, where constraints need to be issued, and so supplements each PRA component definition. It is used in PRA component definitions where solution dependencies would only ever be expressed as constraints (as present in the Spectrum, Vehicle Performance, Mass and Balance, and Operational Rules and Limits components).

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Constraint Management.

Interface

Constraint_Dependency

This interface is the constraint required to be applied to rest of the system in order to execute a solution.

Attributes

component_specific_constraint	A limitation on other parts of a system, including other components, which should be adhered to. This will be something that is of the subject matter of the component. The constraint could come from generating a solution to meet a requirement or from receiving a piece of information as described in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Constraint Management.
temporal_information	Timing information pertaining to the periods of time when the constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the constraint is applicable, e.g. spatial zones in which the constraint applies.
breach	A statement that the constraint has been breached, or is likely to be breached if enforced.

Activities

identify_required_constraint

Identify a constraint which needs to be applied when a solution is enacted.

issue_required_constraint_to_be_enacted

Issue a constraint to be applied as part of enacting a selected solution.

assess_constraint_update

Assess the update to a constraint dependency to determine if the constraint has been adhered to or breached in order to decide whether any further actions need to be taken.

5.4.1.3.2.2.2.2 Authorisation_Dependency

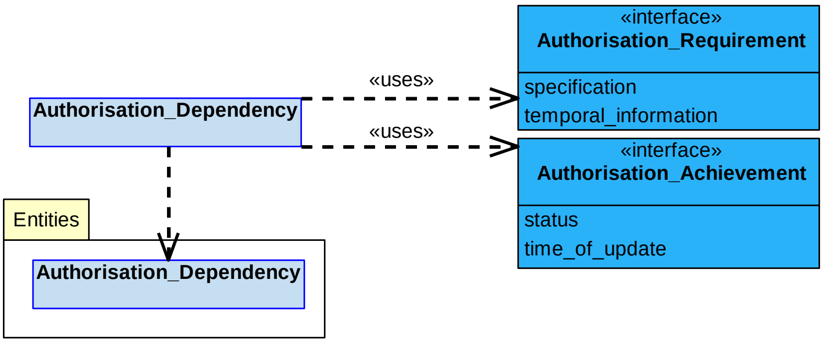
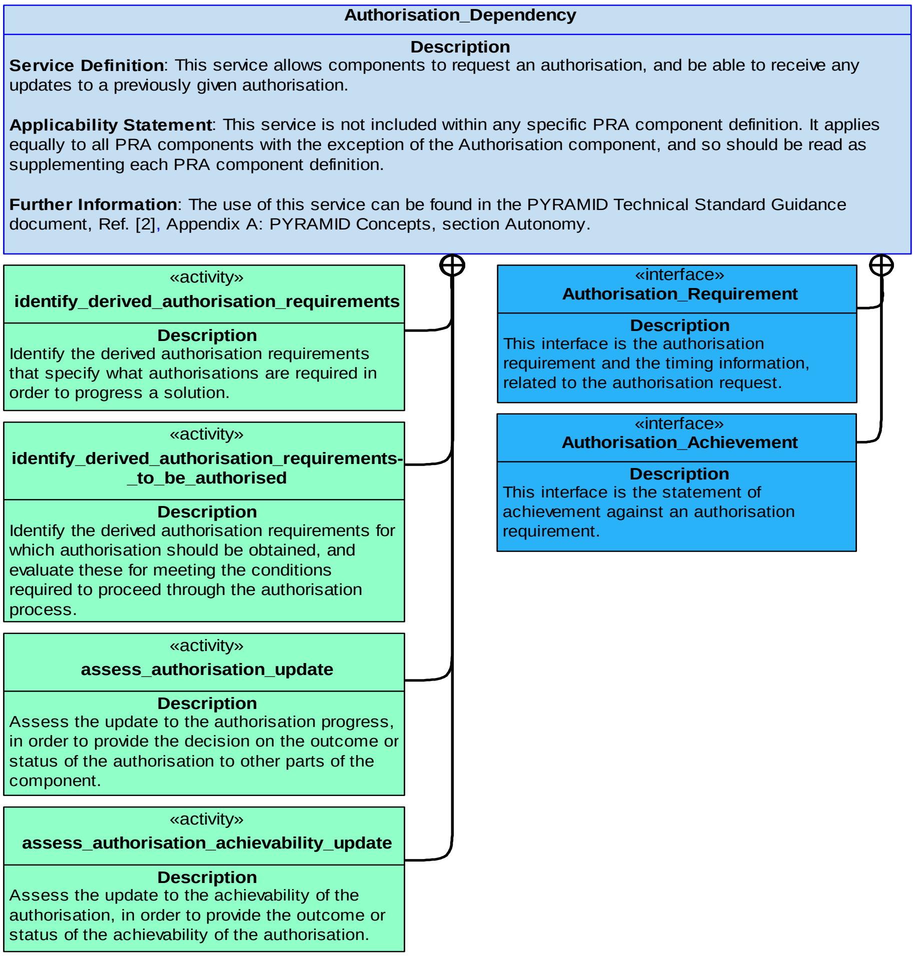


Figure 39: Authorisation_Dependency Service Definition


Authorisation_Dependency

Description

Service Definition: This service allows components to request an authorisation, and be able to receive any updates to a previously given authorisation.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components with the exception of the Authorisation component, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Autonomy.

«activity»
identify_derived_authorisation_requirements

Description
Identify the derived authorisation requirements that specify what authorisations are required in order to progress a solution.

«activity»
identify_derived_authorisation_requirements-_to_be_authorised

Description
Identify the derived authorisation requirements for which authorisation should be obtained, and evaluate these for meeting the conditions required to proceed through the authorisation process.

«activity»
assess_authorisation_update

Description
Assess the update to the authorisation progress, in order to provide the decision on the outcome or status of the authorisation to other parts of the component.

«activity»
assess_authorisation_achievability_update

Description
Assess the update to the achievability of the authorisation, in order to provide the outcome or status of the achievability of the authorisation.

«interface»
Authorisation_Requirement

Description
This interface is the authorisation requirement and the timing information, related to the authorisation request.

«interface»
Authorisation_Achievement

Description
This interface is the statement of achievement against an authorisation requirement.

Figure 40: Authorisation_Dependency Service Policy

Authorisation_Dependency

Service Definition: This service allows components to request an authorisation, and be able to receive any updates to a previously given authorisation.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components with the exception of the Authorisation component, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Autonomy.

Interfaces

Authorisation_Requirement

This interface is the authorisation requirement and the timing information, related to the authorisation request.

Attributes

specification	The definition of the authorisation requirement.
temporal_information	Information covering timing, such as the time period for when authorisation is required, may need to be achieved by or needs to remains relevant for.

Authorisation_Achievement

This interface is the statement of achievement against an authorisation requirement.

Attributes

status	A high-level representation of the achievement in relation to the authorisation (e.g. not started, achievable, in progress, approved, or refused).
time_of_update	The time at which an achievement update occurred.

Activities

identify_derived_authorisation_requirements

Identify the derived authorisation requirements that specify what authorisations are required in order to progress a solution.

identify_derived_authorisation_requirements_to_be_authorised

Identify the derived authorisation requirements for which authorisation should be obtained, and evaluate these for meeting the conditions required to proceed through the authorisation process.

assess_authorisation_update

Assess the update to the authorisation progress, in order to provide the decision on the outcome or status of the authorisation to other parts of the component.

assess_authorisation_achievability_update

Assess the update to the achievability of the authorisation, in order to provide the outcome or status of the achievability of the authorisation.

5.4.1.3.2.2.2.3 Broker_Conflict

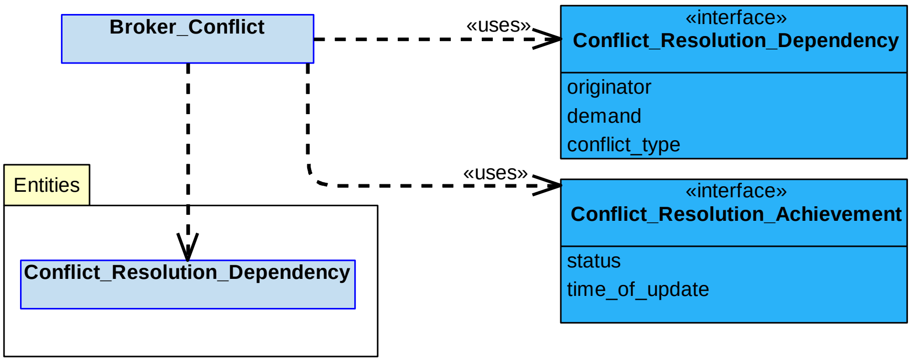
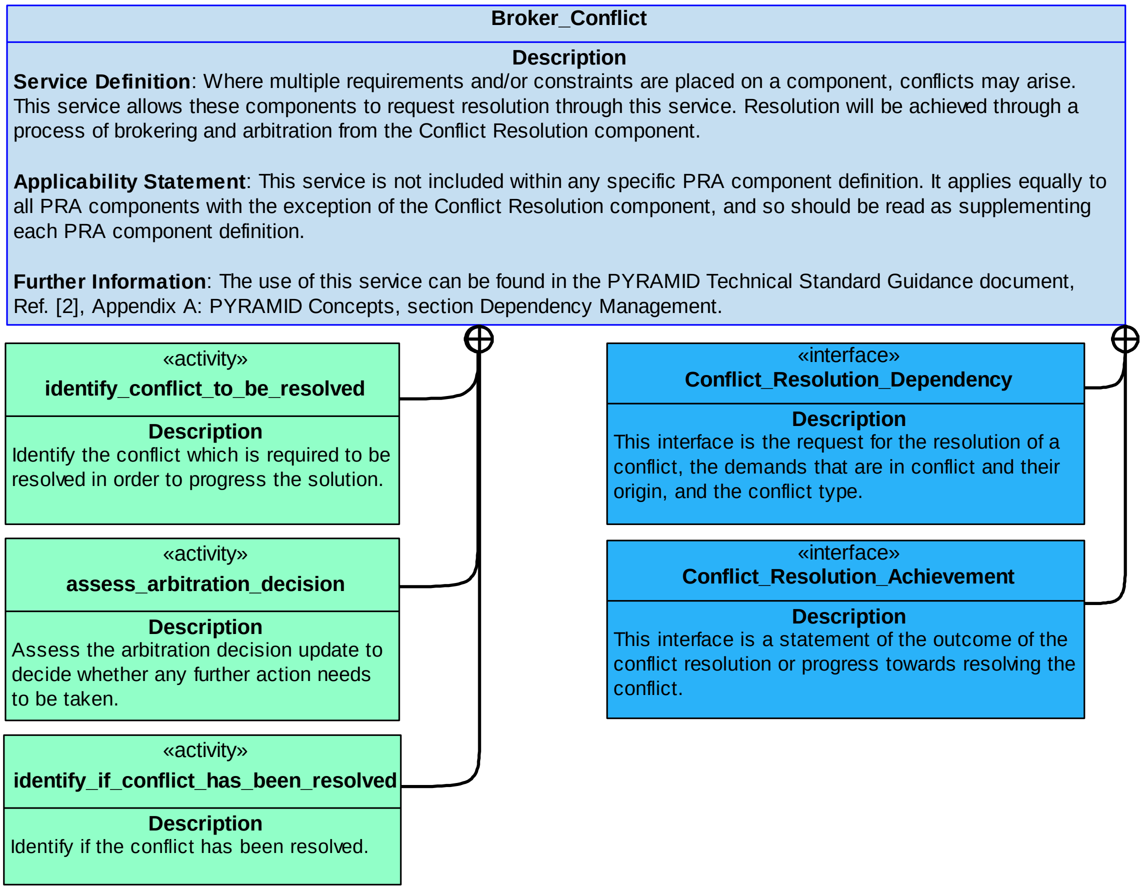


Figure 41: Broker_Conflict Service Definition


Broker_Conflict
«activity»
identify_conflict_to_be_resolved
«activity»
assess_arbitration_decision
«activity»
identify_if_conflict_has_been_resolved
«interface»
Conflict_Resolution_Dependency
«interface»
Conflict_Resolution_Achievement

Broker_Conflict

Description

Service Definition: Where multiple requirements and/or constraints are placed on a component, conflicts may arise. This service allows these components to request resolution through this service. Resolution will be achieved through a process of brokering and arbitration from the Conflict Resolution component.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components with the exception of the Conflict Resolution component, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

«activity» identify_conflict_to_be_resolved

Description

Identify the conflict which is required to be resolved in order to progress the solution.

«activity» assess_arbitration_decision

Description

Assess the arbitration decision update to decide whether any further action needs to be taken.

«activity» identify_if_conflict_has_been_resolved

Description

Identify if the conflict has been resolved.

«interface» Conflict_Resolution_Dependency

Description

This interface is the request for the resolution of a conflict, the demands that are in conflict and their origin, and the conflict type.

«interface» Conflict_Resolution_Achievement

Description

This interface is a statement of the outcome of the conflict resolution or progress towards resolving the conflict.

Figure 42: Broker_Conflict Service Policy

Broker_Conflict

Service Definition: Where multiple requirements and/or constraints are placed on a component, conflicts may arise. This service allows these components to request resolution through this service. Resolution will be achieved through a process of brokering and arbitration from the Conflict Resolution component.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components with the exception of the Conflict Resolution component, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Dependency Management.

Interfaces

Conflict_Resolution_Dependency

This interface is the request for the resolution of a conflict, the demands that are in conflict and their origin, and the conflict type.

Attributes

originator	The originator of a requirement and/or constraint that is in conflict.
demand	The requirement and/or constraint that is in conflict.
conflict_type	The specific nature of the conflict (e.g. two requests to use the same capability simultaneously).

Conflict_Resolution_Achievement

This interface is a statement of the outcome of the conflict resolution or progress towards resolving the conflict.

Attributes

status	A high-level representation of achievement in relation to the requirement (e.g. not started, in progress, or complete).
time_of_update	The time at which an achievement update occurred.

Activities

identify_conflict_to_be_resolved

Identify the conflict which is required to be resolved in order to progress the solution.

assess_arbitration_decision

Assess the arbitration decision update to decide whether any further action needs to be taken.

identify_if_conflict_has_been_resolved

Identify if the conflict has been resolved.

5.4.1.3.2.2.2.4 Storage_Dependency

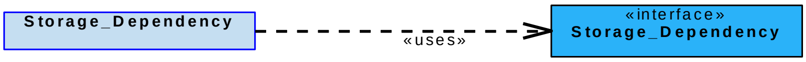
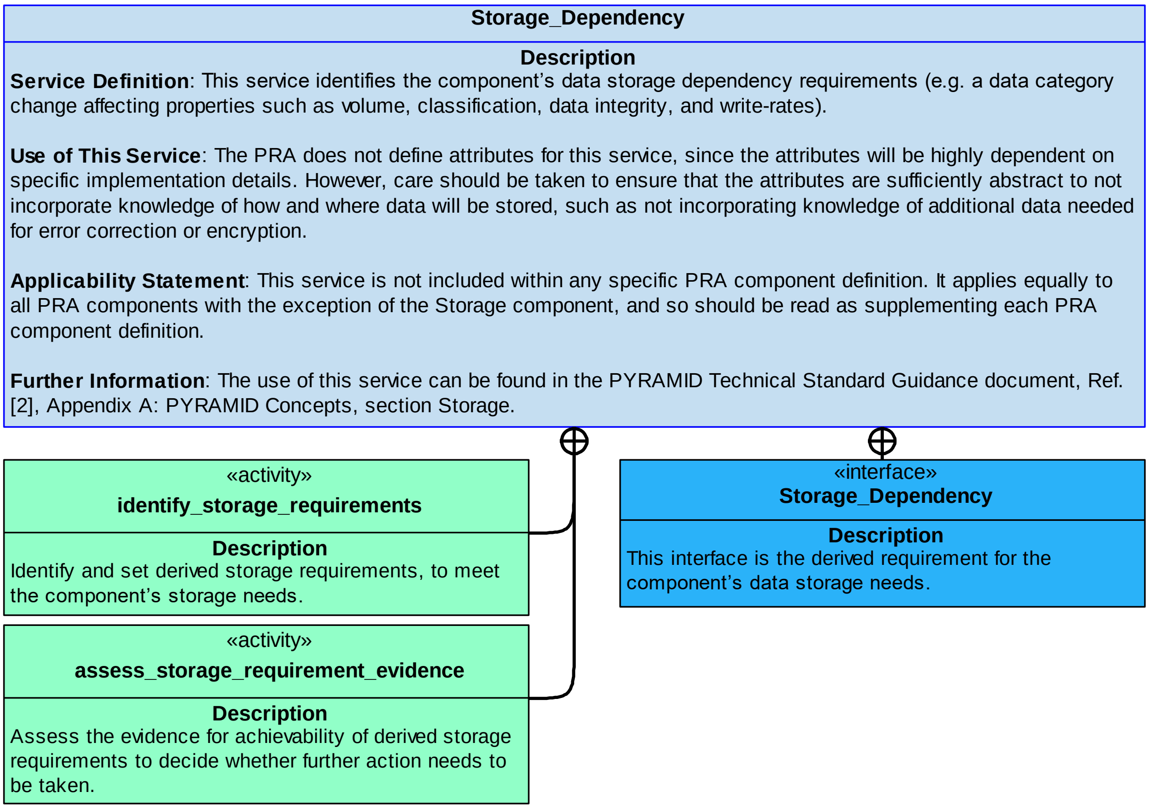


Figure 43: Storage_Dependency Service Definition



Service Definition: This service identifies the component’s data storage dependency requirements (e.g. a data category change affecting properties such as volume, classification, data integrity, and write-rates).

Use of This Service: The PRA does not define attributes for this service, since the attributes will be highly dependent on specific implementation details. However, care should be taken to ensure that the attributes are sufficiently abstract to not incorporate knowledge of how and where data will be stored, such as not incorporating knowledge of additional data needed for error correction or encryption.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components with the exception of the Storage component, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Storage.

Identify and set derived storage requirements, to meet the component’s storage needs.

Assess the evidence for achievability of derived storage requirements to decide whether further action needs to be taken.

This interface is the derived requirement for the component’s data storage needs.

Figure 44: Storage_Dependency Service Policy

Storage_Dependency

Service Definition: This service identifies the component’s data storage dependency requirements (e.g. a data category change affecting properties such as volume, classification, data integrity, and write-rates).

Use of This Service: The PRA does not define attributes for this service, since the attributes will be highly dependent on specific implementation details. However, care should be taken to ensure that the attributes are sufficiently abstract to not incorporate knowledge of how and where data will be stored, such as not incorporating knowledge of additional data needed for error correction or encryption.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components with the exception of the Storage component, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, section Storage.

Interface

Storage_Dependency

This interface is the derived requirement for the component’s data storage needs.

Activities

identify_storage_requirements

Identify and set derived storage requirements, to meet the component’s storage needs.

assess_storage_requirement_evidence

Assess the evidence for achievability of derived storage requirements to decide whether further action needs to be taken.

5.4.1.3.2.2.2.5 Retention_Coordination_Dependency

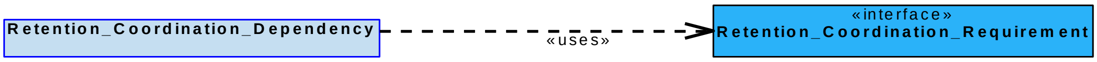
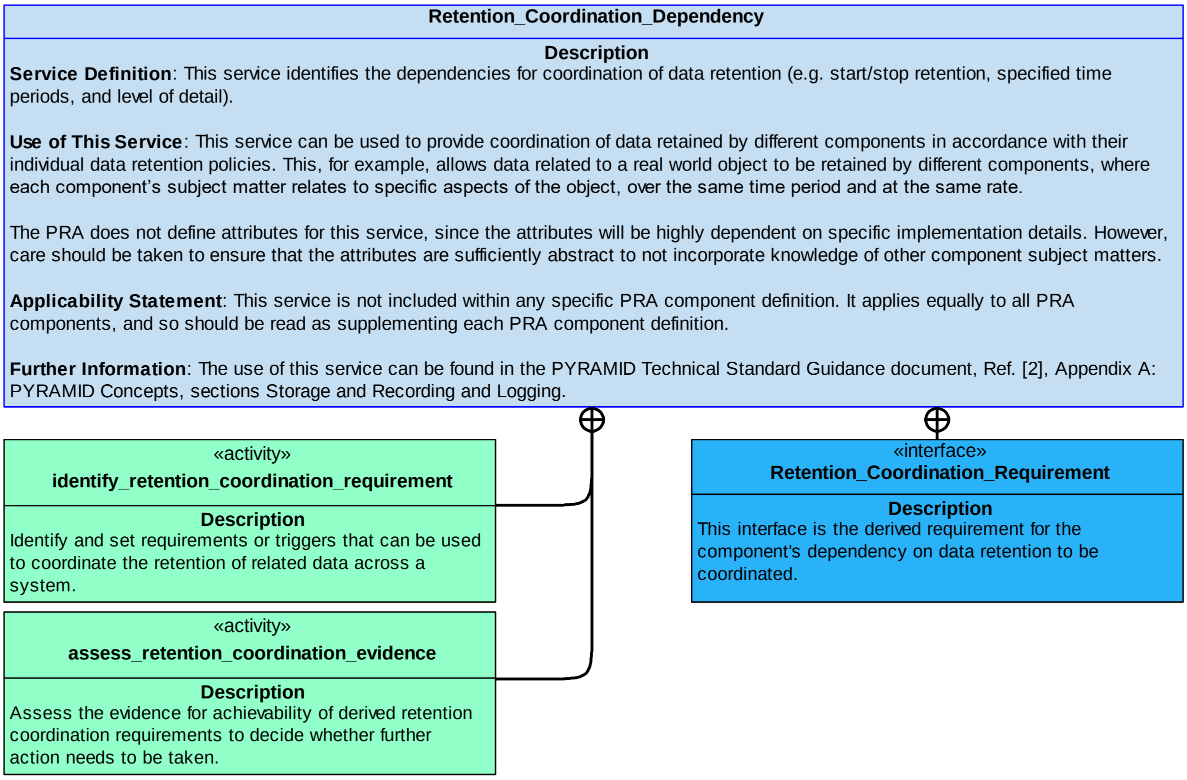

Retention_Coordination_Dependency
«uses»
«interface»
Retention_Coordination_Requirement

Figure 45: Retention_Coordination_Dependency Service Defintion


Retention_Coordination_Dependency
Description
«activity»
identify_retention_coordination_requirement
Description
«activity»
assess_retention_coordination_evidence
Description
«interface»
Retention_Coordination_Requirement
Description

Service Definition: This service identifies the dependencies for coordination of data retention (e.g. start/stop retention, specified time periods, and level of detail).

Use of This Service: This service can be used to provide coordination of data retained by different components in accordance with their individual data retention policies. This, for example, allows data related to a real world object to be retained by different components, where each component's subject matter relates to specific aspects of the object, over the same time period and at the same rate.

The PRA does not define attributes for this service, since the attributes will be highly dependent on specific implementation details. However, care should be taken to ensure that the attributes are sufficiently abstract to not incorporate knowledge of other component subject matters.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, sections Storage and Recording and Logging.

Identify and set requirements or triggers that can be used to coordinate the retention of related data across a system.

Assess the evidence for achievability of derived retention coordination requirements to decide whether further action needs to be taken.

This interface is the derived requirement for the component's dependency on data retention to be coordinated.

Figure 46: Retention_Coordination_Dependency Service Policy

Retention_Coordination_Dependency

Service Definition: This service identifies the dependencies for coordination of data retention (e.g. start/stop retention, specified time periods, and level of detail).

Use of This Service: This service can be used to provide coordination of data retained by different components in accordance with their individual data retention policies. This, for example, allows data related to a real world object to be retained by different components, where each component's subject matter relates to specific aspects of the object, over the same time period and at the same rate.

The PRA does not define attributes for this service, since the attributes will be highly dependent on specific implementation details. However, care should be taken to ensure that the attributes are sufficiently abstract to not incorporate knowledge of other component subject matters.

Applicability Statement: This service is not included within any specific PRA component definition. It applies equally to all PRA components, and so should be read as supplementing each PRA component definition.

Further Information: The use of this service can be found in the PYRAMID Technical Standard Guidance document, Ref. [2], Appendix A: PYRAMID Concepts, sections Storage and Recording and Logging.

Interface

Retention_Coordination_Requirement

This interface is the derived requirement for the component's dependency on data retention to be coordinated.

Activities

identify_retention_coordination_requirement

Identify and set requirements or triggers that can be used to coordinate the retention of related data across a system.

assess_retention_coordination_evidence

Assess the evidence for achievability of derived retention coordination requirements to decide whether further action needs to be taken.

5.4.1.3.3 Service Dependencies

The component composition service dependency diagrams differ from the PRA component service dependency diagrams in the following ways:

- The Component Composition includes multiple service dependency diagrams, whereas PRA components only include one. Each component composition service dependency diagram shows how a specific provided service (the Capability, Constraint, Information and Requirement services) depends on other services. The dependencies for each provided service are shown separately, in order to avoid the diagrams being too complex to easily read.
- The component composition service dependencies diagrams illustrate that the services provided by a component may not only be provided based on a request from outside of the component, but may also be provided based on a pre-loaded definition of the service that is needed, which may then be triggered by a factor external or internal to the component. This is illustrated by showing the dependencies of provided services on 'internal' requirements and constraints, which are present within the component upon initialisation and not placed upon the component via the Requirement or Constraint services. This is not shown on PRA component service dependency diagrams in order to aid readability.

The component composition service dependency diagrams do not show all possible dependencies between services. This is because it is impractical to depict every conceivable dependency on a single diagram or to create a definitive list of all possible dependencies. For example, if a new solution is created by any service, existing solutions may need to be re-planned to ensure they can be achieved.

It should be noted that the services Retention_Requirement, Storage_Dependency and Retention_Coordination_Dependency are not represented within the component composition service dependency diagrams. As the information being stored could apply to information generated by virtually any activity within any service, representing this in a service dependency diagram would not provide meaningful clarity.

5.4.1.3.3.1 Non-Service Activities

Non-Service Activities

These activities define behaviour that is carried out by the component but not by any particular service, which is helpful to show the service activity interactions.

Activities

provide_internal_requirements

Provide pre-loaded and hard-coded requirements to the relevant services/activities within the component.

The solutions generated that satisfy these requirements may be executed automatically or upon request (for example, the component might automatically and continuously generate a solution to an internal ‘get home’ routing requirement that is only enacted when requested externally).

provide_internal_constraints

Provide pre-loaded and hard-coded constraints adhered to by the component.

5.4.1.3.3.2 Activity Overview

Most of the service dependency activity diagrams contain activities that determine a solution, maintain an assessment of whether the solution remains feasible, executes the determined solution, and determines the progress being made throughout the enactment of a solution. The following applies to these activities:

Activities that determine a solution:

- Identify the dependencies of a solution being planned, such as any constraints.
- Identify conflicts that prevent a solution from being determined or influence how a solution is determined.
- Where necessary determine and provide the achievability and predicted quality of the solution.
- Indicate when the enactment of solutions that do not require an external triggering should commence.
- Provide indications of when a solution is modified, is superseded by an alternative solution, or is no longer valid or feasible.

Activities that maintain an assessment of whether the solution remains feasible involve receiving updates from component dependencies. They also account for any new or updated constraints placed upon the component, in order to:

- Determine whether a planned solution that has not been triggered for execution is still feasible.
- Determine whether a solution being executed is still feasible.

Activities that perform the execution of the solution:

- Initiate and perform any non-delegated actions, which are entirely within the component's subject matter.
- Identify dependencies that are to be enacted by other components (through derived requirement solutions, derived constraints, authorisations, and information requests).

Activities that determine the progress being made throughout the enactment of a solution:

- Consume the status of any non-delegated actions.
- Consume dependency updates (for derived requirement progress, constraint adherence, authorisation status, and information).
- Where necessary, provide a measure of the progress made against the requirement.

5.4.1.3.3.3 Requirement Service Dependencies

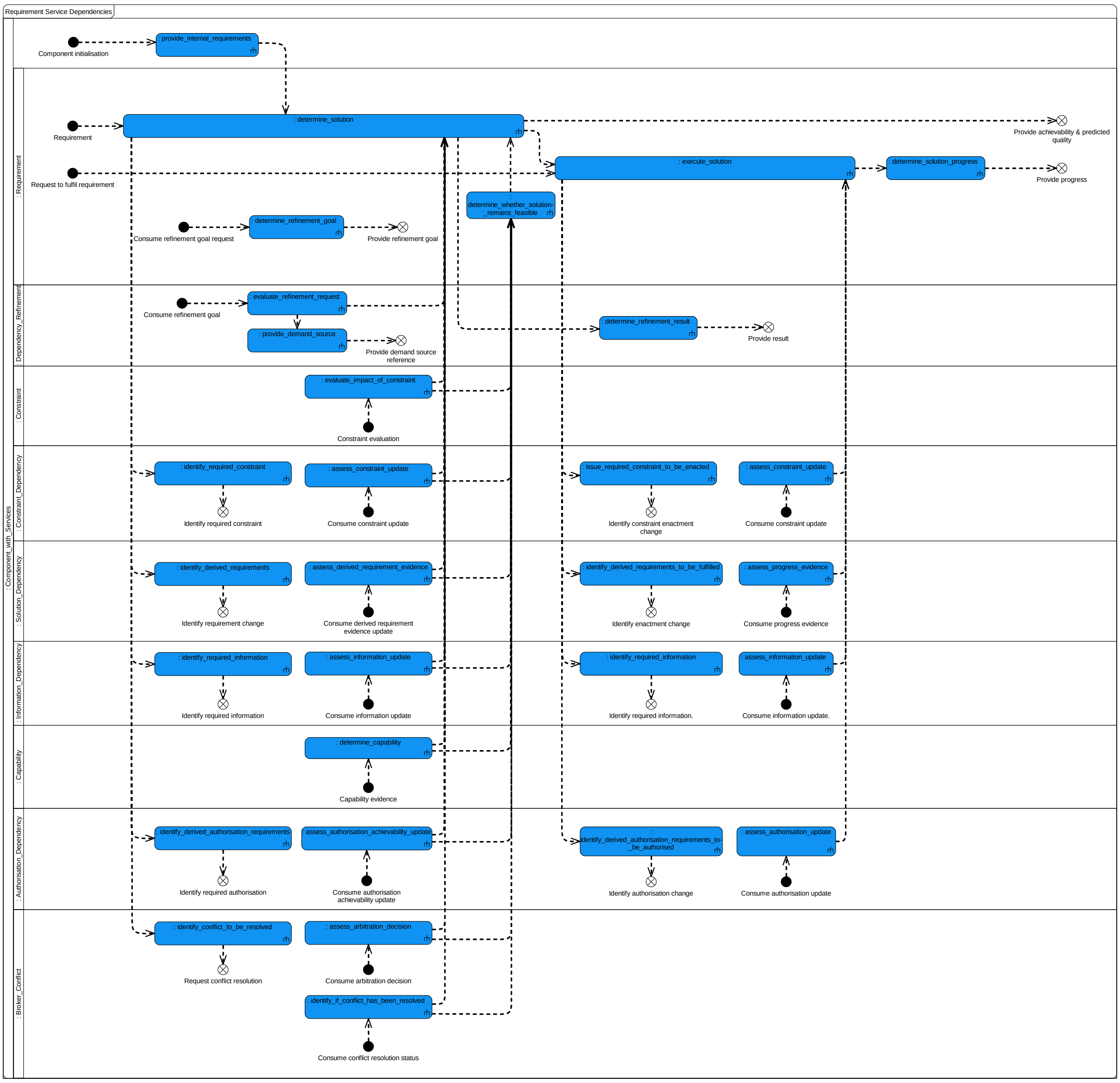

Requirement Service Dependencies
: Component_with_Services
: Requirement
: Dependency_Refinement
: Constraint
: Constraint_Dependency
: Solution_Dependency
: Information_Dependency
: Capability
: Authorisation_Dependency
: Broker_Conflict
Component initialisation
: provide_internal_requirements
Requirement
: determine_solution
Provide achievability & predicted quality
Request to fulfil requirement
: execute_solution
: determine_solution_progress
Provide progress
: determine_whether_solution-_remains_feasible
Consume refinement goal request
: determine_refinement_goal
Provide refinement goal
Consume refinement goal
: evaluate_refinement_request
: provide_demand_source
Provide demand source reference
: determine_refinement_result
Provide result
: evaluate_impact_of_constraint
Constraint evaluation
: identify_required_constraint
Identify required constraint
: assess_constraint_update
Consume constraint update
: issue_required_constraint_to_be_enacted
Identify constraint enactment change
: assess_constraint_update
Consume constraint update
: identify_derived_requirements
Identify requirement change
: assess_derived_requirement_evidence
Consume derived requirement evidence update
: identify_derived_requirements_to_be_fulfilled
Identify enactment change
: assess_progress_evidence
Consume progress evidence
: identify_required_information
Identify required information
: assess_information_update
Consume information update
: identify_required_information
Identify required information.
: assess_information_update
Consume information update.
: determine_capability
Capability evidence
: identify_derived_authorisation_requirements
Identify required authorisation
: assess_authorisation_achievability_update
Consume authorisation achievability update
: identify_derived_authorisation_requirements_to_be_authorised
Identify authorisation change
: assess_authorisation_update
Consume authorisation update
: identify_conflict_to_be_resolved
Request conflict resolution
: assess_arbitration_decision
Consume arbitration decision
: identify_if_conflict_has_been_resolved
Consume conflict resolution status

Figure 47: Requirement Service Dependencies

This diagram shows the dependencies between the Requirement service and other services within the Component Composition, in addition to the various activities within the Requirement service.

5.4.1.3.3.4 Information Service Dependencies

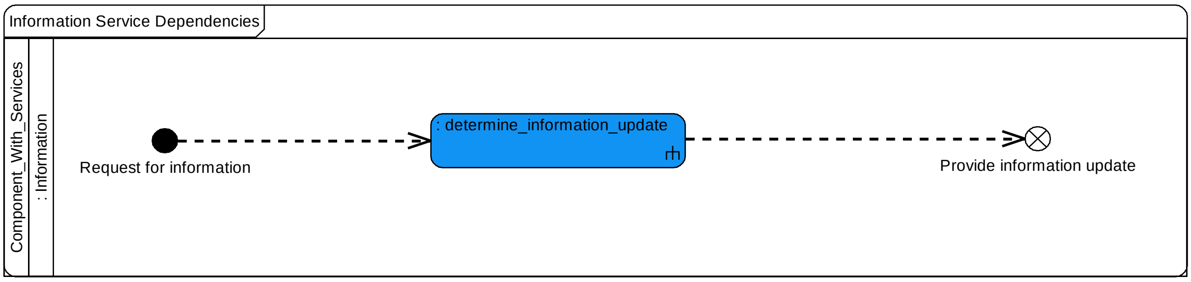

Information Service Dependencies
Component_With_Services
: Information
Request for information
: determine_information_update
Provide information update

Figure 48: Information Service Dependencies

This diagram shows the dependencies between the Information service and other services within the Component Composition (although in this case there are no inter-service dependencies to show), in addition to the single activity within the Information service.

The information provided by this service is information related to the component's subject matter that is either directly captured or derived from data and information from other services.

5.4.1.3.3.5 Constraint Service Dependencies

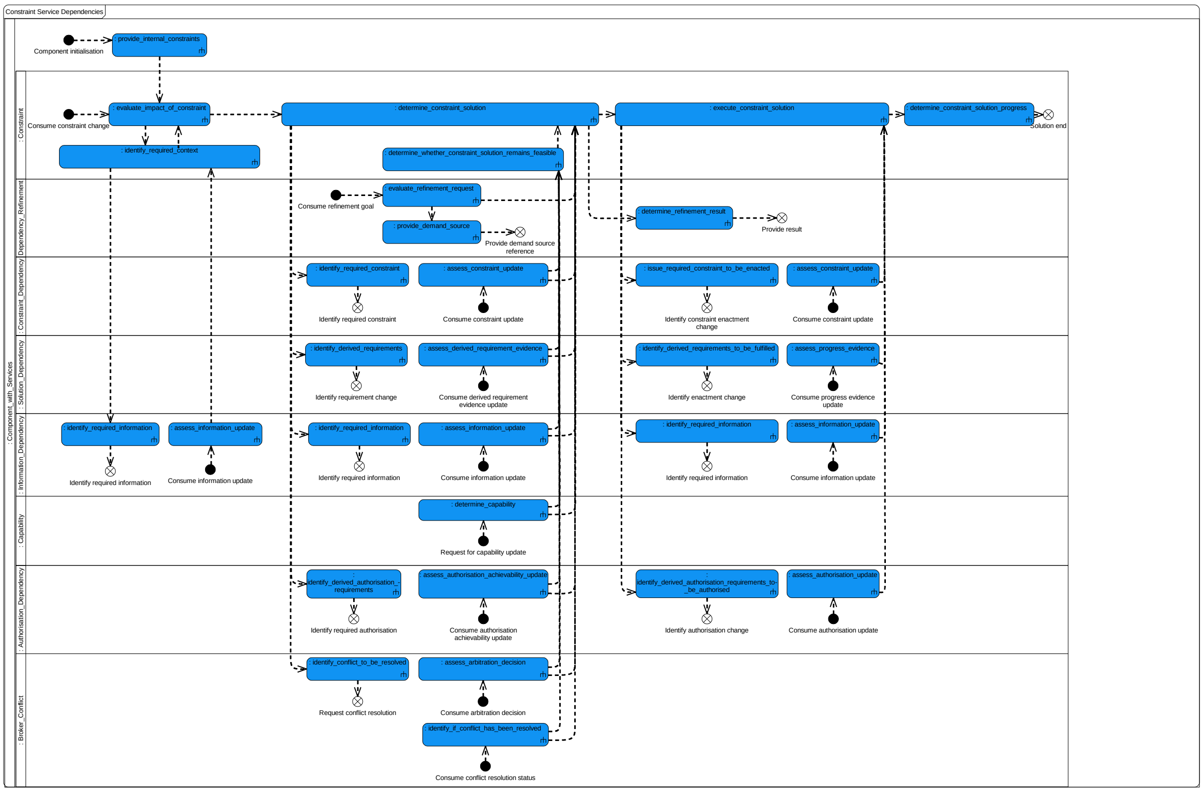

Constraint Service Dependencies
: Component_with_Services
Component initialisation
: provide_internal_constraints
: Constraint
Consume constraint change
: evaluate_impact_of_constraint
: identify_required_context
: determine_constraint_solution
: determine_whether_constraint_solution_remains_feasible
: execute_constraint_solution
: determine_constraint_solution_progress
Solution end
: Dependency_Refinement
Consume refinement goal
: evaluate_refinement_request
: provide_demand_source
Provide demand source reference
: determine_refinement_result
Provide result
: Constraint_Dependency
: identify_required_constraint
Identify required constraint
: assess_constraint_update
Consume constraint update
: issue_required_constraint_to_be_enacted
Identify constraint enactment change
: assess_constraint_update
Consume constraint update
: Solution_Dependency
: identify_derived_requirements
Identify requirement change
: assess_derived_requirement_evidence
Consume derived requirement evidence update
: identify_derived_requirements_to_be_fulfilled
Identify enactment change
: assess_progress_evidence
Consume progress evidence update
: Information_Dependency
: identify_required_information
Identify required information
: assess_information_update
Consume information update
: identify_required_information
Identify required information
: assess_information_update
Consume information update
: identify_required_information
Identify required information
: assess_information_update
Consume information update
: Capability
: determine_capability
Request for capability update
: Authorisation_Dependency
: identify_derived_authorisation_requirements
Identify required authorisation
: assess_authorisation_achievability_update
Consume authorisation achievability update
: identify_derived_authorisation_requirements_to-_be_authorised
Identify authorisation change
: assess_authorisation_update
Consume authorisation update
: Broker_Conflict
: identify_conflict_to_be_resolved
Request conflict resolution
: assess_arbitration_decision
Consume arbitration decision
: identify_if_conflict_has_been_resolved
Consume conflict resolution status

Figure 49: Constraint Service Dependencies

This diagram shows the dependencies between the Constraint service and other services within the Component Composition, in addition to the various activities within the Constraint service.

The evaluate_impact_of_constraint activity has one control flow linking it to the determine_constraint_solution activity, however, it represents both of the following cases where:

- The evaluate_impact_of_constraint activity is evaluating a new constraint that the component needs to produce a solution to in order to meet;
- The evaluate_impact_of_constraint activity is evaluating an update to an existing constraint that does not itself require a solution but must be adhered to by any solution.

5.4.1.3.3.6 Capability Service Dependencies

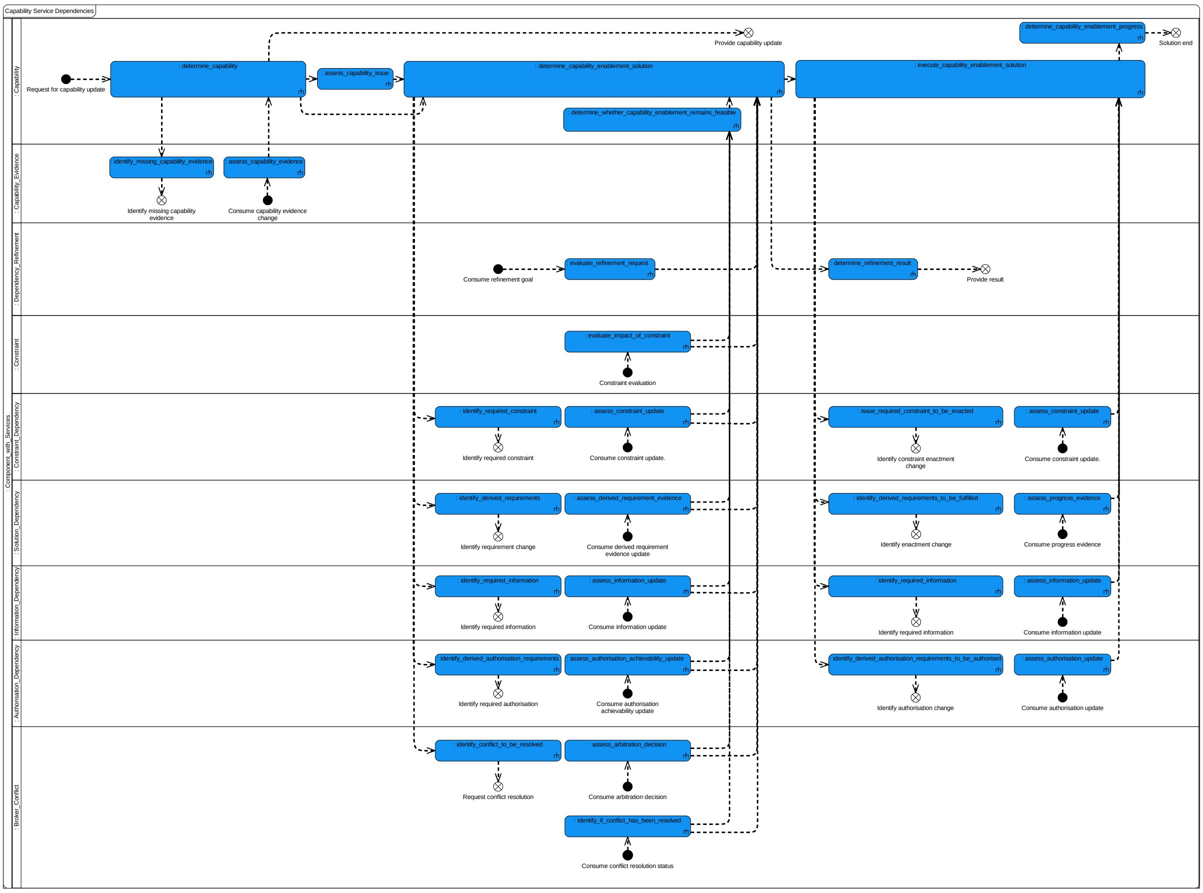

Capability Service Dependencies
: Component_with_Services
: Capability
: Capability_Evidence
: Dependency_Refinement
: Constraint
: Constraint_Dependency
: Solution_Dependency
: Information_Dependency
: Authorisation_Dependency
: Broker_Conflict
Provide capability update
: determine_capability_enablement_progress
Solution end
Request for capability update
: determine_capability
: assess_capability_issue
: determine_capability_enablement_solution
: execute_capability_enablement_solution
: determine_whether_capability_enablement_remains_feasible
: identify_missing_capability_evidence
: assess_capability_evidence
Identify missing capability evidence
Consume capability evidence change
Consume refinement goal
: evaluate_refinement_request
: determine_refinement_result
Provide result
: evaluate_impact_of_constraint
Constraint evaluation
: identify_required_constraint
: assess_constraint_update
: issue_required_constraint_to_be_enacted
: assess_constraint_update
Identify required constraint
Consume constraint update.
Identify constraint enactment change
Consume constraint update.
: identify_derived_requirements
: assess_derived_requirement_evidence
: identify_derived_requirements_to_be_fulfilled
: assess_progress_evidence
Identify requirement change
Consume derived requirement evidence update
Identify enactment change
Consume progress evidence
: identify_required_information
: assess_information_update
: identify_required_information
: assess_information_update
Identify required information
Consume information update
Identify required information
Consume information update
: identify_derived_authorisation_requirements
: assess_authorisation_achievability_update
: identify_derived_authorisation_requirements_to_be_authorised
: assess_authorisation_update
Identify required authorisation
Consume authorisation achievability update
Identify authorisation change
Consume authorisation update
: identify_conflict_to_be_resolved
: assess_arbitration_decision
Request conflict resolution
Consume arbitration decision
: identify_if_conflict_has_been_resolved
Consume conflict resolution status

Figure 50: Capability Service Dependencies

This diagram shows the dependencies between the Capability service and other services within the Component Composition, in addition to the various activities within the Capability service.

The determine_capability_enablement_solution and execute_capability_enablement_solution activities shown relate to determining and executing a solution to address a capability shortfall.

The direct dependency between determine_capability and determine_capability_enablement_solution relates to the capability of the component changing and the solution to address a capability shortfall being updated as a consequence.

5.4.2 Component Set

This section defines the PRA components introduced in the Component Set Overview section.

5.4.2.1 Anomaly Detection

5.4.2.1.1 Role

The role of Anomaly Detection is to detect anomalous behaviours of elements of the system that could be indications of failures, damage, cyber events or other conditions that could affect the capability of the system.

5.4.2.1.2 Overview

Control Architecture

Anomaly Detection is a service component as described in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Anomaly Detection determines the Actual_State of a System_Element using System_Data. It compares this Actual_State to the Expected_State, taking into account any interactions between components or events that would result in a change in State of the System_Element. If the Actual_State and Expected_State differ, according to Rules in place, Anomaly Detection declares an anomaly.

Examples of Use

Anomaly Detection will be used where it is necessary to identify that a System_Element is in an unexpected State, to support health management or identification of a cyber attack. Examples of this include where:

- The physical position of a resource, such as an actuator, is not in an expected position.
- The processing time for an operation falls outside the known timing for that operation.
- There are unexpected changes to user privileges.
- The position and velocity of a detected aircraft does not match received intelligence information for the aircraft.

5.4.2.1.3 Service Summary

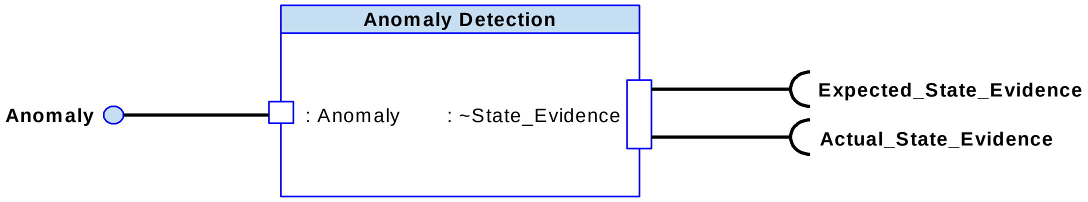


Figure 51: Anomaly Detection Service Summary

5.4.2.1.4 Responsibilities

determine_actual_state

- To interpret and correlate available System_Data to determine the Actual_State of a System_Element.

determine_expected_state

- To interpret and correlate available System_Data to determine the Expected_State of a System_Element.

identify_anomalies

- To identify anomalies.

5.4.2.1.5 Subject Matter Semantics

The subject matter of Anomaly Detection is the expected and actual States of System_Elements and the understanding of which states or sequences of states (i.e. behaviours) are indicative of anomalies.

Exclusions

The subject matter of Anomaly Detection does not include:

- The identification of the cause or effects of anomalies, but does include identification of their existence.

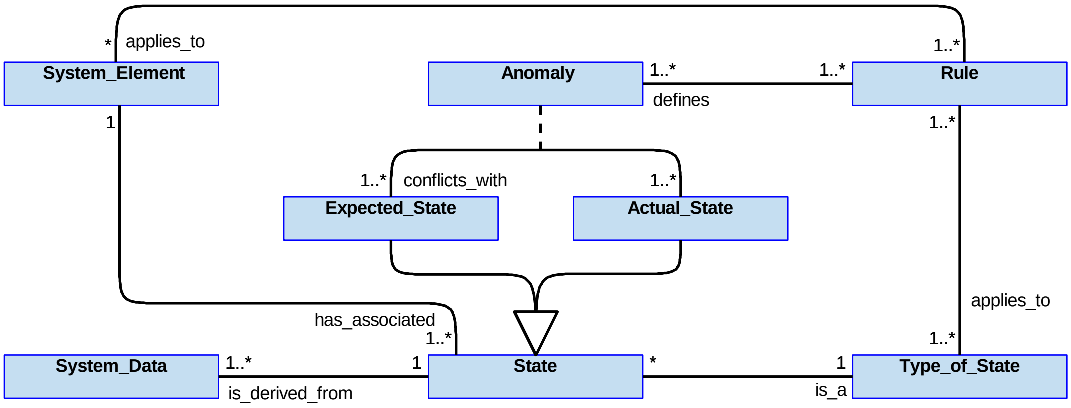


Figure 52: Anomaly Detection Semantics

5.4.2.1.5.1 Entities

Actual_State

A State of an element of a system as observed by the system.

Anomaly

An indication that an element of a system is exhibiting unexpected behaviour that may be a sign of a fault, damage or other degradation. This unexpected behaviour is identified by detecting an unexpected state (or states). An unexpected state could be, but is not limited to, an invalid state, an erroneous progression of state behaviour or an indication that an element of the system is slow to respond.

Expected_State

An expected State of an element of the system based on previous states and commands.

Rule

A rule that governs the conditions under which an anomaly will be identified. For example, a particular actuator should move to the fully open position within 3 seconds of receiving the command to open.

State

A particular instance of a Type_of_State observed in the system.

System_Data

Any data about the system that could be used to deduce the State of an element of the system.

System_Element

A part of the system that needs to be monitored to determine if it is exhibiting anomalous behaviour.

Type_of_State

Some property of an element of the system. It may include, but is not limited to, whether a sensor is on or off, the physical position of an element, the privileges associated with a system user, the estimated velocity of a detected aircraft or bandwidth utilised. A state may characterise behaviour, an example being where quantified spikes that impair sensor data are expected but an increasing number of spikes could be unexpected and cause the sensor data to be declared to have a lower credibility state.

5.4.2.1.6 Design Rationale

5.4.2.1.6.1 Assumptions

- This component will be required to comply with ISO 13374 (Condition Monitoring and Diagnostics of Machines) Ref. [13]. See the Health Management PYRAMID concept.
- A consistent source of time data is available so that the order of commands and sensor readings can be determined precisely.
- Anomaly Detection is expected to work with Cyber Defence to identify States that may be indicative of a cyber attack.
- Anomaly Detection will base its detection of anomalies on System_Data collected from the components being monitored.
- Anomalies are not only caused by hardware failures. Any discrepancy between expected and actual State is an anomaly, and being in an unexpected State may mean the observed behaviour is not as expected.

5.4.2.1.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Anomaly Detection:

- Capability Management - Anomaly Detection supports implementation of the Capability Management PYRAMID concept, although Anomaly Detection itself does not provide an evolving view of its capabilities.
- Health Management - Anomaly Detection follows the Health Management PYRAMID concept. As such, it is likely, except for very simple deployments, that Anomaly Detection components might be deployed in multiple instances or with extensions. Each instance or extension being responsible for detecting anomalies in a given area of responsibility. Anomaly Detection may need to compare states from different components to detect anomalies that emerge at a system level. Not all anomalies can be identified at the hardware level: many anomalies are only apparent when the states of lower-level elements are compared. Therefore, Anomaly Detection is likely to be required to monitor components throughout the system.
- Cyber Defence - Anomaly Detection can identify suspicious states that may indicate the presence of a cyber attack.

Extensions

- Extensions may be used in Anomaly Detection components as an alternative to implementing multiple variants of Anomaly Detection.

Exploitation Considerations

- An instance of Anomaly Detection will be highly specific to the component being monitored. It will need to know how the component is expected to react to various inputs and commands, and under what conditions an anomaly should be reported, e.g. an anomaly is only reported if the conflict between the Actual_State and the Expected_State continues for more than a certain length of time.
- Any component, as it carries out an activity, will necessarily be monitoring any available information to determine whether the activity is being carried out according to requirements. This will lead to some overlap with Anomaly Detection that will have to be resolved at design time. Anomaly Detection will obtain error message information from the operating system. Other components should not bridge the error messages from the operating system to Anomaly Detection. In other cases, Anomaly Detection may be able to provide more detail or to identify additional anomalies, such as where the function is still within its specification, but it is not behaving normally: it may be getting progressively slower, for example.
- For **scalability** and **supportability**, an appropriate solution to determining anomalies could be rule-based, with the specific rules implemented through use of data-driven component behaviour, ideally derived from the system design to minimise errors. Anomaly Detection is unlikely to be complicated, so a bespoke implementation may also be appropriate.
- The anomaly detection rules could be compiled into an algorithm, e.g. an artificial intelligence neural network, which means the rules are not identifiable within the software.
- Anomaly Detection should recognise the limits of precision of time information, to avoid being confused about the order of very closely-timed events: for example, whether a sensor reading was taken before or after a command was received.

- While Anomaly Detection should repeatedly detect known anomalies, in order to recognise the continuation of an anomalous State, care should be taken to avoid the generation of false positive alerts due to the presence of failed elements.

5.4.2.1.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

Failure of this component may result in:

- Failure to recognise that "usage data" is not being collected.
- Failure to detect a genuine hardware fault.
- Spurious fault identification.

Analysis of particular deployments may conclude there is sufficient mitigation external to this component to prevent it directly causing a catastrophic outcome. However, anomaly detection for a safety critical system (e.g. flight controls) may need to be produced at the same DAL as the monitored component (e.g. Vehicle Stability and Control) so it can run on a common computer processor. Therefore, this component may need to be developed to DAL A. However, individual instances may be implemented at lower DALs, for lower criticality systems.

Note: this analysis has assumed that not all instances of Anomaly Detection will be run on a computing infrastructure that allows applications with differing DALs to be run on the same processor, without compromising the claimed assurance level of the high integrity applications.

5.4.2.1.6.4 Security Considerations

The indicative security classification is O-S, however the component(s) with which it is associated will be a significant factor.

It is expected there may be multiple instantiations of this component, each of which will reside in a security domain that reflects the component(s) it assesses. Individual anomaly data is expected to be limited to O-S level, although this could be up to the classification of the component being monitored. Additionally, data may require stricter handling due to concerns with possible aggregation. The confidentiality of information that might divulge additional vulnerabilities should be protected.

The component may be expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** for subsequent forensic examination of anomalies, which might then point to the presence of a cyber attack or other breach.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **System Status and Monitoring** of states and behaviour for possible faults, etc. that might indicate a general problem with integrity or availability of functions.

The component is a cornerstone in detecting behaviour that may be indicative of a cyber attack, and is directly involved in satisfying security enforcing functions relating to:

- **Detecting Security Breaches** by identifying anomalous system states and behaviour, including unauthorised access by software to memory or other resources, etc. that may be due to a security breach. Tamper attempts are also expected to be reported as anomalous behaviour.
- **Preventing Cyber Attacks and Malware** by identifying anomalous system states affecting confidentiality,integrity and availability, etc. that may be caused by a cyber attack.
- **Verifying Integrity of Software** through detection of changes in programmable content, etc. following start-up and during operation.

5.4.2.1.7 Services

5.4.2.1.7.1 Service Definitions

5.4.2.1.7.1.1 Anomaly

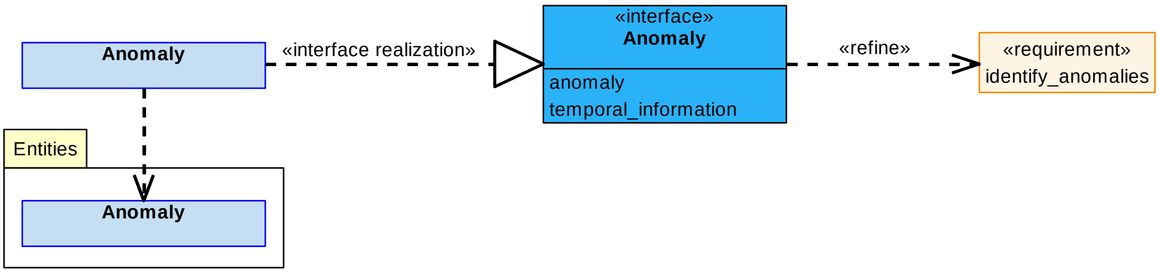


Figure 53: Anomaly Service Definition

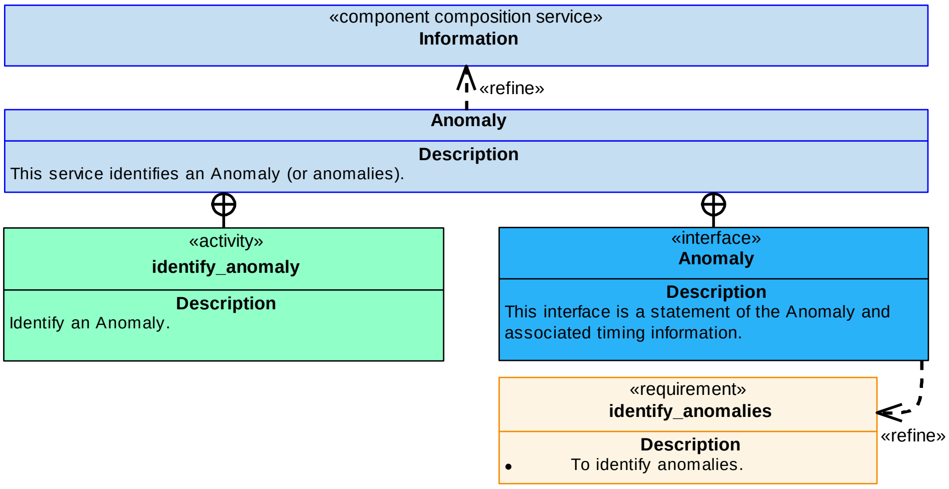


Figure 54: Anomaly Service Policy

Anomaly

This service identifies an Anomaly (or anomalies).

Interface

Anomaly

This interface is a statement of the Anomaly and associated timing information.

Attributes

anomaly	The information about an Anomaly that has taken place, e.g. a particular System_Element is not in the Expected_State.
temporal_information	Temporal information, such as the persistence of an Anomaly (for example, it has occurred x times during the flight), or time of occurrence.

Activity

identify_anomaly

Identify an Anomaly.

5.4.2.1.7.1.2 State_Evidence

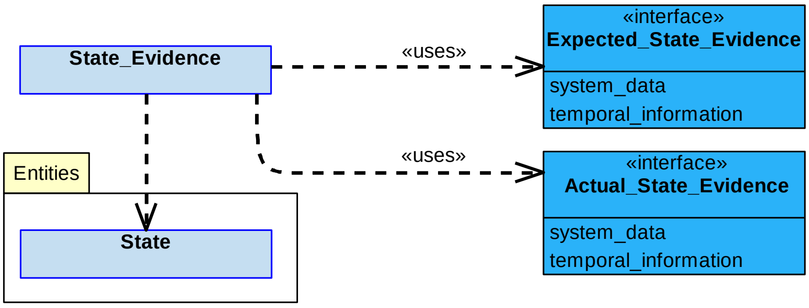


Figure 55: State_Evidence Service Definition

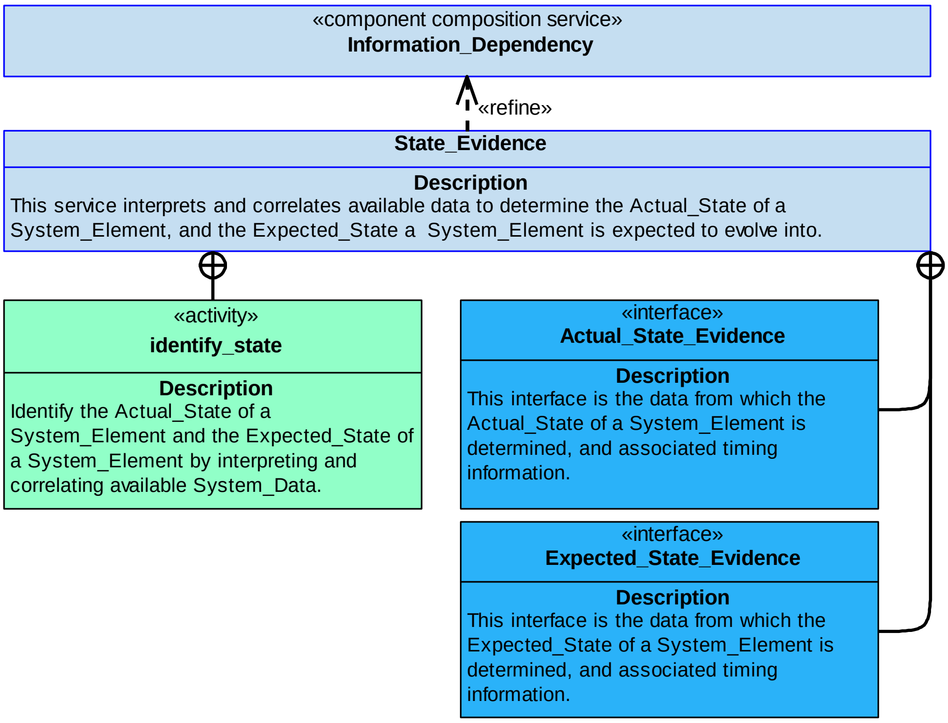

«component composition service»
Information_Dependency
«refine»
State_Evidence
Description
This service interprets and correlates available data to determine the Actual_State of a System_Element, and the Expected_State a System_Element is expected to evolve into.
«activity»
identify_state
Description
Identify the Actual_State of a System_Element and the Expected_State of a System_Element by interpreting and correlating available System_Data.
«interface»
Actual_State_Evidence
Description
This interface is the data from which the Actual_State of a System_Element is determined, and associated timing information.
«interface»
Expected_State_Evidence
Description
This interface is the data from which the Expected_State of a System_Element is determined, and associated timing information.

Figure 56: State_Evidence Service Policy

State_Evidence

This service interprets and correlates available data to determine the Actual_State of a System_Element, and the Expected_State a System_Element is expected to evolve into.

Interfaces

Expected_State_Evidence

This interface is the data from which the Expected_State of a System_Element is determined, and associated timing information.

Attributes

system_data	The System_Data, including commands, sensor data, etc. from which the expected state of a System_Element is determined.
temporal_information	The time at which the System_Data information being used was determined or the time it is valid for.

Actual_State_Evidence

This interface is the data from which the Actual_State of a System_Element is determined, and associated timing information.

Attributes

system_data	The System_Data from which the actual state of a System_Element is determined.
temporal_information	The time at which the System_Data did change.

Activity

identify_state

Identify the Actual_State of a System_Element and the Expected_State of a System_Element by interpreting and correlating available System_Data.

5.4.2.1.7.2 Service Dependencies

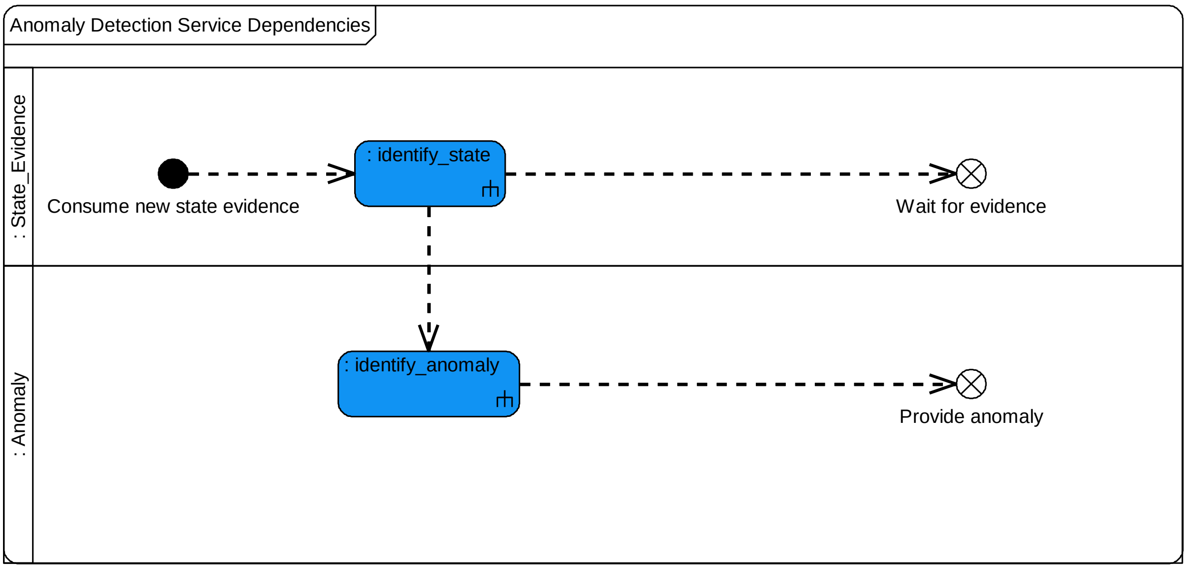


Figure 57: Anomaly Detection Service Dependencies

5.4.2.2 Asset Transitions

5.4.2.2.1 Role

The role of Asset Transitions is to coordinate the transition between states to enable or maintain capabilities.

5.4.2.2.2 Overview

Control Architecture

Asset Transitions is an action component as defined by the Control Architecture PYRAMID concept.

Standard Pattern of Use

On receipt of a Transition_Requirement, Asset Transitions determines a Transition_Solution which involves the coordination of Transition_Steps to achieve a desired State change of an Asset. To do so, the component will request an Asset to perform the necessary Transition_Steps in the prescribed order. The Transition_Steps will be limited by the Available_Transitions the Asset can perform.

Examples of Use

Asset Transitions is used when there is a need to coordinate the establishment and maintenance of high level equipment/system States (including states of the equipment software) that are broader than those directly under the control of other PRA components. For example, where the relative timing of power application, cooling, and data transfer activities is important to transition equipment from an off state to steady state operation.

This includes two main example categories:

- Where a PYRAMID based system needs to interact with another system or equipment that does not fully articulate its high level interoperability needs in real time; therefore, Asset Transitions is used to hold the understanding of these needs. This is applicable when integrating with:
  - Legacy parts of a system that have not been redeveloped to fully integrate with the PYRAMID based part of the system.
  - ‘Off the shelf’ equipment/systems, such as external sensor systems or deployable assets.
- Where major state changes of the exploiting platform need to be coordinated, typically relating to supporting the execution platform, including:
  - The overall system state - such as to enable system initialisation, system shutdown, and maintenance modes.
  - System reconfiguration - such as, when not handled exclusively by the execution platform, to activate reversionary systems, including requesting reversionary processor capability, to maintain safety-critical functions.

5.4.2.2.3 Service Summary

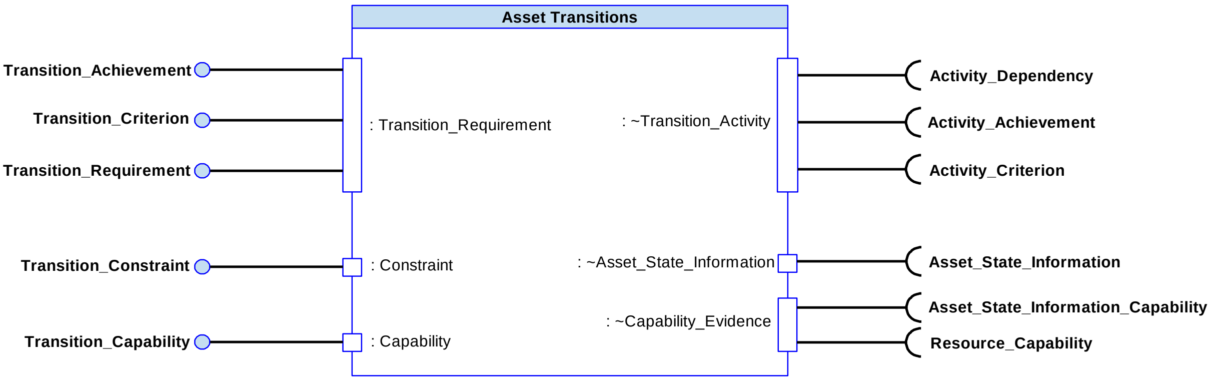


Figure 58: Asset Transitions Service Summary

5.4.2.2.4 Responsibilities

capture_transition_requirements

- To capture given Transition_Requirements to enable a wider system capability.

capture_measurement_criteria

- To capture given Measurement_Criterion/criteria.

capture_transition_constraints

- To capture given Transition_Constraints on a Transition_Solution.

determine_transition_solution

- To determine a Transition_Solution to achieve desired States.

determine_predicted_quality_of_transition_solution

- To determine the predicted quality of a proposed Transition_Solution against the given Measurement_Criterion/criteria.

determine_if_transition_solution_remains_feasible

- To determine the feasibility of a planned or on-going Transition_Solution.

identify_states

- To maintain a view of the current States.

determine_infrastructure_states

- To determine the desired States that offer the required system capability.

implement_transition_solution

- To implement a Transition_Solution.

identify_transition_solution_progress

- To identify the progress of a Transition_Solution against the Transition_Requirement.

determine_actual_quality_of_transition_deliverables

- To determine the actual quality of the Transition_Solution against the given Measurement_Criterion/criteria.

assess_capability

- To assess the Transition_Capability, taking into account available resources, system health and observed anomalies.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Transition_Capability assessment.

predict_capability_progression

- To predict the progression of the Transition_Capability over time and with use.

5.4.2.2.5 Subject Matter Semantics

The subject matter of Asset Transitions is the transitioning of one State into another.

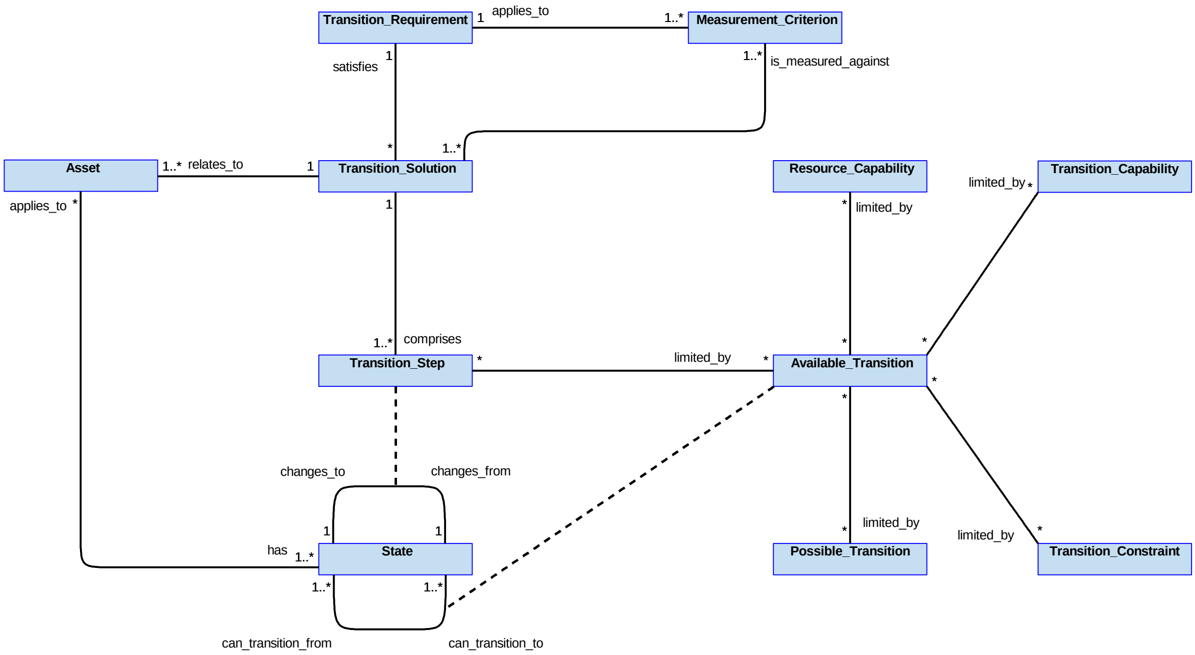


Figure 59: Asset Transitions Semantics

5.4.2.2.5.1 Entities

Asset

A system element (e.g. bay doors, sensors, software configurations) or deployable asset that can transition from one State to another.

Available_Transition

A State transition that an Asset is able to perform under a specific set of circumstances; e.g. transitioning from 'Standby Mode' to an 'Active Mode'.

Measurement_Criterion

A criterion that the quality of the Transition_Solution will be measured against, e.g. the speed of a handover from a primary to a secondary process.

Possible_Transition

Any State transition that an Asset can legally undertake (e.g. from a 'Standby' to an 'Operational' state), even if it is not possible to do so in a specific situation.

Resource_Capability

The capability of a resource to assist in a State transition, e.g. having available power to supply to an engine.

State

A condition of an Asset, e.g. 'On' or 'Off', 'Primary' or 'Secondary'.

Transition_Requirement

A requirement to coordinate transitions of Asset(s) between States.

Transition_Solution

A sequence of Transition_Steps that are required to meet the Transition_Requirement, e.g. the need to transition from ‘Off’ to ‘Standby’ to ‘On’.

Transition_Step

A State transition that contributes to achieving the desired end State, e.g. transitioning from ‘Off’ to ‘Standby’.

Transition_Constraint

An externally imposed restriction that limits the Available_Transitions of an Asset under specific circumstances, e.g. preventing the landing gear from being deployed if there is a need to reduce the observability of the Exploiting Platform.

Transition_Capability

The capability of an Asset to perform its Available_Transitions.

5.4.2.2.6 Design Rationale

5.4.2.2.6.1 Assumptions

- A single event may impact multiple system or equipment States depending on how it is viewed.
- An Exploiting Platform's resources will change due to a significant upgrade or when new role fit equipment is installed.

5.4.2.2.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Asset Transitions:

- Interfacing with Deployable Assets - A deployable asset may have States that need to be transitioned in order to fulfil its purpose.

Extensions

- Extensions can be added to the parent component in order to address mission variations and enable special case behaviours as needed.

Exploitation Considerations

- It is expected that different variants or extensions of the component may be utilised, necessitating coordination between these components to handle various Assets or groups of Assets.

5.4.2.2.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component can coordinate the transitions between States of an Exploiting Platform or deployable asset. For example opening a door or configuring an air vehicle for landing. Failure of this component resulting in inadvertent changes may be protected by other components (e.g. Interlocks). However, failure to change the Exploiting Platform configuration or changing the configuration inappropriately when required could result in the Exploiting Platform not being in a safe configuration for the particular scenario. Configuration changes may be required to accommodate normal occurrences (e.g. weather or landing) or failures (e.g. equipment failure or fire). Therefore, failure of this component may lead to uncontrolled flight of the Exploiting Platform and an uncontrolled crash. This would result in loss of the Exploiting Platform and potentially fatalities.

Where instances of this component contribute to hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.2.6.4 Security Considerations

The indicative security classification is O.

This component coordinates the transition between States to enable or maintain the capabilities of the Exploiting Platform, which as a minimum is considered O. However, where the necessary level of understanding of the vehicle infrastructure, performance, capability and, in particular, redundancy is present this may be increased to SNEO. This component is expected to have rigorous confidentiality requirements where the information it holds on infrastructure and redundancy, etc. would allow a much more targeted attack should it be divulged. Loss of integrity or availability of this component may have a detrimental effect on the adaptability of the Exploiting Platform.

The component may be expected to at least partially satisfy security related functions by:

- **Identifying Data Sources**, ensuring transitions are only instigated following input from valid sources.
- **Logging of Security Data** relating to transitions, access requests to different resources or reversionary functions, shut-down, start-up and warm starts, etc.
- **Maintaining Audit Records** relating to Assets throughout the mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** of States against demands, with unexpected transitions or failure to transition being a sign of possible cyber attack.

The component is unlikely to implement Security Enforcing Functions, but may implement the transitions necessary to prevent cyber attacks and malware from taking hold of the system, e.g. switching from a primary Asset that has been compromised to a reversionary Asset. This component would not identify nor understand the cyber attack.

5.4.2.2.7 Services

5.4.2.2.7.1 Service Definitions

5.4.2.2.7.1.1 Transition_Requirement

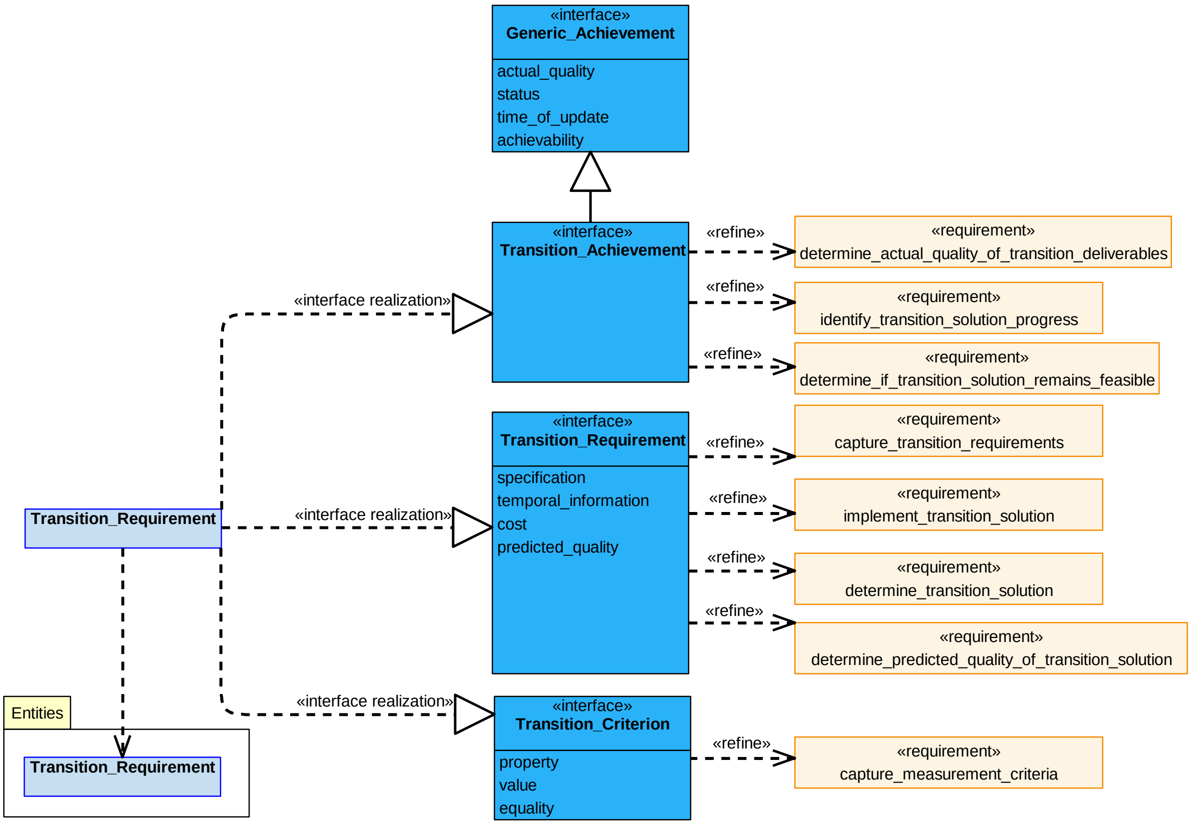


Figure 60: Transition_Requirement Service Definition

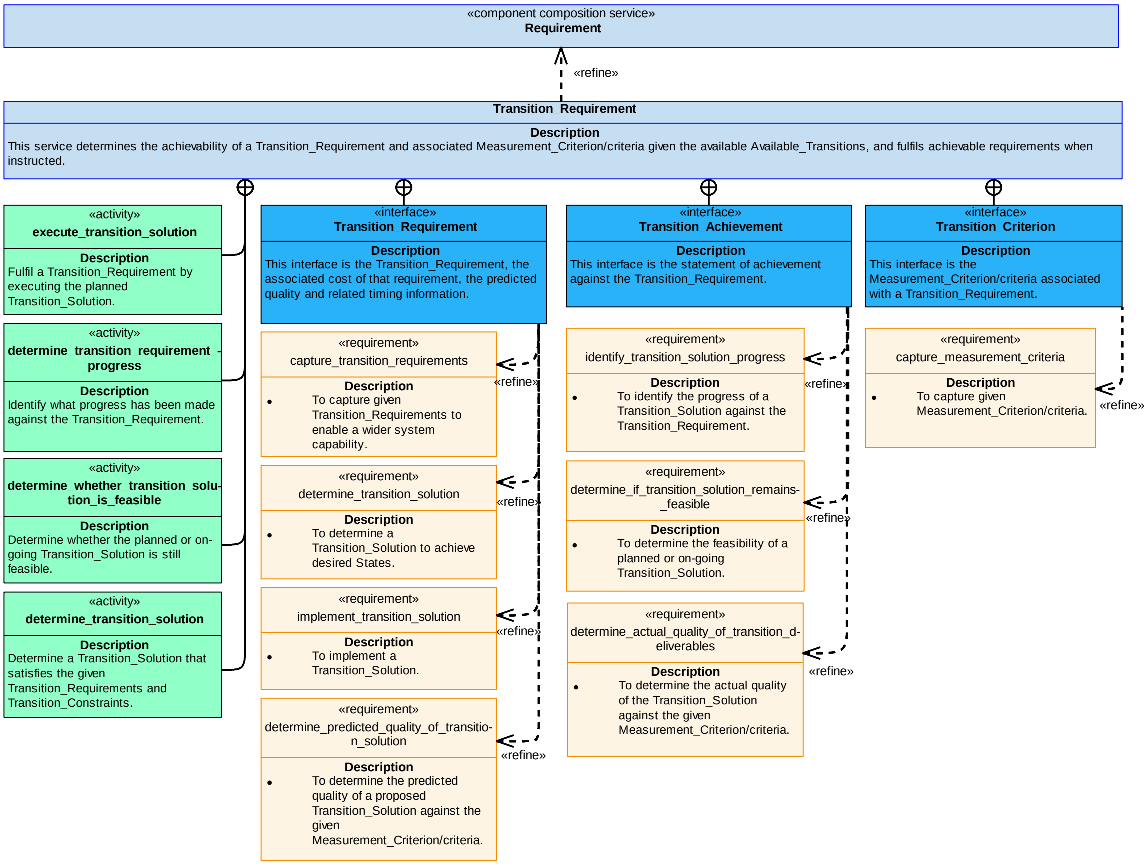

«component composition service»
Requirement
«refine»
Transition_Requirement
Description
This service determines the achievability of a Transition_Requirement and associated Measurement_Criterion/criteria given the available Available_Transitions, and fulfils achievable requirements when instructed.
«activity»
execute_transition_solution
Description
Fulfil a Transition_Requirement by executing the planned Transition_Solution.
«activity»
determine_transition_requirement_progress
Description
Identify what progress has been made against the Transition_Requirement.
«activity»
determine_whether_transition_solution_is_feasible
Description
Determine whether the planned or on-going Transition_Solution is still feasible.
«activity»
determine_transition_solution
Description
Determine a Transition_Solution that satisfies the given Transition_Requirements and Transition_Constraints.
«interface»
Transition_Requirement
Description
This interface is the Transition_Requirement, the predicted associated cost of that requirement, the predicted quality and related timing information.
«requirement»
capture_transition_requirements
Description
To capture given Transition_Requirements to enable a wider system capability.
«refine»
«requirement»
determine_transition_solution
Description
To determine a Transition_Solution to achieve desired States.
«refine»
«requirement»
implement_transition_solution
Description
To implement a Transition_Solution.
«refine»
«requirement»
determine_predicted_quality_of_transition_solution
Description
To determine the predicted quality of a proposed Transition_Solution against the given Measurement_Criterion/criteria.
«refine»
«interface»
Transition_Achievement
Description
This interface is the statement of achievement against the Transition_Requirement.
«requirement»
identify_transition_solution_progress
Description
To identify the progress of a Transition_Solution against the Transition_Requirement.
«refine»
«requirement»
determine_if_transition_solution_remains_feasible
Description
To determine the feasibility of a planned or on-going Transition_Solution.
«refine»
«requirement»
determine_actual_quality_of_transition_deliverables
Description
To determine the actual quality of the Transition_Solution against the given Measurement_Criterion/criteria.
«refine»
«interface»
Transition_Criterion
Description
This interface is the Measurement_Criterion/criteria associated with a Transition_Requirement.
«requirement»
capture_measurement_criteria
Description
To capture given Measurement_Criterion/criteria.
«refine»

Figure 61: Transition_Requirement Service Policy

Transition_Requirement

This service determines the achievability of a Transition_Requirement and associated Measurement_Criterion/criteria given the available Available_Transitions, and fulfils achievable requirements when instructed.

Interfaces

Transition_Achievement

This interface is the statement of achievement against the Transition_Requirement.

Transition_Criterion

This interface is the Measurement_Criterion/criteria associated with a Transition_Requirement.

Attributes

property	The property to be measured, e.g. transition time.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Transition_Requirement

This interface is the Transition_Requirement, the associated cost of that requirement, the predicted quality and related timing information.

Attributes

specification	The definition of the Transition_Requirement, e.g. to change the State of an Asset from on to off.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the Transition_Solution, e.g. resources used or time taken.
predicted_quality	How well the planned Transition_Solution is predicted to satisfy the Transition_Requirement.

Activities

execute_transition_solution

Fulfil a Transition_Requirement by executing the planned Transition_Solution.

determine_whether_transition_solution_is_feasible

Determine whether the planned or on-going Transition_Solution is still feasible.

determine_transition_requirement_progress

Identify what progress has been made against the Transition_Requirement.

determine_transition_solution

Determine a Transition_Solution that satisfies the given Transition_Requirements and Transition_Constraints.

5.4.2.2.7.1.2 Transition_Activity

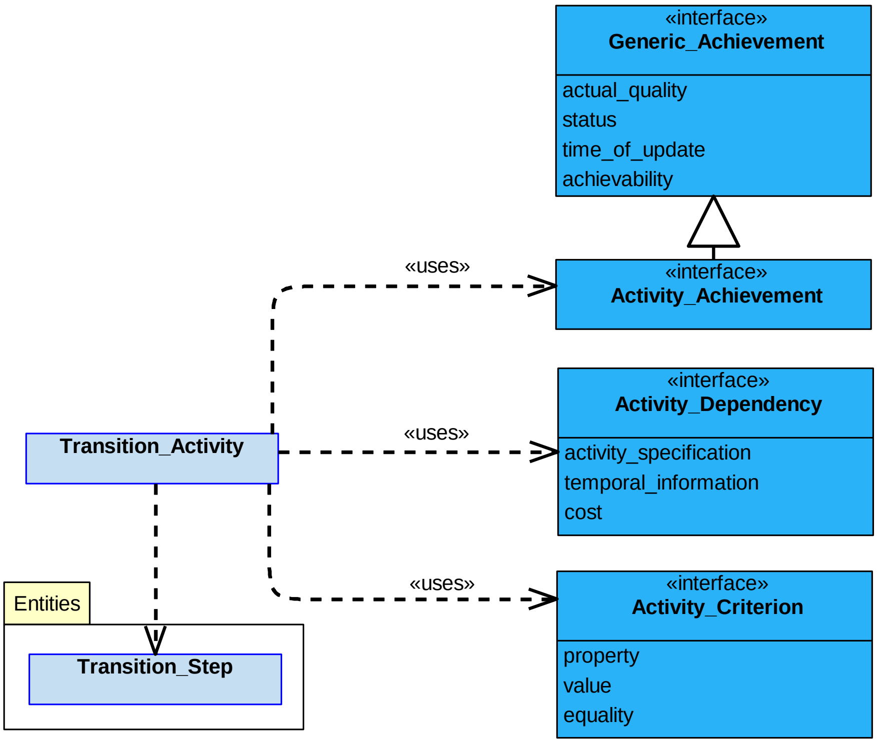


Figure 62: Transition_Activity Service Definition

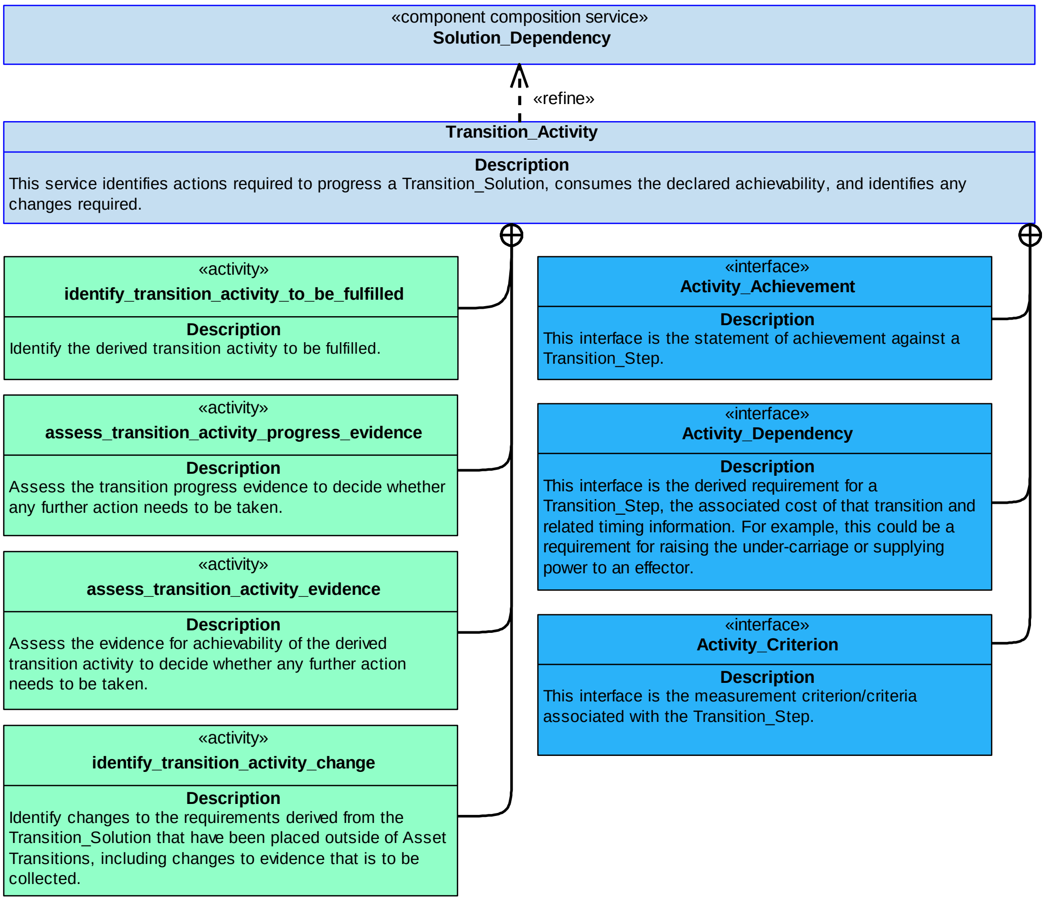

«component composition service»
Solution_Dependency
«refine»
Transition_Activity
Description
This service identifies actions required to progress a Transition_Solution, consumes the declared achievability, and identifies any changes required.
«activity»
identify_transition_activity_to_be_fulfilled
Description
Identify the derived transition activity to be fulfilled.
«activity»
assess_transition_activity_progress_evidence
Description
Assess the transition progress evidence to decide whether any further action needs to be taken.
«activity»
assess_transition_activity_evidence
Description
Assess the evidence for achievability of the derived transition activity to decide whether any further action needs to be taken.
«activity»
identify_transition_activity_change
Description
Identify changes to the requirements derived from the Transition_Solution that have been placed outside of Asset Transitions, including changes to evidence that is to be collected.
«interface»
Activity_Achievement
Description
This interface is the statement of achievement against a Transition_Step.
«interface»
Activity_Dependency
Description
This interface is the derived requirement for a Transition_Step, the associated cost of that transition and related timing information. For example, this could be a requirement for raising the under-carriage or supplying power to an effector.
«interface»
Activity_Criterion
Description
This interface is the measurement criterion/criteria associated with the Transition_Step.

Figure 63: Transition_Activity Service Policy

Transition_Activity

This service identifies actions required to progress a Transition_Solution, consumes the declared achievability, and identifies any changes required.

Interfaces

Activity_Dependency

This interface is the derived requirement for a Transition_Step, the associated cost of that transition and related timing information. For example, this could be a requirement for raising the under-carriage or supplying power to an effector.

Attributes

activity_specification	The derived activity within a set of activities that are to be coordinated in order to achieve a desired State, e.g. enabling the provision of power, the movement of fluids or a physical change to the Exploiting Platform.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the activity_specification, e.g. resources used or time taken.

Activity_Criterion

This interface is the measurement criterion/criteria associated with the Transition_Step.

Attributes

property	The property to be measured, e.g. transition time.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activity_Achievement

This interface is the statement of achievement against a Transition_Step.

Activities

assess_transition_activity_progress_evidence

Assess the transition progress evidence to decide whether any further action needs to be taken.

assess_transition_activity_evidence

Assess the evidence for achievability of the derived transition activity to decide whether any further action needs to be taken.

identify_transition_activity_to_be_fulfilled

Identify the derived transition activity to be fulfilled.

identify_transition_activity_change

Identify changes to the requirements derived from the Transition_Solution that have been placed outside of Asset Transitions, including changes to evidence that is to be collected.

5.4.2.2.7.1.3 Asset_State_Information

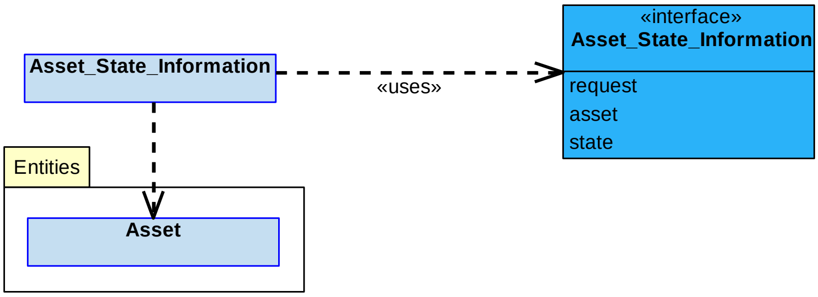


Figure 64: Asset_State_Information Service Definition

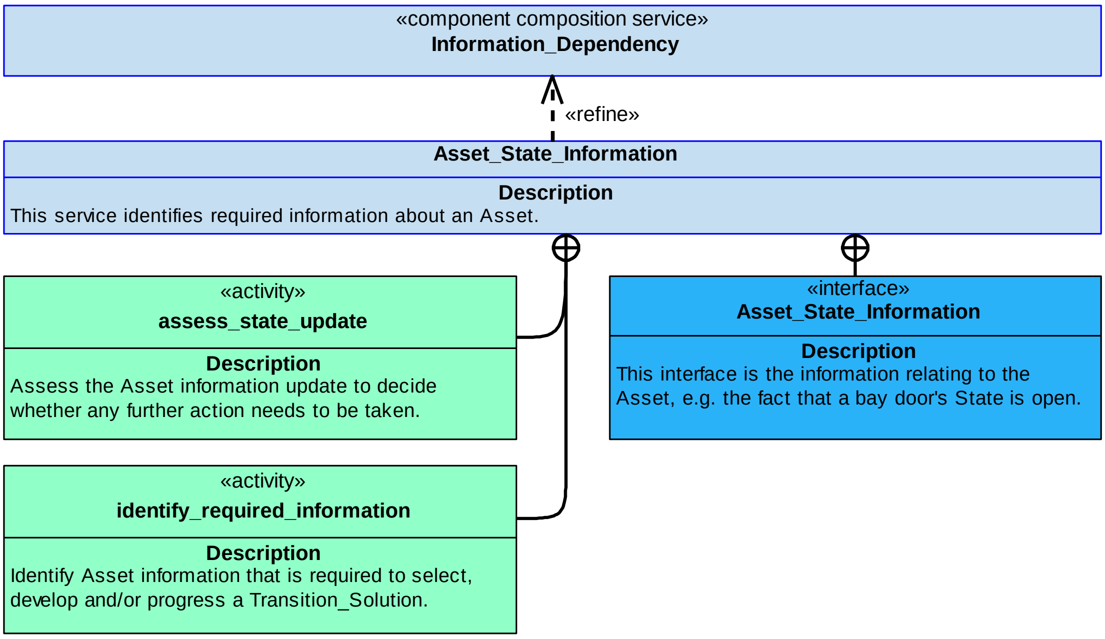

«component composition service»
Information_Dependency
«refine»
Asset_State_Information
Description
This service identifies required information about an Asset.
«activity»
assess_state_update
Description
Assess the Asset information update to decide whether any further action needs to be taken.
«interface»
Asset_State_Information
Description
This interface is the information relating to the Asset, e.g. the fact that a bay door's State is open.
«activity»
identify_required_information
Description
Identify Asset information that is required to select, develop and/or progress a Transition_Solution.

Figure 65: Asset_State_Information Service Policy

Asset_State_Information

This service identifies required information about an Asset.

<u>**Interface**</u>

Asset_State_Information

This interface is the information relating to the Asset, e.g. the fact that a bay door's State is open.

<u>Attributes</u>

request	The request for information about an Asset.
asset	A system element (e.g. bay doors, sensors, software configurations) or deployable asset that can transition from one State to another.
state	A condition of an Asset, e.g. on or off, primary or secondary.

<u>**Activities**</u>

assess_state_update

Assess the Asset information update to decide whether any further action needs to be taken.

identify_required_information

Identify Asset information that is required to select, develop and/or progress a Transition_Solution.

5.4.2.2.7.1.4 Constraint

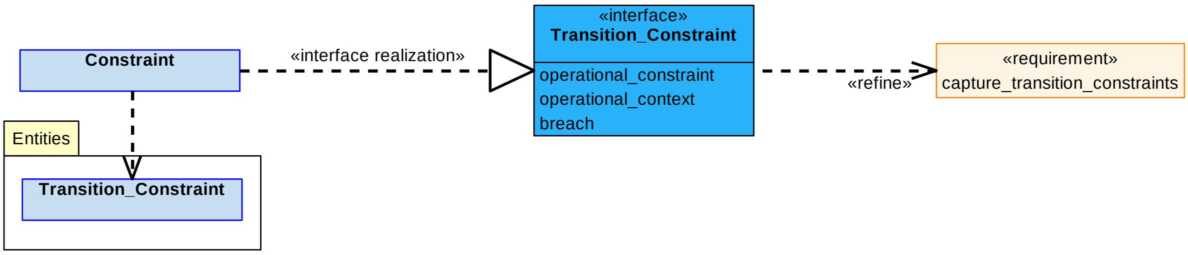


Figure 66: Constraint Service Definition

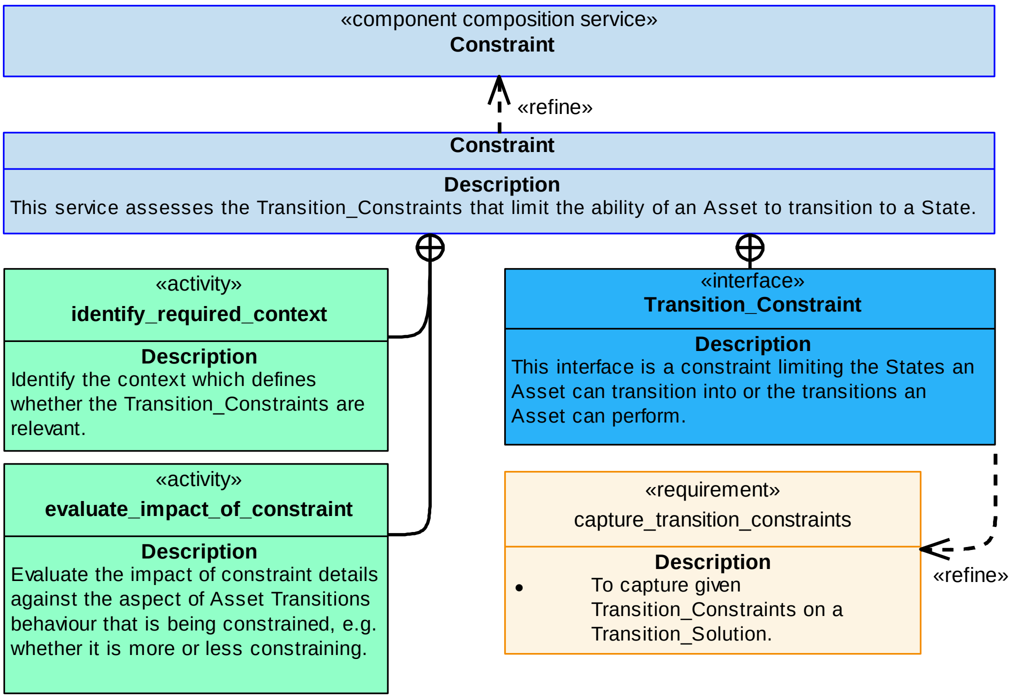

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Transition_Constraints that limit the ability of an Asset to transition to a State.
«activity»
identify_required_context
Description
Identify the context which defines whether the Transition_Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of constraint details against the aspect of Asset Transitions behaviour that is being constrained, e.g. whether it is more or less constraining.
«interface»
Transition_Constraint
Description
This interface is a constraint limiting the States an Asset can transition into or the transitions an Asset can perform.
«requirement»
capture_transition_constraints
Description
To capture given Transition_Constraints on a Transition_Solution.
«refine»

Figure 67: Constraint Service Policy

Constraint

This service assesses the Transition_Constraints that limit the ability of an Asset to transition to a State.

Interface

Transition_Constraint

This interface is a constraint limiting the States an Asset can transition into or the transitions an Asset can perform.

Attributes

operational_constraint	A constraint which prevents a Available_Transition, e.g. to prevent a bay door from being opened so that RCS is minimised.
operational_context	The context in which the constraint is applicable, e.g. operating on the ground.

breach	A statement that the Transition_Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of constraint details against the aspect of Asset Transitions behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Transition_Constraints are relevant.

5.4.2.2.7.1.5 Capability

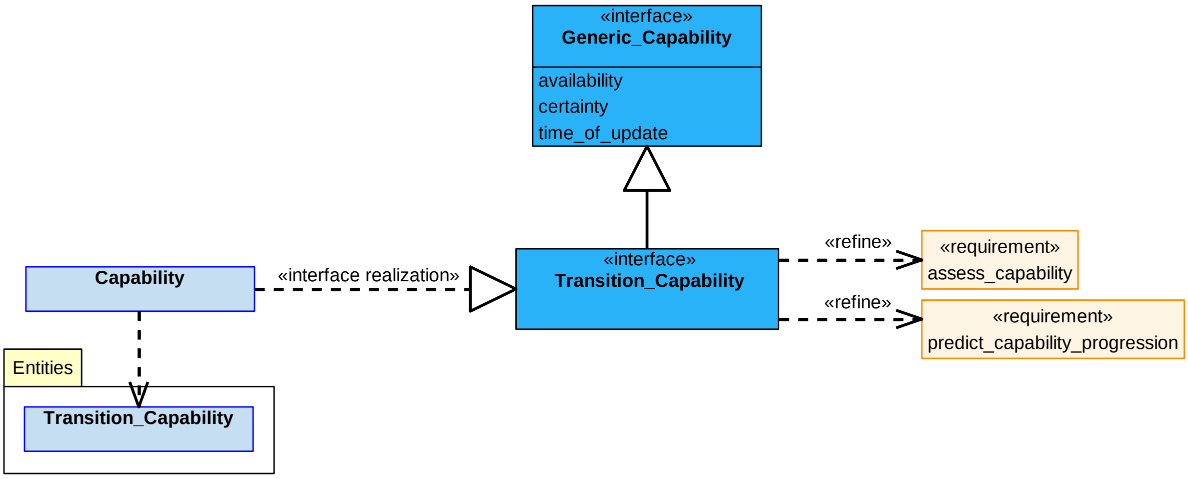


Figure 68: Capability Service Definition

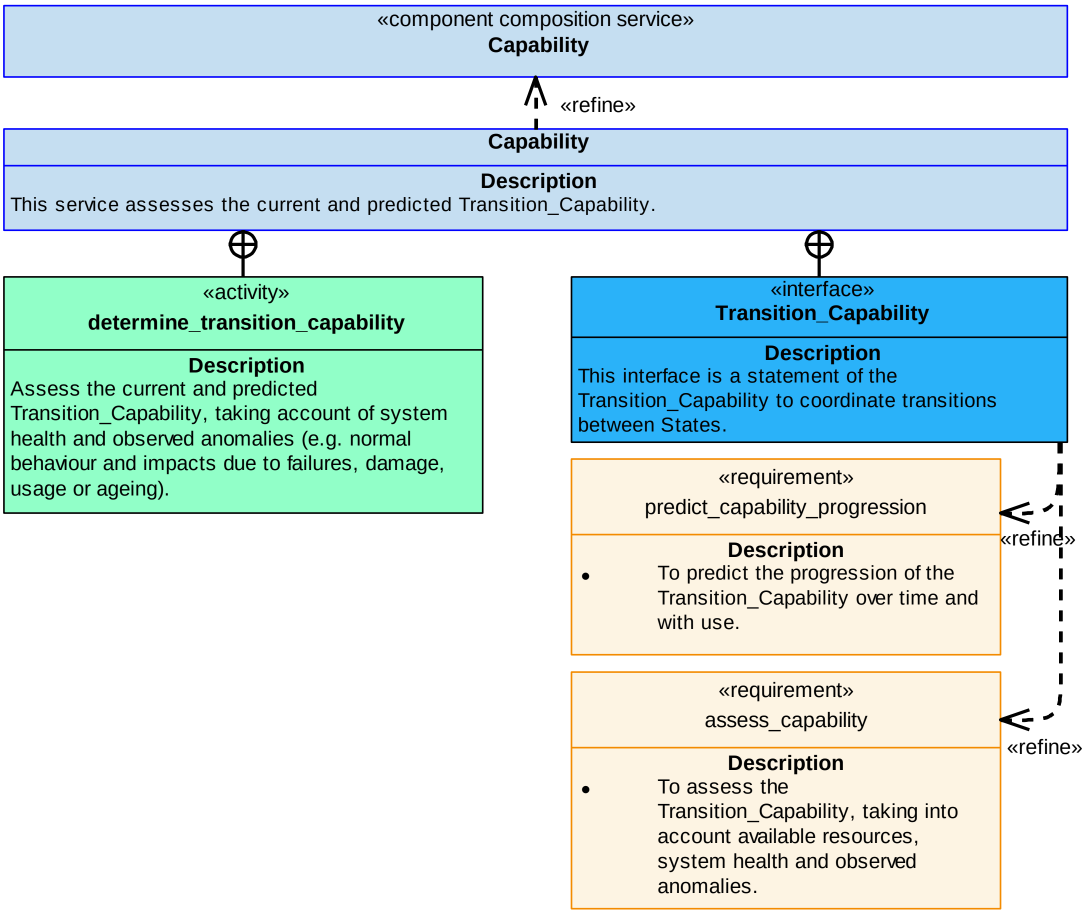

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Transition_Capability.
«activity»
determine_transition_capability
Description
Assess the current and predicted Transition_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Transition_Capability
Description
This interface is a statement of the Transition_Capability to coordinate transitions between States.
«requirement»
predict_capability_progression
Description
• To predict the progression of the Transition_Capability over time and with use.
«refine»
«requirement»
assess_capability
Description
• To assess the Transition_Capability, taking into account available resources, system health and observed anomalies.
«refine»

Figure 69: Capability Service Policy

Capability

This service assesses the current and predicted Transition_Capability.

<u>Interface</u>

Transition_Capability

This interface is a statement of the Transition_Capability to coordinate transitions between States.

<u>Activity</u>

determine_transition_capability

Assess the current and predicted Transition_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.2.7.1.6 Capability_Evidence

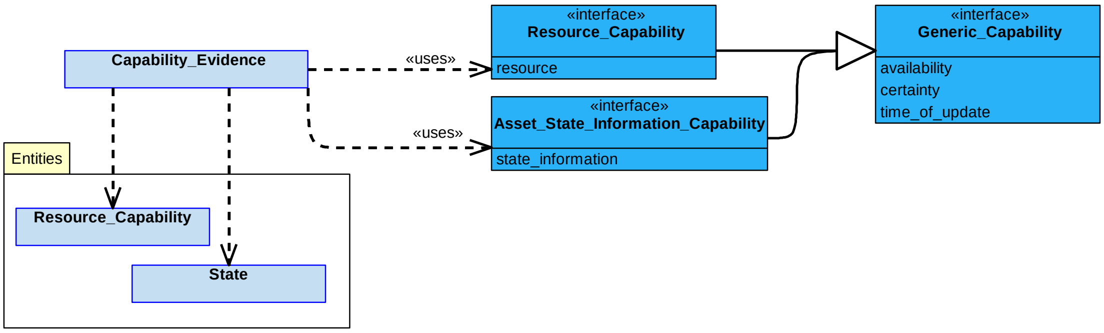


Figure 70: Capability_Evidence Service Definition

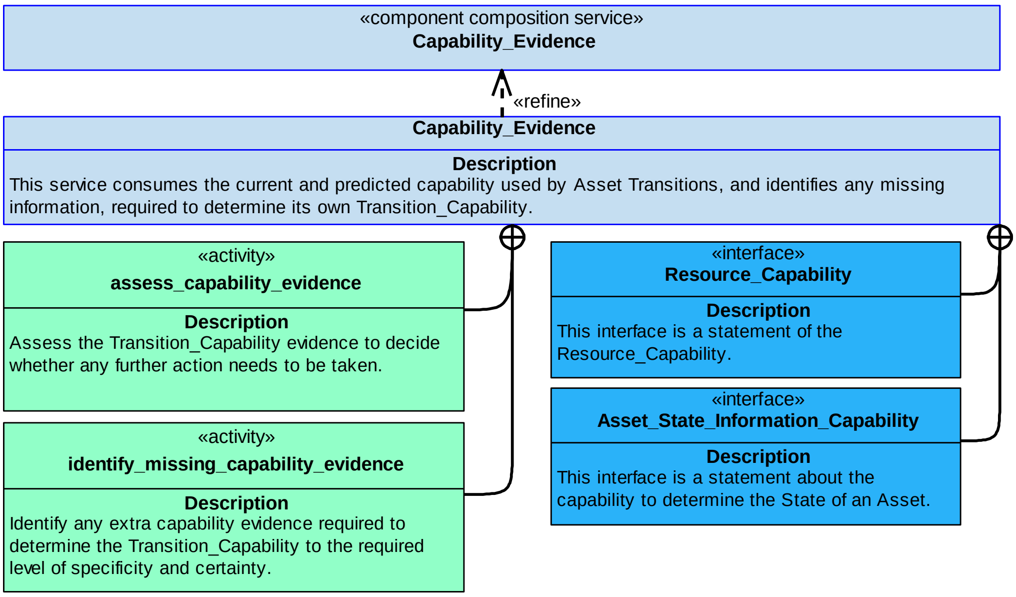

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted capability used by Asset Transitions, and identifies any missing information, required to determine its own Transition_Capability.
«activity»
assess_capability_evidence
Description
Assess the Transition_Capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Transition_Capability to the required level of specificity and certainty.
«interface»
Resource_Capability
Description
This interface is a statement of the Resource_Capability.
«interface»
Asset_State_Information_Capability
Description
This interface is a statement about the capability to determine the State of an Asset.

Figure 71: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capability used by Asset Transitions, and identifies any missing information, required to determine its own Transition_Capability.

<u>**Interfaces**</u>

Resource_Capability

This interface is a statement of the Resource_Capability.

<u>Attribute</u>

resource	The resource for which the capability assessment is provided.

Asset_State_Information_Capability

This interface is a statement about the capability to determine the State of an Asset.

Attribute

state_information	The Asset State information for which the capability assessment is provided.

Activities

assess_capability_evidence

Assess the Transition_Capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Transition_Capability to the required level of specificity and certainty.

5.4.2.2.7.2 Service Dependencies

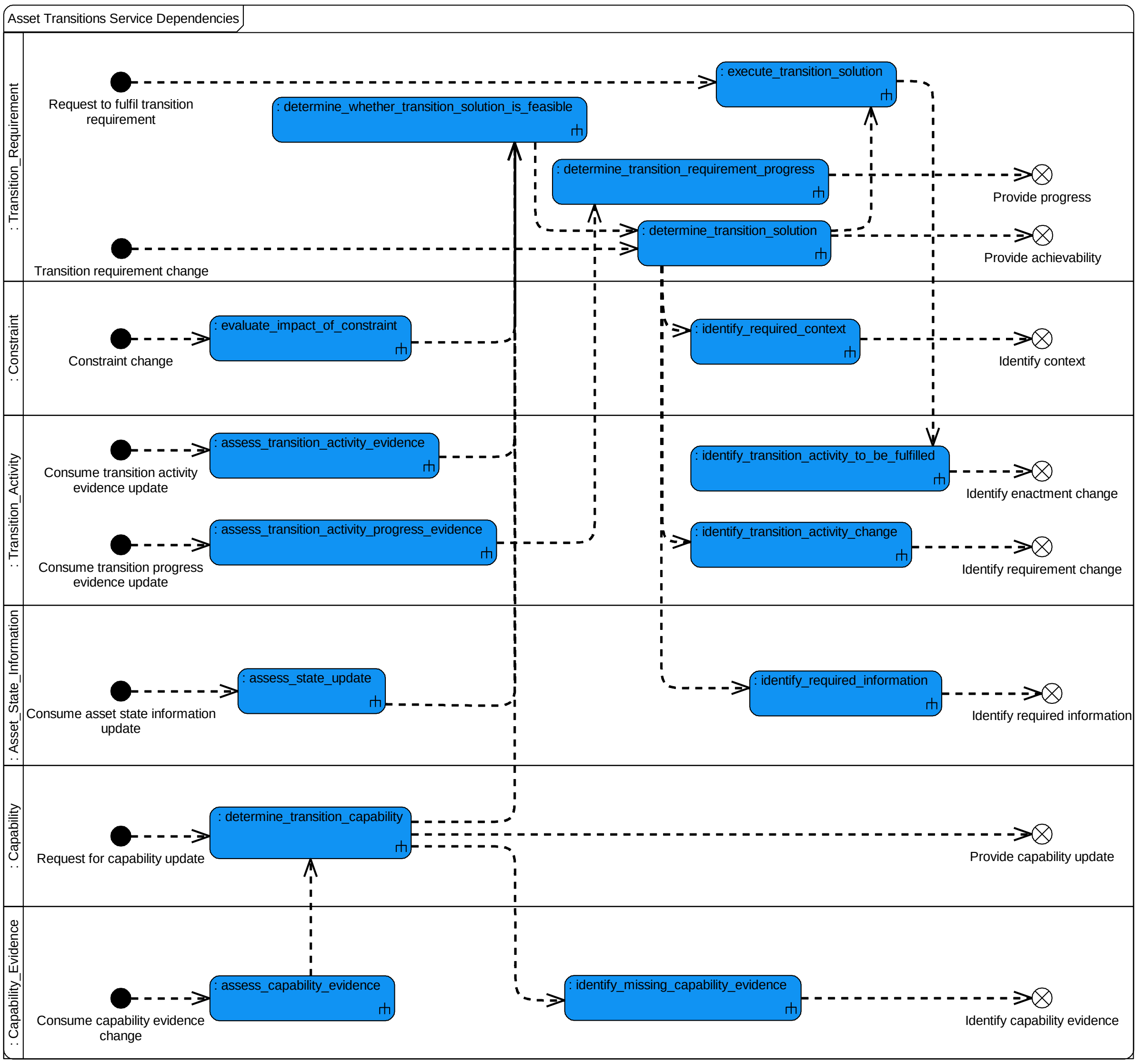

Asset Transitions Service Dependencies
: Transition_Requirement
: Constraint
: Transition_Activity
: Asset_State_Information
: Capability
: Capability_Evidence
Request to fulfil transition requirement
: execute_transition_solution
: determine_whether_transition_solution_is_feasible
: determine_transition_requirement_progress
Provide progress
: determine_transition_solution
Provide achievability
Transition requirement change
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume transition activity evidence update
: assess_transition_activity_evidence
: identify_transition_activity_to_be_fulfilled
Identify enactment change
Consume transition progress evidence update
: assess_transition_activity_progress_evidence
: identify_transition_activity_change
Identify requirement change
Consume asset state information update
: assess_state_update
: identify_required_information
Identify required information
Request for capability update
: determine_transition_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 72: Asset Transitions Service Dependencies

5.4.2.3 Authorisation

5.4.2.3.1 Role

The role of Authorisation is to obtain and manage authorisation for the execution of activities.

5.4.2.3.2 Overview

Control Architecture

Authorisation is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When an Authorisation requester plans an activity that requires explicit Authorisation, it will request to obtain such Authorisation. This component will follow the Authorisation_Policy to identify the necessary Authorisation and the Authoriser(s) that can grant it. The Authorisation component will send requests to the Authoriser(s). When all the requests have been approved, the Authorisation component will inform the Authorisation requester. The component monitors the validity of the approved (or planned) Authorisation until it is no longer required.

Examples of Use

The Authorisation component may be required for activities where a level of approval is required, e.g. weapon release or entering restricted zones.

5.4.2.3.3 Service Summary

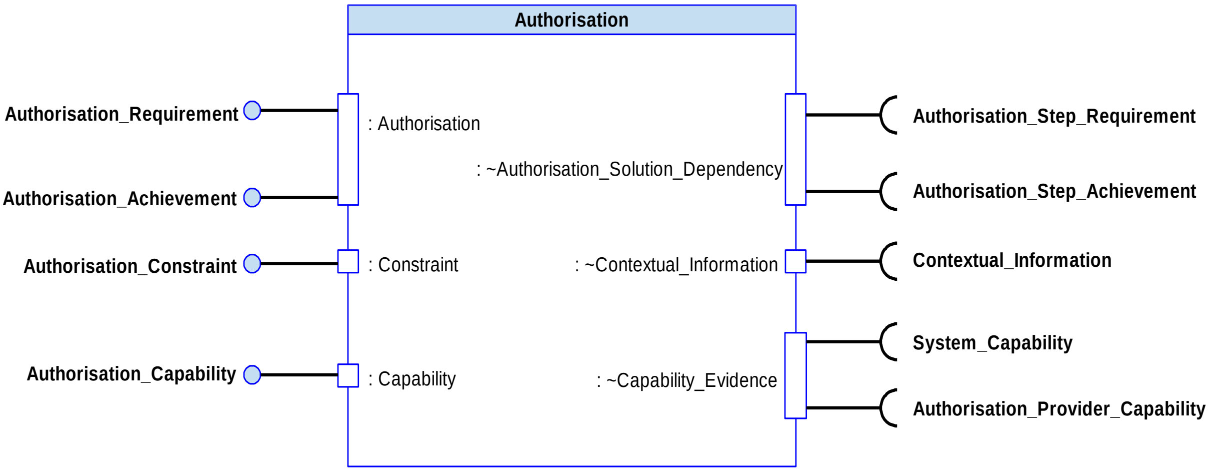

Authorisation
Authorisation_Requirement
Authorisation_Achievement
Authorisation_Constraint
Authorisation_Capability
: Authorisation
: ~Authorisation_Solution_Dependency
: Constraint
: ~Contextual_Information
: Capability
: ~Capability_Evidence
Authorisation_Step_Requirement
Authorisation_Step_Achievement
Contextual_Information
System_Capability
Authorisation_Provider_Capability

Figure 73: Authorisation Service Summary

5.4.2.3.4 Responsibilities

capture_requirements_for_authorisations

- To capture Authorisation_Requirements.

capture_constraints_for_authorisations

- To capture Constraints related to obtaining Authorisations.

determine_authorisation_solution

- To determine the Steps required to obtain an Authorisation that meets the given Authorisation_Requirements, using available Authorisers in accordance with the applicable Authorisation_Policy.

determine_authorisation_policy

- To determine the conditions under which a Authorisation_Policy will be active.

determine_permitted_authorisers

- To determine which Authorisers are permitted (e.g. as defined in the Authorisation_Policy) to provide a given Authorisation.

determine_if_authorisation_remains_feasible

- To determine the feasibility of a planned or on-going Authorisation.

capture_authoriser_availability

- To capture available Authorisers (e.g. Authorisers to which or whom there is a communications link).

coordinate_authorisation_solution

- To coordinate the execution of the Steps required to obtain Authorisation.

identify_progress_of_authorisation

- To identify the progress of an Authorisation against the Authorisation_Requirement.

determine_solution_dependencies

- To determine dependencies required to support Authorisation or a Step of the solution.

assess_authorisation_capability

- To assess the Capability to obtain Authorisation taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Capability assessment of the Authorisation component (e.g. communication capability).

predict_capability_progression

- To predict the progression of the Authorisation component's Capability over time and with use.

5.4.2.3.5 Subject Matter Semantics

The subject matter of Authorisation is the conditions that govern explicit Authorisation of activities: when it is required, when it is valid (including any relationships between authorisations), and how to obtain it.

Exclusions

The subject matter of Authorisation does not include:

- The calculation of dynamic limits and allowable activities.

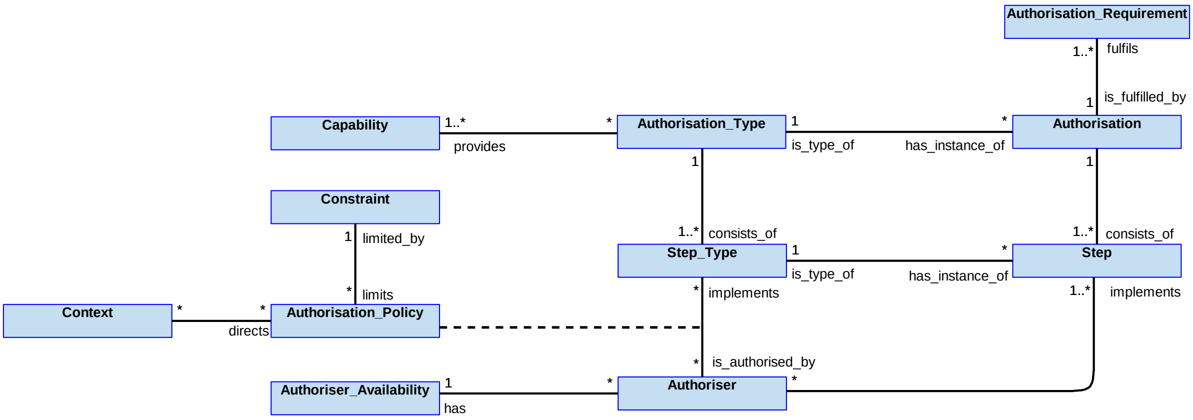


Figure 74: Authorisation Semantics

5.4.2.3.5.1 Entities

Authorisation

Approval to perform activities and the status of that approval, including limitations on its validity.

Authorisation_Policy

A set of rules defining the Authorisers allowed to authorise a Step_Type.

Authorisation_Requirement

A requirement to obtain Authorisation for an activity or group of activities.

Authorisation_Type

A type of Authorisation (e.g. release a weapon, a report, or a piece of data).

Authoriser

An entity that is able to provide an Authorisation.

Authoriser_Availability

Whether an Authoriser is available to provide authorisation.

Capability

The capability to request different Authorisation_Types.

Constraint

A limitation on the applicability of an Authorisation_Policy.

Context

Information about the conditions in which an Authorisation_Policy will be active, e.g. within controlled or uncontrolled airspace.

Step

A step in the lifecycle of Authorisation, by which approval will be obtained.

Step_Type

A type of Step (e.g. inform a role or request to a role).

5.4.2.3.6 Design Rationale

5.4.2.3.6.1 Assumptions

- A single act of Authorisation may be applied to a group of related activities, e.g. authorising an attack plan could authorise each activity that makes up that plan.
- The process for granting Authorisation may vary according to what type of authorisation is being requested and the types of activities it authorises.
- Explicit Authorisation by an operator will be required for a range of autonomous activities, others may be granted a general Authorisation within the rules embedded within the design.

5.4.2.3.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Authorisation:

- Data Driving - This component has been designed to permit data driving of a variable authorisation process (i.e. Authorisation_Types, Authorisers, Step_Types, Constraints and Authorisation_Policy).
- Autonomy - This component enables a clear understanding of what is authorised and under what conditions, which is critical for the implementation of autonomy.
- Recording and Logging - the process for obtaining Authorisation will require logging of Authorisers for security purposes.

Extensions

- It is unlikely that extensions will be appropriate as the basic methods of determining the required Authorisation are not likely to change.

Exploitation Considerations

- Explicit Authorisation can be granted in advance (e.g. during mission planning) or as required in response to events occurring during a mission.
- Context (as described in the Autonomy PYRAMID concept) is critical for the Authorisation of activities by this component.

5.4.2.3.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- The activities that this component could provide Authorisation for would cover many air system functions and would include safety related functions such as authorised stores release or jettison areas, routing, disabling protection functions (e.g. disabling Mode C transponders, ACAS and radio transceivers) and overriding contingency actions required to maintain safety in the event of failures (e.g. continue mission rather than RTB).
- If this component fails then activities could be authorised that result in catastrophic accidents. Where an activity could result in a catastrophic outcome it is expected that additional barriers (within the system) would also prevent the activity taking place. However, DAL A is considered appropriate for this component as it allows other, more complex components, to be DAL C.

Where instances of this component contribute to hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.3.6.4 Security Considerations

The indicative security classification is O-S.

Whilst, in its own right, the component is thought unlikely to be above O-S, it will be required to authorise those activities performed autonomously by the system and may therefore need some access to higher classifications of data, e.g. pertaining to the mission plan, which will raise its classification. Authorisations may be required across the security domains of the Exploiting Platform and communication by instances in different domains may be required, depending on the architecture implemented.

As the component provides both safety and mission related authorisations this component is considered a subject of interest for an adversary and a likely target for a cyber attack; additional protection should be provided to ensure continued confidentiality and availability, and particularly for the integrity and authenticity of requests for and provision of the authorisation.

The component is expected to satisfy security related functions relating to:

- **Identifying Data Sources** of external authorisation.
- **Logging of Security Data** of authorisation successes and failures for later forensic examination.
- **Maintaining Audit Records** to support non-repudiation of pre-authorisations and authorisations granted in the course of a mission.

- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

Where pre-authorisation can be provided, this component is **Supporting Secure Remote Operation**, i.e. the UAV does not need to seek authorisation over communications links.

The component is expected to perform some aspects of security enforcing functions relating to:

- **Verifying Integrity of Data** where the data has been provided by external authorisers through a level of authentication checks for the data/external authorisers.

5.4.2.3.7 Services

5.4.2.3.7.1 Service Definitions

5.4.2.3.7.1.1 Authorisation

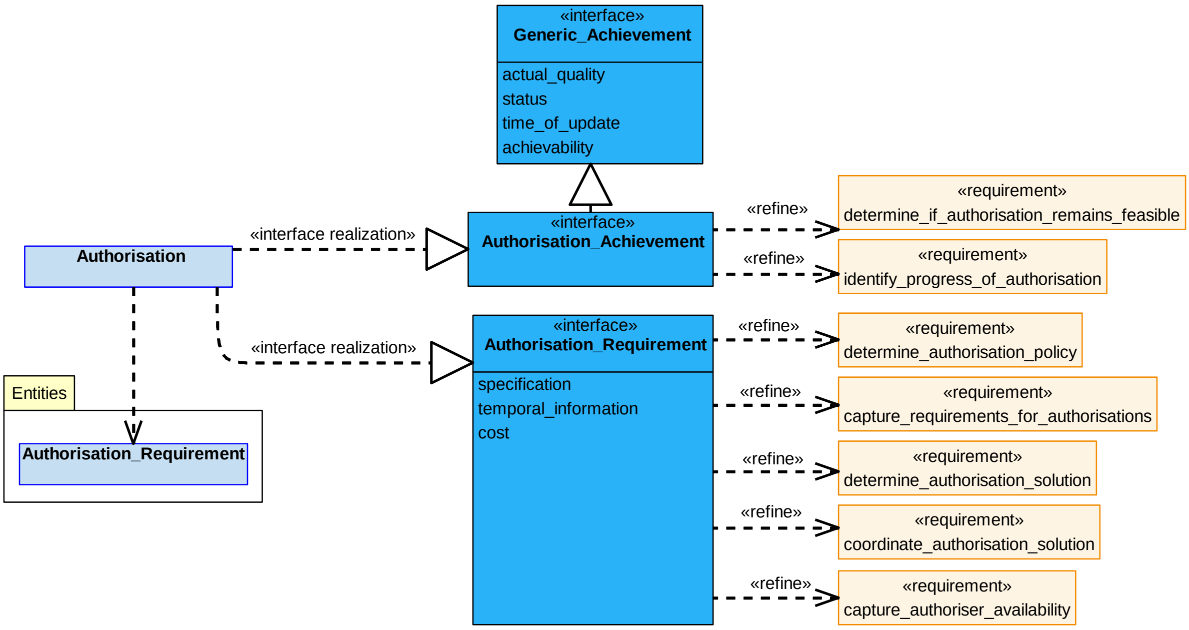
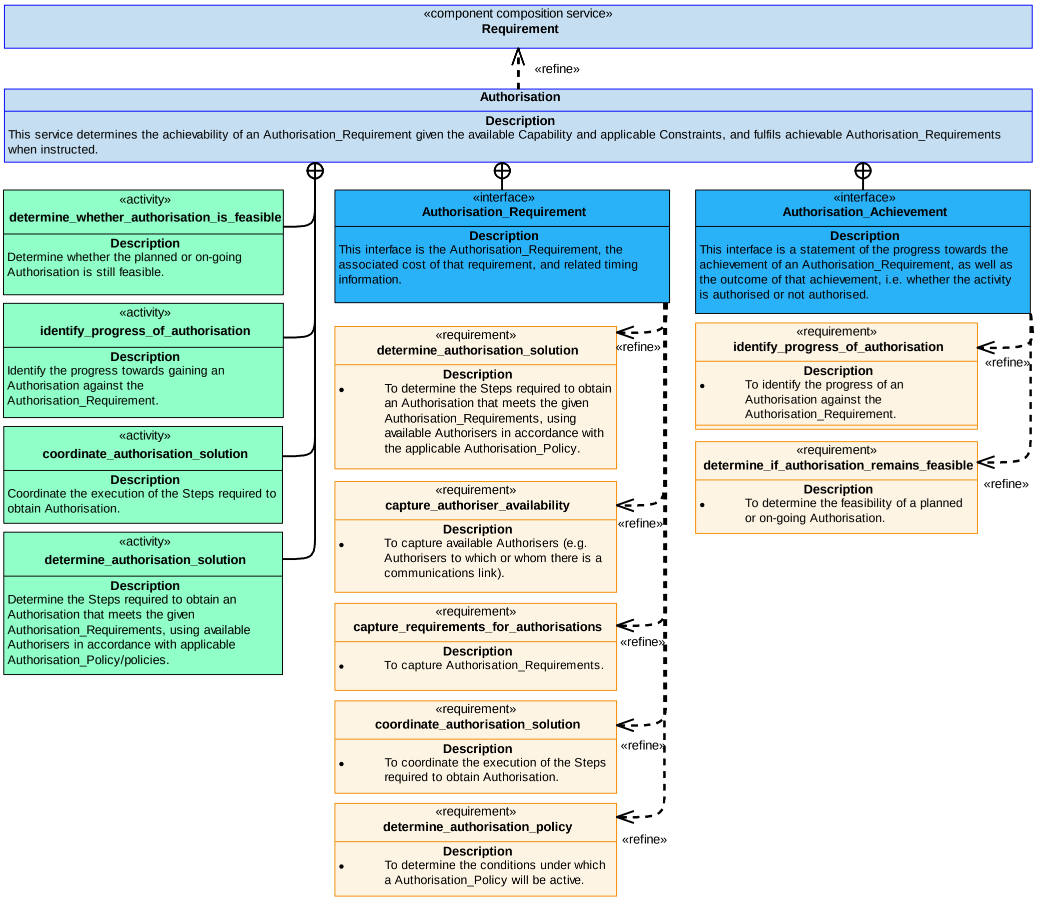


Figure 75: Authorisation Service Definition


«component composition service»
Requirement
«refine»
Authorisation
Description
«activity»
determine_whether_authorisation_is_feasible
Description
«activity»
identify_progress_of_authorisation
Description
«activity»
coordinate_authorisation_solution
Description
«activity»
determine_authorisation_solution
Description
«interface»
Authorisation_Requirement
Description
«interface»
Authorisation_Achievement
Description
«requirement»
determine_authorisation_solution
Description
«refine»
«requirement»
capture_authoriser_availability
Description
«refine»
«requirement»
capture_requirements_for_authorisations
Description
«refine»
«requirement»
coordinate_authorisation_solution
Description
«refine»
«requirement»
determine_authorisation_policy
Description
«refine»
«requirement»
identify_progress_of_authorisation
Description
«refine»
«requirement»
determine_if_authorisation_remains_feasible
Description
«refine»

Authorisation: This service determines the achievability of an Authorisation_Requirement given the available Capability and applicable Constraints, and fulfils achievable Authorisation_Requirements when instructed.

determine_whether_authorisation_is_feasible: Determine whether the planned or on-going Authorisation is still feasible.

identify_progress_of_authorisation: Identify the progress towards gaining an Authorisation against the Authorisation_Requirement.

coordinate_authorisation_solution: Coordinate the execution of the Steps required to obtain Authorisation.

determine_authorisation_solution: Determine the Steps required to obtain an Authorisation that meets the given Authorisation_Requirements, using available Authorisers in accordance with applicable Authorisation_Policy/policies.

Authorisation_Requirement: This interface is the Authorisation_Requirement, the associated cost of that requirement, and related timing information.

Authorisation_Achievement: This interface is a statement of the progress towards the achievement of an Authorisation_Requirement, as well as the outcome of that achievement, i.e. whether the activity is authorised or not authorised.

determine_authorisation_solution:
- To determine the Steps required to obtain an Authorisation that meets the given Authorisation_Requirements, using available Authorisers in accordance with the applicable Authorisation_Policy.

capture_authoriser_availability:
- To capture available Authorisers (e.g. Authorisers to which or whom there is a communications link).

capture_requirements_for_authorisations:
- To capture Authorisation_Requirements.

coordinate_authorisation_solution:
- To coordinate the execution of the Steps required to obtain Authorisation.

determine_authorisation_policy:
- To determine the conditions under which a Authorisation_Policy will be active.

identify_progress_of_authorisation:
- To identify the progress of an Authorisation against the Authorisation_Requirement.

determine_if_authorisation_remains_feasible:
- To determine the feasibility of a planned or on-going Authorisation.

Figure 76: Authorisation Service Policy

Authorisation

This service determines the achievability of an Authorisation_Requirement given the available Capability and applicable Constraints, and fulfils achievable Authorisation_Requirements when instructed.

Interfaces

Authorisation_Requirement

This interface is the Authorisation_Requirement, the associated cost of that requirement, and related timing information.

Attributes

specification	The definition of the Authorisation_Requirement, e.g. to authorise the deployment of a specific weapon in a particular location at a specified time.
temporal_information	Information covering timing, such as the time period for when Authorisation is required, has been obtained for, or if the achieved Authorisation is a subset of the required period.
cost	The cost of executing the solution, for example: resources used, time taken.

Authorisation_Achievement

This interface is a statement of the progress towards the achievement of an Authorisation_Requirement, as well as the outcome of that achievement, i.e. whether the activity is authorised or not authorised.

Activities

identify_progress_of_authorisation

Identify the progress towards gaining an Authorisation against the Authorisation_Requirement.

determine_authorisation_solution

Determine the Steps required to obtain an Authorisation that meets the given Authorisation_Requirements, using available Authorisers in accordance with applicable Authorisation_Policy/policies.

coordinate_authorisation_solution

Coordinate the execution of the Steps required to obtain Authorisation.

determine_whether_authorisation_is_feasible

Determine whether the planned or on-going Authorisation is still feasible.

5.4.2.3.7.1.2 Authorisation_Solution_Dependency

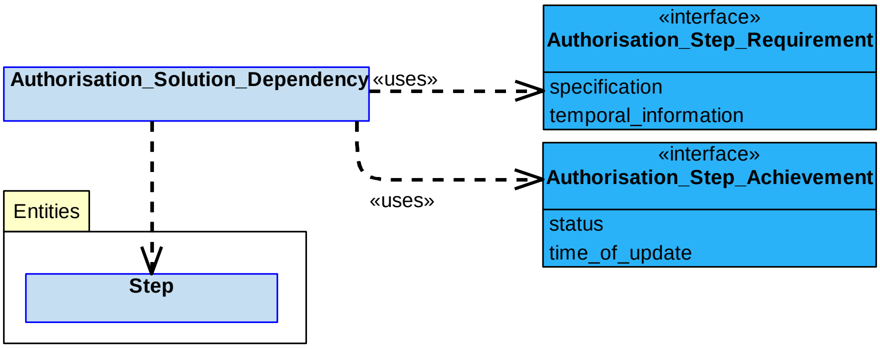
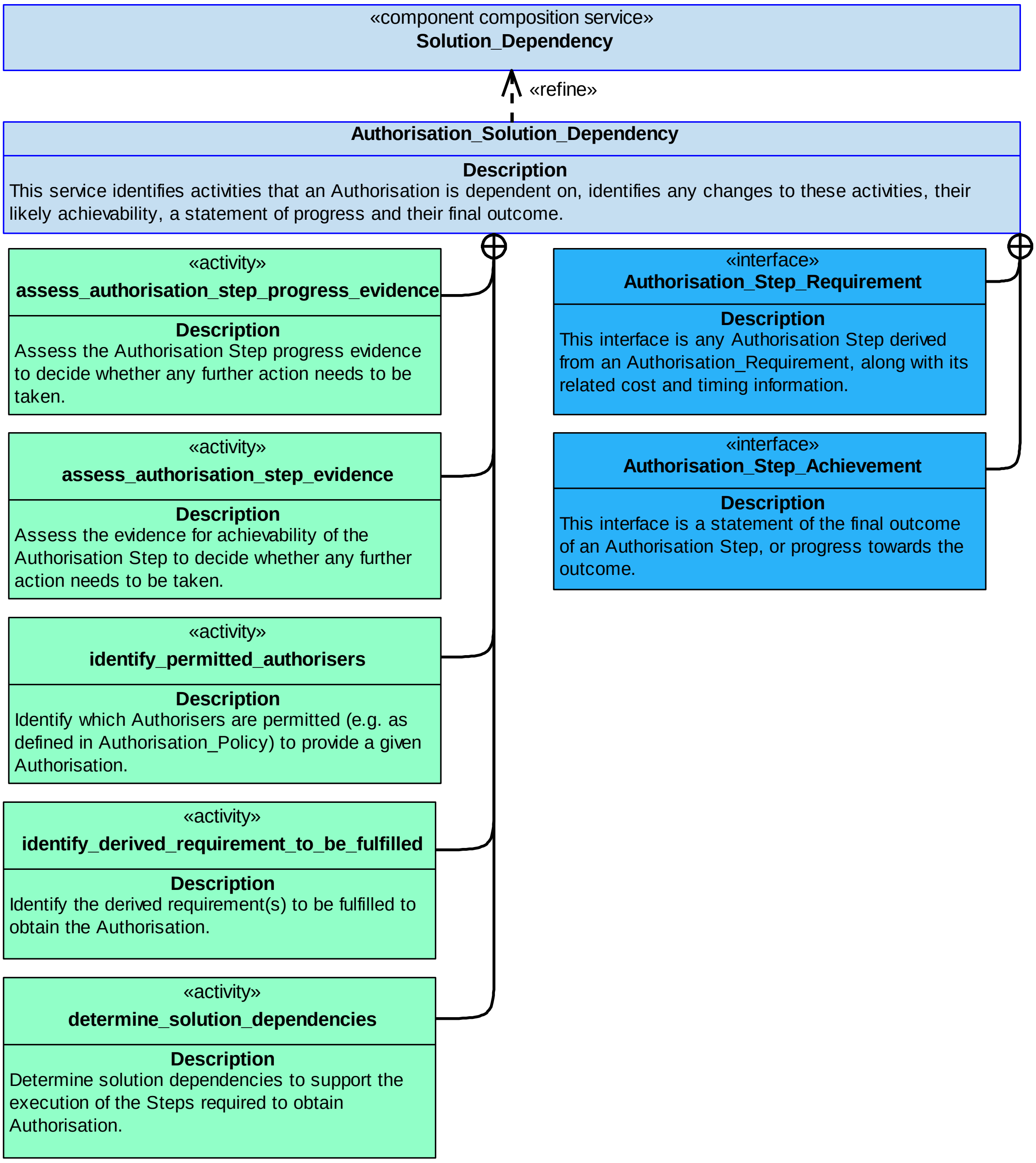


Figure 77: Authorisation_Solution_Dependency Service Definition


«component composition service»
Solution_Dependency
«refine»
Authorisation_Solution_Dependency
Description
«activity»
assess_authorisation_step_progress_evidence
Description
«activity»
assess_authorisation_step_evidence
Description
«activity»
identify_permitted_authorisers
Description
«activity»
identify_derived_requirement_to_be_fulfilled
Description
«activity»
determine_solution_dependencies
Description
«interface»
Authorisation_Step_Requirement
Description
«interface»
Authorisation_Step_Achievement
Description

This service identifies activities that an Authorisation is dependent on, identifies any changes to these activities, their likely achievability, a statement of progress and their final outcome.

Assess the Authorisation Step progress evidence to decide whether any further action needs to be taken.

Assess the evidence for achievability of the Authorisation Step to decide whether any further action needs to be taken.

Identify which Authorisers are permitted (e.g. as defined in Authorisation_Policy) to provide a given Authorisation.

Identify the derived requirement(s) to be fulfilled to obtain the Authorisation.

Determine solution dependencies to support the execution of the Steps required to obtain Authorisation.

This interface is any Authorisation Step derived from an Authorisation_Requirement, along with its related cost and timing information.

This interface is a statement of the final outcome of an Authorisation Step, or progress towards the outcome.

Figure 78: Authorisation_Solution_Dependency Service Policy

Authorisation_Solution_Dependency

This service identifies activities that an Authorisation is dependent on, identifies any changes to these activities, their likely achievability, a statement of progress and their final outcome.

<u>Interfaces</u>

Authorisation_Step_Requirement

This interface is any Authorisation Step derived from an Authorisation_Requirement, along with its related cost and timing information.

Attributes

specification	The definition of the Authorisation Step requirement, e.g. the activity needing to be achieved to allow an Authorisation, which could be an approval from a human authorised operator, or confirmation from another part of the system of some state or condition.
temporal_information	Information covering timing, such as start and end times within which the Step is required, is in place for, or if the achieved step is for a subset of the required period.

Authorisation_Step_Achievement

This interface is a statement of the final outcome of an Authorisation Step, or progress towards the outcome.

Attributes

status	A high-level representation of the achievement in relation to the authorisation (e.g. not started, achievable, in progress, approved, or refused).
time_of_update	The time at which an achievement update occurred.

Activities

assess_authorisation_step_evidence

Assess the evidence for achievability of the Authorisation Step to decide whether any further action needs to be taken.

assess_authorisation_step_progress_evidence

Assess the Authorisation Step progress evidence to decide whether any further action needs to be taken.

determine_solution_dependencies

Determine solution dependencies to support the execution of the Steps required to obtain Authorisation.

identify_permitted_authorisers

Identify which Authorisers are permitted (e.g. as defined in Authorisation_Policy) to provide a given Authorisation.

identify_derived_requirement_to_be_fulfilled

Identify the derived requirement(s) to be fulfilled to obtain the Authorisation.

5.4.2.3.7.1.3 Contextual_Information

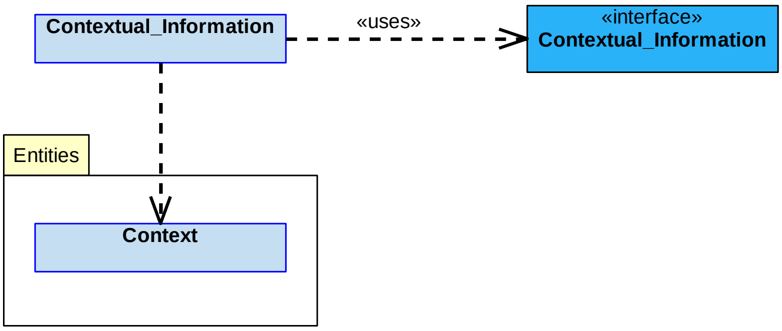


Figure 79: Contextual_Information Service Definition

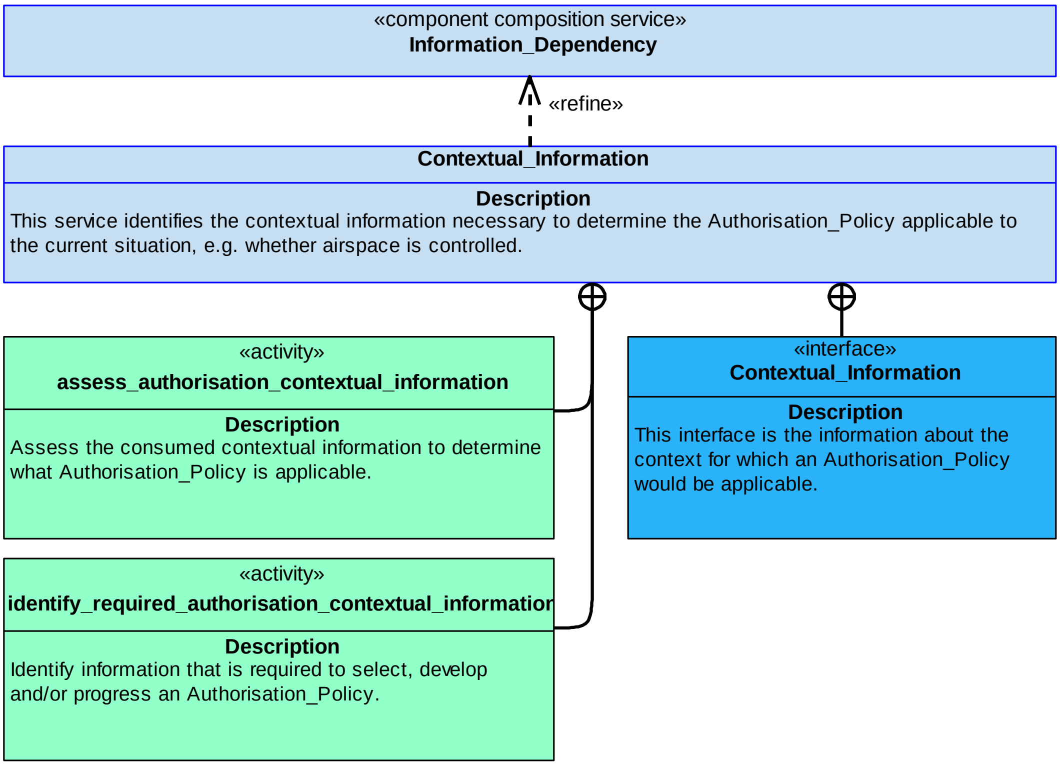

«component composition service»
Information_Dependency
«refine»
Contextual_Information
Description
This service identifies the contextual information necessary to determine the Authorisation_Policy applicable to the current situation, e.g. whether airspace is controlled.
«activity»
assess_authorisation_contextual_information
Description
Assess the consumed contextual information to determine what Authorisation_Policy is applicable.
«interface»
Contextual_Information
Description
This interface is the information about the context for which an Authorisation_Policy would be applicable.
«activity»
identify_required_authorisation_contextual_informatior
Description
Identify information that is required to select, develop and/or progress an Authorisation_Policy.

Figure 80: Contextual_Information Service Policy

Contextual_Information

This service identifies the contextual information necessary to determine the Authorisation_Policy applicable to the current situation, e.g. whether airspace is controlled.

Interface

Contextual_Information

This interface is the information about the context for which an Authorisation_Policy would be applicable.

Activities

assess_authorisation_contextual_information

Assess the consumed contextual information to determine what Authorisation_Policy is applicable.

identify_required_authorisation_contextual_information

Identify information that is required to select, develop and/or progress an Authorisation_Policy.

5.4.2.3.7.1.4 Constraint

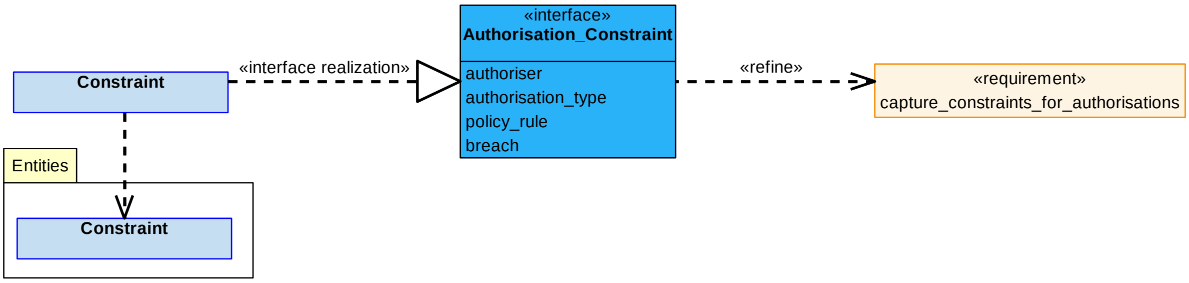


Figure 81: Constraint Service Definition

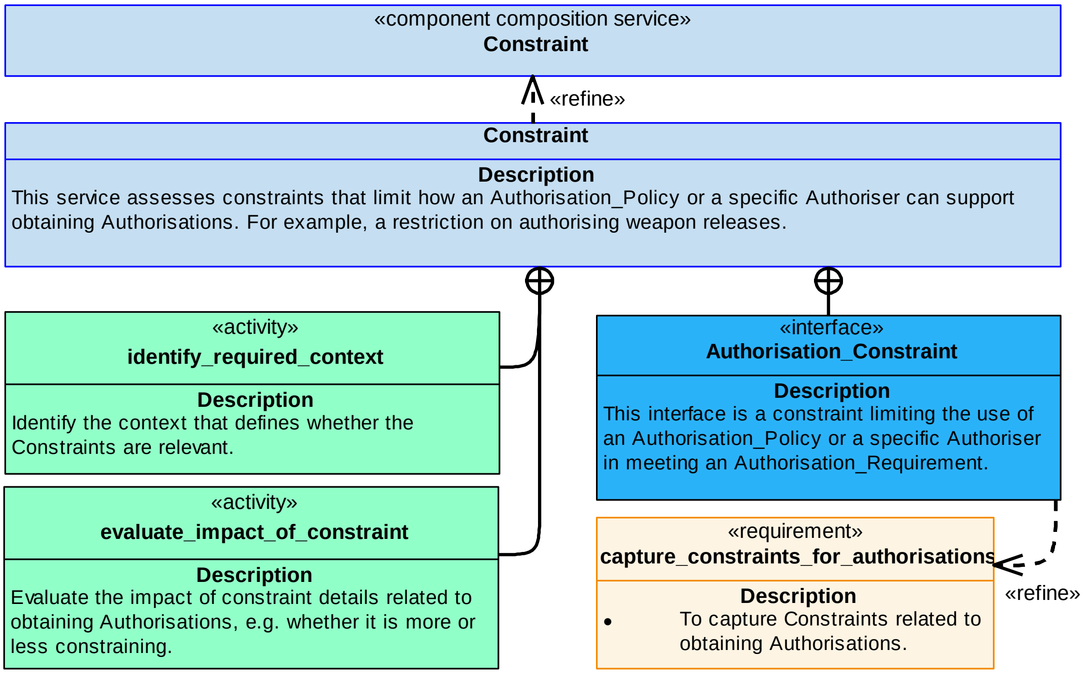

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses constraints that limit how an Authorisation_Policy or a specific Authoriser can support obtaining Authorisations. For example, a restriction on authorising weapon releases.
«activity»
identify_required_context
Description
Identify the context that defines whether the Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of constraint details related to obtaining Authorisations, e.g. whether it is more or less constraining.
«interface»
Authorisation_Constraint
Description
This interface is a constraint limiting the use of an Authorisation_Policy or a specific Authoriser in meeting an Authorisation_Requirement.
«requirement»
capture_constraints_for_authorisations
Description
• To capture Constraints related to obtaining Authorisations.
«refine»

Figure 82: Constraint Service Policy

Constraint

This service assesses constraints that limit how an Authorisation_Policy or a specific Authoriser can support obtaining Authorisations. For example, a restriction on authorising weapon releases.

Interface

Authorisation_Constraint

This interface is a constraint limiting the use of an Authorisation_Policy or a specific Authoriser in meeting an Authorisation_Requirement.

Attributes

authoriser	Specific Authorisers that are restricted by the Constraint.
authorisation_type	Authorisation_Type affected by the Constraint, e.g. a restriction on authorising weapon releases or information release.
policy_rule	Particular Authorisation_Policy rules that are restricted by the Constraint, e.g. certain Authorisers prevented from carrying out certain Step_Types.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of constraint details related to obtaining Authorisations, e.g. whether it is more or less constraining.

identify_required_context

Identify the context that defines whether the Constraints are relevant.

5.4.2.3.7.1.5 Capability

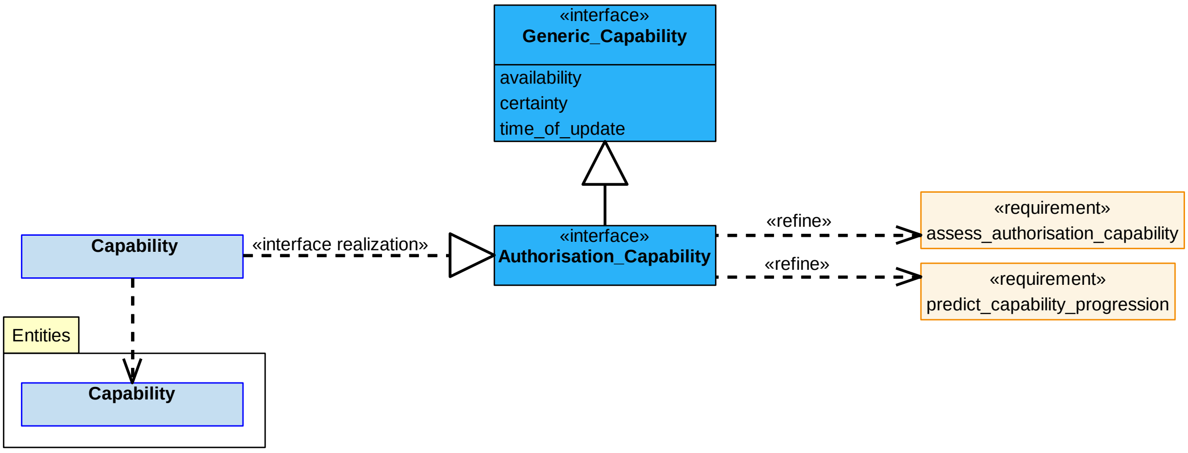


Figure 83: Capability Service Definition

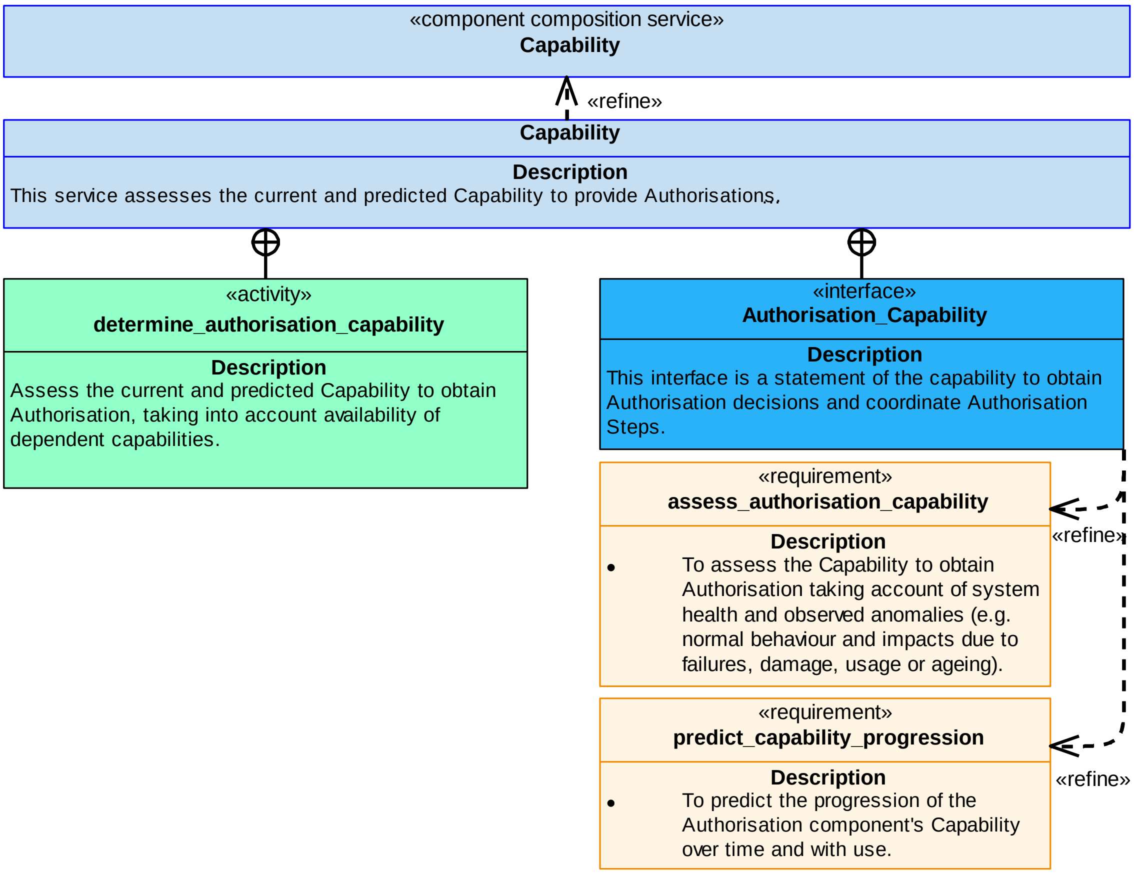

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to provide Authorisations.
«activity»
determine_authorisation_capability
Description
Assess the current and predicted Capability to obtain Authorisation, taking into account availability of dependent capabilities.
«interface»
Authorisation_Capability
Description
This interface is a statement of the capability to obtain Authorisation decisions and coordinate Authorisation Steps.
«requirement»
assess_authorisation_capability
Description
To assess the Capability to obtain Authorisation taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the Authorisation component's Capability over time and with use.
«refine»

Figure 84: Capability Service Policy

Capability

This service assesses the current and predicted Capability to provide Authorisations.

Interface

Authorisation_Capability

This interface is a statement of the capability to obtain Authorisation decisions and coordinate Authorisation Steps.

Activity

determine_authorisation_capability

Assess the current and predicted Capability to obtain Authorisation, taking into account availability of dependent capabilities.

5.4.2.3.7.1.6 Capability_Evidence

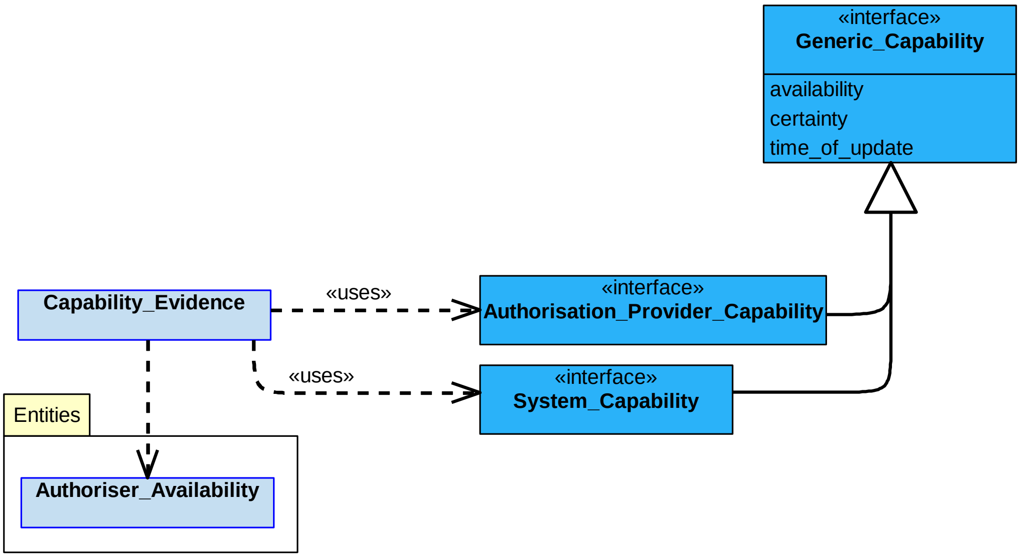
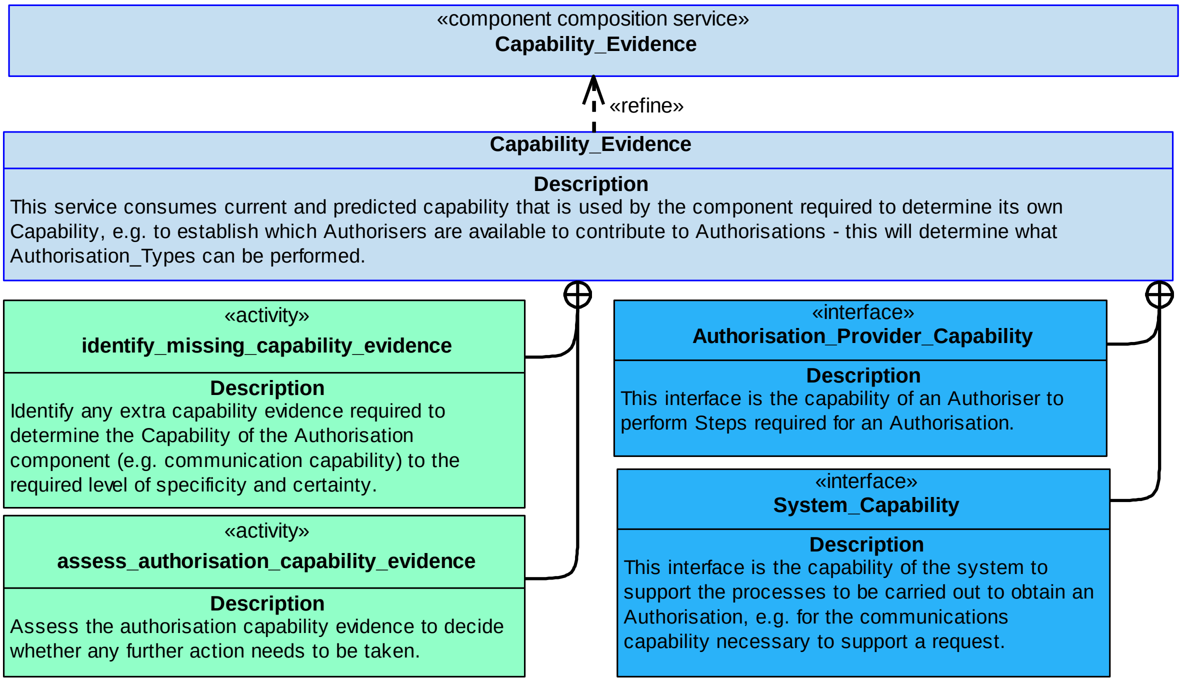


Figure 85: Capability_Evidence Service Definition



This service consumes current and predicted capability that is used by the component required to determine its own Capability, e.g. to establish which Authorisers are available to contribute to Authorisations - this will determine what Authorisation_Types can be performed.

Identify any extra capability evidence required to determine the Capability of the Authorisation component (e.g. communication capability) to the required level of specificity and certainty.

Assess the authorisation capability evidence to decide whether any further action needs to be taken.

This interface is the capability of an Authoriser to perform Steps required for an Authorisation.

This interface is the capability of the system to support the processes to be carried out to obtain an Authorisation, e.g. for the communications capability necessary to support a request.

Figure 86: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability that is used by the component required to determine its own Capability, e.g. to establish which Authorisers are available to contribute to Authorisations - this will determine what Authorisation_Types can be performed.

Interfaces

Authorisation_Provider_Capability

This interface is the capability of an Authoriser to perform Steps required for an Authorisation.

System_Capability

This interface is the capability of the system to support the processes to be carried out to obtain an Authorisation, e.g. for the communications capability necessary to support a request.

Activities

assess_authorisation_capability_evidence

Assess the authorisation capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability of the Authorisation component (e.g. communication capability) to the required level of specificity and certainty.

5.4.2.3.7.2 Service Dependencies

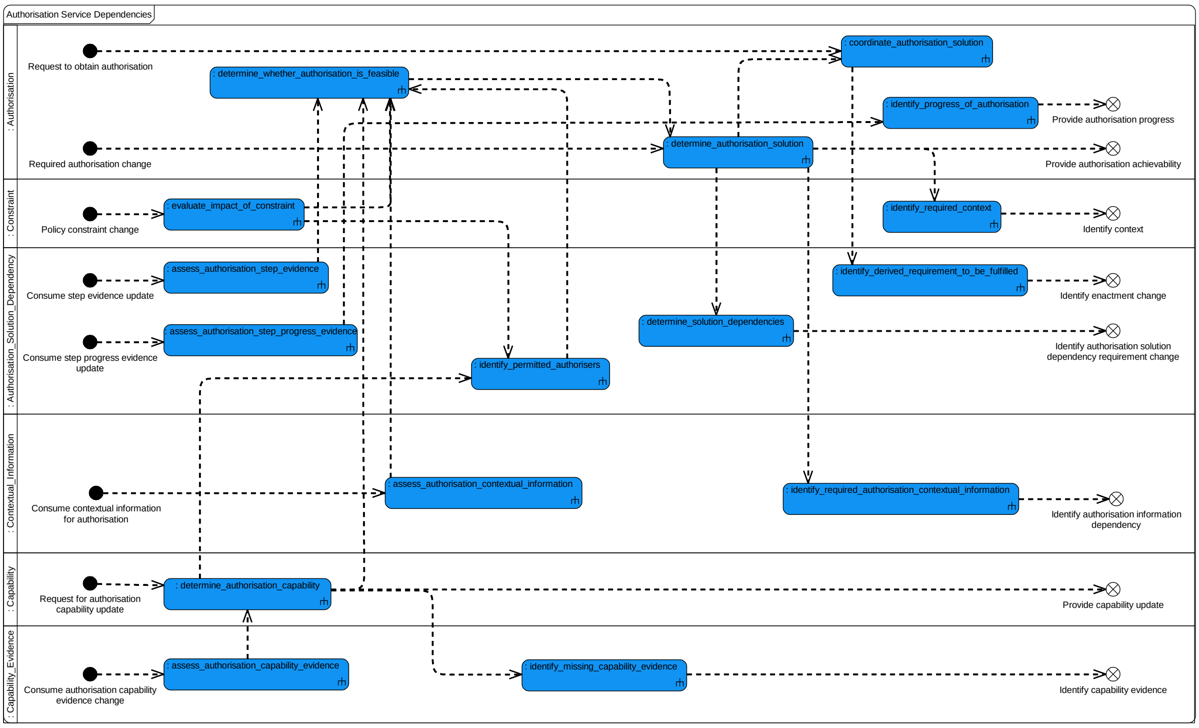

Authorisation Service Dependencies
: Authorisation
: Constraint
: Authorisation_Solution_Dependency
: Contextual_Information
: Capability
: Capability_Evidence
Request to obtain authorisation
Required authorisation change
Policy constraint change
Consume step evidence update
Consume step progress evidence update
Consume contextual information for authorisation
Request for authorisation capability update
Consume authorisation capability evidence change
: determine_whether_authorisation_is_feasible
: coordinate_authorisation_solution
: identify_progress_of_authorisation
: determine_authorisation_solution
: evaluate_impact_of_constraint
: identify_required_context
: assess_authorisation_step_evidence
: assess_authorisation_step_progress_evidence
: identify_derived_requirement_to_be_fulfilled
: determine_solution_dependencies
: identify_permitted_authorisers
: assess_authorisation_contextual_information
: identify_required_authorisation_contextual_information
: determine_authorisation_capability
: assess_authorisation_capability_evidence
: identify_missing_capability_evidence
Provide authorisation progress
Provide authorisation achievability
Identify context
Identify enactment change
Identify authorisation solution dependency requirement change
Identify authorisation information dependency
Provide capability update
Identify capability evidence

Figure 87: Authorisation Service Dependencies

5.4.2.4 Collision Avoidance

5.4.2.4.1 Role

The role of Collision Avoidance is to determine the solution required to avoid a predicted collision, or to exit a separation breach.

5.4.2.4.2 Overview

Control Architecture

Collision Avoidance is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

On receipt of a Separation_Breach, Collision Avoidance will determine the necessary Avoidance_Measure, based on Controllable_Vehicle status (speed, bearing, etc.), the Obstruction status (object type, speed, bearing, etc.) and the applicable Ruleset, taking into account Avoidance_Capability and Constraints.

Examples of Use

Collision Avoidance can be used:

- To provide manoeuvring cues for an operator to avoid a collision (e.g. with another vehicle or a feature such as terrain or weather).
- Where autonomous manoeuvres are required to avoid collisions with Obstructions in the path of the vehicle.
- As part of an ACAS implementation (with other components).

5.4.2.4.3 Service Summary

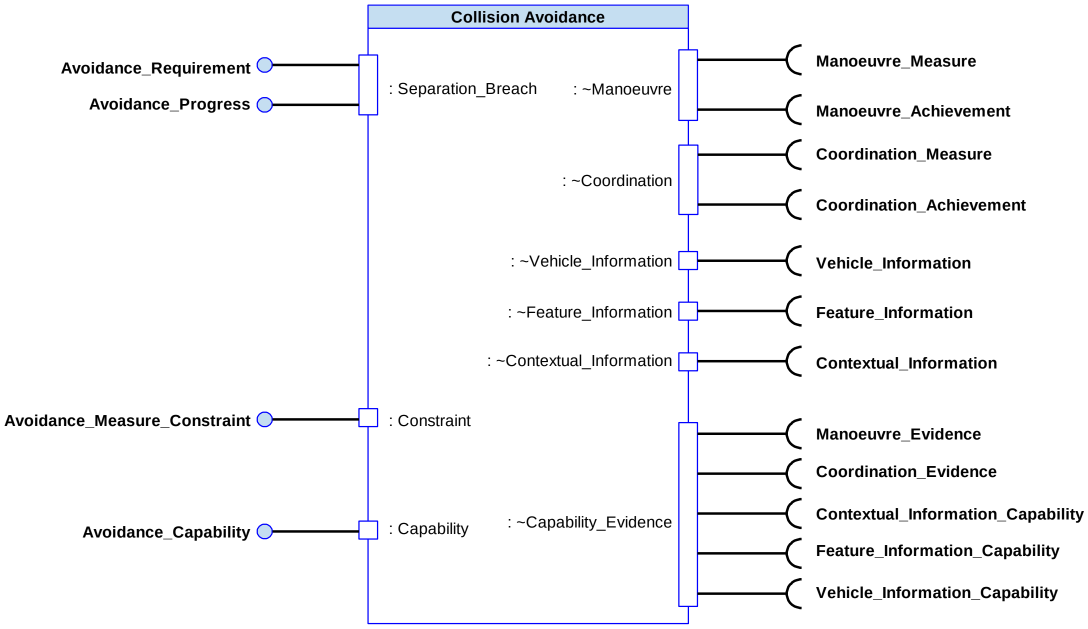


Figure 88: Collision Avoidance Service Summary

5.4.2.4.4 Responsibilities

capture_separation_breach

- To capture the provided Separation_Breach for which Avoidance_Measures are required.

capture_avoidance_constraints

- To capture provided Constraints that limit the ability to avoid a collision.

determine_avoidance_measure

- To determine the Avoidance_Measure required to avoid a collision (either cooperatively or non-cooperatively), or to exit a separation breach.

identify_progress_of_avoidance_measure

- To identify the progress of Avoidance_Measures in addressing the Separation_Breach.

assess_avoidance_capability

- To assess the Avoidance_Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Avoidance_Capability assessment.

predict_progression_of_avoidance_capability

- To predict the progression of the Avoidance_Capability over time and with use.

identify_avoidance_measure_in_progress_remains_feasible

- To identify if the Avoidance_Measure in progress remains feasible given current Avoidance_Capability and Constraints.

5.4.2.4.5 Subject Matter Semantics

The subject matter of Collision Avoidance is any activity (i.e. manoeuvring or getting others to manoeuvre) to avoid a collision.

Exclusions

The subject matter of Collision Avoidance does not include:

- How Avoidance_Measures are enacted or managed, only that they are to be performed.

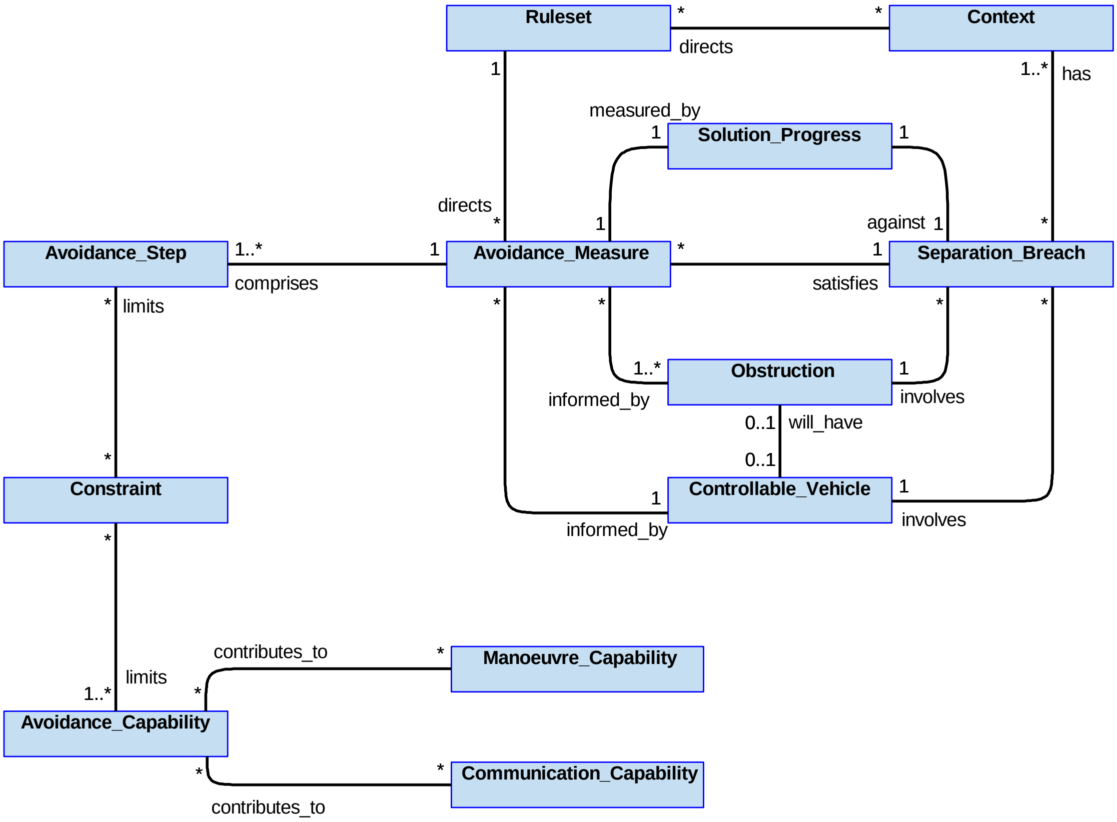


Figure 89: Collision Avoidance Semantics

5.4.2.4.5.1 Entities

Avoidance_Capability

The capability to determine Avoidance_Measures that will avoid a collision.

Avoidance_Measure

A sequence of steps to be performed to avoid a collision, e.g. in a TCAS implementation, a climb manoeuvre accompanied by communications to coordinate with the Obstruction.

Avoidance_Step

A specific activity undertaken in order to avoid a collision, e.g. to perform a manoeuvre or coordinate a complimentary manoeuvre with an Obstruction in accordance with the active ruleset.

Communication_Capability

The capability to communicate (and therefore cooperate) with Obstructions when performing manoeuvres.

Constraint

An externally placed limit, e.g. a restriction on the use of a particular avoidance technique.

Context

Information about the conditions in which the breach has occurred that may affect how it may be averted, e.g. within controlled or uncontrolled airspace.

Controllable_Vehicle

The state of the directly controllable vehicle, e.g. speed, bearing or altitude.

A controllable vehicle may be own vehicle, a drone or a swarm, etc.

Manoeuvre_Capability

The capability to perform manoeuvres.

Obstruction

An object with which a breach in separation has occurred, this could be another vehicle or a feature such as terrain or weather.

Ruleset

The rules that govern the choice of action to avoid the collision.

Separation_Breach

A requirement to avoid a potential collision, e.g. with another vehicle or feature, such as terrain or weather.

Solution_Progress

Evidence of the progress made to avert a breach.

5.4.2.4.6 Design Rationale

5.4.2.4.6.1 Assumptions

- This component will require some knowledge of vehicle performance to determine the manoeuvre to perform.
- When used as part of an ACAS II or similar future system, coordinated manoeuvres may be presented. Whilst it is assumed that coordinated manoeuvres will be adopted by both parties

of a breach in separation (as per the pilots' responsibilities), due to mission priorities or other conditions, it is acknowledged this may not always be the case.

5.4.2.4.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Collision Avoidance:

- Data Driving - Manoeuvre logic (Rulesets) can be data-driven at build time (i.e. development) or later in order to maintain compliance with regulatory standards.

Extensions

- Extensions may be useful for implementing differing manoeuvre Rulesets, e.g. for avoidance of terrain, of objects in flight or objects during taxi.

Other Factors that were Taken into Account

- Specific Rulesets may be required to conform to regulatory standards, e.g. to meet FAA Collision Avoidance System logic version *"X",* thus supporting flight certifiability. ACAS II has been considered whilst defining this component, although it does not constrain or limit the component to only being developed that way.

Exploitation Considerations

- A vehicle may be both a Controllable_Vehicle and an Obstruction, e.g. if a manned ownship has a breach of separation with its support drone. In such a case, cooperative avoidance may include direct control of both ownship and the drone.
- Collision Avoidance will determine the most appropriate avoidance manoeuvre, based on pre-determined rules and constrained by the location and current manoeuvring profile of the vehicle, the available resources and the information provided about the predicted collision. Where an avoidance manoeuvre is not available that can satisfy the avoidance of multiple objects, the rules can also dictate which object takes priority to be avoided.
- Collision Avoidance may receive contextual information for different scenarios, such as an air vehicle landing. For scenarios like this, data would need to be input into the component so that it would understand that it does not need to generate an avoidance manoeuvre, in this example a ground avoidance manoeuvre.
- Once Collision Avoidance has identified an avoidance manoeuvre to be performed, it will be executed by Vehicle Guidance. However, should the avoidance manoeuvre result in conflicting requests on Vehicle Guidance, the Conflict Resolution component will be invoked to resolve the conflict, typically arbitrating in favour of the avoidance manoeuvre. During this process, Conflict Resolution will take into account the context of the current situation (such as the mission goals) when determining if the collision avoidance manoeuvre is appropriate. For example, Collision Avoidance may determine an avoidance manoeuvre to prevent a collision between an air vehicle and a building, but Conflict Resolution will reject the avoidance manoeuvre if the goal is to fly the air vehicle into the building.

- Where necessary to conform to regulatory standards such as ACAS II, Collision Avoidance supports cooperative avoidance manoeuvres with an appropriately equipped intruder (the Obstruction).

5.4.2.4.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

This component generates a manoeuvre to avoid a collision. If a manoeuvre is incorrectly generated, it could, for example, lead directly to inadvertent flight into terrain. In this example, the worst case would be generating a dive rather than a climb. This would result in loss of the air vehicle and potential fatalities.

Where instances of this component contribute to hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.4.6.4 Security Considerations

The indicative security classification is O.

This component will determine the appropriate avoidance measures when requested. Whilst it is involved in any cooperative manoeuvre negotiations carried out via clear communications, it also requires knowledge of performance data (speed, climb rates, etc.) in order to determine possible manoeuvres. This data is considered SNEO for military platforms; to avoid possible loss of availability when crossing domain boundaries, use of a declassified subset of performance data may be possible in order to lower this rating. The availability (timeliness) and integrity (correctness) of the determined avoidance activities for those required to act upon them will need protecting.

The ability to influence manoeuvres means that this component is considered a subject of interest for an adversary and a likely target for a cyber attack. Additionally “gaming” the avoidance function by hostile forces to invoke a predictable manoeuvre is a concern.

The component may be expected to at least partially satisfy security related functions relating to:

- **Identifying Data Sources**, supporting the authentication of cooperative entities (e.g. where part of an ACAS installation) to protect against spoofing of cooperative negotiations.
- **Maintaining Audit Records**, providing accountability for any actions performed (or not) in avoidance of a collision, and in support of filing Mandatory Occurrence Reports, Voluntary Occurrence Reports and Airprox incidents.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is not expected to directly implement security enforcing functions, but will rely on the integrity of externally-sourced information, e.g. from other ACAS users.

5.4.2.4.7 Services

5.4.2.4.7.1 Service Definitions

5.4.2.4.7.1.1 Separation_Breach

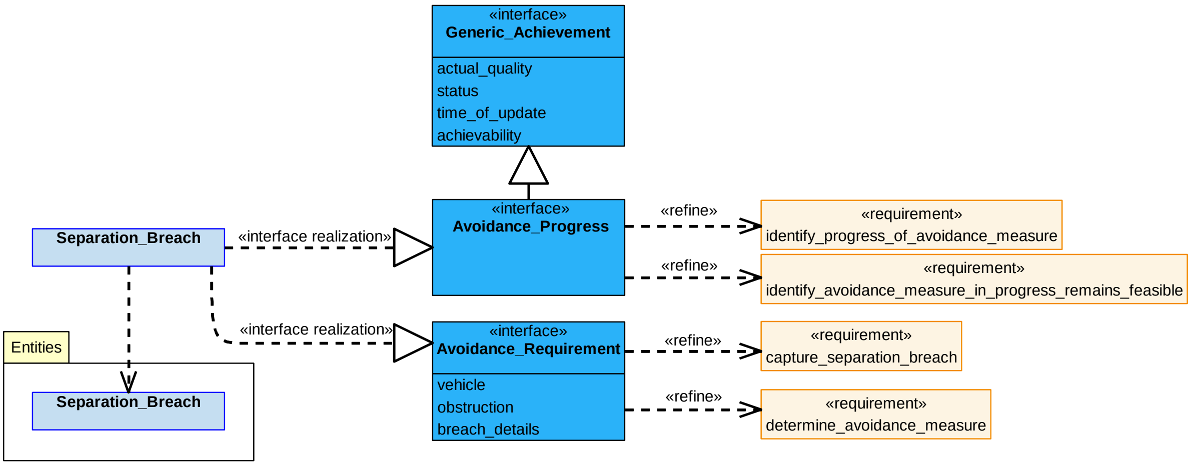
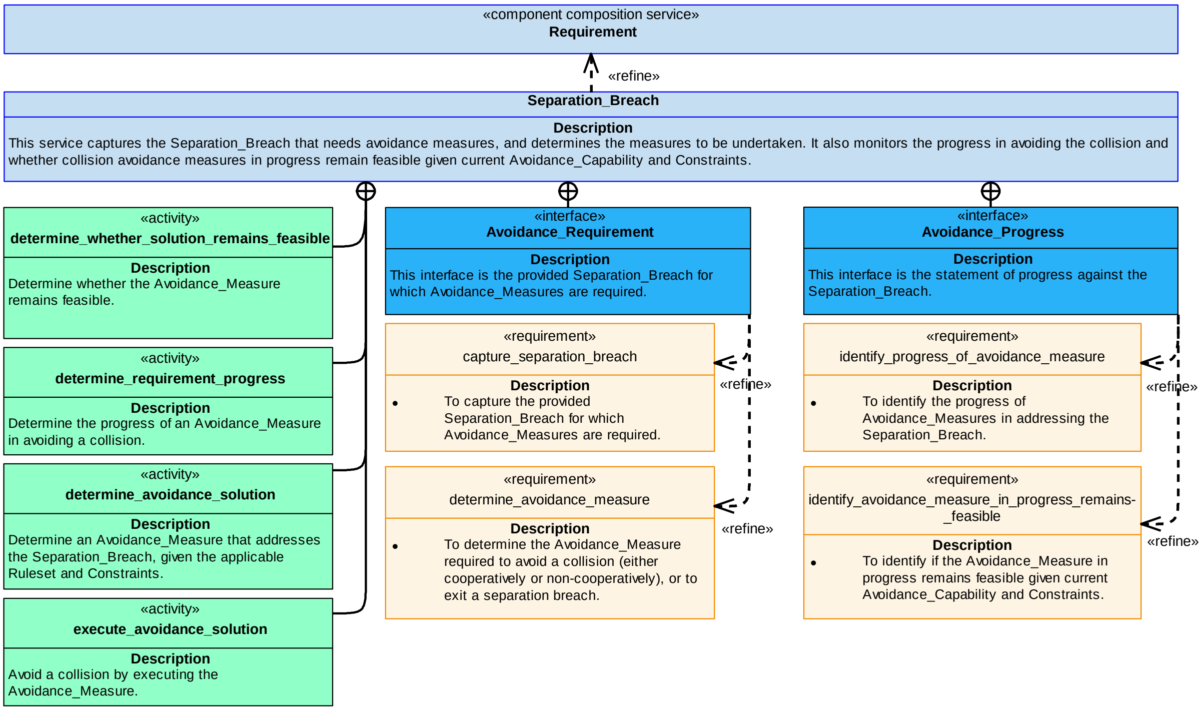


Figure 90: Separation_Breach Service Definition


«component composition service»
Requirement
«refine»
Separation_Breach
«activity»
determine_whether_solution_remains_feasible
«activity»
determine_requirement_progress
«activity»
determine_avoidance_solution
«activity»
execute_avoidance_solution
«interface»
Avoidance_Requirement
«interface»
Avoidance_Progress
«requirement»
capture_separation_breach
«refine»
«requirement»
determine_avoidance_measure
«refine»
«requirement»
identify_progress_of_avoidance_measure
«refine»
«requirement»
identify_avoidance_measure_in_progress_remains-_feasible
«refine»

Separation_Breach

Description

This service captures the Separation_Breach that needs avoidance measures, and determines the measures to be undertaken. It also monitors the progress in avoiding the collision and whether collision avoidance measures in progress remain feasible given current Avoidance_Capability and Constraints.

determine_whether_solution_remains_feasible

Description

Determine whether the Avoidance_Measure remains feasible.

determine_requirement_progress

Description

Determine the progress of an Avoidance_Measure in avoiding a collision.

determine_avoidance_solution

Description

Determine an Avoidance_Measure that addresses the Separation_Breach, given the applicable Ruleset and Constraints.

execute_avoidance_solution

Description

Avoid a collision by executing the Avoidance_Measure.

Avoidance_Requirement

Description

This interface is the provided Separation_Breach for which Avoidance_Measures are required.

capture_separation_breach

Description

- To capture the provided Separation_Breach for which Avoidance_Measures are required.

determine_avoidance_measure

Description

- To determine the Avoidance_Measure required to avoid a collision (either cooperatively or non-cooperatively), or to exit a separation breach.

Avoidance_Progress

Description

This interface is the statement of progress against the Separation_Breach.

identify_progress_of_avoidance_measure

Description

- To identify the progress of Avoidance_Measures in addressing the Separation_Breach.

identify_avoidance_measure_in_progress_remains-_feasible

Description

- To identify if the Avoidance_Measure in progress remains feasible given current Avoidance_Capability and Constraints.

Figure 91: Separation_Breach Service Policy

Separation_Breach

This service captures the Separation_Breach that needs avoidance measures, and determines the measures to be undertaken. It also monitors the progress in avoiding the collision and whether collision avoidance measures in progress remain feasible given current Avoidance_Capability and Constraints.

Interfaces

Avoidance_Requirement

This interface is the provided Separation_Breach for which Avoidance_Measures are required.

Attributes

vehicle	The Controllable_Vehicle that the requirement applies to, e.g. ownship or a drone.
obstruction	The obstruction with which a Separation_Breach has been identified, e.g. an aircraft or terrain.
breach_details	Details about the Separation_Breach, e.g. its severity, location, whether self-detected or externally notified through cooperative means and the standards (e.g. TCAS) to be applied.

Avoidance_Progress

This interface is the statement of progress against the Separation_Breach.

Activities

determine_requirement_progress

Determine the progress of an Avoidance_Measure in avoiding a collision.

determine_avoidance_solution

Determine an Avoidance_Measure that addresses the Separation_Breach, given the applicable Ruleset and Constraints.

execute_avoidance_solution

Avoid a collision by executing the Avoidance_Measure.

determine_whether_solution_remains_feasible

Determine whether the Avoidance_Measure remains feasible.

5.4.2.4.7.1.2 Manoeuvre

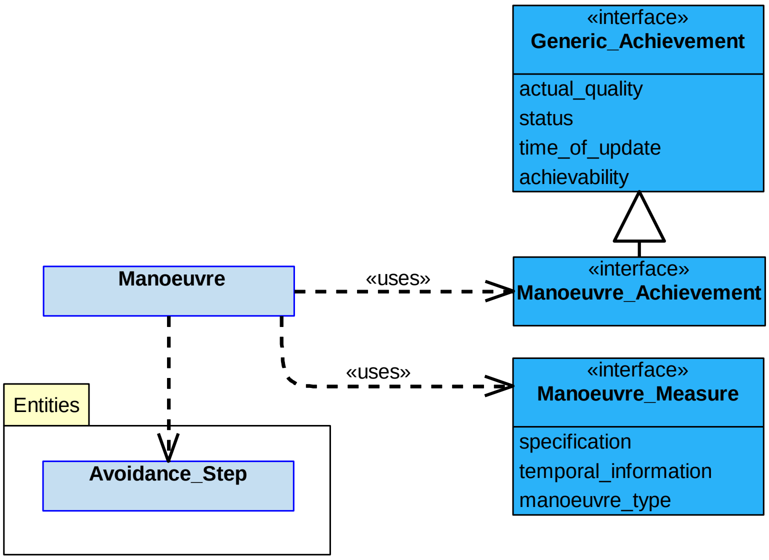


Figure 92: Manoeuvre Service Definition

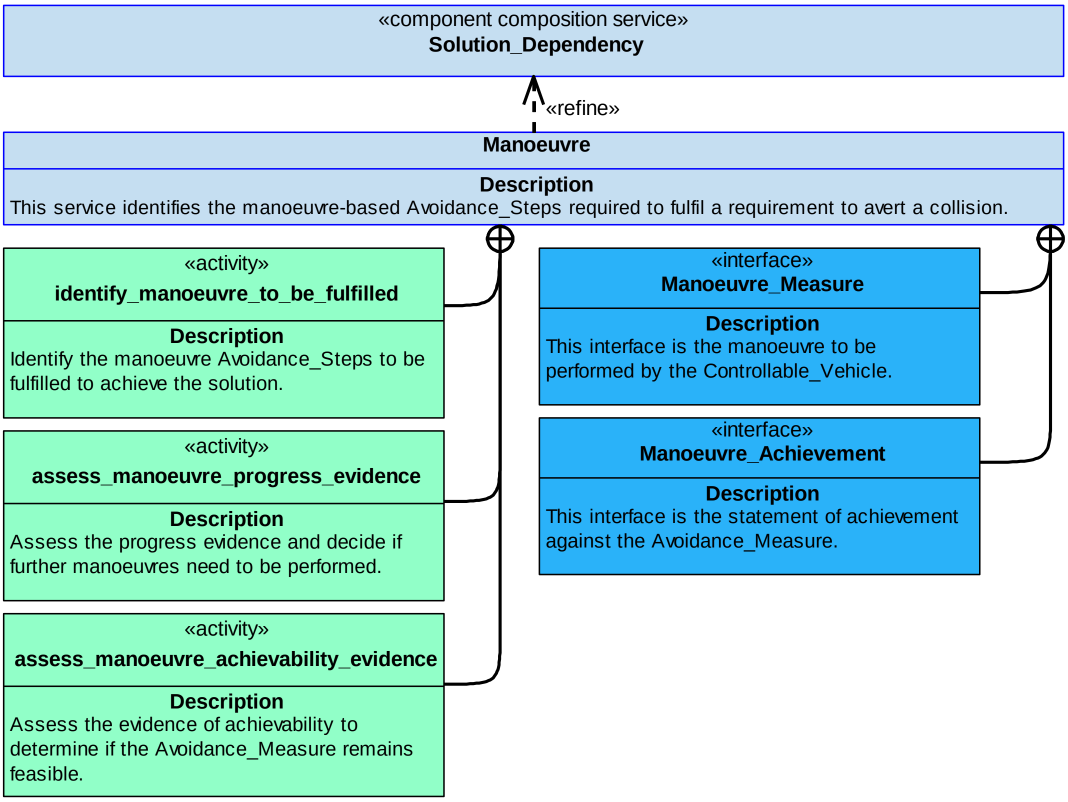

«component composition service»
Solution_Dependency
«refine»
Manoeuvre
Description
This service identifies the manoeuvre-based Avoidance_Steps required to fulfil a requirement to avert a collision.
«activity»
identify_manoeuvre_to_be_fulfilled
Description
Identify the manoeuvre Avoidance_Steps to be fulfilled to achieve the solution.
«activity»
assess_manoeuvre_progress_evidence
Description
Assess the progress evidence and decide if further manoeuvres need to be performed.
«activity»
assess_manoeuvre_achievability_evidence
Description
Assess the evidence of achievability to determine if the Avoidance_Measure remains feasible.
«interface»
Manoeuvre_Measure
Description
This interface is the manoeuvre to be performed by the Controllable_Vehicle.
«interface»
Manoeuvre_Achievement
Description
This interface is the statement of achievement against the Avoidance_Measure.

Figure 93: Manoeuvre Service Policy

Manoeuvre

This service identifies the manoeuvre-based Avoidance_Steps required to fulfil a requirement to avert a collision.

Interfaces

Manoeuvre_Measure

This interface is the manoeuvre to be performed by the Controllable_Vehicle.

Attributes

specification	The definition of the manoeuvre, e.g. the rate of climb or dive.
temporal_information	Information covering timing, such as start or end times of a climb.
manoeuvre_type	The type of manoeuvre required, e.g. climb or dive.

Manoeuvre_Achievement

This interface is the statement of achievement against the Avoidance_Measure.

Activities

identify_manoeuvre_to_be_fulfilled

Identify the manoeuvre Avoidance_Steps to be fulfilled to achieve the solution.

assess_manoeuvre_progress_evidence

Assess the progress evidence and decide if further manoeuvres need to be performed.

assess_manoeuvre_achievability_evidence

Assess the evidence of achievability to determine if the Avoidance_Measure remains feasible.

5.4.2.4.7.1.3 Coordination

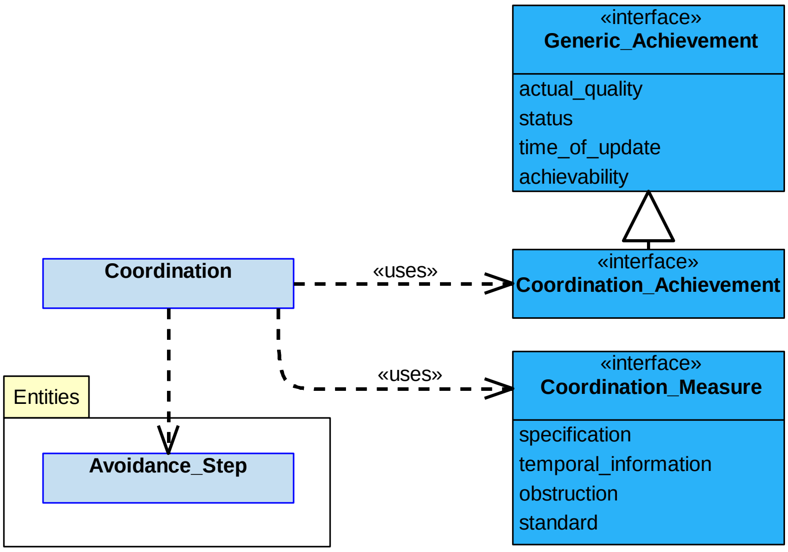


Figure 94: Coordination Service Definition

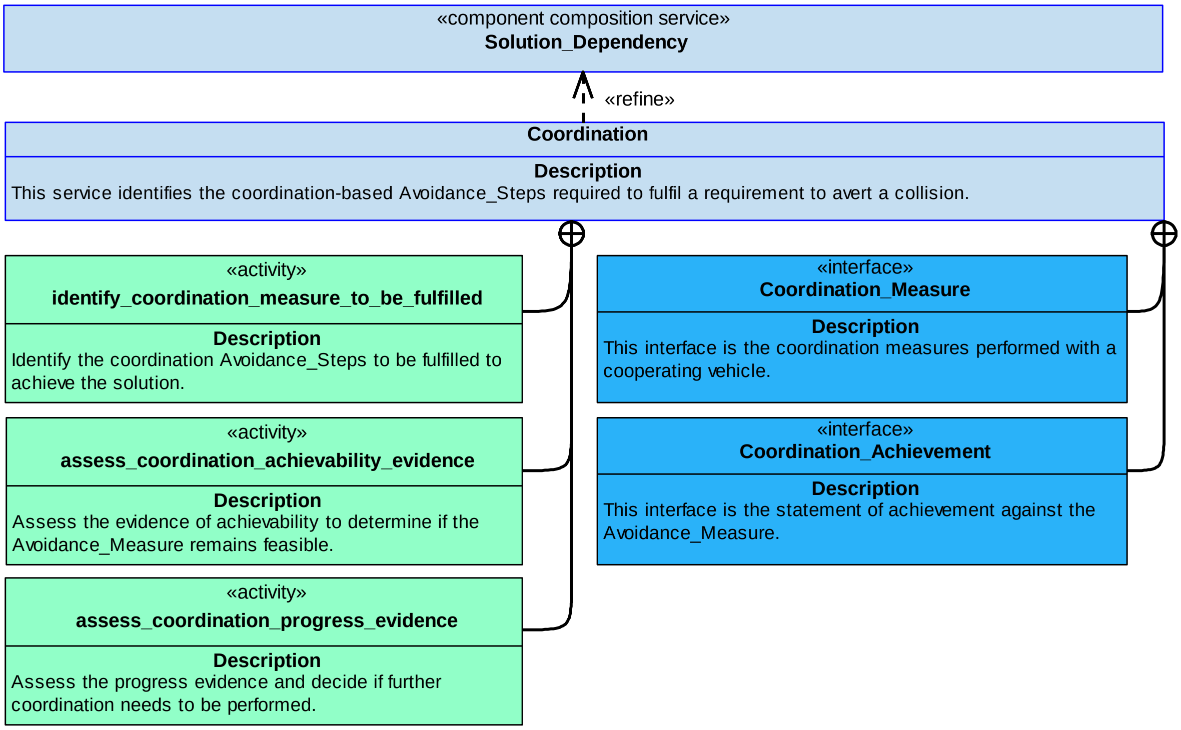

«component composition service»
Solution_Dependency
«refine»
Coordination
Description
This service identifies the coordination-based Avoidance_Steps required to fulfil a requirement to avert a collision.
«activity»
identify_coordination_measure_to_be_fulfilled
Description
Identify the coordination Avoidance_Steps to be fulfilled to achieve the solution.
«activity»
assess_coordination_achievability_evidence
Description
Assess the evidence of achievability to determine if the Avoidance_Measure remains feasible.
«activity»
assess_coordination_progress_evidence
Description
Assess the progress evidence and decide if further coordination needs to be performed.
«interface»
Coordination_Measure
Description
This interface is the coordination measures performed with a cooperating vehicle.
«interface»
Coordination_Achievement
Description
This interface is the statement of achievement against the Avoidance_Measure.

Figure 95: Coordination Service Policy

Coordination

This service identifies the coordination-based Avoidance_Steps required to fulfil a requirement to avert a collision.

Interfaces

Coordination_Measure

This interface is the coordination measures performed with a cooperating vehicle.

Attributes

specification	The definition of the coordination measures to be undertaken, e.g. to request the obstruction climbs to complement an ownship dive.
temporal_information	Information covering timing, such as start or end times.
obstruction	The Obstruction that is to be coordinated with.
standard	The standard being use for the coordination, e.g. TCAS.

Coordination_Achievement

This interface is the statement of achievement against the Avoidance_Measure.

Activities

identify_coordination_measure_to_be_fulfilled

Identify the coordination Avoidance_Steps to be fulfilled to achieve the solution.

assess_coordination_progress_evidence

Assess the progress evidence and decide if further coordination needs to be performed.

assess_coordination_achievability_evidence

Assess the evidence of achievability to determine if the Avoidance_Measure remains feasible.

5.4.2.4.7.1.4 Vehicle_Information

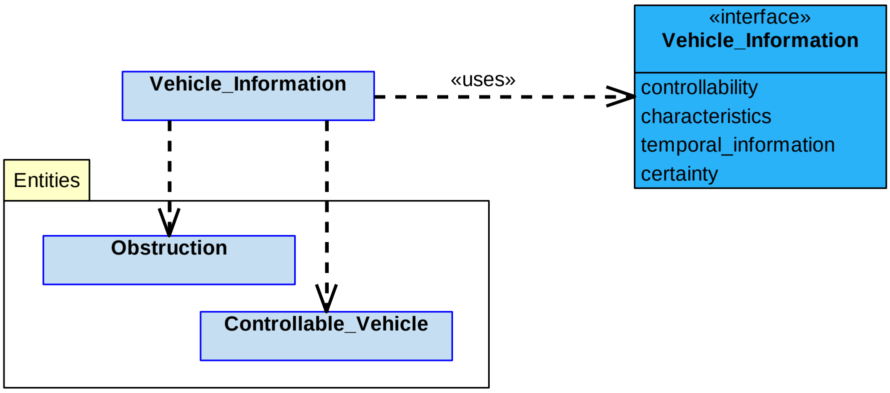


Figure 96: Vehicle_Information Service Definition

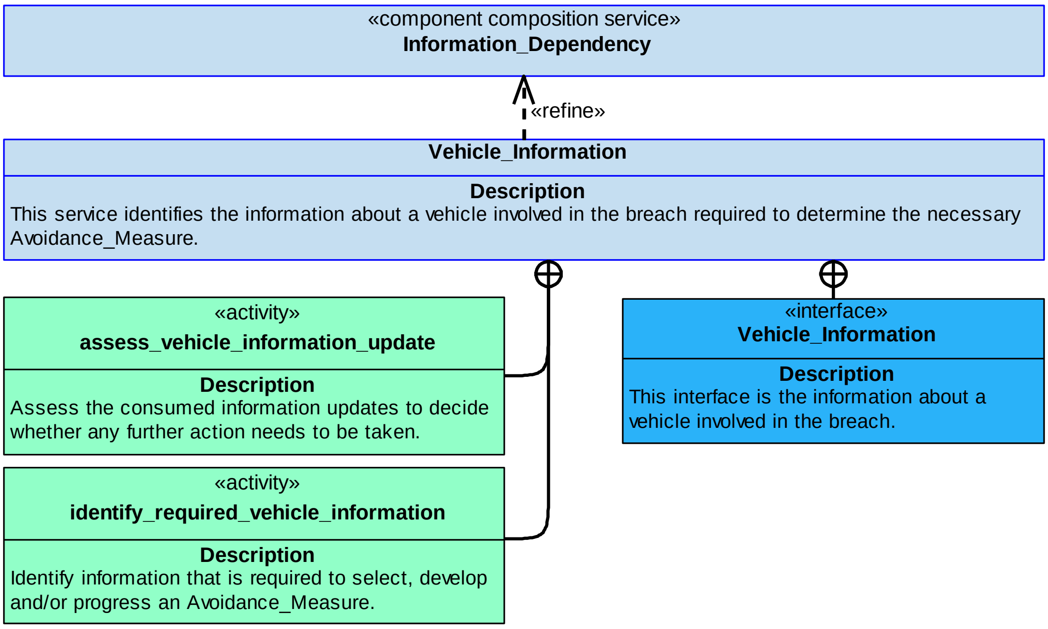

«component composition service»
Information_Dependency
«refine»
Vehicle_Information
Description
This service identifies the information about a vehicle involved in the breach required to determine the necessary Avoidance_Measure.
«activity»
assess_vehicle_information_update
Description
Assess the consumed information updates to decide whether any further action needs to be taken.
«activity»
identify_required_vehicle_information
Description
Identify information that is required to select, develop and/or progress an Avoidance_Measure.
«interface»
Vehicle_Information
Description
This interface is the information about a vehicle involved in the breach.

Figure 97: Vehicle_Information Service Policy

Vehicle_Information

This service identifies the information about a vehicle involved in the breach required to determine the necessary Avoidance_Measure.

Interface

Vehicle_Information

This interface is the information about a vehicle involved in the breach.

Attributes

controllability	Whether the vehicle is controllable or influenceable, or not.
characteristics	Information about the vehicle, e.g. altitude, range, bearing and speed.
temporal_information	Information covering the timing of the information being reported.
certainty	The level of confidence in the reported information.

Activities

assess_vehicle_information_update

Assess the consumed information updates to decide whether any further action needs to be taken.

identify_required_vehicle_information

Identify information that is required to select, develop and/or progress an Avoidance_Measure.

5.4.2.4.7.1.5 Feature_Information

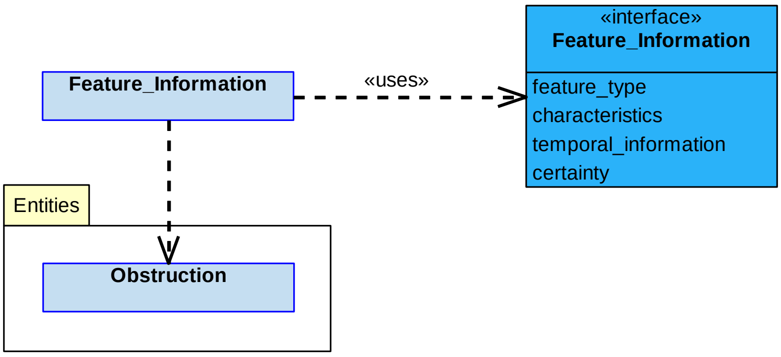


Figure 98: Feature_Information Service Definition

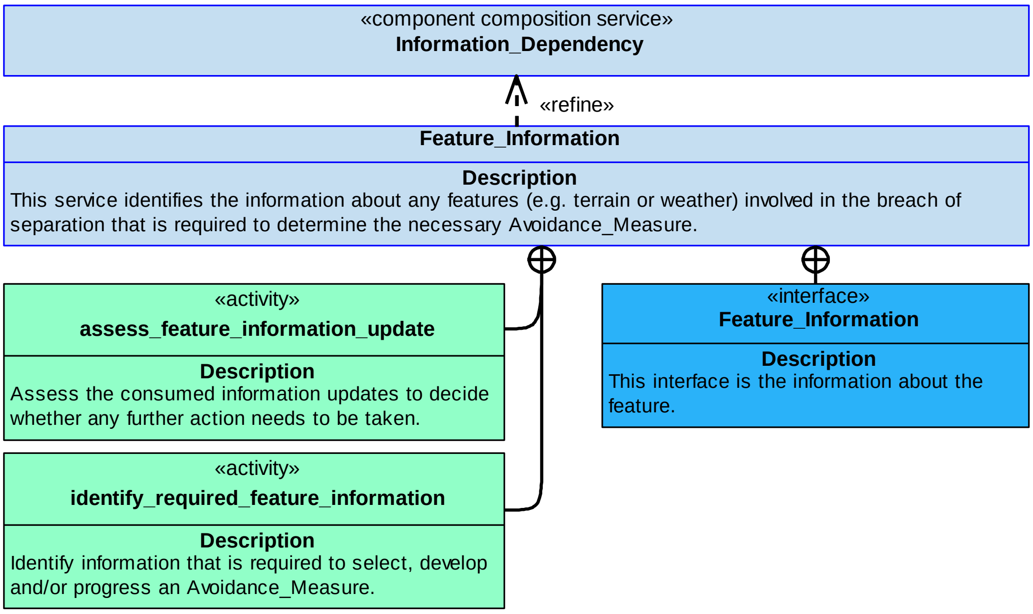

«component composition service»
Information_Dependency
«refine»
Feature_Information
Description
This service identifies the information about any features (e.g. terrain or weather) involved in the breach of separation that is required to determine the necessary Avoidance_Measure.
«activity»
assess_feature_information_update
Description
Assess the consumed information updates to decide whether any further action needs to be taken.
«activity»
identify_required_feature_information
Description
Identify information that is required to select, develop and/or progress an Avoidance_Measure.
«interface»
Feature_Information
Description
This interface is the information about the feature.

Figure 99: Feature Information Service Policy

Feature_Information

This service identifies the information about any features (e.g. terrain or weather) involved in the breach of separation that is required to determine the necessary Avoidance_Measure.

<u>Interface</u>

Feature_Information

This interface is the information about the feature.

<u>Attributes</u>

feature_type	The type of feature, e.g. terrain, a building, weather or a no-fly zone.
characteristics	Information about the feature, e.g. size or range.
temporal_information	Information covering the timing of the information being reported.
certainty	The level of confidence in the reported information.

<u>Activities</u>

assess_feature_information_update

Assess the consumed information updates to decide whether any further action needs to be taken.

identify_required_feature_information

Identify information that is required to select, develop and/or progress an Avoidance_Measure.

5.4.2.4.7.1.6 Contextual_Information

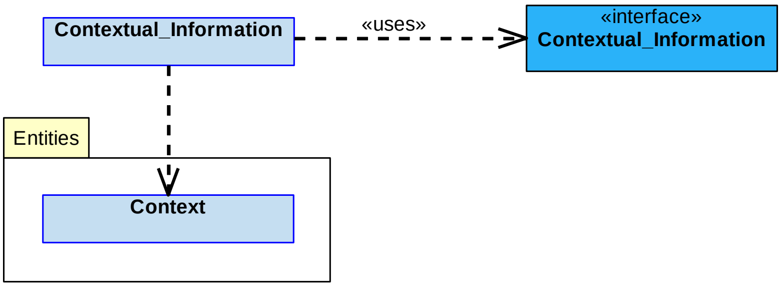


Figure 100: Contextual_Information Service Definition

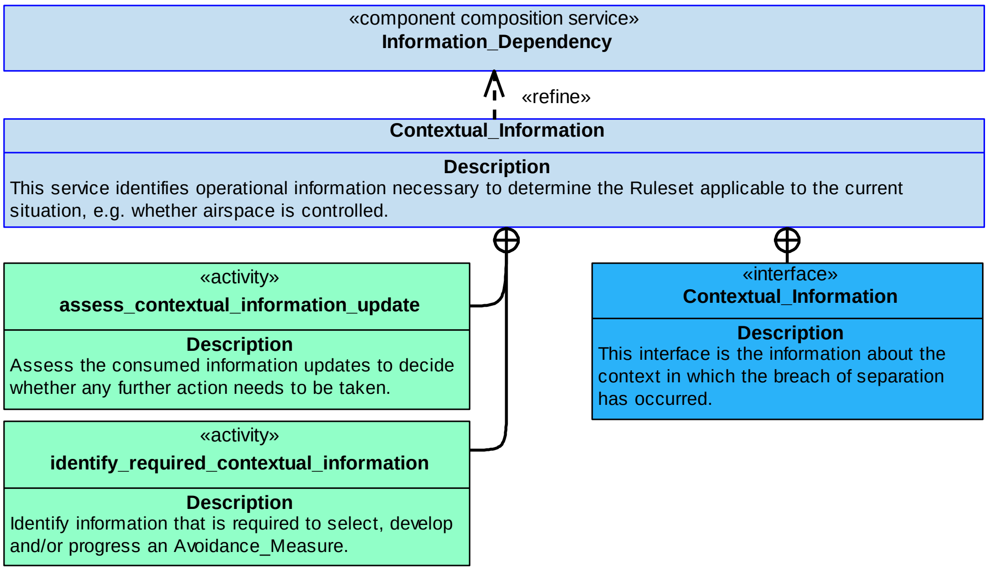

«component composition service»
Information_Dependency
«refine»
Contextual_Information
Description
This service identifies operational information necessary to determine the Ruleset applicable to the current situation, e.g. whether airspace is controlled.
«activity»
assess_contextual_information_update
Description
Assess the consumed information updates to decide whether any further action needs to be taken.
«activity»
identify_required_contextual_information
Description
Identify information that is required to select, develop and/or progress an Avoidance_Measure.
«interface»
Contextual_Information
Description
This interface is the information about the context in which the breach of separation has occurred.

Figure 101: Contextual_Information Service Policy

Contextual_Information

This service identifies operational information necessary to determine the Ruleset applicable to the current situation, e.g. whether airspace is controlled.

Interface

Contextual_Information

This interface is the information about the context in which the breach of separation has occurred.

Activities

assess_contextual_information_update

Assess the consumed information updates to decide whether any further action needs to be taken.

identify_required_contextual_information

Identify information that is required to select, develop and/or progress an Avoidance_Measure.

5.4.2.4.7.1.7 Constraint

Assess the current and predicted capability to avoid collisions

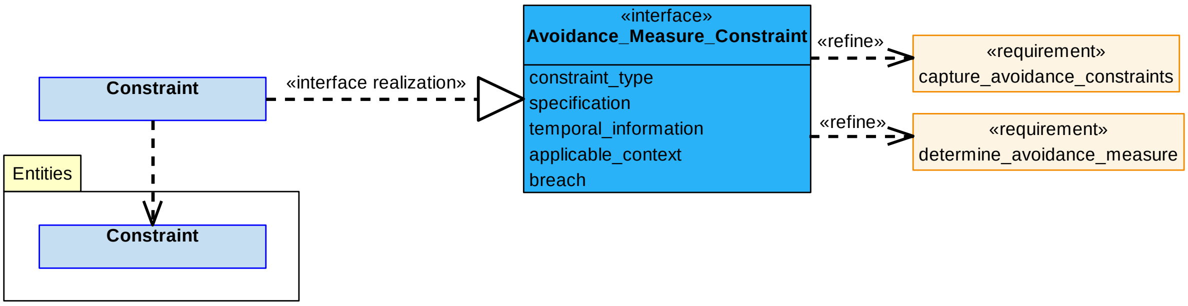


Figure 102: Constraint Service Definition

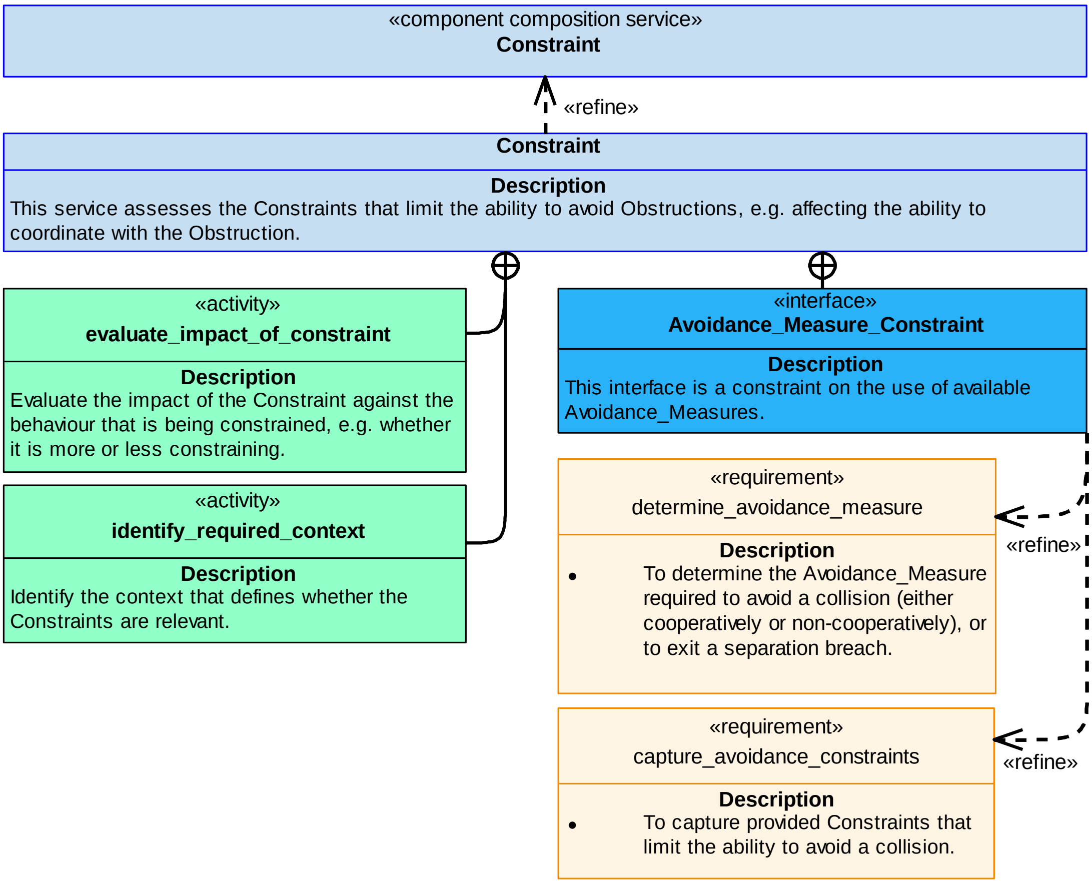

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Constraints that limit the ability to avoid Obstructions, e.g. affecting the ability to coordinate with the Obstruction.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of the Constraint against the behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context that defines whether the Constraints are relevant.
«interface»
Avoidance_Measure_Constraint
Description
This interface is a constraint on the use of available Avoidance_Measures.
«requirement»
determine_avoidance_measure
Description
To determine the Avoidance_Measure required to avoid a collision (either cooperatively or non-cooperatively), or to exit a separation breach.
«refine»
«requirement»
capture_avoidance_constraints
Description
To capture provided Constraints that limit the ability to avoid a collision.
«refine»

Figure 103: Constraint Service Policy

Constraint

This service assesses the Constraints that limit the ability to avoid Obstructions, e.g. affecting the ability to coordinate with the Obstruction.

Interface

Avoidance_Measure_Constraint

This interface is a constraint on the use of available Avoidance_Measures.

Attributes

constraint_type	The type of constraint to be applied to the Avoidance_Measure, e.g. restricting possible manoeuvres or means of coordination.
specification	The definition of the constraint, e.g. restricting climb rate to 5000ft/minute or preventing transmission of transponder information.
temporal_information	Information covering timing, such as start and end times.
applicable_context	The context in which the constraint is applicable.
breach	A statement that the constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of the Constraint against the behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context that defines whether the Constraints are relevant.

5.4.2.4.7.1.8 Capability

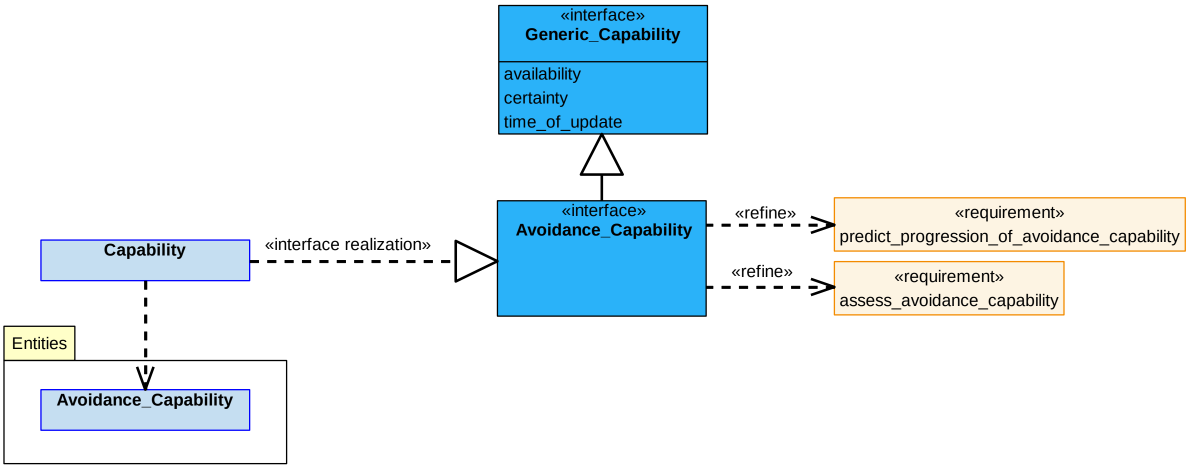


Figure 104: Capability Service Definition

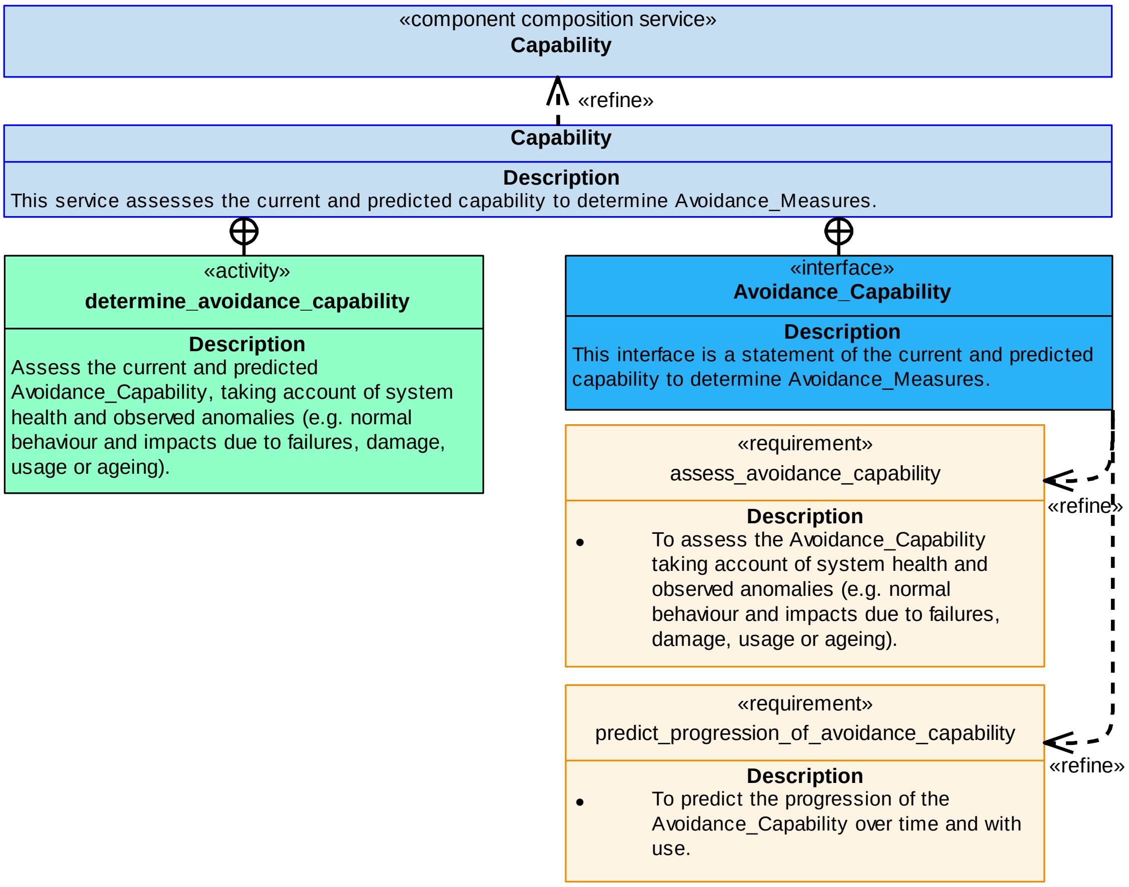

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability to determine Avoidance_Measures.
«activity»
determine_avoidance_capability
Description
Assess the current and predicted Avoidance_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Avoidance_Capability
Description
This interface is a statement of the current and predicted capability to determine Avoidance_Measures.
«requirement»
assess_avoidance_capability
Description
To assess the Avoidance_Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_progression_of_avoidance_capability
Description
To predict the progression of the Avoidance_Capability over time and with use.
«refine»

Figure 105: Capability Service Policy

Capability

This service assesses the current and predicted capability to determine Avoidance_Measures.

Interface

Avoidance_Capability

This interface is a statement of the current and predicted capability to determine Avoidance_Measures.

Activity

determine_avoidance_capability

Assess the current and predicted Avoidance_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.4.7.1.9 Capability_Evidence

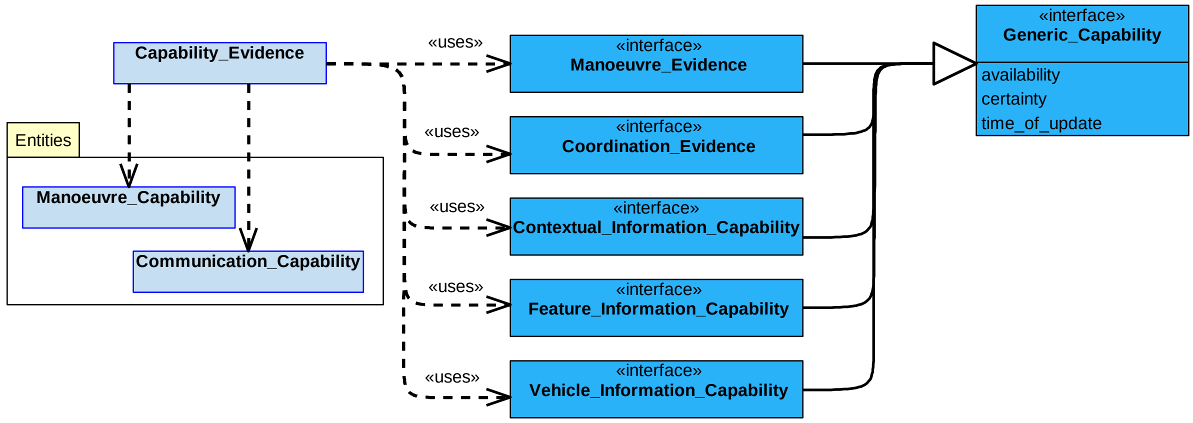
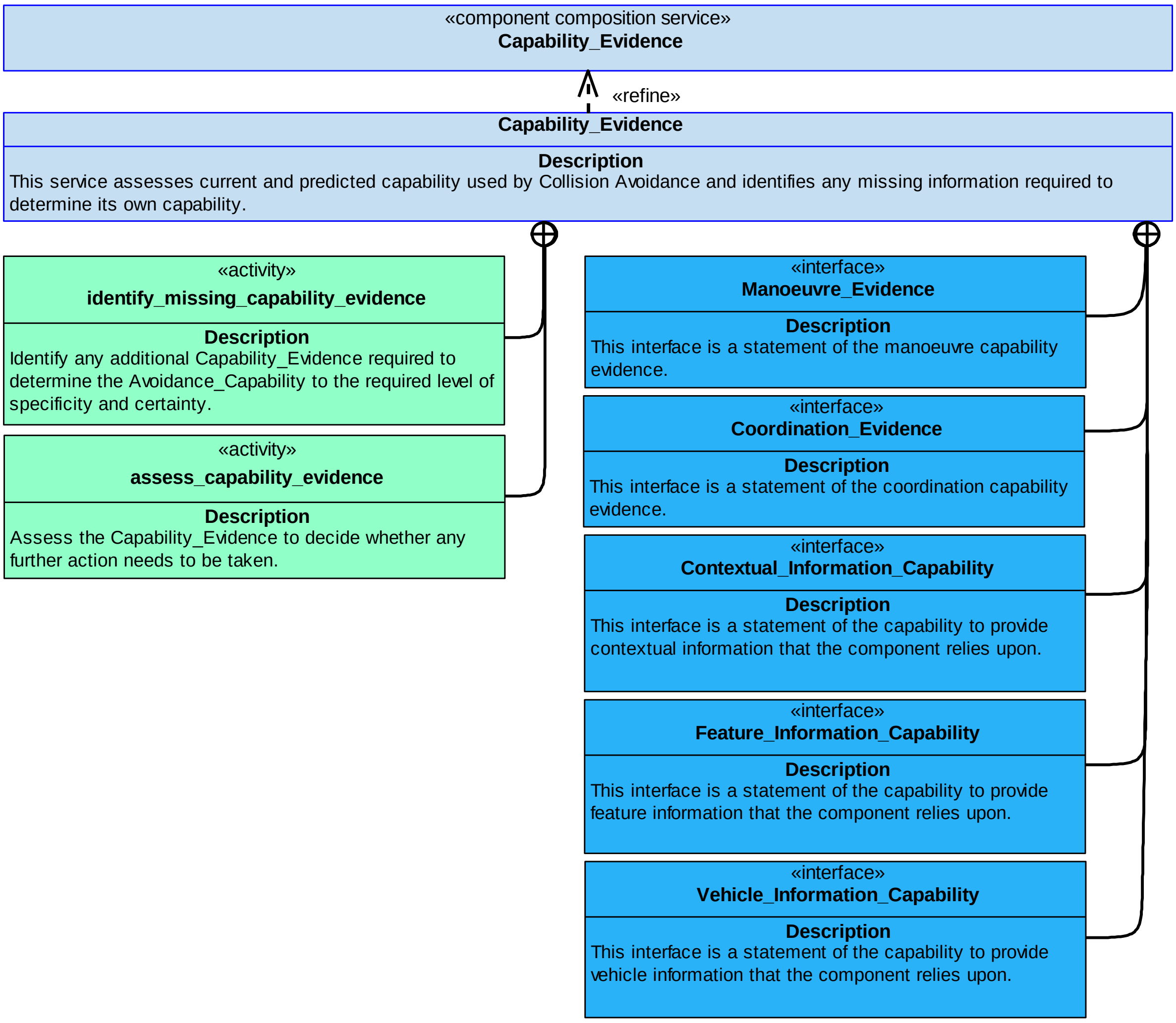


Figure 106: Capability_Evidence Service Definition



This service assesses current and predicted capability used by Collision Avoidance and identifies any missing information required to determine its own capability.

Identify any additional Capability_Evidence required to determine the Avoidance_Capability to the required level of specificity and certainty.

Assess the Capability_Evidence to decide whether any further action needs to be taken.

This interface is a statement of the manoeuvre capability evidence.

This interface is a statement of the coordination capability evidence.

This interface is a statement of the capability to provide contextual information that the component relies upon.

This interface is a statement of the capability to provide feature information that the component relies upon.

This interface is a statement of the capability to provide vehicle information that the component relies upon.

Figure 107: Capability_Evidence Service Policy

Capability_Evidence

This service assesses current and predicted capability used by Collision Avoidance and identifies any missing information required to determine its own capability.

Interfaces

Manoeuvre_Evidence

This interface is a statement of the manoeuvre capability evidence.

Coordination_Evidence

This interface is a statement of the coordination capability evidence.

Vehicle_Information_Capability

This interface is a statement of the capability to provide vehicle information that the component relies upon.

Feature_Information_Capability

This interface is a statement of the capability to provide feature information that the component relies upon.

Contextual_Information_Capability

This interface is a statement of the capability to provide contextual information that the component relies upon.

Activities

identify_missing_capability_evidence

Identify any additional Capability_Evidence required to determine the Avoidance_Capability to the required level of specificity and certainty.

assess_capability_evidence

Assess the Capability_Evidence to decide whether any further action needs to be taken.

5.4.2.4.7.2 Service Dependencies

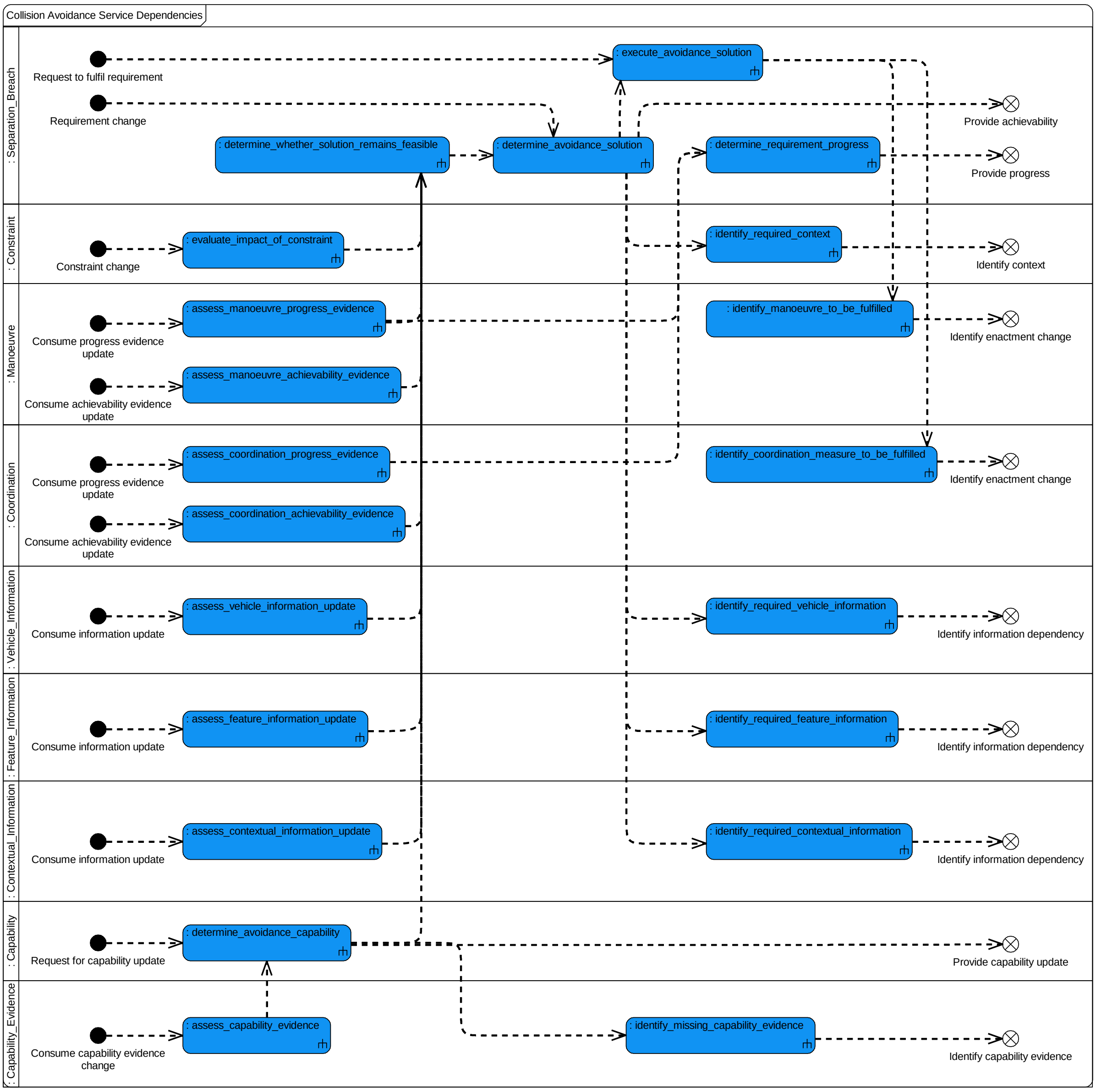

Collision Avoidance Service Dependencies
: Separation_Breach
Request to fulfil requirement
Requirement change
: execute_avoidance_solution
: determine_whether_solution_remains_feasible
: determine_avoidance_solution
: determine_requirement_progress
Provide achievability
Provide progress
: Constraint
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
: Manoeuvre
Consume progress evidence update
: assess_manoeuvre_progress_evidence
: identify_manoeuvre_to_be_fulfilled
Identify enactment change
Consume achievability evidence update
: assess_manoeuvre_achievability_evidence
: Coordination
Consume progress evidence update
: assess_coordination_progress_evidence
: identify_coordination_measure_to_be_fulfilled
Identify enactment change
Consume achievability evidence update
: assess_coordination_achievability_evidence
: Vehicle_Information
Consume information update
: assess_vehicle_information_update
: identify_required_vehicle_information
Identify information dependency
: Feature_Information
Consume information update
: assess_feature_information_update
: identify_required_feature_information
Identify information dependency
: Contextual_Information
Consume information update
: assess_contextual_information_update
: identify_required_contextual_information
Identify information dependency
: Capability
Request for capability update
: determine_avoidance_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 108: Collision Avoidance Service Dependencies

5.4.2.5 Collision Prediction

5.4.2.5.1 Role

The role of Collision Prediction is to predict collisions between the protected object and hazards such as terrain, another vehicle, a building or area of extreme weather.

5.4.2.5.2 Overview

Control Architecture

Collision Prediction is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Collision Prediction will perform a Proximity assessment for a Protected_Object in relation to a Hazardous_Object (such as terrain, a building, another vehicle, or area of extreme weather), taking into consideration the operational Context, to determine if a Breach has occurred or is expected to occur.

Examples of Use

Collision Prediction can be used:

- Where there is a requirement to predict collisions between Protected_Objects and Hazardous_Objects during the execution phase of a mission.
- As part of an ACAS implementation (with other components).

5.4.2.5.3 Service Summary

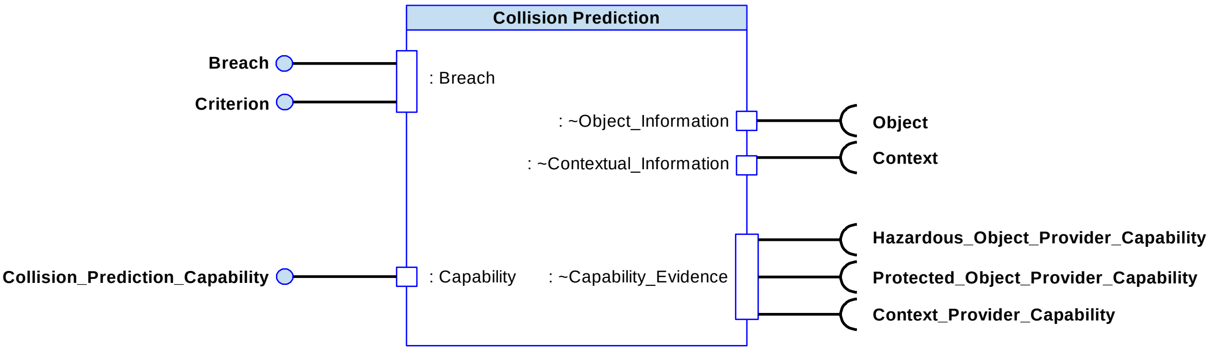

Collision Prediction
Breach
Criterion
: Breach
: ~Object_Information
: ~Contextual_Information
Object
Context
Hazardous_Object_Provider_Capability
Collision_Prediction_Capability
: Capability
: ~Capability_Evidence
Protected_Object_Provider_Capability
Context_Provider_Capability

Figure 109: Collision Prediction Service Summary

5.4.2.5.4 Responsibilities

capture_prediction_requirements

- To capture Requirements for collision prediction (e.g. during taxi, transit or air-to-air refuelling).

capture_measurement_criteria_for_collision_prediction

- To capture provided Measurement_Criterion/criteria for collision prediction (e.g. confidence of prediction).

determine_breach

- To determine an actual or predicted Breach.

determine_breach_status

- To determine the status of an actual or predicted Breach, e.g. cleared or active.

identify_rules

- To identify the rules that apply when determining if a Breach has occurred or is predicted.

determine_quality_of_deliverables

- To determine the quality of the collision prediction, measured against given Requirements and Measurement_Criterion/criteria.

assess_prediction_capability

- To assess the Prediction_Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Prediction_Capability assessment.

predict_capability_progression

- To predict the progression of the component's Prediction_Capability over time and with use.

5.4.2.5.5 Subject Matter Semantics

The subject matter of Collision Prediction is predicting whether Protected_Object Proximity to Hazardous_Objects will constitute a Breach.

Exclusions

The subject matter of Collision Prediction does not include:

- The avoidance of a collision, only the prediction that one may be imminent.
- The planning of a route to avoid a conflict with terrain or other known Hazardous_Objects.

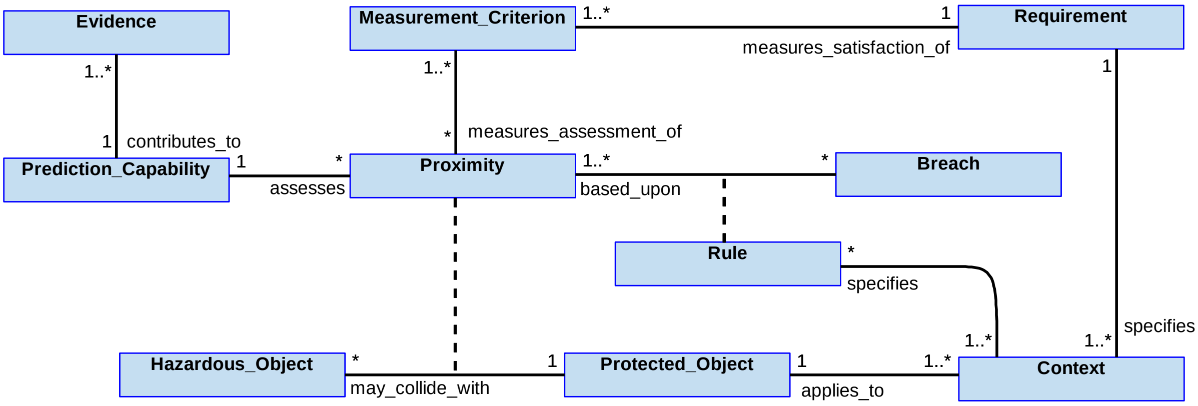


Figure 110: Collision Prediction Semantics

5.4.2.5.5.1 Entities

Breach

A violation of the acceptable Proximity. For example, an aircraft is too close to a hillside whilst using a ruleset associated with terrain following, or that an advisory alert is required when another aircraft crosses a zone boundary.

Context

A condition under which Proximity assessments are to be made, e.g. the current phase of flight, particular mission activity, or formation.

Evidence

The information (e.g. from transponders, maps, or sensors) that enables Proximity to be determined.

Hazardous_Object

Something that has the potential to breach the acceptable proximity of the Protected_Object, such as terrain, a building or another vehicle. It can also represent a non-solid entity such as weather or a no-fly zone.

Measurement_Criterion

A criterion used to measure the success of the component's activities, e.g. confidence or timeliness.

Prediction_Capability

The component's capability to determine an actual or predicted Breach.

Protected_Object

An object for which collisions are being predicted.

Proximity

The current and predicted spatial closeness of the Hazardous_Object to a Protected_Object.

Requirement

A requirement to perform ongoing collision prediction in a particular Context, e.g. during transit or air-to-air refuelling.

Rule

A rule that states what constitutes a Breach.

5.4.2.5.6 Design Rationale

5.4.2.5.6.1 Assumptions

- A group of vehicles, including less-capable support drones (e.g. those with fewer or no sensors) or swarms can be considered a Protected_Object.
- There can be multiple Protected_Objects.
- Information on Hazardous_Objects that need to be avoided could be provided by multiple sources, e.g. sensors, Geography and Tactical Objects.
- Rules which define what constitutes a Breach may depend on the Context, e.g. the definition of a Breach for an aircraft undergoing air-to-air refuelling may differ from a Breach for a transiting aircraft.
- A vehicle may be both a Protected_Object and a Hazardous_Object. For example, ownship may be predicted to collide with its support drone.

5.4.2.5.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Collision Prediction:

- Data Driving - Prediction algorithms and Breach rules could be data-driven at build time or later in order to maintain compliance with regulatory standards, etc.

Extensions

- Extensions may be useful for implementing differing ways of determining if a collision may be imminent, e.g. for different prediction algorithms for avoidance of terrain, of objects in flight or objects during taxi.

Other Factors that were Taken into Account

- Specific prediction algorithms may be required to conform to regulatory standards, e.g. to meet FAA Collision Avoidance System logic version *"X",* thus supporting flight certifiability. ACAS II has been considered whilst defining this component, although it does not constrain or limit the component to only being developed that way.

Exploitation Considerations

- This component may need to understand the characteristics (including dynamics) of the Protected_Object and Hazardous_Object in order to predict a Breach.

- Prediction of collisions during ground operations is likely to require a higher resolution knowledge of object geometry.
- Although it is expected that this component will mostly be used to predict collisions between an Exploiting Platform and terrain or other vehicles, it could also be used to identify a potential Breach with non-solid entities, such as an area of extreme weather or a no-fly zone.
- Different levels of Breach may be associated with different levels of prediction accuracy or alert, e.g. to be aware of a Hazardous_Object coming into Proximity, or to advise immediate action is required to avoid a collision.
- Rules for collision prediction will need to be tailored for some operational circumstances (e.g. close formation flying, follow-me taxi, formation take-offs, or tanking). This may be driven by the currently employed tactics (see the Tasks component).
- Collision Prediction is not involved in the planning phase of a mission lifecycle. Tactical constraints applied by planning components should be more conservative than the appropriate Breach definitions, e.g. a terrain-following path should not be planned that will trigger constant Proximity alerts from this component.

5.4.2.5.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

Failure of this component would mean that an impending collision is not detected and so no avoiding action would be taken. Potential collisions would include:

- Terrain or obstacles.
- Other aircraft (in flight or on the ground).
- Obstructions during operation on the ground.

Whilst failure to deliver this function may conceivably lead to a collision, with the result likely to be loss of the air vehicle and fatalities, additional failures would have previously occurred to cause the intended path of the air vehicle to be on a collision course with something. For example, one of the following expectations not being met:

- The intended path is expected to be planned to maintain separation from terrain and obstacles.
- When in a region of airspace where the air vehicle is receiving a deconfliction service, ATC are expected to manage flightpaths so that the host air vehicle maintains separation from other aircraft.
- When on the ground and under the control of ATC, ATC are expected to manage groundpaths so that the host air vehicle maintains separation from other aircraft, ground vehicles and buildings.

This component is therefore considered to be DAL B. This is consistent with the requirements on civil aircraft for Airborne Collision Avoidance Systems (ACAS).

Where instances of this component contribute to hazards that are less severe or more reliance can be placed on other barriers to an accident, then the Exploiting Platform may require a less onerous DAL.

Whilst this component would identify that an avoidance manoeuvre is required, generating the manoeuvre is the responsibility of the Collision Avoidance component.

5.4.2.5.6.4 Security Considerations

The indicative security classification is O.

This component determines when the Proximity between the Protected_Object and a Hazardous_Object indicates a collision is possible. This is deemed O (and may form part of an ACAS installation), however the component will need to know the protected object's current speed and bearing information, etc. the aggregation of which may provide knowledge of performance capabilities or flight envelope; this data is considered SNEO for military platforms. Additionally, where determination of incursion into geographical features is performed, including terrain or no-fly zones, etc. then it seems likely that component will be processing positional information; during everyday operations performed in civil airspace this is considered O, but may be SNEO whilst on an active mission. To avoid possible loss of availability when crossing domain boundaries, use of relative, rather than absolute, positioning data should be explored and care taken to avoid aggregation of data.

The availability (timeliness) and integrity (correctness) of the determined avoidance activities for those that are required to act on them will need protecting. “Gaming” the prediction function by hostile forces to invoke a manoeuvre is a concern.

The component may be expected to at least partially satisfy security related functions relating to:

- **Identifying Data Sources**, supporting the authentication of identified Hazardous_Objects. Spoofing of transponder transmissions used by a TCAS installation is a particular concern.
- **Maintaining Audit Records**, providing accountability for any predicted collisions (thus supporting that of any avoidance manoeuvres), and in support of filing Mandatory Occurrence Reports, Voluntary Occurrence Reports and Airprox incidents.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- The system is expected to generate **Warnings and Notifications** in response to detected breaches, however spurious or excessive notifications may provide awareness of unexpected activity and therefore possible cyber attack.

The component is not expected to directly implement security enforcing functions, but will rely on the integrity of externally-sourced information, e.g. from sensors and other ACAS users.

5.4.2.5.7 Services

5.4.2.5.7.1 Service Definitions

5.4.2.5.7.1.1 Breach

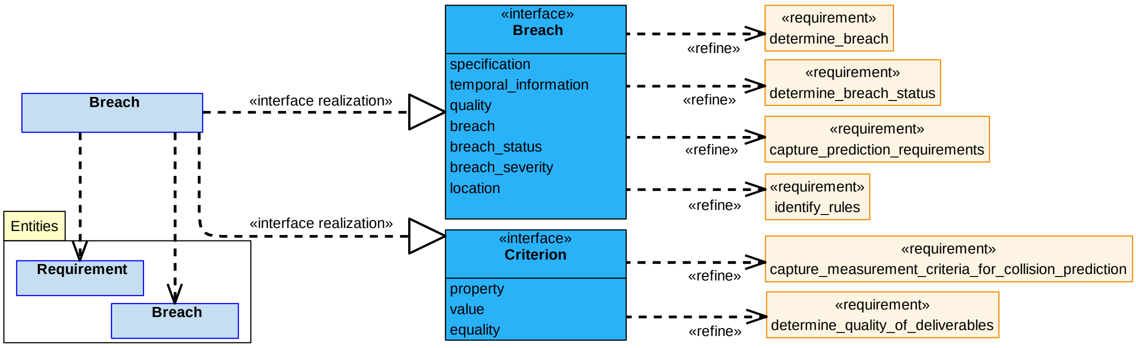
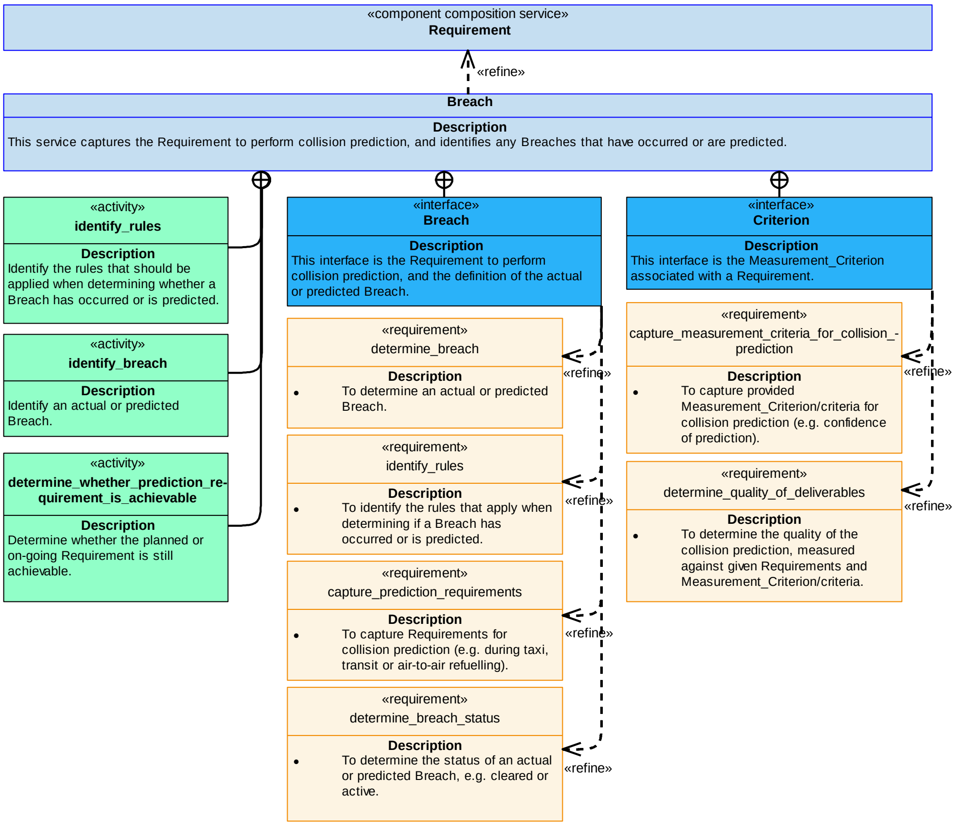


Figure 111: Breach Service Definition



Breach: This service captures the Requirement to perform collision prediction, and identifies any Breaches that have occurred or are predicted.

identify_rules: Identify the rules that should be applied when determining whether a Breach has occurred or is predicted.

identify_breach: Identify an actual or predicted Breach.

determine_whether_prediction_requirement_is_achievable: Determine whether the planned or on-going Requirement is still achievable.

«interface» Breach: This interface is the Requirement to perform collision prediction, and the definition of the actual or predicted Breach.

«interface» Criterion: This interface is the Measurement_Criterion associated with a Requirement.

determine_breach:
- To determine an actual or predicted Breach.

identify_rules:
- To identify the rules that apply when determining if a Breach has occurred or is predicted.

capture_prediction_requirements:
- To capture Requirements for collision prediction (e.g. during taxi, transit or air-to-air refuelling).

determine_breach_status:
- To determine the status of an actual or predicted Breach, e.g. cleared or active.

capture_measurement_criteria_for_collision_prediction:
- To capture provided Measurement_Criterion/criteria for collision prediction (e.g. confidence of prediction).

determine_quality_of_deliverables:
- To determine the quality of the collision prediction, measured against given Requirements and Measurement_Criterion/criteria.

Figure 112: Breach Service Policy

Breach

This service captures the Requirement to perform collision prediction, and identifies any Breaches that have occurred or are predicted.

Interfaces

Criterion

This interface is the Measurement_Criterion associated with a Requirement.

Attributes

property	The property to be measured, e.g. timeliness, confidence, or proximity.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Breach

This interface is the Requirement to perform collision prediction, and the definition of the actual or predicted Breach.

Attributes

specification	The definition of the Requirement, e.g. to perform ongoing collision prediction for a specific Protected_Object and a type of Hazardous_Object.
temporal_information	Temporal information, such as the actual or expected time of Breach.
quality	The quality of the Breach declaration or prediction, including the confidence and timeliness.
breach	The definition of the Breach that has been identified in response to a requirement to perform collision prediction, i.e. the Hazardous_Object and Protected_Object that are the subject of the breach, and their Proximity.
breach_status	The status of the Breach, e.g. cleared or live.
breach_severity	The severity of the Breach, e.g. traffic advisory or resolution advisory.
location	Where the Breach has occurred or is predicted to occur.

Activities

identify_breach

Identify an actual or predicted Breach.

identify_rules

Identify the rules that should be applied when determining whether a Breach has occurred or is predicted.

determine_whether_prediction_requirement_is_achievable

Determine whether the planned or on-going Requirement is still achievable.

5.4.2.5.7.1.2 Object_Information

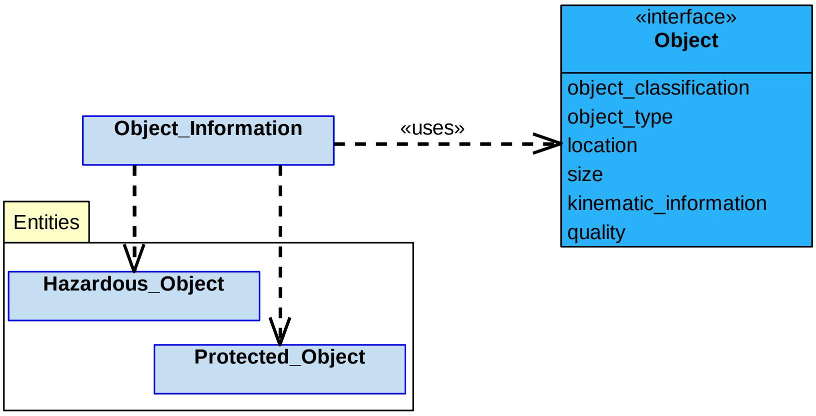


Figure 113: Object_Information Service Definition

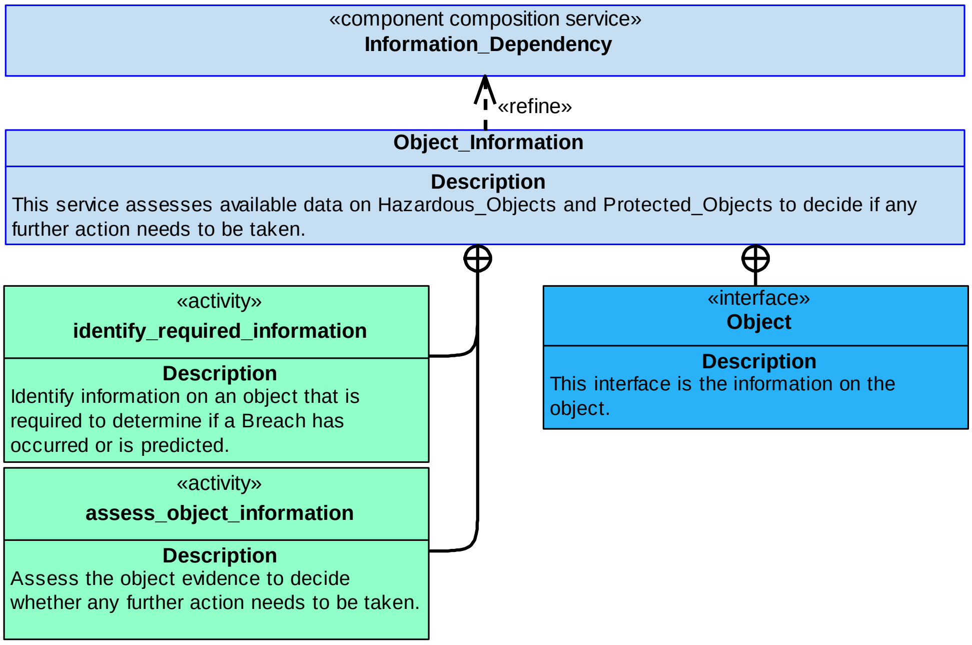

«component composition service»
Information_Dependency
«refine»
Object_Information
Description
This service assesses available data on Hazardous_Objects and Protected_Objects to decide if any further action needs to be taken.
«activity»
identify_required_information
Description
Identify information on an object that is required to determine if a Breach has occurred or is predicted.
«activity»
assess_object_information
Description
Assess the object evidence to decide whether any further action needs to be taken.
«interface»
Object
Description
This interface is the information on the object.

Figure 114: Object_Information Service Policy

Object_Information

This service assesses available data on Hazardous_Objects and Protected_Objects to decide if any further action needs to be taken.

Interface

Object

This interface is the information on the object.

Attributes

object_classification	Whether the object is a Hazardous_Object or a Protected_Object.
object_type	The type of object, e.g. building, aircraft, drone, or ownship.

location	Where the object is located, e.g. latitude / longitude / altitude.
size	The size and extent of the object.
kinematic_information	A set of information relating to the object's motion. This may include trajectory, or be separated into course, speed, acceleration (x/y/z), etc.
quality	The quality of the object information, e.g. confidence or resolution.

Activities

assess_object_information

Assess the object evidence to decide whether any further action needs to be taken.

identify_required_information

Identify information on an object that is required to determine if a Breach has occurred or is predicted.

5.4.2.5.7.1.3 Contextual_Information

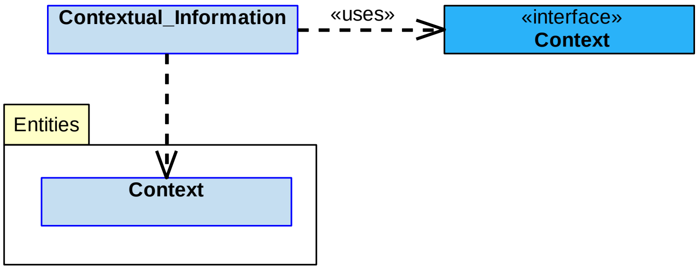


Figure 115: Contextual_Information Service Definition

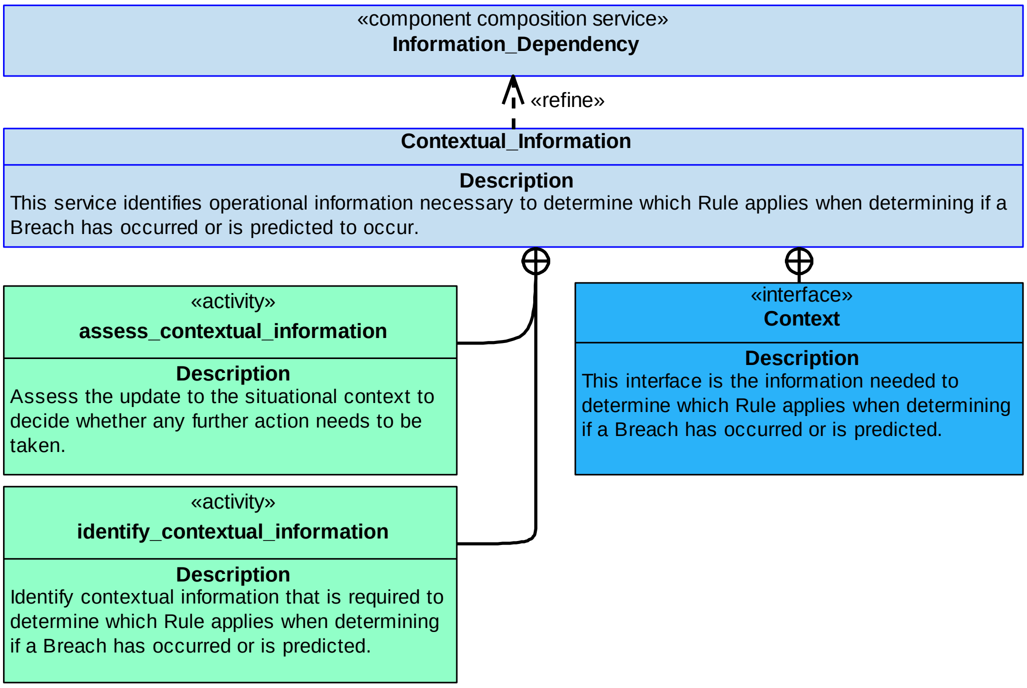

«component composition service»
Information_Dependency
«refine»
Contextual_Information
Description
This service identifies operational information necessary to determine which Rule applies when determining if a Breach has occurred or is predicted to occur.
«activity»
assess_contextual_information
Description
Assess the update to the situational context to decide whether any further action needs to be taken.
«activity»
identify_contextual_information
Description
Identify contextual information that is required to determine which Rule applies when determining if a Breach has occurred or is predicted.
«interface»
Context
Description
This interface is the information needed to determine which Rule applies when determining if a Breach has occurred or is predicted.

Figure 116: Contextual_Information Service Policy

Contextual_Information

This service identifies operational information necessary to determine which Rule applies when determining if a Breach has occurred or is predicted to occur.

Interface

Context

This interface is the information needed to determine which Rule applies when determining if a Breach has occurred or is predicted.

Activities

assess_contextual_information

Assess the update to the situational context to decide whether any further action needs to be taken.

identify_contextual_information

Identify contextual information that is required to determine which Rule applies when determining if a Breach has occurred or is predicted.

5.4.2.5.7.1.4 Capability

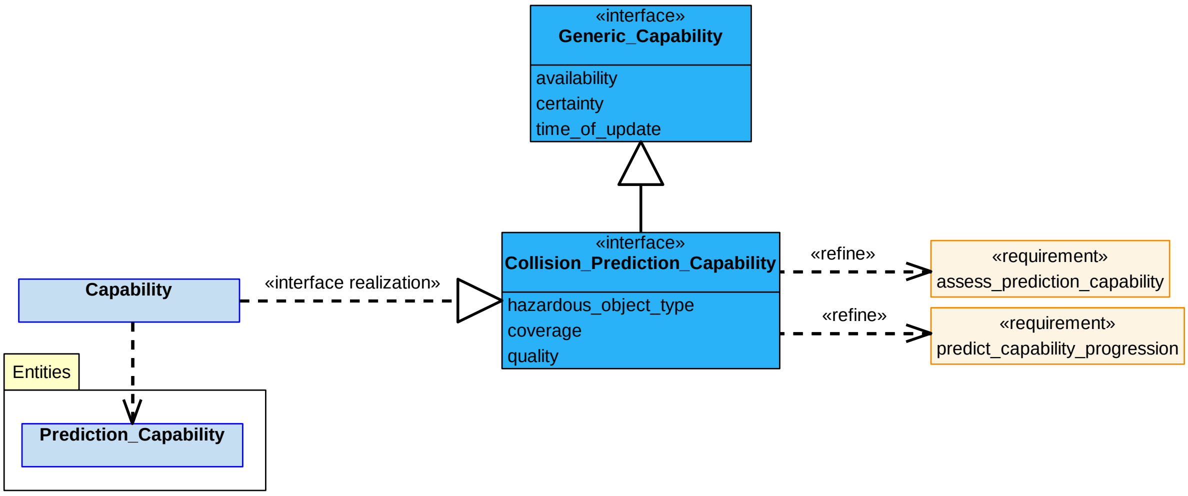


Figure 117: Capability Service Definition

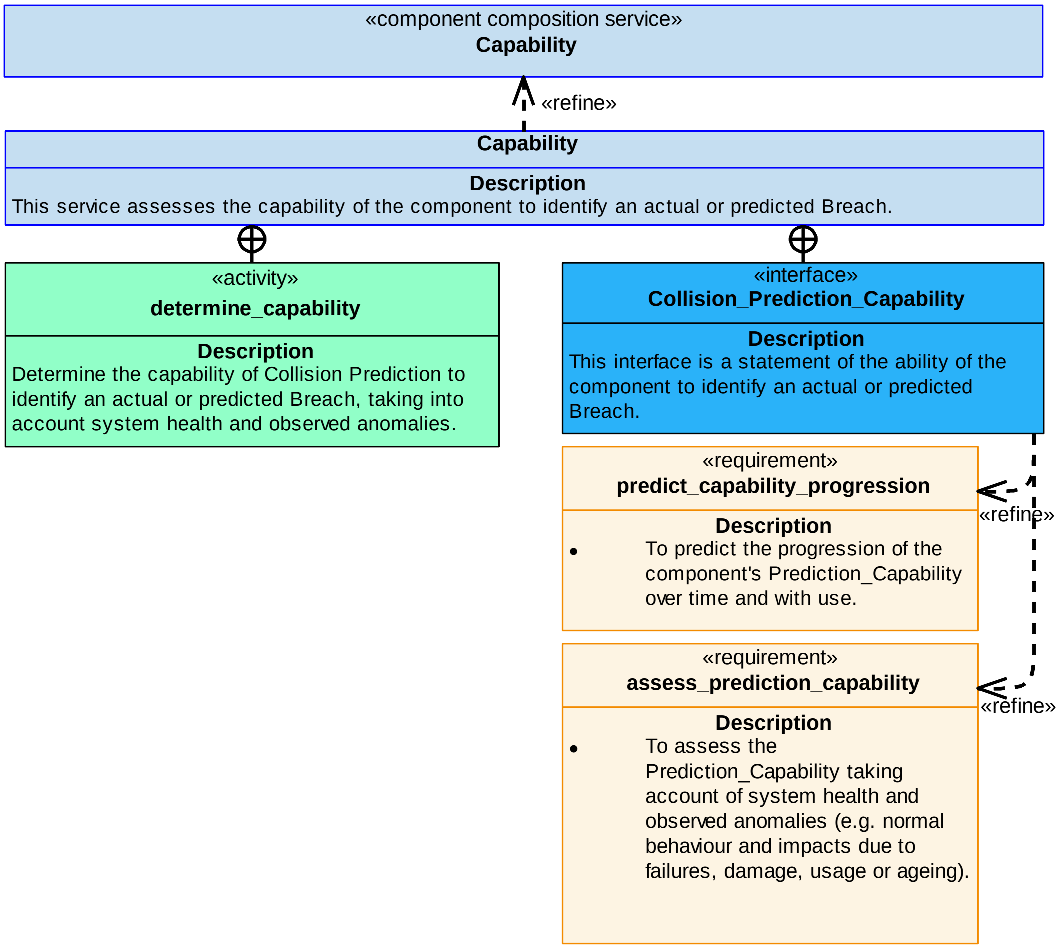

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the capability of the component to identify an actual or predicted Breach.
«activity»
determine_capability
Description
Determine the capability of Collision Prediction to identify an actual or predicted Breach, taking into account system health and observed anomalies.
«interface»
Collision_Prediction_Capability
Description
This interface is a statement of the ability of the component to identify an actual or predicted Breach.
«requirement»
predict_capability_progression
Description
To predict the progression of the component's Prediction_Capability over time and with use.
«refine»
«requirement»
assess_prediction_capability
Description
To assess the Prediction_Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 118: Capability Service Policy

Capability

This service assesses the capability of the component to identify an actual or predicted Breach.

Interface

Collision_Prediction_Capability

This interface is a statement of the ability of the component to identify an actual or predicted Breach.

Attributes

hazardous_object_type	The type of Hazardous_Objects for which a collision prediction service can be provided.
coverage	An absolute or relative volume in which a collision prediction service can be provided.
quality	The quality of collision prediction service.

Activity

determine_capability

Determine the capability of Collision Prediction to identify an actual or predicted Breach, taking into account system health and observed anomalies.

5.4.2.5.7.1.5 Capability_Evidence

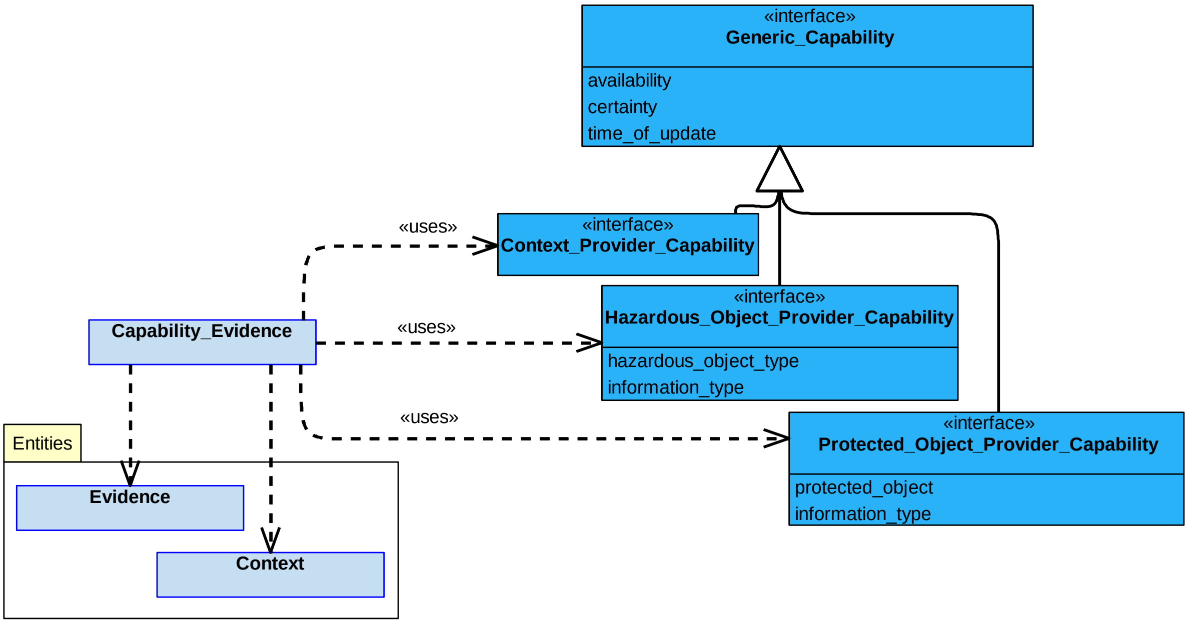


Figure 119: Capability_Evidence Service Definition

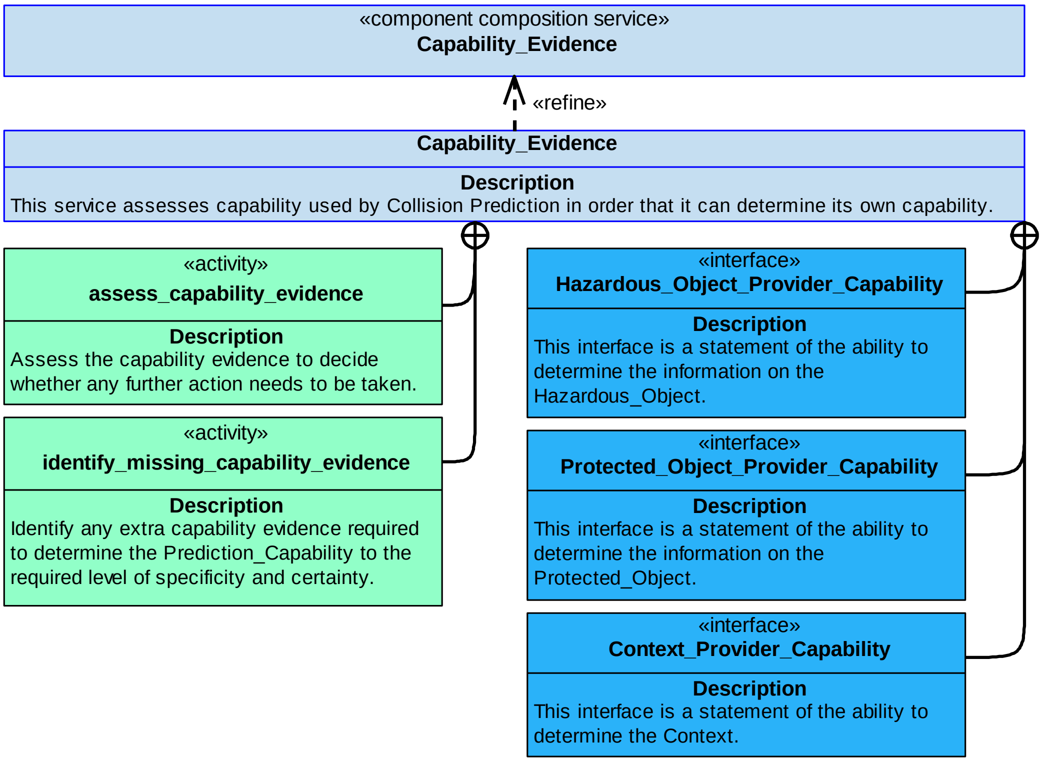

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service assesses capability used by Collision Prediction in order that it can determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Prediction_Capability to the required level of specificity and certainty.
«interface»
Hazardous_Object_Provider_Capability
Description
This interface is a statement of the ability to determine the information on the Hazardous_Object.
«interface»
Protected_Object_Provider_Capability
Description
This interface is a statement of the ability to determine the information on the Protected_Object.
«interface»
Context_Provider_Capability
Description
This interface is a statement of the ability to determine the Context.

Figure 120: Capability_Evidence Service Policy

Capability_Evidence

This service assesses capability used by Collision Prediction in order that it can determine its own capability.

Interfaces

Hazardous_Object_Provider_Capability

This interface is a statement of the ability to determine the information on the Hazardous_Object.

Attributes

hazardous_object_type	A type of Hazardous_Object on which information can be provided, e.g. terrain, ground vehicles, or other aircraft.
information_type	The type of information that can be provided, e.g. location, speed, or bearing.

Protected_Object_Provider_Capability

This interface is a statement of the ability to determine the information on the Protected_Object.

Attributes

protected_object	A specific Protected_Object on which information can be provided.
information_type	The type of information that can be provided, e.g. location, speed, or bearing.

Context_Provider_Capability

This interface is a statement of the ability to determine the Context.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Prediction_Capability to the required level of specificity and certainty.

5.4.2.5.7.2 Service Dependencies

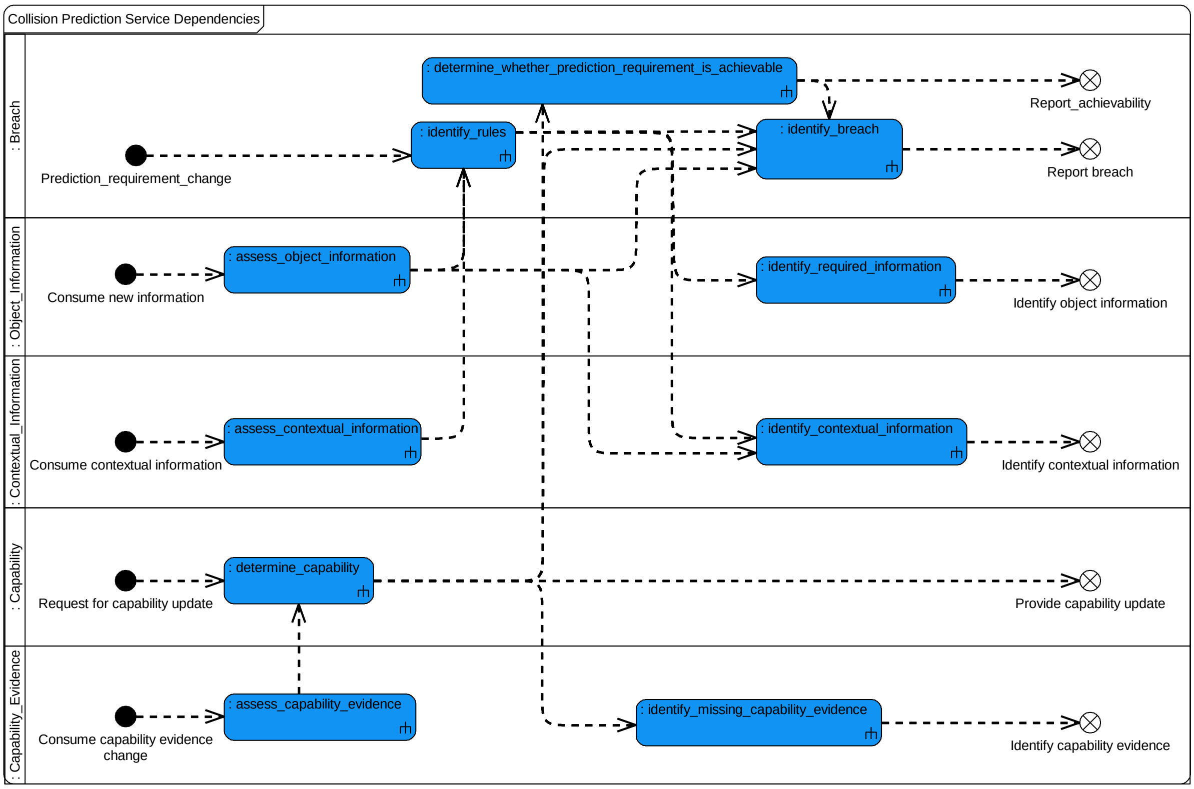

Collision Prediction Service Dependencies
: Breach
: Object_Information
: Contextual_Information
: Capability
: Capability_Evidence
: determine_whether_prediction_requirement_is_achievable
Report_achievability
: identify_rules
: identify_breach
Prediction_requirement_change
Report breach
: assess_object_information
Consume new information
: identify_required_information
Identify object information
: assess_contextual_information
Consume contextual information
: identify_contextual_information
Identify contextual information
: determine_capability
Request for capability update
Provide capability update
: assess_capability_evidence
Consume capability evidence change
: identify_missing_capability_evidence
Identify capability evidence

Figure 121: Collision Prediction Service Dependencies

5.4.2.6 Communication Links

5.4.2.6.1 Role

The role of Communication Links is to establish, maintain and optimise direct communication links between nodes.

5.4.2.6.2 Overview

Control Architecture

Communication Links is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Communication Links will receive a request to determine the feasibility of creating a link to a Node that meets the specified Link_Requirements. Communication Links will determine if such a link can be made and the appropriate set of parameters to use from the available Link_Options. The Link_Options, provided by the Link_Resources are limited by the Pre-conditions and Constraints. The set of selected Link_Options forms the Link_Solution. If the proposed Link_Solution is acceptable, Communication Links can then be tasked to establish the Link.

Examples of Use

- Communication Links will be required when communication between vehicles is required.
- Communication Links will be required when communication between a PRA Exploiting Platform and another platform is required.

5.4.2.6.3 Service Summary

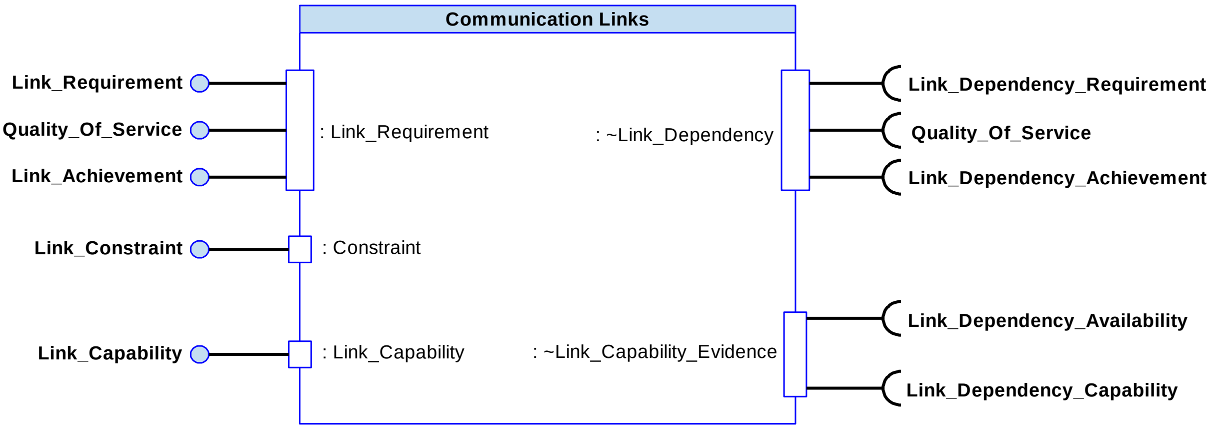


Figure 122: Communication Links Service Summary

5.4.2.6.4 Responsibilities

capture_link_requirements

- To capture given communication Link_Requirements (e.g. endpoint, throughput, reliability and latency).

capture_link_measurement_criteria

- To capture the criteria by which Link_Quality will be measured (e.g. reliability, throughput, and latency) for Link_Solutions.

capture_link_constraints

- To capture given communication link Constraints (e.g. maximum power level and spectrum usage).

determine_link_solution

- To determine a communication Link_Solution (i.e. a set of Link_Options), which includes the planning and configuration of the links, that meets the given requirements and Constraints using available Link_Resources.

determine_quality_of_link_performance_solution

- To determine the quality of a proposed communication Link_Solution against given required Link_Quality (i.e. determine the theoretical performance for a link solution).

determine_if_link_solution_remains_feasible

- To determine the feasibility of a planned or on-going Link_Solution.

identify_link_pre-conditions

- To identify Pre-conditions (e.g. to achieve or maintain a zone with sufficient signal visibility) to support a communication Link_Solution.

identify_link_performance_deviation

- To identify the deviation from expected performance of a Link_Solution against given criteria.

establish_and_maintain_links

- To establish and maintain a communication Link (including termination).

determine_link_cost

- To determine the Link_Cost of a communication Link_Solution for a given required Link_Quality (i.e. determining the viability of a link to support a need).

assess_link_capability

- To assess the system Capability to establish and maintain links using available Link_Resources within given Constraints.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the link management Capability assessment (e.g. observability/LoS assessment).

predict_communication_link_capability_progression

- To predict the progression of the Communication Links component's Capability over time and with use (i.e. if the communications link is failing, provide predictions on how long the capability is capable of functioning before it fails).

5.4.2.6.5 Subject Matter Semantics

The subject matter of Communication Links is the communication links between Nodes.

Exclusions

The subject matter of Communication Links does not include:

- The sequence of steps required to operate a specific communication resource.
- The handling or manipulation of any data to be transmitted through the communications link.
- The determination of the line of sight to a communications receiver.
- The determination of the required power signal level to transmit to a communications receiver.
- The correction for platform position when directing antenna.
- Whilst Communication Links will determine how Link_Requirements are met, it does not determine the need for a Link, which is the subject of another component (e.g. Networks).

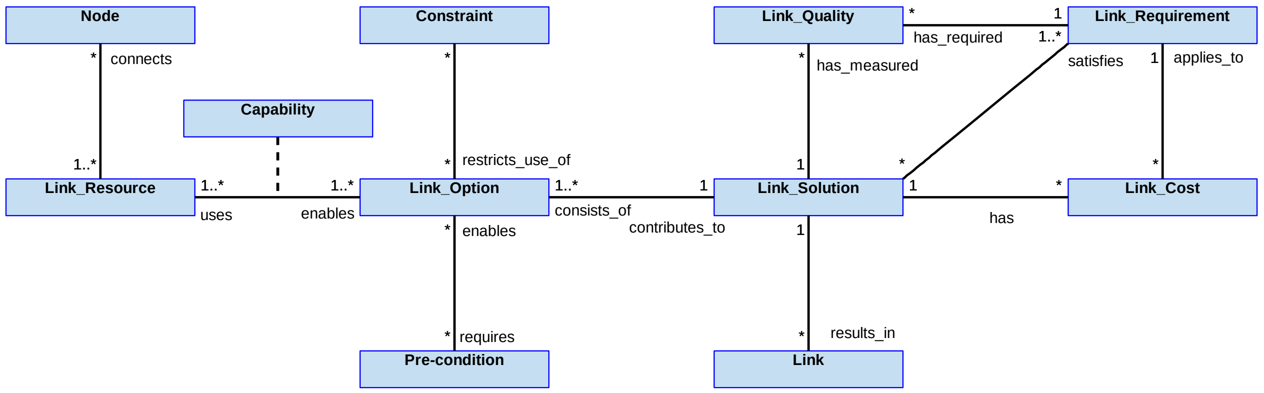


Figure 123: Communication Links Semantics

5.4.2.6.5.1 Entities

Capability

The capability of Communication Links to create or maintain a link.

Constraint

An externally imposed restriction that limits when or how a link can be used (e.g. EMCON, allowed power and transmission direction in terms of receiver equipment safety).

Link

An established communication link.

Link_Cost

The cost of providing the link (e.g. power usage or needing to maintain line of sight).

Link_Option

An option that may be selected when establishing or maintaining a communication link (e.g. frequency band, security, and antenna direction).

Link_Quality

The quality of a link (e.g. a measure of the reliability, and quality of service).

Link_Requirement

A requirement for a communication connection (e.g. Nodes to be connected, latency, and reliability).

Link_Resource

A resource that is used in providing a communication link. This could be a hardware device (e.g. voice radio, tactical datalink, modem, or antenna) or the additional resources that are needed for their operation (e.g. power or frequency reservation).

Link_Solution

A selected set of parameters and actions that can be used to establish a new link or maintain an existing link (e.g. resources that need powering and frequency to operate on).

Node

A source or sink of the communication link (e.g. a ground station or air vehicle).

Pre-condition

A condition that must be true before a link can be established (e.g. maximum distance or lack of obstacles between nodes).

5.4.2.6.6 Design Rationale

5.4.2.6.6.1 Assumptions

- Communication Links is responsible for controlling the links, but not for handling or processing the data that traverse those links.
- The introduction of any novel types of communication links is expected to occur during the development process, not during mission fit or mission execution.
- The type of capability will be known during mission fit, it is unlikely that during mission execution that the communication link capability can be significantly improved (e.g. through the replacement of hardware equipment or component software).
- Constraints and limitations of links may be set to conform to policies. Link constraints will also be defined by the Exploiting Platform and fit.
- During mission execution, the availability and usability of the actual communication resources is likely to change throughout the operation, which has a direct impact on establishing and maintaining links between nodes.
- The component will support and manage the use of link encryption. However this component will not handle the key material for link cryptography, which will be directed to the cryptographic devices by the Cryptographic Materials component.
- Determination of the required power levels to communicate to a third party is delegated to the Observability component.
- Managing corrections for platform position when directing antennae is delegated to the Pointing component.
- Communication Links will have knowledge of communications configuration, including channel frequencies and security policies (TRANSEC and COMSEC) where required.

5.4.2.6.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were taken into account when defining Communication Links:

- Use of Communications - This specifies how the communication Links between Nodes are managed by components, such as this one.
- Multi-Vehicle Coordination - This PYRAMID concept shows how tasks can be coordinated across different vehicles. This is useful in coordinating changes to communication links.
- Data Driving - This PYRAMID concept specifies the data driving principles to be used when configuring the Link_Resources.

Extensions

- It is not expected that extension components will be needed.

Other Factors that were Taken into Account

- Communication Links will need to represent the capability of links that are available to the system.

5.4.2.6.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component may result in the inability to transfer data, between, for example, a ground based control station and the air vehicle. This is primarily a concern for a UAS, but may apply to manned air vehicles where some functions are controlled by external users. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) then the air vehicle will have been designed to mitigate a loss of communications. For a UAS this would be achieved by relying on pre-determined automatic or autonomous behaviour. For this failure mode it is concluded that failure of this component may result a "significant reduction in safety margins", which has a major severity. Therefore the indicative DAL is C.

This component does not handle the data being transferred. Therefore, this component cannot corrupt data.

5.4.2.6.6.4 Security Considerations

The indicative security classification is O but will vary according to the datalink.

This component establishes and maintains communication Links between the Exploiting Platform and other entities; it does not handle the data being communicated. The communications policies and communications plan (including frequencies etc.) required for tactical datalinks will typically be SCEO/SNEO, however communications links with entities such as Air Traffic Control (e.g. for CPDLC) will be O. Instances of this component may therefore be required in differing security domains at each Node. Loss of confidentiality, integrity or availability of the communications links will have a detrimental impact on communications capability, and will need appropriate protection.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to security options for the connections, excessive use of a resource or changes to the policies, etc.
- **Maintaining Audit Records** of links established and broken down during the course of the mission.
- **Supporting Secure Remote Operation** by means of establishing and maintaining the control links necessary.
- Carrying out **System Status and Monitoring**, poor link performance is a possible indicator of jamming or DoS cyber attack.

The component is expected to at least partially satisfy security enforcing functions by:

- **Preventing Cyber Attacks and Malware**; determining actions to mitigate some types of cyber attacks, such as changing encoding to counter jamming.

- **Securing Communications** through the management and application of policies, triggering changes to frequencies to counter jamming, etc. This component will be cognisant of security classification in order to select the security options necessary for a particular link, e.g. whether link encryption is required to maintain confidentiality, although it does not perform the encryption.

5.4.2.6.7 Services

5.4.2.6.7.1 Service Definitions

5.4.2.6.7.1.1 Link_Requirement

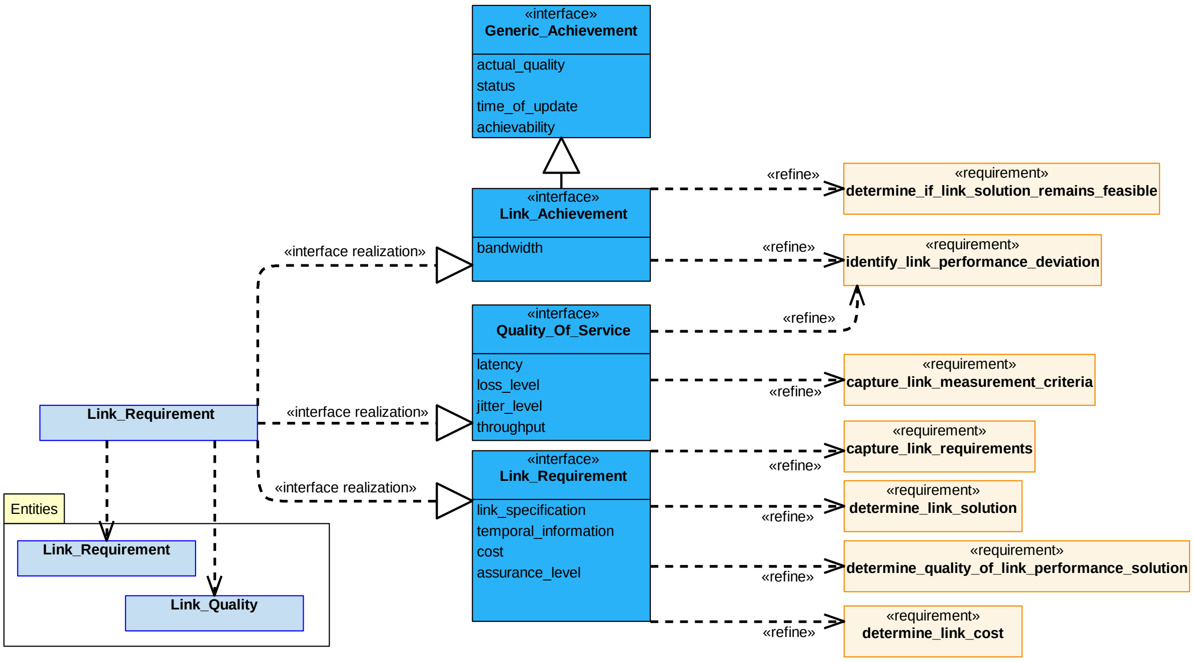


Figure 124: Link_Requirement Service Definition

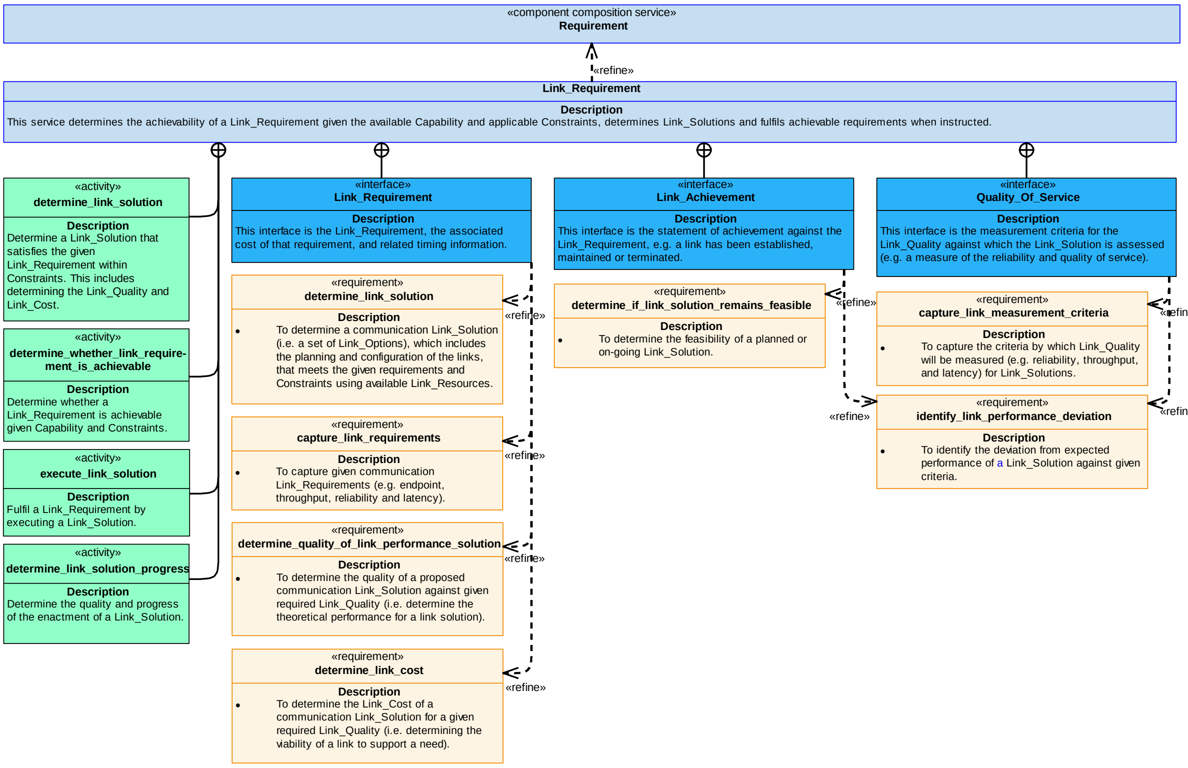

«component composition service»
Requirement
«refine»
Link_Requirement
Description
This service determines the achievability of a Link_Requirement given the available Capability and applicable Constraints, determines Link_Solutions and fulfils achievable requirements when instructed.
«activity»
determine_link_solution
Description
Determine a Link_Solution that satisfies the given Link_Requirement within Constraints. This includes determining the Link_Quality and Link_Cost.
«activity»
determine_whether_link_requirement_is_achievable
Description
Determine whether a Link_Requirement is achievable given Capability and Constraints.
«activity»
execute_link_solution
Description
Fulfil a Link_Requirement by executing a Link_Solution.
«activity»
determine_link_solution_progress
Description
Determine the quality and progress of the enactment of a Link_Solution.
«interface»
Link_Requirement
Description
This interface is the Link_Requirement, the associated cost of that requirement, and related timing information.
«requirement»
determine_link_solution
Description
To determine a communication Link_Solution (i.e. a set of Link_Options), which includes the planning and configuration of the links, that meets the given requirements and Constraints using available Link_Resources.
«refine»
«requirement»
capture_link_requirements
Description
To capture given communication Link_Requirements (e.g. endpoint, throughput, reliability and latency).
«refine»
«requirement»
determine_quality_of_link_performance_solution
Description
To determine the quality of a proposed communication Link_Solution against given required Link_Quality (i.e. determine the theoretical performance for a link solution).
«refine»
«requirement»
determine_link_cost
Description
To determine the Link_Cost of a communication Link_Solution for a given required Link_Quality (i.e. determining the viability of a link to support a need).
«refine»
«interface»
Link_Achievement
Description
This interface is the statement of achievement against the Link_Requirement, e.g. a link has been established, maintained or terminated.
«requirement»
determine_if_link_solution_remains_feasible
Description
To determine the feasibility of a planned or on-going Link_Solution.
«refine»
«refine»
«interface»
Quality_Of_Service
Description
This interface is the measurement criteria for the Link_Quality against which the Link_Solution is assessed (e.g. a measure of the reliability and quality of service).
«requirement»
capture_link_measurement_criteria
Description
To capture the criteria by which Link_Quality will be measured (e.g. reliability, throughput, and latency) for Link_Solutions.
«refine»
«requirement»
identify_link_performance_deviation
Description
To identify the deviation from expected performance of a Link_Solution against given criteria.
«refine»

Figure 125: Link_Requirement Service Policy

Link_Requirement

This service determines the achievability of a Link_Requirement given the available Capability and applicable Constraints, determines Link_Solutions and fulfils achievable requirements when instructed.

Interfaces

Link_Requirement

This interface is the Link_Requirement, the associated cost of that requirement, and related timing information.

Attributes

link_specification	The definition of the needs from a Link, e.g. the Nodes which it connects.
temporal_information	Timing information, e.g. the time to establish a Link.
cost	The Link_Cost of the Link_Solution, e.g. power usage, emissions level, or limits to movement.
assurance_level	The level of assurance required for a Link, e.g. whether the Link needs to be approved for safety critical traffic.

Quality_Of_Service

This interface is the measurement criteria for the Link_Quality against which the Link_Solution is assessed (e.g. a measure of the reliability and quality of service).

Attributes

latency	The level of delay.
loss_level	The rate of data being dropped.
jitter_level	The variability in latency.
throughput	The amount of data that can be sent and received within a specific timeframe.

Link_Achievement

This interface is the statement of achievement against the Link_Requirement, e.g. a link has been established, maintained or terminated.

Attribute

bandwidth	The amount of traffic supported on a Link.

Activities

determine_link_solution

Determine a Link_Solution that satisfies the given Link_Requirement within Constraints. This includes determining the Link_Quality and Link_Cost.

determine_whether_link_requirement_is_achievable

Determine whether a Link_Requirement is achievable given Capability and Constraints.

execute_link_solution

Fulfil a Link_Requirement by executing a Link_Solution.

determine_link_solution_progress

Determine the quality and progress of the enactment of a Link_Solution.

5.4.2.6.7.1.2 Link_Dependency

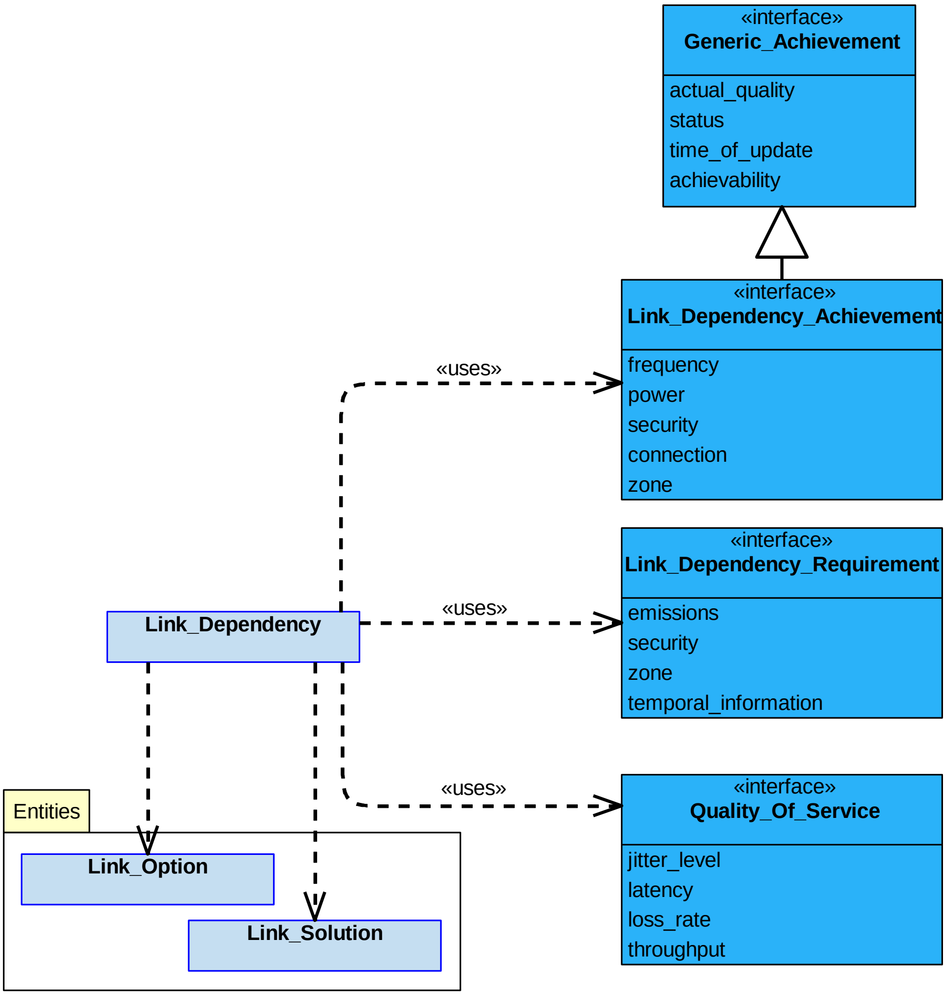


Figure 126: Link_Dependency Service Definition

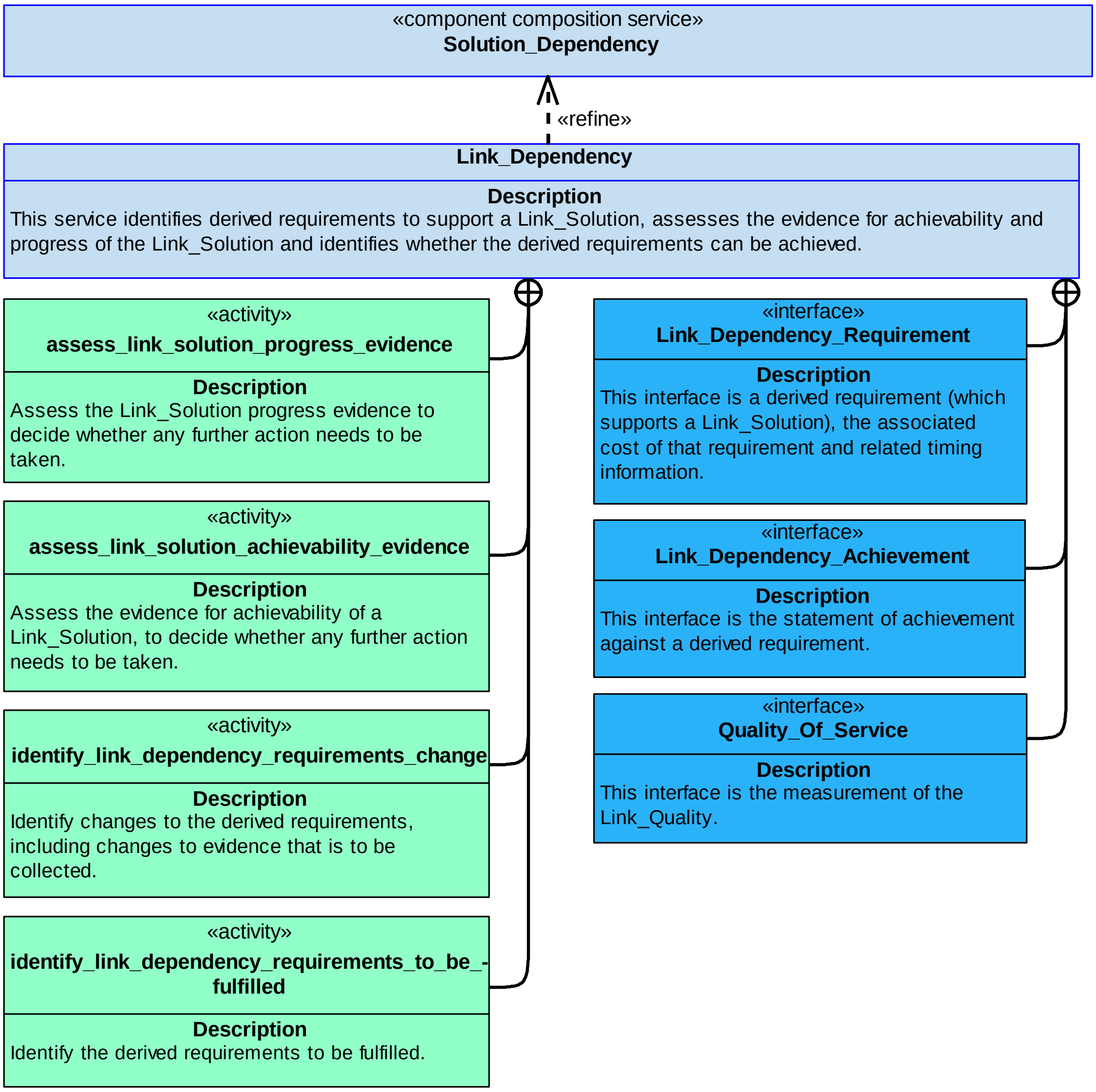

«component composition service»
Solution_Dependency
«refine»
Link_Dependency
Description
This service identifies derived requirements to support a Link_Solution, assesses the evidence for achievability and progress of the Link_Solution and identifies whether the derived requirements can be achieved.
«activity»
assess_link_solution_progress_evidence
Description
Assess the Link_Solution progress evidence to decide whether any further action needs to be taken.
«activity»
assess_link_solution_achievability_evidence
Description
Assess the evidence for achievability of a Link_Solution, to decide whether any further action needs to be taken.
«activity»
identify_link_dependency_requirements_change
Description
Identify changes to the derived requirements, including changes to evidence that is to be collected.
«activity»
identify_link_dependency_requirements_to_be_-fulfilled
Description
Identify the derived requirements to be fulfilled.
«interface»
Link_Dependency_Requirement
Description
This interface is a derived requirement (which supports a Link_Solution), the associated cost of that requirement and related timing information.
«interface»
Link_Dependency_Achievement
Description
This interface is the statement of achievement against a derived requirement.
«interface»
Quality_Of_Service
Description
This interface is the measurement of the Link_Quality.

Figure 127: Link_Dependency Service Policy

Link_Dependency

This service identifies derived requirements to support a Link_Solution, assesses the evidence for achievability and progress of the Link_Solution and identifies whether the derived requirements can be achieved.

Interfaces

Link_Dependency_Requirement

This interface is a derived requirement (which supports a Link_Solution), the associated cost of that requirement and related timing information.

Attributes

emissions	The required frequency and power emissions, e.g. antenna transmitted power.
security	The required confidentiality, e.g. encryption.

zone	The requirement to achieve or maintain a zone with sufficient signal visibility.
temporal_information	Information covering timing, e.g. start and end times.

Link_Dependency_Achievement

This interface is the statement of achievement against a derived requirement.

Attributes

frequency	Frequency related achievement, e.g. the status of a frequency band.
power	Power achievement, e.g. the status of the power provision.
security	Confidentiality achievement, e.g. the confidentiality level provided.
connection	The status of the Link establishment.
zone	The status of the achievement of a zone with sufficient signal visibility.

Quality_Of_Service

This interface is the measurement of the Link_Quality.

Attributes

jitter_level	The variability in latency.
latency	The level of delay.
loss_rate	The rate of data being dropped or the level of degradation.
throughput	The amount of data that can be sent and received within a specific timeframe.

Activities

assess_link_solution_achievability_evidence

Assess the evidence for achievability of a Link_Solution, to decide whether any further action needs to be taken.

assess_link_solution_progress_evidence

Assess the Link_Solution progress evidence to decide whether any further action needs to be taken.

identify_link_dependency_requirements_change

Identify changes to the derived requirements, including changes to evidence that is to be collected.

identify_link_dependency_requirements_to_be_fulfilled

Identify the derived requirements to be fulfilled.

5.4.2.6.7.1.3 Constraint

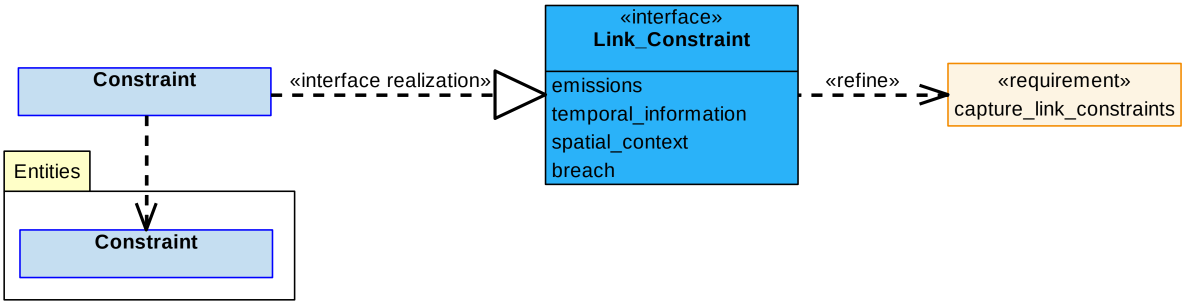


Figure 128: Constraint Service Definition

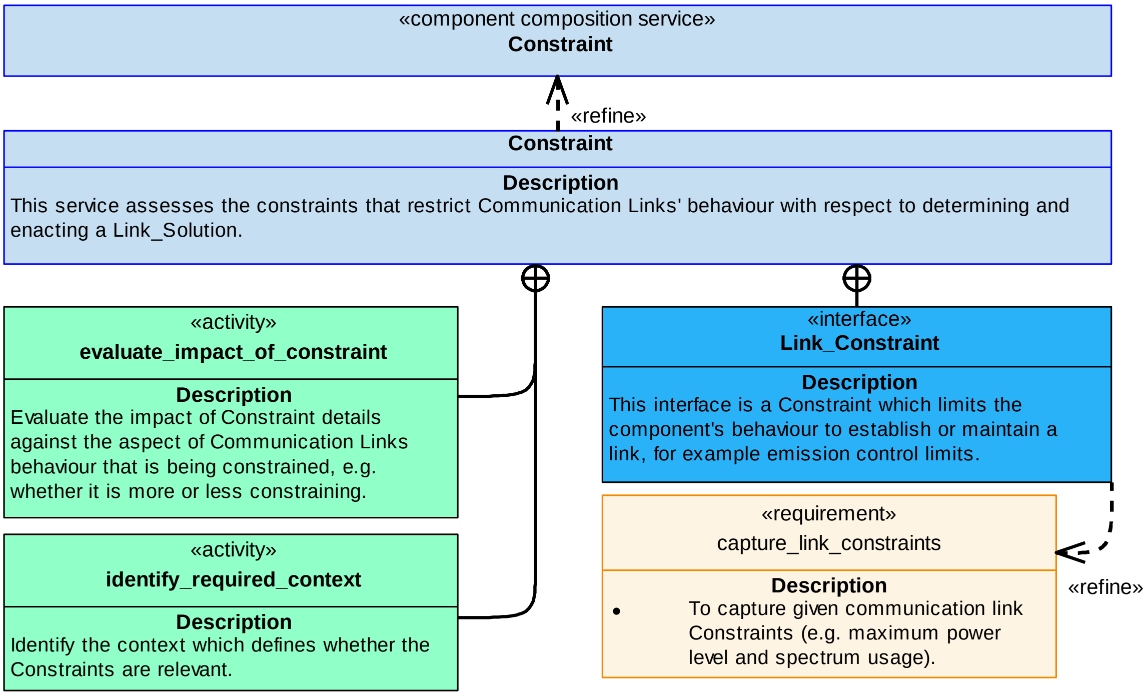

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the constraints that restrict Communication Links' behaviour with respect to determining and enacting a Link_Solution.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of Communication Links behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Link_Constraint
Description
This interface is a Constraint which limits the component's behaviour to establish or maintain a link, for example emission control limits.
«requirement»
capture_link_constraints
Description
To capture given communication link Constraints (e.g. maximum power level and spectrum usage).
«refine»

Figure 129: Constraint Service Policy

Constraint

This service assesses the constraints that restrict Communication Links' behaviour with respect to determining and enacting a Link_Solution.

Interface

Link_Constraint

This interface is a Constraint which limits the component's behaviour to establish or maintain a link, for example emission control limits.

Attributes

emissions	An emissions constraint, e.g. the maximum power level usage or disallowed frequency range.

temporal_information	Timing information pertaining to the periods of time when a constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
spatial_context	The spatial information for the zones in which a constraint is applicable or a constraint on the transmission direction (e.g. for receiver equipment safety).
breach	A statement that the constraint has been breached, e.g. that an emission has exceeded current constraints limits.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of Communication Links behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.6.7.1.4 Link_Capability

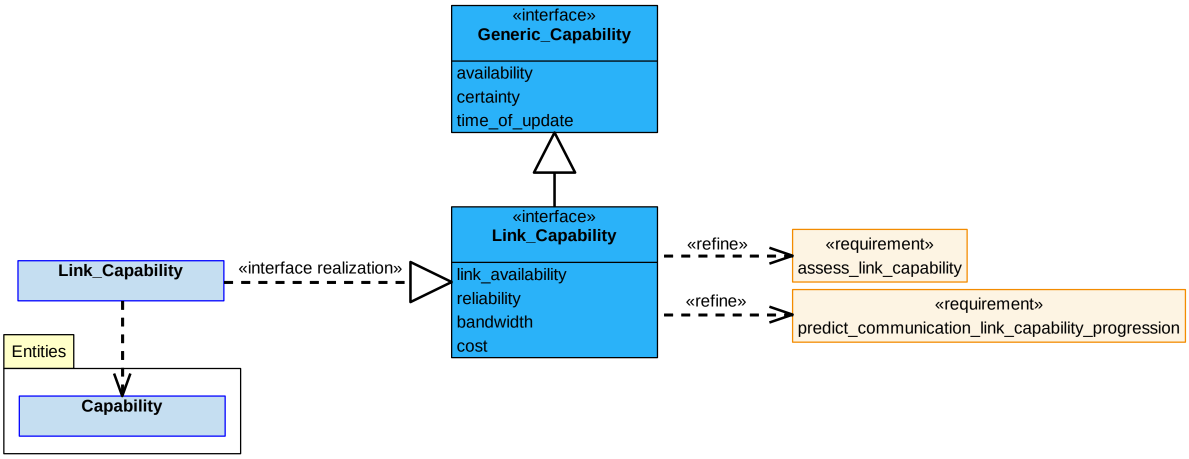


Figure 130: Link_Capability Service Definition

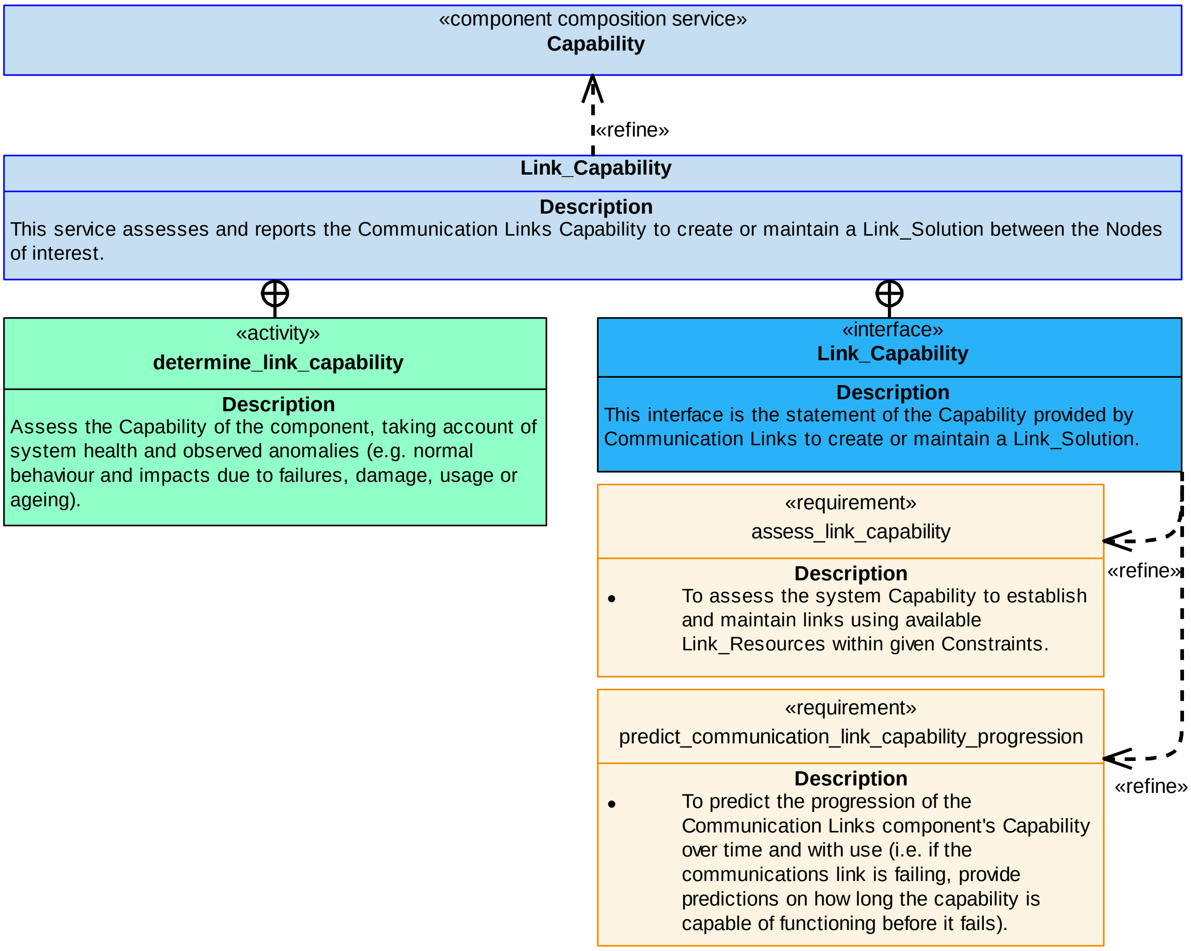

«component composition service»
Capability
«refine»
Link_Capability
Description
This service assesses and reports the Communication Links Capability to create or maintain a Link_Solution between the Nodes of interest.
«activity»
determine_link_capability
Description
Assess the Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Link_Capability
Description
This interface is the statement of the Capability provided by Communication Links to create or maintain a Link_Solution.
«requirement»
assess_link_capability
Description
To assess the system Capability to establish and maintain links using available Link_Resources within given Constraints.
«refine»
«requirement»
predict_communication_link_capability_progression
Description
To predict the progression of the Communication Links component's Capability over time and with use (i.e. if the communications link is failing, provide predictions on how long the capability is capable of functioning before it fails).
«refine»

Figure 131: Link_Capability Service Policy

Link_Capability

This service assesses and reports the Communication Links Capability to create or maintain a Link_Solution between the Nodes of interest.

<u>Interface</u>

Link_Capability

This interface is the statement of the Capability provided by Communication Links to create or maintain a Link_Solution.

<u>Attributes</u>

link_availability	The availability of specific Links between Nodes.
reliability	The likelihood for the Link to be maintained.
bandwidth	The maximum amount of traffic that can be supported on a Link.
cost	The cost of the Link capability (e.g. power usage, emissions level, or limits to movement).

Activity

determine_link_capability

Assess the Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.6.7.1.5 Link_Capability_Evidence

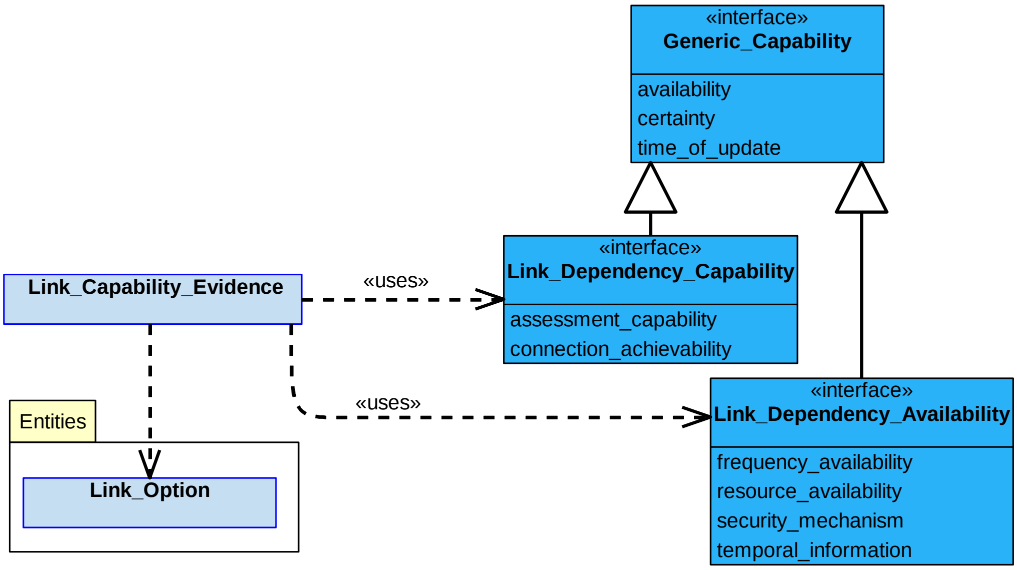


Figure 132: Link_Capability Evidence Service Definition

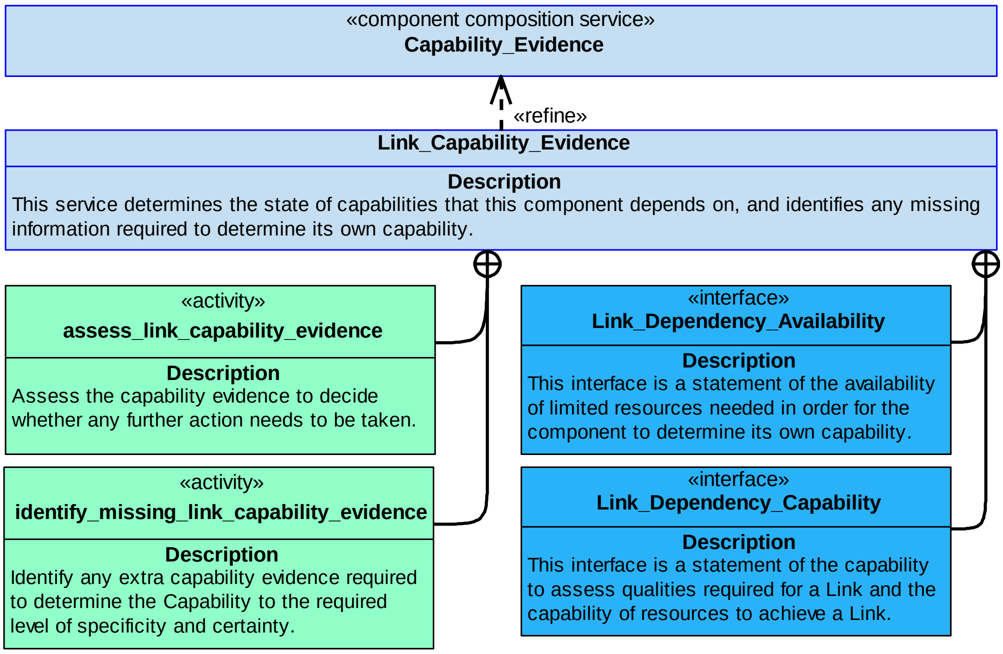

«component composition service»
Capability_Evidence
«refine»
Link_Capability_Evidence
Description
This service determines the state of capabilities that this component depends on, and identifies any missing information required to determine its own capability.
«activity»
assess_link_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_link_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Link_Dependency_Availability
Description
This interface is a statement of the availability of limited resources needed in order for the component to determine its own capability.
«interface»
Link_Dependency_Capability
Description
This interface is a statement of the capability to assess qualities required for a Link and the capability of resources to achieve a Link.

Figure 133: Link_Capability Evidence Service Policy

Link_Capability_Evidence

This service determines the state of capabilities that this component depends on, and identifies any missing information required to determine its own capability.

Interfaces

Link_Dependency_Availability

This interface is a statement of the availability of limited resources needed in order for the component to determine its own capability.

Attributes

frequency_availability	The available frequency bands.
resource_availability	The availability of non-consumable resources (e.g. a radio or antenna).
security_mechanism	The availability of provision of Link security.
temporal_information	Information covering timing of the capability, e.g. the availability of the capability.

Link_Dependency_Capability

This interface is a statement of the capability to assess qualities required for a Link and the capability of resources to achieve a Link.

Attributes

assessment_capability	A statement of the ability to determine qualities required for a Link, e.g. the observability between Nodes.
connection_achievability	A statement of the capability of Link providing resources, e.g. the frequency capabilities available at a Node.

Activities

assess_link_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_link_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.6.7.2 Service Dependencies

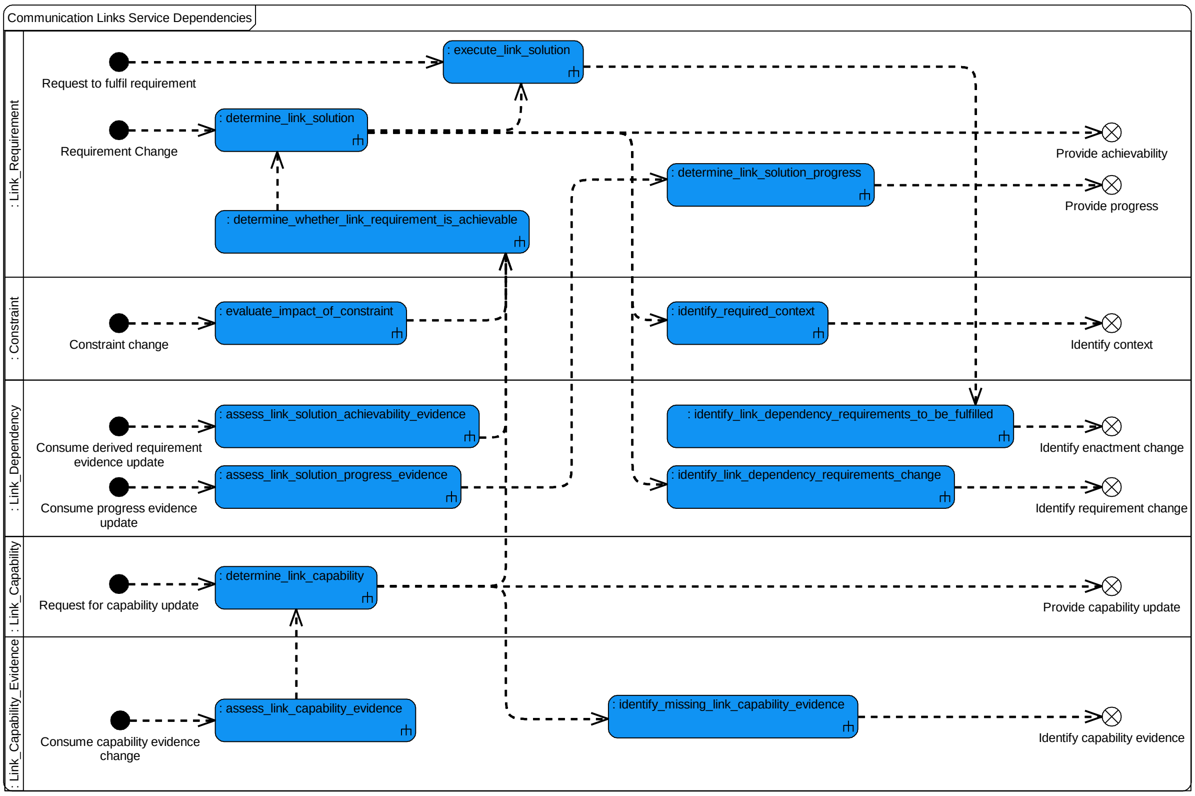

Communication Links Service Dependencies
: Link_Requirement
: Constraint
: Link_Dependency
: Link_Capability
: Link_Capability_Evidence
Request to fulfil requirement
: execute_link_solution
Requirement Change
: determine_link_solution
Provide achievability
: determine_link_solution_progress
Provide progress
: determine_whether_link_requirement_is_achievable
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume derived requirement evidence update
: assess_link_solution_achievability_evidence
: identify_link_dependency_requirements_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_link_solution_progress_evidence
: identify_link_dependency_requirements_change
Identify requirement change
Request for capability update
: determine_link_capability
Provide capability update
Consume capability evidence change
: assess_link_capability_evidence
: identify_missing_link_capability_evidence
Identify capability evidence

Figure 134: Communication Links Service Dependencies

5.4.2.7 Communicator

5.4.2.7.1 Role

The role of Communicator is to provide an interface to manage communication resources.

5.4.2.7.2 Overview

Control Architecture

Communicator is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When there is requirement to use a Communicator_Resource (for example send or receive a transmission using a transceiver) by the Exploiting Platform, Communicator will determine how best to achieve the action with the available resources. Communicator will then use the selected resources to enact the solution. Communicator can also monitor the communication resources whilst the solution is being enacted to report progress.

Examples of Use

Communicator will be used whenever a communication related activity forms part of a deployment, such as:

- Receiving by or sending from the Communicator_Resource by the Exploiting Platform (for example using a line of sight radio link or a transponder).
- Determining the direction of a transmission (for example signal lock).

5.4.2.7.3 Service Summary

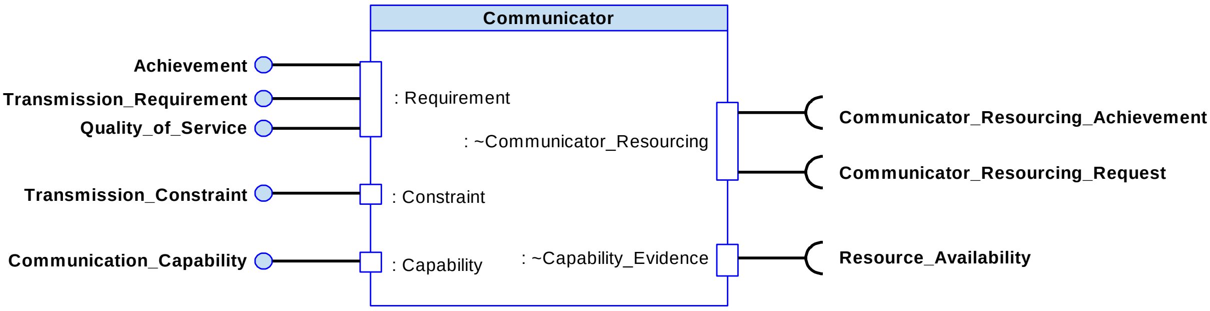


Figure 135: Communicator Service Summary

5.4.2.7.4 Responsibilities

capture_requirements_for_communicator_resources

- To capture provided Requirements (e.g. power, latency, loss rate, direction and throughput) for the use of Communicator_Resources.

capture_constraints_for_communicator_resources

- To capture provided Constraints for use of Communicator_Resources (e.g. maximum power or frequency).

determine_transmission_sequence

- To determine a Transmission_Sequence for the use of Communicator_Resources that will meet given Requirements, including forward error correction, frequency migration, frequency and spatial diversity.

identify_transmission_sequence_remains_feasible

- To identify if a Transmission_Sequence in progress remains feasible given current Communicator_Resources.

coordinate_use_of_resources

- To coordinate the use of Communicator_Resources (e.g. determining signal direction including signal lock).

manage_transmissions

- To manage incoming and outgoing Transmissions to an external entity, including initiation and termination (e.g. transceiver handshaking, keep-alive and timing synchronisation).

identify_progress_of_transmission_sequence

- To identify the progress of a Communicator_Resource Transmission_Sequence against the Requirements.

determine_quality_of_transmission_sequence

- To determine the quality of a Communicator_Resource Transmission_Sequence against a given Quality_of_Service.

determine_quality_of_transmission

- To determine the quality of the Transmission provided by Communicator_Resources during execution, measured against the given Requirements and Quality_of_Service.

assess_communicator_resource_capability

- To assess the Capability provided by Communicator_Resources, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Communicator_Resource Capability assessment.

predict_capability_progression

- To predict the progression of the Communicator_Resources' Capability over time and with use.

5.4.2.7.5 Subject Matter Semantics

The subject matter of Communicator is the resources that enable communications.

Exclusions

The subject matter of Communicator does not include:

- Encryption, including the frequency selection of frequency hopping and scramble techniques.
- Planning and knowledge of "possible links".
- Knowledge of "multiple nodes".
- Initial directivity of connections.
- The determination of mitigation for degradation of communications.
- Predicting link loss.

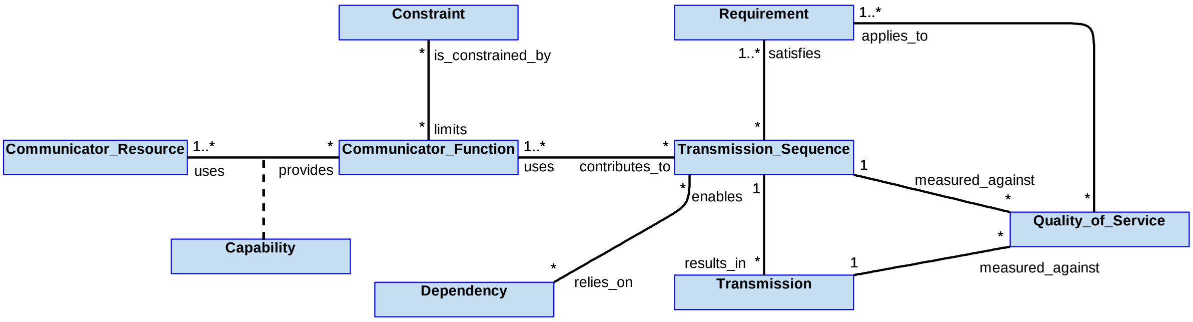


Figure 136: Communicator Semantics

5.4.2.7.5.1 Entities

Capability

A piece of communications functional capability that can be provided by the Communicator_Resources.

Communicator_Function

An operation that can be performed by the Communicator_Resource (for example being able to send or receive a transmission).

Communicator_Resource

A resource that can be used by the Communicator component, e.g. transmitters, receivers or transceivers.

Constraint

An externally imposed restriction that limits when or how a Communicator_Function can be used.

Dependency

Something that the component may rely on to successfully perform its Transmission_Sequence (e.g. power and cooling needs to support a Communicator_Function).

Quality_of_Service

The quality of Transmission_Sequence and Transmission that will be measured, e.g. delay, signal-to-noise ratio, error rate, throughput, or received power.

Requirement

A demand for the Communicator component to achieve a communication action, e.g. transmission direction, or radio communication set to a particular channel.

Transmission

What is transmitted or received as part of the communication Transmission_Sequence, e.g. the data exchanged.

Transmission_Sequence

The sequence of steps needed to be taken, using the available Communicator_Functions, in order to satisfy the Requirement(s) (e.g. synchronisation, a transmission, termination of a transmission, or signal lock).

5.4.2.7.6 Design Rationale

5.4.2.7.6.1 Assumptions

- Communicator will have knowledge of the communications configuration, including channel frequencies and security policies (TRANSEC and COMSEC) where required.
- Communications will require high integrity and availability checking.
- Communicator will be aware of any available fine tuning capabilities, and be able to provide feedback.
- Radio interference (including jamming) may be detected by this component, but resultant actions will be determined by another component.

5.4.2.7.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Communicator:

- Interaction with Equipment - This PYRAMID concept specifies how this component supports new and different pieces of equipment to interact with the system.
- Use of Communications - This PYRAMID concept specifies how communications are managed by components such as this one.

Extensions

The Communicator component may be implemented using extensions to cater for:

- Alternative link protocols.
- Alternative signal processing algorithms (including frequency hopping).

- Alternative rule sets for forward error correction.

Exploitation Considerations

- Define clear demarcation of the function of the component and equipment under its control.

5.4.2.7.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

- This component provides information to ATS and proximate aircraft relating to identity, location and emergency codes by interfacing directly with equipment (e.g. transponders). A failure of the component would result in loss of or erroneous transmission of data to ATS and proximate aircraft, increasing the risk of mid-air collision. This is considered a "large reduction in safety margins" (critical severity) and so the indicative IDAL is DAL B. This is consistent with the requirements on civil aircraft ADS-B or transponder systems.

Communication may use transmitters that may cause harm or damage to nearby people or objects. However, inadvertent transmission is not expected to result in a more onerous DAL for this component. This is because:

- Low power transmitters are expected to cause no worse than minor injury if ground crew are directly radiated. DAL C is appropriate for this major severity hazard.
- Where low power transmitters can cause more severe accidents in particular circumstances (e.g. whilst installing Electronic Explosive Devices (EEDs) during stores loading), it is expected that transmissions would be prevented by removing electrical power to transmitting hardware.
- If more powerful transmitters were used by this component, potentially leading to more severe consequences, it is expected that the Interlocks and Authorisation components would inhibit transmissions, whilst the air vehicle is in the wrong configuration or in the wrong location, independently of this component.

Where instances of this component contribute to hazards that are less severe or more reliance may be placed on other barriers to an accident, then the Exploiting Platform may require a less onerous DAL. For example, where this component is used to communicate between a UAV and UCS the indicative IDAL is expected to be DAL C.

5.4.2.7.6.4 Security Considerations

The indicative classification is O but may vary according to the communicator capability.

This component manages the Communicator_Resources available to the Exploiting Platform. It does not select communications policies or frequencies, etc. but they will be applied as directed. To the extent these are applicable in use, these are considered O, as will all clear communications with entities such as air traffic control. However, in mission planning, future frequency use may be considered SNEO and therefore have different confidentiality requirements. Instances of this component may therefore be required in differing security domains. Loss of confidentiality, integrity or availability of the communicator resources will have a detrimental impact on mission capability, and will need appropriate protection.

The component is expected to at least partially satisfy security related functions by:

- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Providing **System Status and Monitoring** of QoS, configuration and error correction, etc. This component may detect and react to some radio interference, however it will not analyse and counter an attack by moving to a different frequency spread in the event of jamming.

The component is expected to at least partially satisfy security enforcing functions by:

- **Protecting Integrity of Data** using techniques such as forward error correction to maintain data accuracy and completeness.
- **Securing Communications** through the application of provided security measures to achieve low probability of detection or interception, etc.

5.4.2.7.7 Services

5.4.2.7.7.1 Service Definitions

5.4.2.7.7.1.1 Requirement

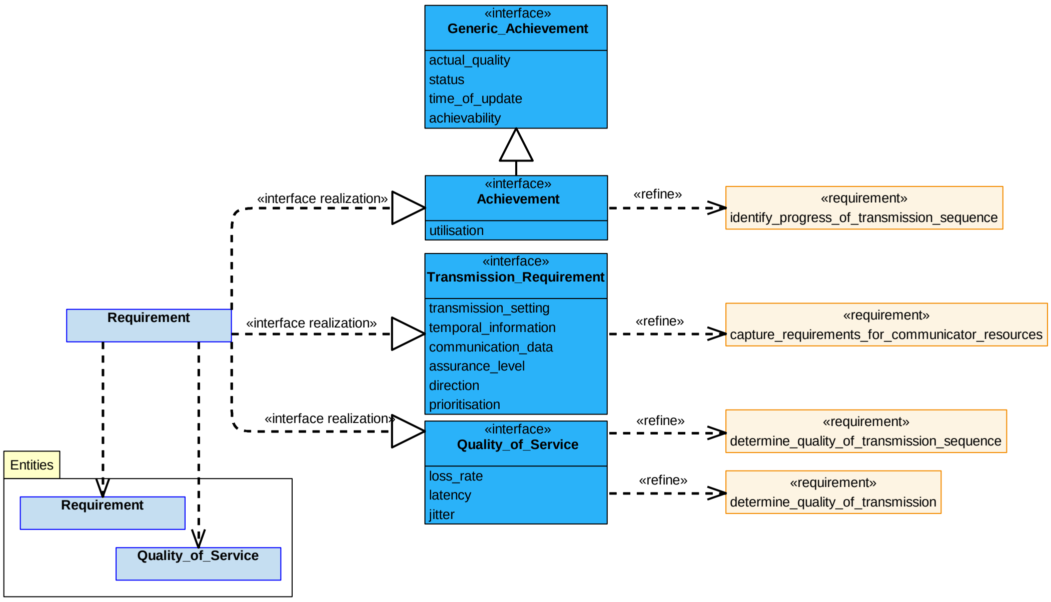


Figure 137: Requirement Service Definition

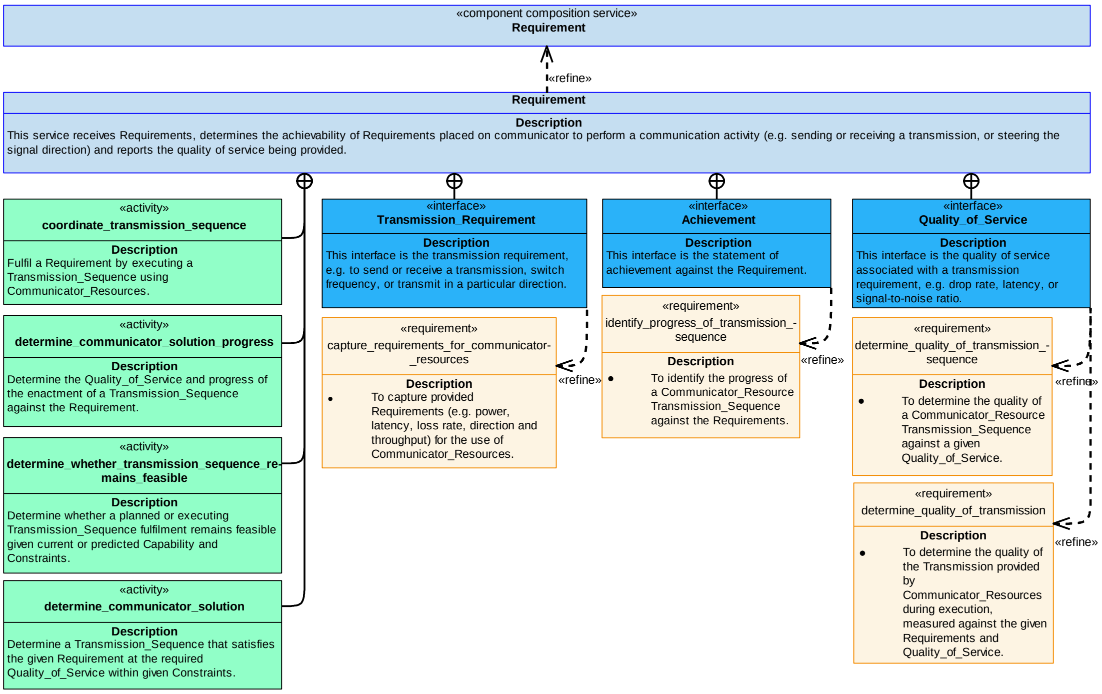

«component composition service»
Requirement
«refine»
Requirement
Description
This service receives Requirements, determines the achievability of Requirements placed on communicator to perform a communication activity (e.g. sending or receiving a transmission, or steering the signal direction) and reports the quality of service being provided.
«activity»
coordinate_transmission_sequence
Description
Fulfil a Requirement by executing a Transmission_Sequence using Communicator_Resources.
«activity»
determine_communicator_solution_progress
Description
Determine the Quality_of_Service and progress of the enactment of a Transmission_Sequence against the Requirement.
«activity»
determine_whether_transmission_sequence_remains_feasible
Description
Determine whether a planned or executing Transmission_Sequence fulfilment remains feasible given current or predicted Capability and Constraints.
«activity»
determine_communicator_solution
Description
Determine a Transmission_Sequence that satisfies the given Requirement at the required Quality_of_Service within given Constraints.
«interface»
Transmission_Requirement
Description
This interface is the transmission requirement, e.g. to send or receive a transmission, switch frequency, or transmit in a particular direction.
«requirement»
capture_requirements_for_communicator_resources
Description
To capture provided Requirements (e.g. power, latency, loss rate, direction and throughput) for the use of Communicator_Resources.
«refine»
«interface»
Achievement
Description
This interface is the statement of achievement against the Requirement.
«requirement»
identify_progress_of_transmission_sequence
Description
To identify the progress of a Communicator_Resource Transmission_Sequence against the Requirements.
«refine»
«interface»
Quality_of_Service
Description
This interface is the quality of service associated with a transmission requirement, e.g. drop rate, latency, or signal-to-noise ratio.
«requirement»
determine_quality_of_transmission_sequence
Description
To determine the quality of a Communicator_Resource Transmission_Sequence against a given Quality_of_Service.
«refine»
«requirement»
determine_quality_of_transmission
Description
To determine the quality of the Transmission provided by Communicator_Resources during execution, measured against the given Requirements and Quality_of_Service.
«refine»

Figure 138: Requirement Service Policy

Requirement

This service receives Requirements, determines the achievability of Requirements placed on communicator to perform a communication activity (e.g. sending or receiving a transmission, or steering the signal direction) and reports the quality of service being provided.

Interfaces

Achievement

This interface is the statement of achievement against the Requirement.

Attribute

utilisation	The actual level of usage of Communicator_Resources.

Transmission_Requirement

This interface is the transmission requirement, e.g. to send or receive a transmission, switch frequency, or transmit in a particular direction.

Attributes

transmission_setting	The required settings for a Transmission, e.g. a particular channel, frequency or power.
temporal_information	The time period over which a Transmission is to be made or over which receipt of a possible Transmission is allowed.
communication_data	The volume of data to be transmitted using the Communicator_Resource.
assurance_level	The level of assurance required of a Transmission, e.g. whether the communication needs a specified level of protection.

direction	The direction in which a signal needs to be steered.
prioritisation	The priority of the data to be transmitted.

Quality_of_Service

This interface is the quality of service associated with a transmission requirement, e.g. drop rate, latency, or signal-to-noise ratio.

Attributes

loss_rate	The rate of data being lost in a Transmission, e.g. packet loss, or being unable to interpret a received signal due to attenuation or signal-to-noise ratio.
latency	The level of delay of a Transmission.
jitter	The variability in delay of delivery.

Activities

determine_communicator_solution

Determine a Transmission_Sequence that satisfies the given Requirement at the required Quality_of_Service within given Constraints.

determine_communicator_solution_progress

Determine the Quality_of_Service and progress of the enactment of a Transmission_Sequence against the Requirement.

determine_whether_transmission_sequence_remains_feasible

Determine whether a planned or executing Transmission_Sequence fulfilment remains feasible given current or predicted Capability and Constraints.

coordinate_transmission_sequence

Fulfil a Requirement by executing a Transmission_Sequence using Communicator_Resources.

5.4.2.7.7.1.2 Communicator_Resourcing

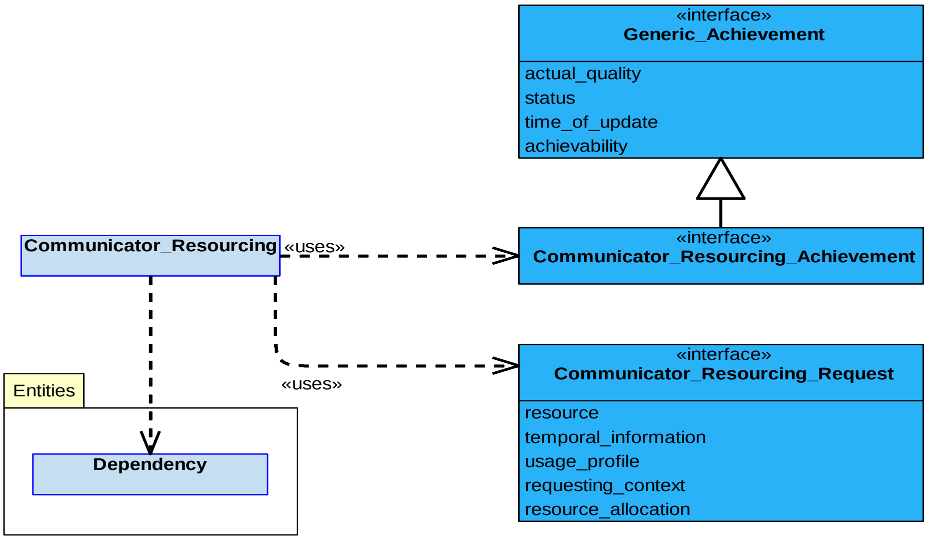
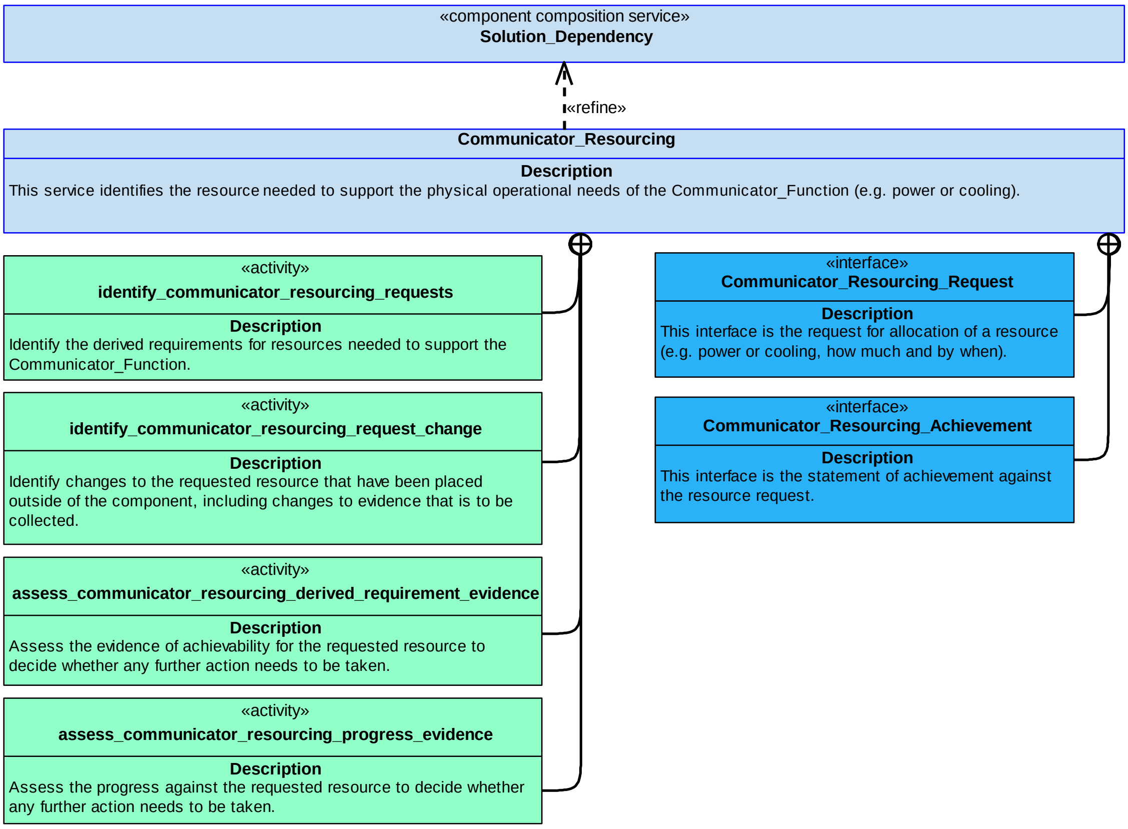


Figure 139: Communicator Resourcing Service Definition


«component composition service»
Solution_Dependency
«refine»
Communicator_Resourcing
Description
«activity»
identify_communicator_resourcing_requests
Description
«activity»
identify_communicator_resourcing_request_change
Description
«activity»
assess_communicator_resourcing_derived_requirement_evidence
Description
«activity»
assess_communicator_resourcing_progress_evidence
Description
«interface»
Communicator_Resourcing_Request
Description
«interface»
Communicator_Resourcing_Achievement
Description

This service identifies the resource needed to support the physical operational needs of the Communicator_Function (e.g. power or cooling).

Identify the derived requirements for resources needed to support the Communicator_Function.

Identify changes to the requested resource that have been placed outside of the component, including changes to evidence that is to be collected.

Assess the evidence of achievability for the requested resource to decide whether any further action needs to be taken.

Assess the progress against the requested resource to decide whether any further action needs to be taken.

This interface is the request for allocation of a resource (e.g. power or cooling, how much and by when).

This interface is the statement of achievement against the resource request.

Figure 140: Communicator Resourcing Service Policy

Communicator_Resourcing

This service identifies the resource needed to support the physical operational needs of the Communicator_Function (e.g. power or cooling).

Interfaces

Communicator_Resourcing_Request

This interface is the request for allocation of a resource (e.g. power or cooling, how much and by when).

Attributes

resource	The resource being requested (e.g. power or cooling).
temporal_information	Information covering timing for the requested resource, such as start and end times. This might include segments of a requested time window that must not be interrupted.
usage_profile	The quantity of resource requested for use, e.g. a one-off amount or a variable amount, an example being 10 kW for a specified period of time.
requesting_context	The information that identifies the source or reason for the request.
resource_allocation	The actual allocated resource quantity required to meet the usage_profile.

Communicator_Resourcing_Achievement

This interface is the statement of achievement against the resource request.

Activities

identify_communicator_resourcing_requests

Identify the derived requirements for resources needed to support the Communicator_Function.

identify_communicator_resourcing_request_change

Identify changes to the requested resource that have been placed outside of the component, including changes to evidence that is to be collected.

assess_communicator_resourcing_derived_requirement_evidence

Assess the evidence of achievability for the requested resource to decide whether any further action needs to be taken.

assess_communicator_resourcing_progress_evidence

Assess the progress against the requested resource to decide whether any further action needs to be taken.

5.4.2.7.7.1.3 Constraint

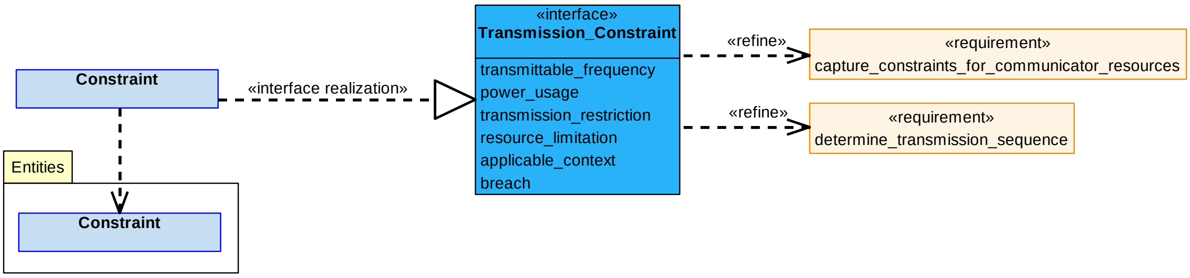


Figure 141: Constraint Service Definition

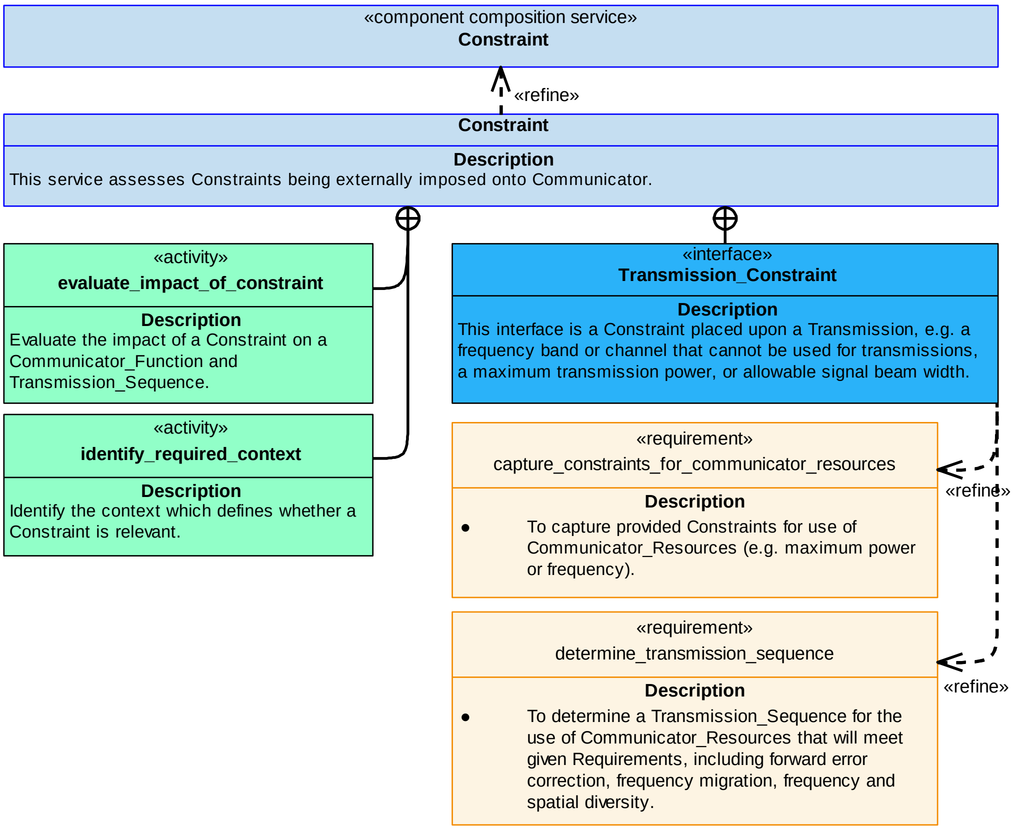

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints being externally imposed onto Communicator.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of a Constraint on a Communicator_Function and Transmission_Sequence.
«activity»
identify_required_context
Description
Identify the context which defines whether a Constraint is relevant.
«interface»
Transmission_Constraint
Description
This interface is a Constraint placed upon a Transmission, e.g. a frequency band or channel that cannot be used for transmissions, a maximum transmission power, or allowable signal beam width.
«requirement»
capture_constraints_for_communicator_resources
Description
To capture provided Constraints for use of Communicator_Resources (e.g. maximum power or frequency).
«refine»
«requirement»
determine_transmission_sequence
Description
To determine a Transmission_Sequence for the use of Communicator_Resources that will meet given Requirements, including forward error correction, frequency migration, frequency and spatial diversity.
«refine»

Figure 142: Constraint Service Policy

Constraint

This service assesses Constraints being externally imposed onto Communicator.

Interface

Transmission_Constraint

This interface is a Constraint placed upon a Transmission, e.g. a frequency band or channel that cannot be used for transmissions, a maximum transmission power, or allowable signal beam width.

Attributes

transmittable_frequency	A limit on which frequencies are permitted to be used for Transmission.
power_usage	A limit on the amount of power the Communicator_Resource can use.
transmission_restriction	A restriction on Transmissions, e.g. a limit on the radiated power, or preventing all but essential Transmission.
resource_limitation	A limitation on which Communicator_Resource can be used, e.g. only equipment on one side of the platform due to a sensor being used on the other side.
applicable_context	The context in which the Constraint is applicable.
breach	A statement that a Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of a Constraint on a Communicator_Function and Transmission_Sequence.

identify_required_context

Identify the context which defines whether a Constraint is relevant.

5.4.2.7.7.1.4 Capability

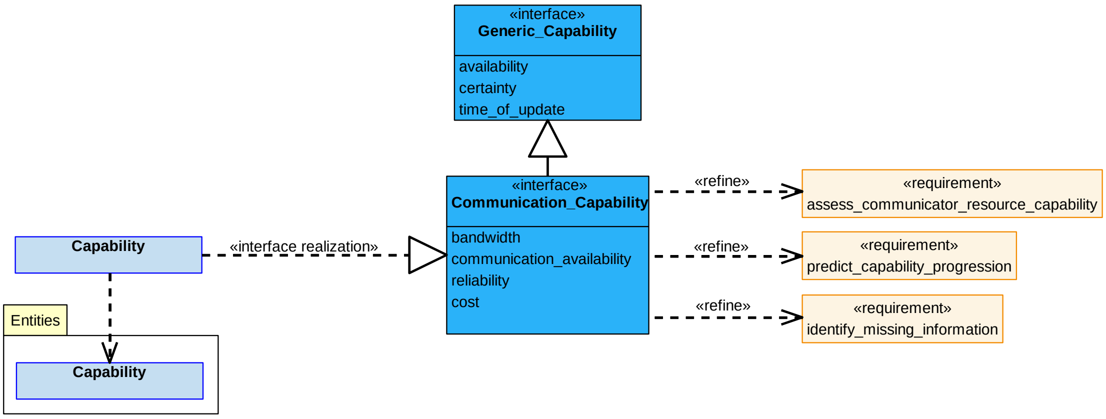
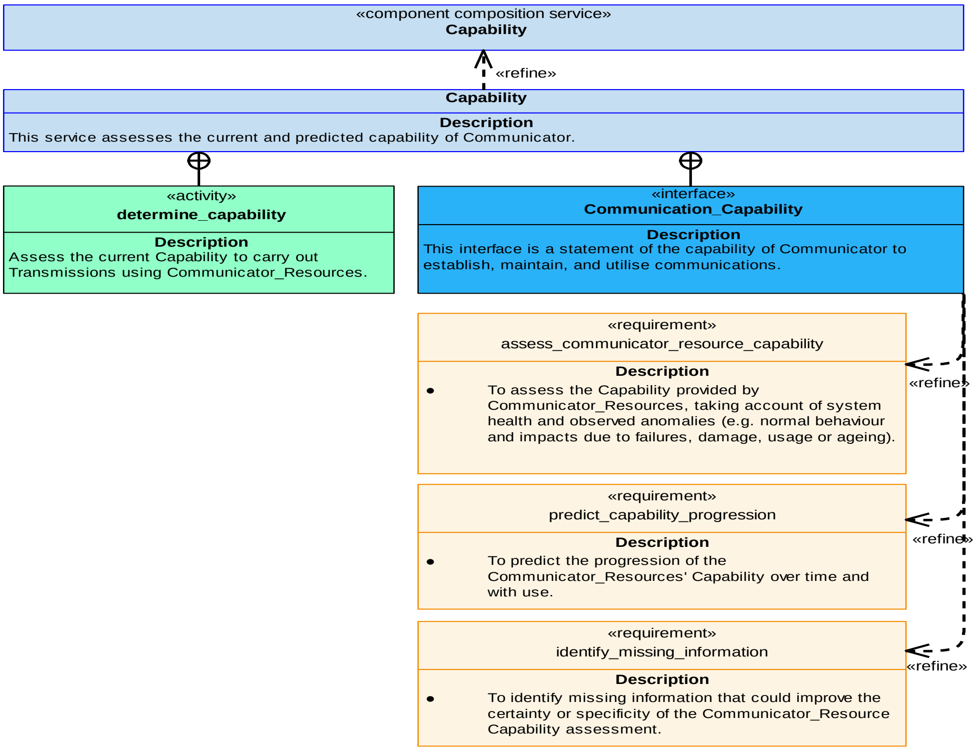


Figure 143: Capability Service Definition



This service assesses the current and predicted capability of Communicator.

Assess the current Capability to carry out Transmissions using Communicator_Resources.

This interface is a statement of the capability of Communicator to establish, maintain, and utilise communications.

- To assess the Capability provided by Communicator_Resources, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

- To predict the progression of the Communicator_Resources' Capability over time and with use.

- To identify missing information that could improve the certainty or specificity of the Communicator_Resource Capability assessment.

Figure 144: Capability Service Policy

Capability

This service assesses the current and predicted capability of Communicator.

Interface

Communication_Capability

This interface is a statement of the capability of Communicator to establish, maintain, and utilise communications.

Attributes

bandwidth	The maximum amount of traffic that can be supported.
communication_availability	The availability of specific Communicator_Resource(s) able to be used for communications.
reliability	The reliability of a communication, e.g. the likelihood for the signal to remain directed such that a Transmission is maintained.
cost	The cost of the communication capability (e.g. the power needs of a specific resource).

Activity

determine_capability

Assess the current Capability to carry out Transmissions using Communicator_Resources.

5.4.2.7.7.1.5 Capability Evidence

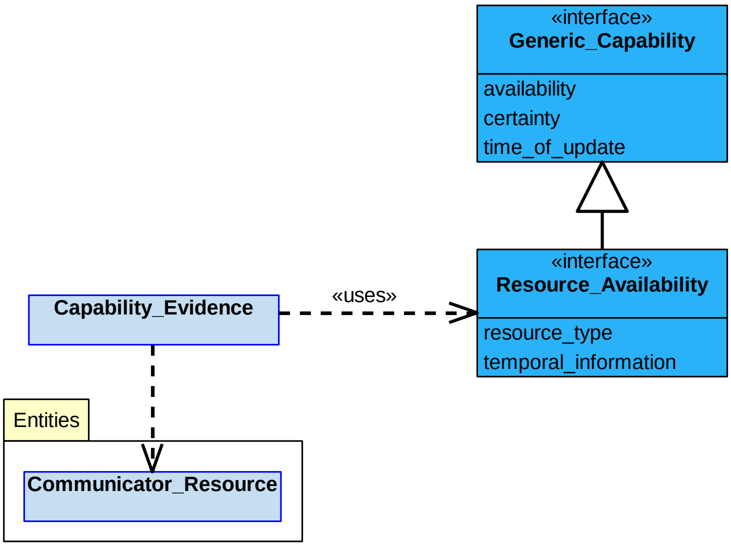


Figure 145: Capability_Evidence Service Definition

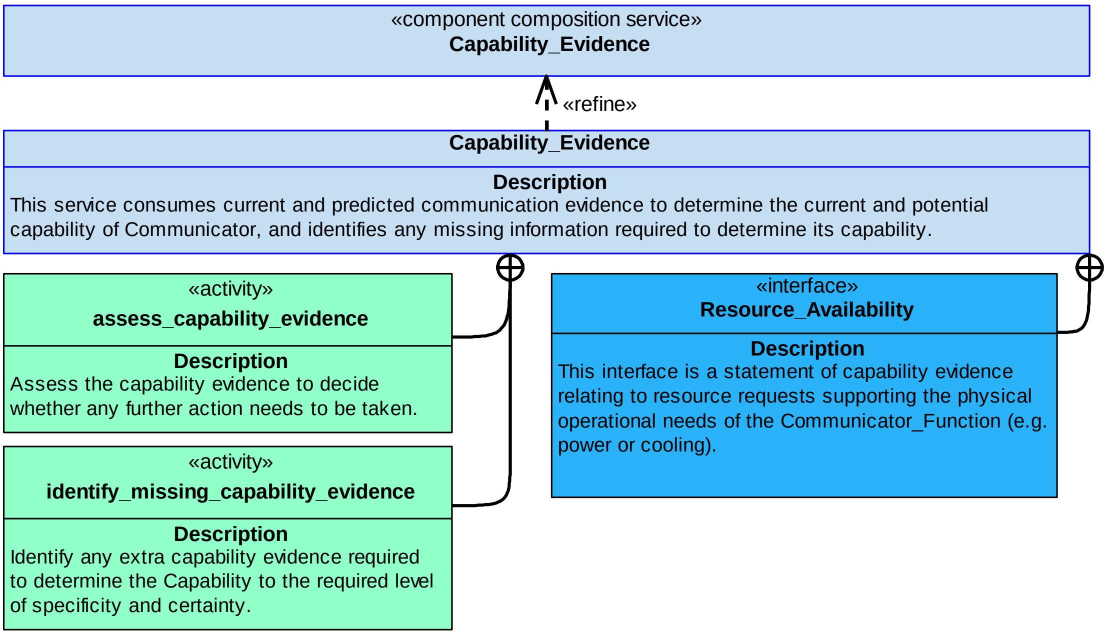

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted communication evidence to determine the current and potential capability of Communicator, and identifies any missing information required to determine its capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Resource_Availability
Description
This interface is a statement of capability evidence relating to resource requests supporting the physical operational needs of the Communicator_Function (e.g. power or cooling).

Figure 146: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted communication evidence to determine the current and potential capability of Communicator, and identifies any missing information required to determine its capability.

Interface

Resource_Availability

This interface is a statement of capability evidence relating to resource requests supporting the physical operational needs of the Communicator_Function (e.g. power or cooling).

Attributes

resource_type	The specific resource type (e.g. power or cooling) to which capability statements apply.
temporal_information	Timing information related to the availability of a resource, e.g. how long power capacity will be available for.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.7.7.2 Service Dependencies

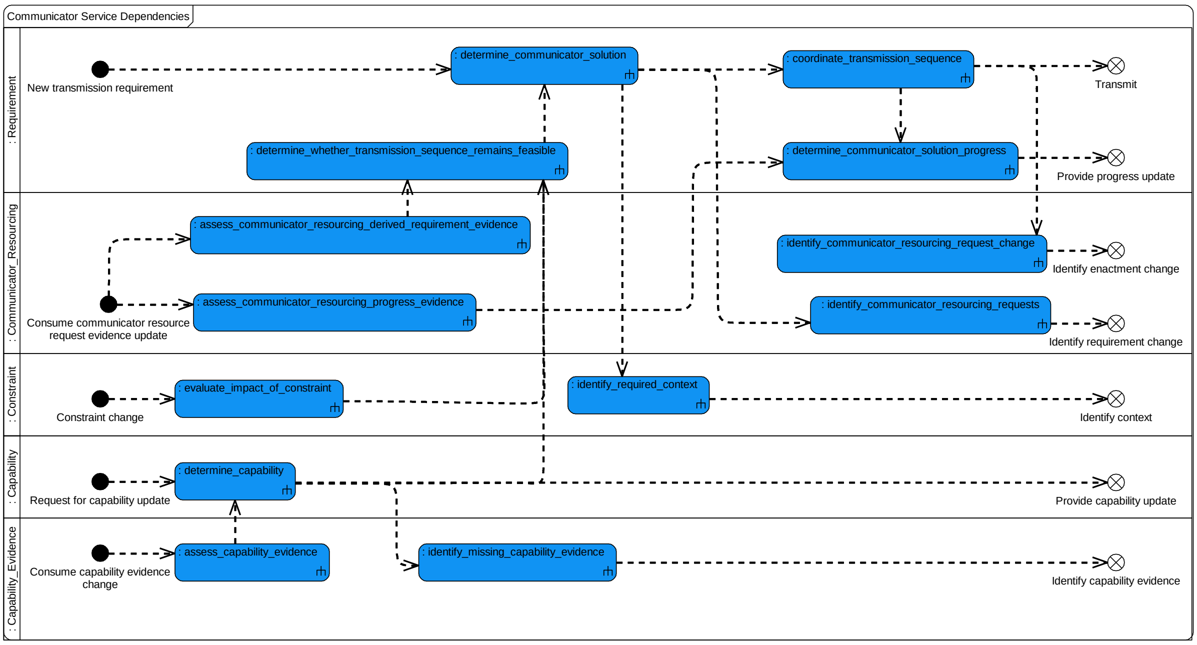

Communicator Service Dependencies
: Requirement
: Communicator_Resourcing
: Constraint
: Capability
: Capability_Evidence
New transmission requirement
: determine_communicator_solution
: coordinate_transmission_sequence
Transmit
: determine_whether_transmission_sequence_remains_feasible
: determine_communicator_solution_progress
Provide progress update
: assess_communicator_resourcing_derived_requirement_evidence
: identify_communicator_resourcing_request_change
Identify enactment change
Consume communicator resource request evidence update
: assess_communicator_resourcing_progress_evidence
: identify_communicator_resourcing_requests
Identify requirement change
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 147: Communicator Service Dependencies

5.4.2.8 Conflict Resolution

5.4.2.8.1 Role

The role of Conflict Resolution is to resolve conflicts between requirements and constraints through a process of brokering and arbitration.

5.4.2.8.2 Overview

Control Architecture

Conflict Resolution is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Conflict Resolution receives a request for resolution of a Conflict from a Recipient of conflicting Demands that cannot be resolved locally by the Recipient. Conflict Resolution determines the approach to resolving the Conflict, in accordance with the defined resolution Scheme, given the current Context and Resolution_Constraints. Conflict Resolution identifies the Originators whose Demands result in the Conflict and, in accordance with the resolution Scheme, requests that one or more Demand Originators attempt to refine their solution. Subject to successful refinement, Conflict Resolution subsequently receives notification that the Conflict is resolved.

Where a Conflict cannot be resolved through the brokering process described above, Conflict Resolution seeks arbitration from an Arbitration_Authority. According to the nature of the Conflict and the Context, Conflict Resolution itself may act as the Arbitration_Authority in accordance with the defined resolution Scheme.

Examples of Use

Conflict Resolution will be used in any system where the system may receive conflicting demands, or where conflicting demands may emerge between components in the system, for example:

- Where two Demands relating to the state change of an aircraft are conflicting, the higher level requirements forming the root cause of the conflict can be identified allowing for the arbitration of the conflicting higher level requirements and an appropriate state change to be determined.
- Where two activities generate routing requirements that result in a combined route exceeding the available fuel budget, a brokering process may be used to seek modification of the activities in a way that enables a viable route to be established.
- Where two activities result in conflicting zonal constraints (such as due to maintaining a separation distance from hostile vehicles to avoid detection) that prevent the generation of a viable route, a brokering process may be used to seek modification of the activities in a way that enables the zonal constraints to be removed or sufficiently reduced in size.
- Where a constraint on transmission, in a particular timeframe, to prevent interference with a sensing activity, makes a communications requirement unachievable, a brokering process may be used to seek refinement of the communications and sensing solutions so that both Demands can be achieved.

- Where a predefined 'high priority' activity needs to interrupt a mutually exclusive lower priority ongoing activity, Conflict Resolution may act as the Arbitration_Authority, allowing the high priority activity to occur (e.g. collision avoidance manoeuvres would take priority over the planned mission flight path).
- Where fundamentally incompatible Demands are placed on an air vehicle, such as the need to be at one place at one time and another place at a different time, that is fundamentally unachievable due to the maximum velocity capability of the vehicle. Alternatively, the Demands may not be fundamentally incompatible, but cannot both be achieved within the operating conditions, such as where the vehicle velocity is limited by the current environmental Context. In either case the relevant user may be required to act as Arbitration_Authority on which request should be satisfied.

5.4.2.8.3 Service Summary

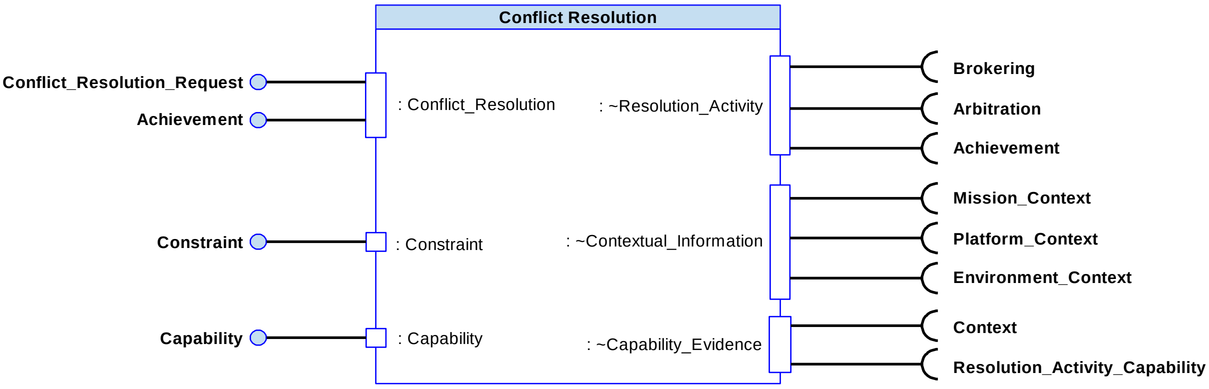


Figure 148: Conflict Resolution Service Summary

5.4.2.8.4 Responsibilities

capture_conflicts

- To capture requirements and requirement relationships for Conflict resolution.

capture_constraints

- To capture Resolution_Constraints on the brokering and arbitration of Conflicts.

determine_applicable_scheme

- To determine the applicable Scheme to apply to resolve a Conflict.

monitor_progress

- To monitor the progress of a Conflict resolution.

arbitrate_resource_conflicts

- To arbitrate on a Conflict in accordance with the applicable Scheme.

coordinate_conflict_resolution

- To coordinate the activities required to broker and arbitrate a resolution to a Conflict in accordance with the applicable Scheme and Context.

assess_capability

- To assess the Capability of the component to broker and arbitrate Conflicts.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the components capability assessment for being able to broker and arbitrate Conflicts.

5.4.2.8.5 Subject Matter Semantics

The subject matter of Conflict Resolution is the processes and rules for brokering and arbitration of Conflicts between Demands on Recipients.

Exclusions

The subject matter of Conflict Resolution does not include:

- The solutions ultimately generated by Recipients.
- The identification of conflicting demands.
- Approaches to the resolution of conflicting demands that fall entirely within the subject matter of a single component (e.g. a Recipient resolving the conflict by changing its solution to one that is able to satisfy all demands placed on it).

Note: All components have the responsibility to attempt to revise their approach to satisfying individual demands as needed so that they can satisfy all demands placed upon them. Only where this is not practical, because of a Conflicting Demand, is the subject matter of Conflict Resolution applicable to invoke strategies that result in modification to those demands or result in the most important demand(s) being identified to take precedence.

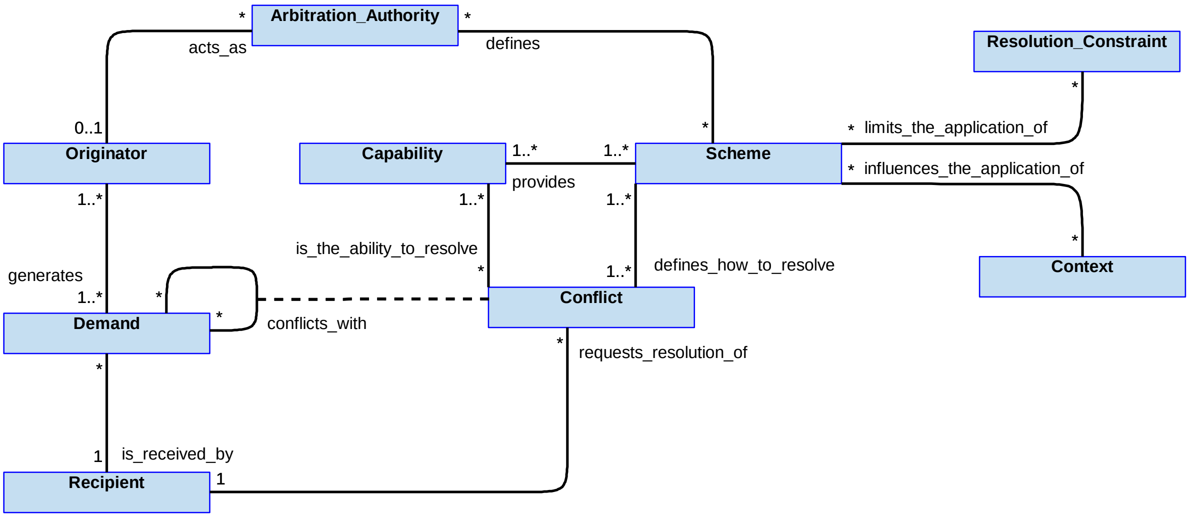


Figure 149: Conflict Resolution Semantics

5.4.2.8.5.1 Entities

Arbitration_Authority

An authority for making arbitration decisions, e.g. a common Originator of Demands, an operator, or this component.

Capability

The range of Conflicts that can be resolved (in terms of the Recipients involved, the required Arbitration_Authority and other relevant factors).

Conflict

The inability to satisfy a Demand as a result of one or more other Demands (including two or more Demands being fundamentally incompatible).

Context

Information that influences the process of brokering and arbitration. For example, the state of the mission, platform or environment.

Demand

A requirement or constraint placed on a Recipient that the Recipient is attempting to satisfy or comply with.

Originator

The source of a requirement that, directly or indirectly, results in a Demand.

Recipient

A part of the system, such as a component, which has received Demands from one or more Originators.

Resolution_Constraint

A limitation on the process or rules for brokering or arbitrating a Conflict. For example, a constraint may apply if an Originator has been compromised due to a cyber attack.

Scheme

The process and rules for brokering and arbitration including requirement relationships defining whether or not conflicting requirements occur within the same solution space and whether or not they contribute to the same fundamental source requirement.

5.4.2.8.6 Design Rationale

5.4.2.8.6.1 Assumptions

There are no assumptions.

5.4.2.8.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Conflict Resolution:

- Data Driving - data driving is a recommended approach to designing this component as the policies and strategies, which define the processes and rules for brokering and arbitration for particular Recipients, Demands, Originators or system Context, will vary with exploitation and operation.
- Component Extensions - This component may be supported by extensions that define the processes and rules for brokering and arbitration for specific Recipients, Demands, Originators or system Context. For example, Recipients that control resources that can be consumed, shared or replenished.
- Resource Management - This PYRAMID concept describes how the Conflict Resolution component is intended to be used to support Recipients which control the allocation of resources.
- Dependency Management - This PYRAMID concept discusses the wider considerations of dependency management, of which conflicting Demands on Recipients are one aspect.

Other Factors that were Taken into Account

- This component ensures that Recipients do not need to manage the processes and rules for brokering and arbitration and are only concerned with the complexities of their own subject matter.

Exploitation Considerations

- The choice between the use of (or combinations of) multiple instances of this component, extensions and data driving to address conflict resolution for different types of Originator, Recipient and Conflict will be a key choice for the Exploiter.
- The Exploiter will be required to define the mechanism by which the component determines the traceability of Demands to enable the brokering and arbitration process. To do this the component will need to have a level of understanding of planning context solution spaces and will need to be able to: (i) Determine whether or not the conflicts occur within the same solution space and whether or not they contribute to the same fundamental source requirement. (ii) Determine the originator of the fundamental source requirement(s) (e.g. the specific user).
- The component is not responsible for the full conflict resolution process, only determining the traceability of Demands and relevant Arbitration_Authority and facilitating the brokering and arbitration processes.
- The use of the component does not diminish, and should not be used to bypass, the PRA principle that problems should be resolved at the lowest ‘level’ possible. The Conflict Resolution component should be used to help facilitate this principle.

5.4.2.8.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Failure of this component could result in failure to correctly resolve a Conflict, when required, through the process of brokering or arbitration. Such a failure could lead to a critical Recipient either not being able to reach a solution, or to incorrectly prioritise Demands. This could cause the functionality dependant on the Recipient to be unavailable. For example:
  - The inability to correctly arbitrate on conflicting vehicle movement demands where one of the demands is for emergency collision avoidance.
  - Failure to immediately arbitrate in favour of power provision to a highly critical system. Whilst some high criticality systems (e.g. flight control system) may have permanent access to power this may not be the case for all safety related systems.
  - The failure to prioritise mission-critical health assessment activities over planned maintenance-related tasks.
  - The inability to place a refinement request against the Originator of a constraint that is preventing any viable solution to a requirement.
  - Depletion of resources that will be required later (such as the batteries on an air vehicle where batteries are the sole source of power, for both flight control and propulsion).

Therefore, it is assessed, conservatively, that the indicative IDAL for this component should be DAL A.

Where instances of this component contribute to hazards that are less severe, or more reliance may be placed on other barriers to an accident, then the Exploiting Platform may require a less onerous DAL.

5.4.2.8.6.4 Security Considerations

The indicative security classification is SNEO.

This component will have defined Schemes that define the policies and rules for brokering and arbitration of Conflicts. This information may, directly or indirectly, include information on system performance, system capability and mission objectives, which form the basis of prioritisation decisions. The indicative security classification is therefore considered to be SNEO. The integrity and availability of this component will affect the capability of the Exploiting Platform; if the component is prevented from performing its role, the Recipients will not be able to correctly react correctly to Conflicting Demands and may not be able to generate viable solutions at all.

The component is expected to at least partially satisfy security related functions relating to:

- **Identifying Data Sources** with regards to the demands on Recipients only being brokered and arbitrated upon if submitted by trusted Originators.

- **Maintaining Audit Records** of the Conflicts brokered and arbitrated upon for accountability purposes.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) which may therefore need to be protected to assure continued airworthiness.
- **System Status and Monitoring** through the monitoring of the system Conflict status.

This component is considered unlikely to directly implement any security enforcing function.

5.4.2.8.7 Services

5.4.2.8.7.1 Service Definitions

5.4.2.8.7.1.1 Conflict_Resolution

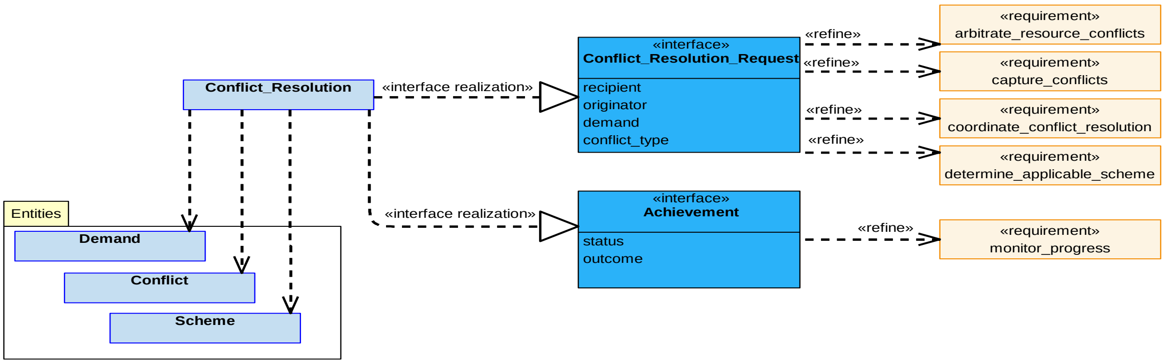


Figure 150: Conflict_Resolution Service Definition

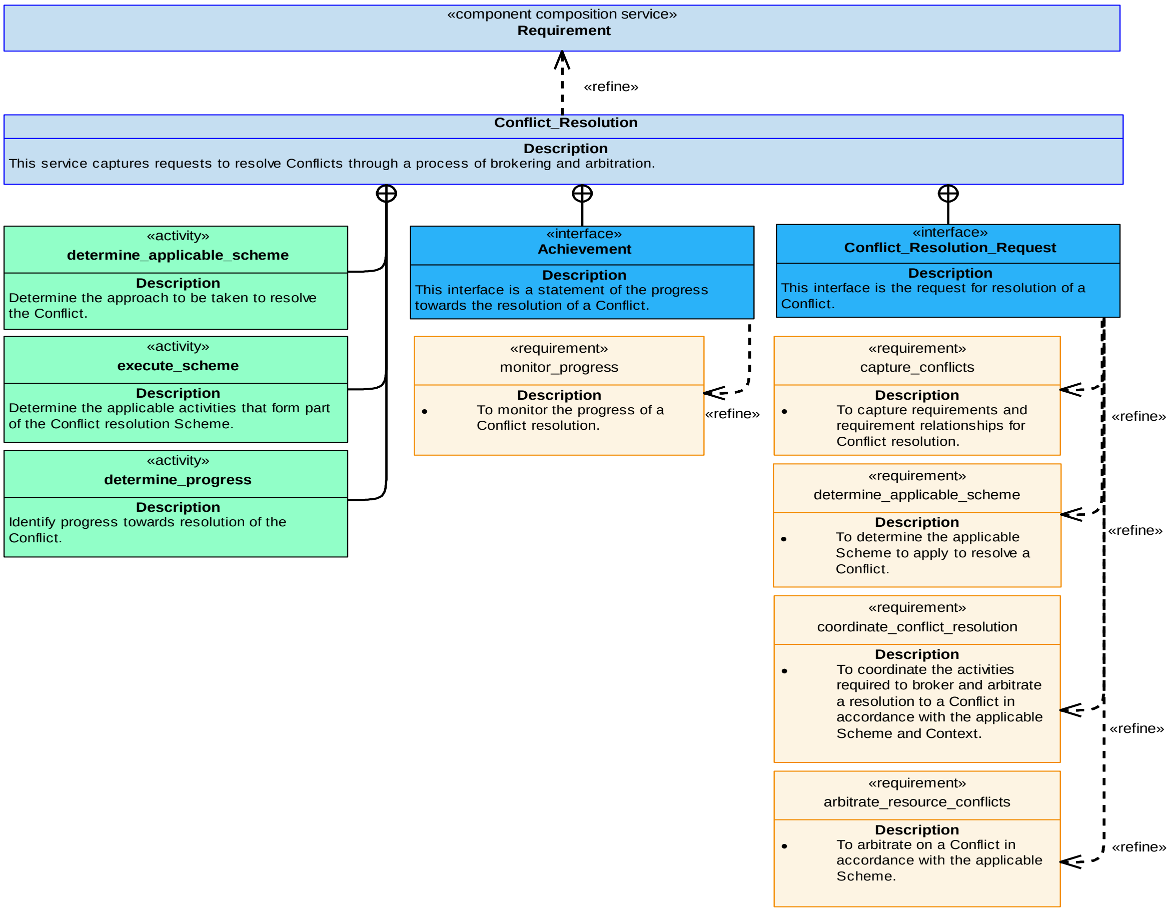

«component composition service»
Requirement
«refine»
Conflict_Resolution
Description
This service captures requests to resolve Conflicts through a process of brokering and arbitration.
«activity»
determine_applicable_scheme
Description
Determine the approach to be taken to resolve the Conflict.
«activity»
execute_scheme
Description
Determine the applicable activities that form part of the Conflict resolution Scheme.
«activity»
determine_progress
Description
Identify progress towards resolution of the Conflict.
«interface»
Achievement
Description
This interface is a statement of the progress towards the resolution of a Conflict.
«requirement»
monitor_progress
Description
To monitor the progress of a Conflict resolution.
«refine»
«interface»
Conflict_Resolution_Request
Description
This interface is the request for resolution of a Conflict.
«requirement»
capture_conflicts
Description
To capture requirements and requirement relationships for Conflict resolution.
«refine»
«requirement»
determine_applicable_scheme
Description
To determine the applicable Scheme to apply to resolve a Conflict.
«refine»
«requirement»
coordinate_conflict_resolution
Description
To coordinate the activities required to broker and arbitrate a resolution to a Conflict in accordance with the applicable Scheme and Context.
«refine»
«requirement»
arbitrate_resource_conflicts
Description
To arbitrate on a Conflict in accordance with the applicable Scheme.
«refine»

Figure 151: Conflict_Resolution Service Policy

Conflict_Resolution

This service captures requests to resolve Conflicts through a process of brokering and arbitration.

Interfaces

Conflict_Resolution_Request

This interface is the request for resolution of a Conflict.

Attributes

recipient	The Recipient that is experiencing a Conflict.
originator	The Originator of a Demand that is in Conflict.
demand	A Demand that is in Conflict.
conflict_type	The specific nature of the Conflict (e.g. two requests to use the same capability simultaneously).

Achievement

This interface is a statement of the progress towards the resolution of a Conflict.

Attributes

status	The status of a Conflict.

outcome	The result of a Conflict resolution request.

Activities

determine_applicable_scheme

Determine the approach to be taken to resolve the Conflict.

execute_scheme

Determine the applicable activities that form part of the Conflict resolution Scheme.

determine_progress

Identify progress towards resolution of the Conflict.

5.4.2.8.7.1.2 Resolution_Activity

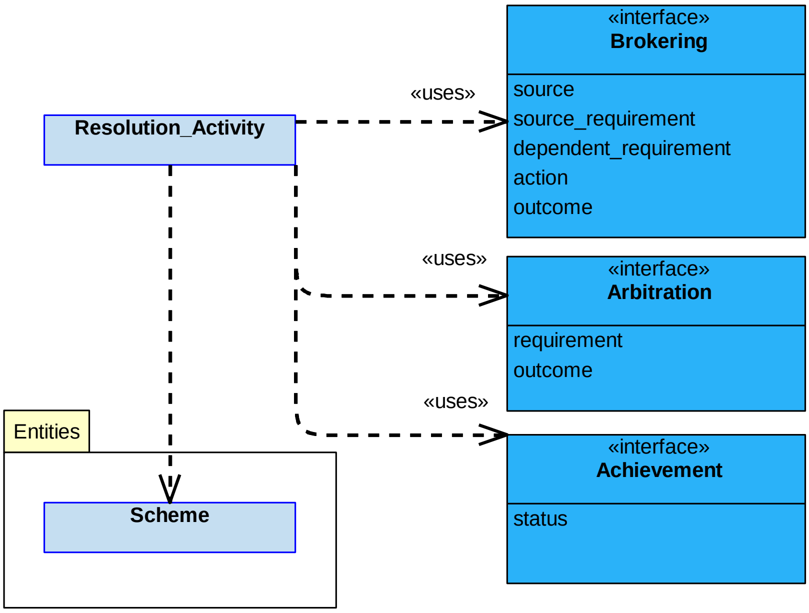


Figure 152: Resolution_Activity Service Definition

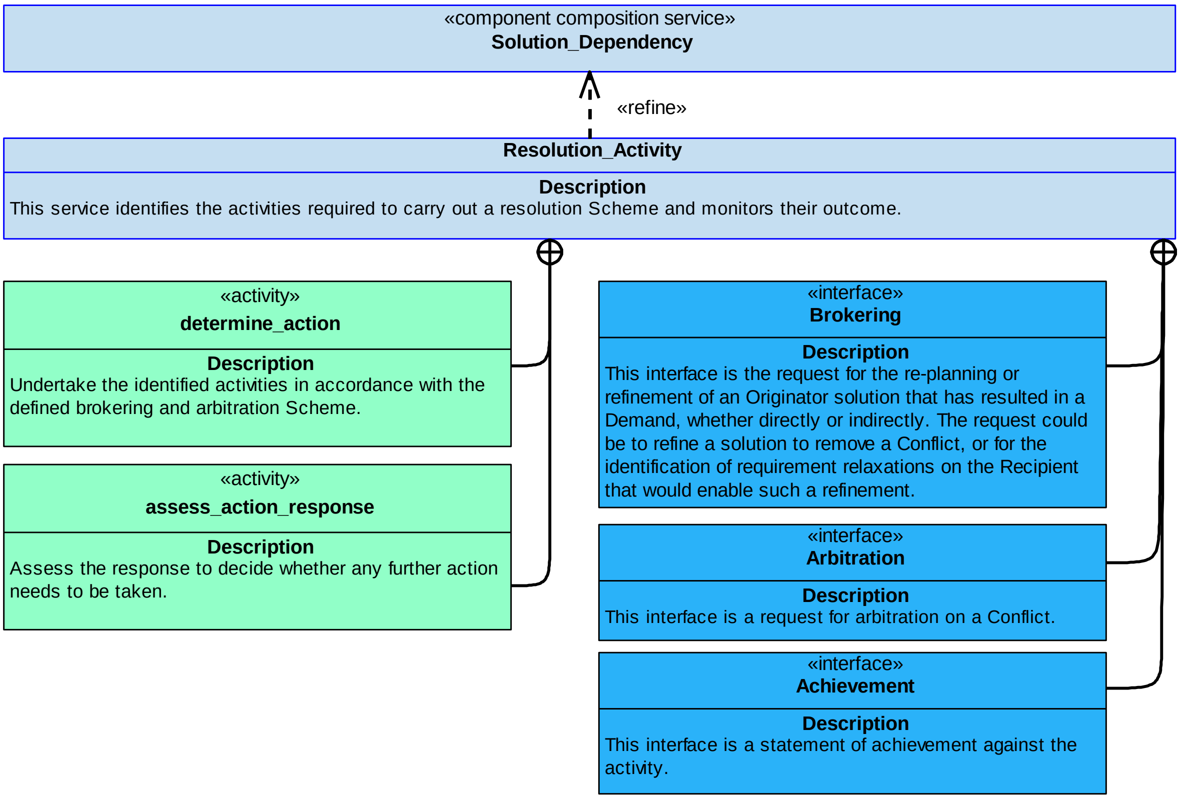

«component composition service»
Solution_Dependency
«refine»
Resolution_Activity
Description
This service identifies the activities required to carry out a resolution Scheme and monitors their outcome.
«activity»
determine_action
Description
Undertake the identified activities in accordance with the defined brokering and arbitration Scheme.
«activity»
assess_action_response
Description
Assess the response to decide whether any further action needs to be taken.
«interface»
Brokering
Description
This interface is the request for the re-planning or refinement of an Originator solution that has resulted in a Demand, whether directly or indirectly. The request could be to refine a solution to remove a Conflict, or for the identification of requirement relaxations on the Recipient that would enable such a refinement.
«interface»
Arbitration
Description
This interface is a request for arbitration on a Conflict.
«interface»
Achievement
Description
This interface is a statement of achievement against the activity.

Figure 153: Resolution_Activity Service Policy

Resolution_Activity

This service identifies the activities required to carry out a resolution Scheme and monitors their outcome.

Interfaces

Brokering

This interface is the request for the re-planning or refinement of an Originator solution that has resulted in a Demand, whether directly or indirectly. The request could be to refine a solution to remove a Conflict, or for the identification of requirement relaxations on the Recipient that would enable such a refinement.

Attributes

source	The source of the requirement or constraint, placed on the Originator, that relates to the Conflict.
source_requirement	The requirement or constraint, placed on the Originator, that relates to the Conflict.
dependent_requirement	The Demand, generated by the Originator, that relates to the Conflict.
action	The activity required to address the Conflict, e.g. to re-plan (including possible additional activities to change the Recipient's capability, such as resource replenishment), optimise or identify relaxations.
outcome	The result of an action, e.g. the solution has been refined to remove a Conflict.

Arbitration

This interface is a request for arbitration on a Conflict.

Attributes

requirement	A requirement or constraint that relates to the Conflict and for which arbitration is required.
outcome	The result of an arbitration decision.

Achievement

This interface is a statement of achievement against the activity.

Attribute

status	The status of an activity.

Activities

determine_action

Undertake the identified activities in accordance with the defined brokering and arbitration Scheme.

assess_action_response

Assess the response to decide whether any further action needs to be taken.

5.4.2.8.7.1.3 Contextual_Information

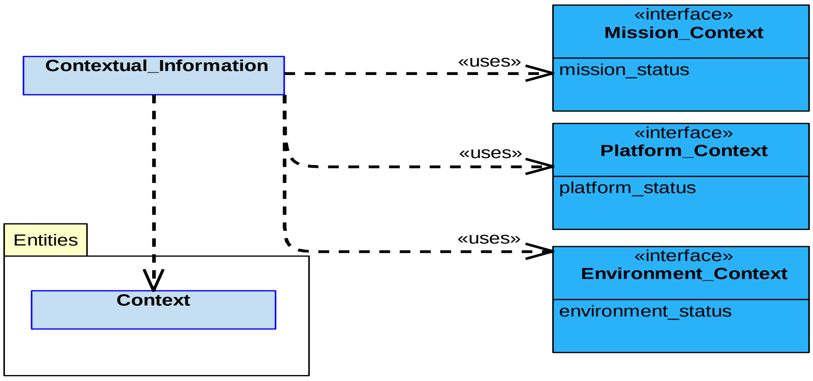


Figure 154: Contextual_Information Service Definition

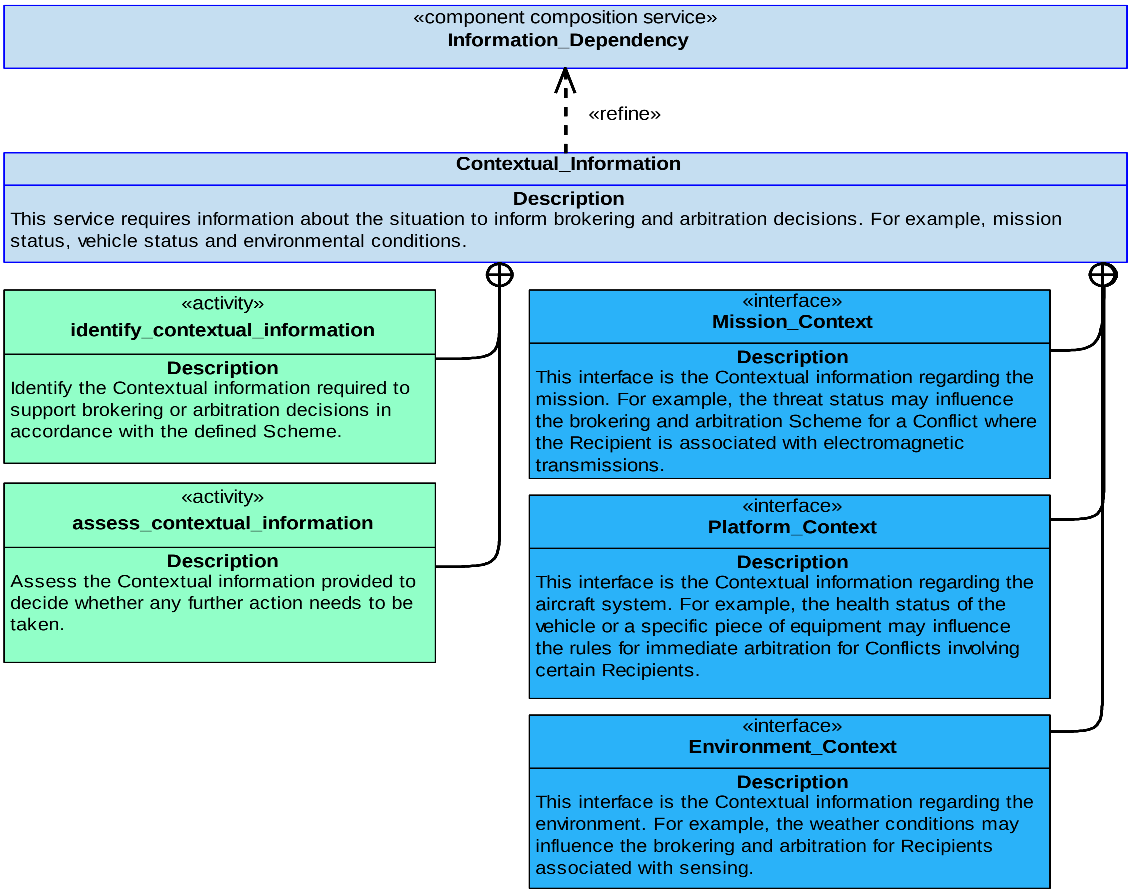

«component composition service»
Information_Dependency
«refine»
Contextual_Information
Description
This service requires information about the situation to inform brokering and arbitration decisions. For example, mission status, vehicle status and environmental conditions.
«activity»
identify_contextual_information
Description
Identify the Contextual information required to support brokering or arbitration decisions in accordance with the defined Scheme.
«activity»
assess_contextual_information
Description
Assess the Contextual information provided to decide whether any further action needs to be taken.
«interface»
Mission_Context
Description
This interface is the Contextual information regarding the mission. For example, the threat status may influence the brokering and arbitration Scheme for a Conflict where the Recipient is associated with electromagnetic transmissions.
«interface»
Platform_Context
Description
This interface is the Contextual information regarding the aircraft system. For example, the health status of the vehicle or a specific piece of equipment may influence the rules for immediate arbitration for Conflicts involving certain Recipients.
«interface»
Environment_Context
Description
This interface is the Contextual information regarding the environment. For example, the weather conditions may influence the brokering and arbitration for Recipients associated with sensing.

Figure 155: Contextual_Information Service Policy

Contextual_Information

This service requires information about the situation to inform brokering and arbitration decisions. For example, mission status, vehicle status and environmental conditions.

Interfaces

Mission_Context

This interface is the Contextual information regarding the mission. For example, the threat status may influence the brokering and arbitration Scheme for a Conflict where the Recipient is associated with electromagnetic transmissions.

Attribute

mission_status	The state of the mission.

Platform_Context

This interface is the Contextual information regarding the aircraft system. For example, the health status of the vehicle or a specific piece of equipment may influence the rules for immediate arbitration for Conflicts involving certain Recipients.

Attribute

platform_status	The state of the aircraft system.

Environment_Context

This interface is the Contextual information regarding the environment. For example, the weather conditions may influence the brokering and arbitration for Recipients associated with sensing.

Attribute

environment_status	The state of the environment.

Activities

identify_contextual_information

Identify the Contextual information required to support brokering or arbitration decisions in accordance with the defined Scheme.

assess_contextual_information

Assess the Contextual information provided to decide whether any further action needs to be taken.

5.4.2.8.7.1.4 Constraint

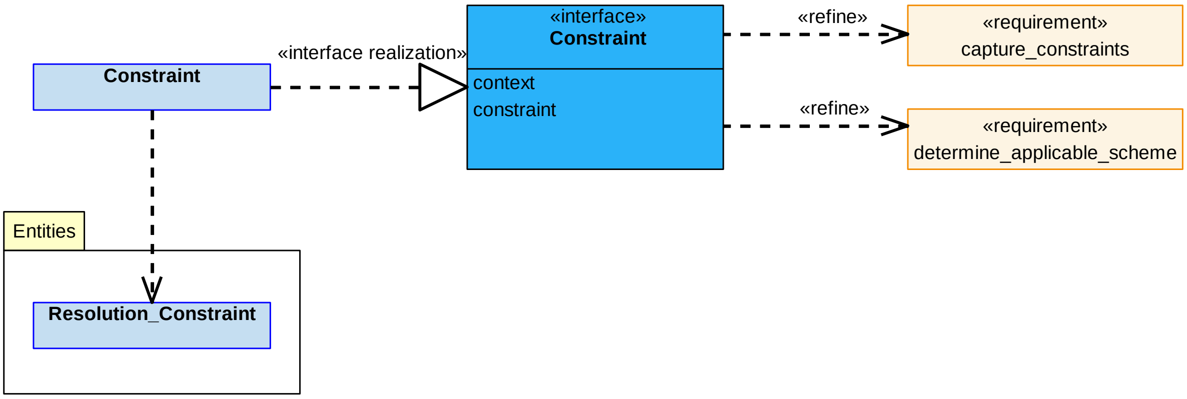


Figure 156: Constraint Service Definition

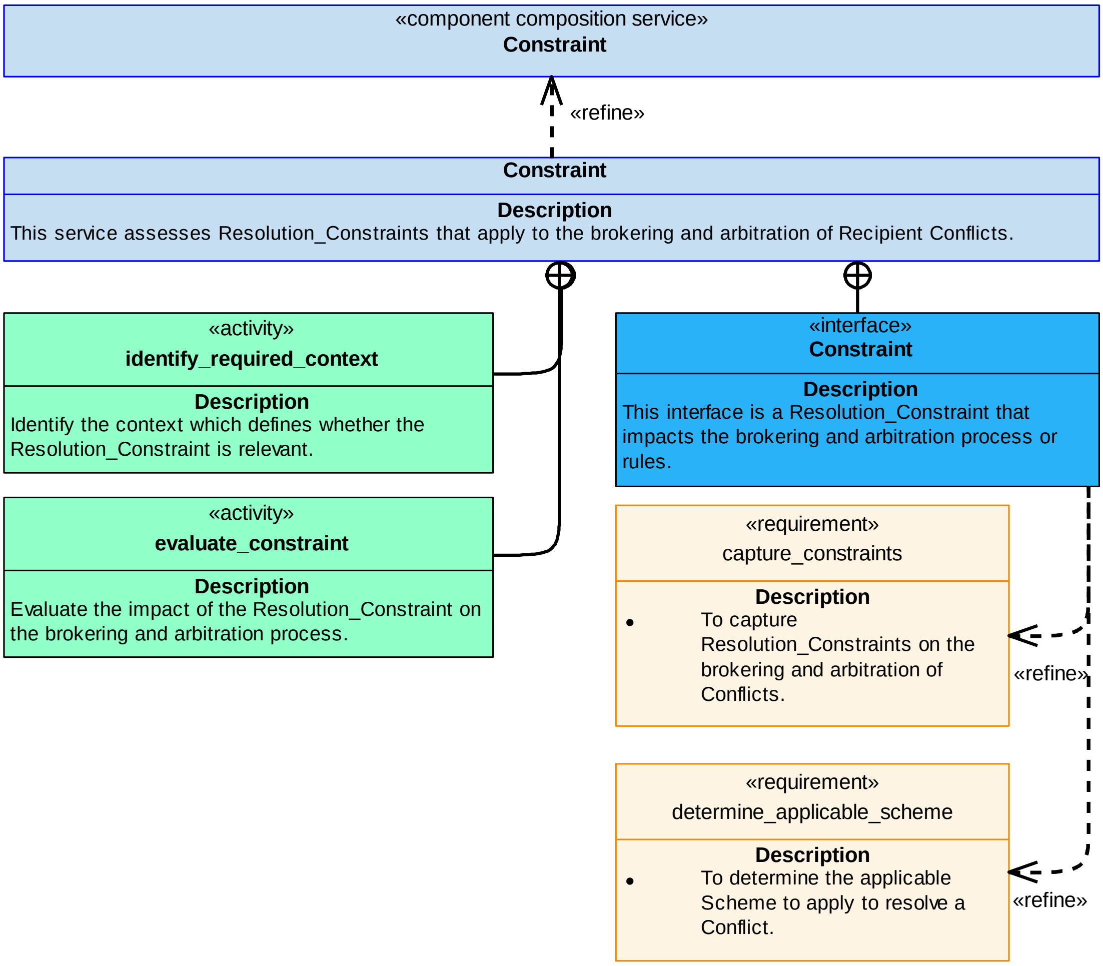

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Resolution_Constraints that apply to the brokering and arbitration of Recipient Conflicts.
«activity»
identify_required_context
Description
Identify the context which defines whether the Resolution_Constraint is relevant.
«activity»
evaluate_constraint
Description
Evaluate the impact of the Resolution_Constraint on the brokering and arbitration process.
«interface»
Constraint
Description
This interface is a Resolution_Constraint that impacts the brokering and arbitration process or rules.
«requirement»
capture_constraints
Description
To capture Resolution_Constraints on the brokering and arbitration of Conflicts.
«refine»
«requirement»
determine_applicable_scheme
Description
To determine the applicable Scheme to apply to resolve a Conflict.
«refine»

Figure 157: Constraint Service Policy

Constraint

This service assesses Resolution_Constraints that apply to the brokering and arbitration of Recipient Conflicts.

Interface

Constraint

This interface is a Resolution_Constraint that impacts the brokering and arbitration process or rules.

Attributes

context	The context in which the Resolution_Constraint is applicable.
constraint	The nature of the Resolution_Constraint that applies, e.g. to exclude an Originator from acting in the role of Arbitration_Authority.

Activities

identify_required_context

Identify the context which defines whether the Resolution_Constraint is relevant.

evaluate_constraint

Evaluate the impact of the Resolution_Constraint on the brokering and arbitration process.

5.4.2.8.7.1.5 Capability

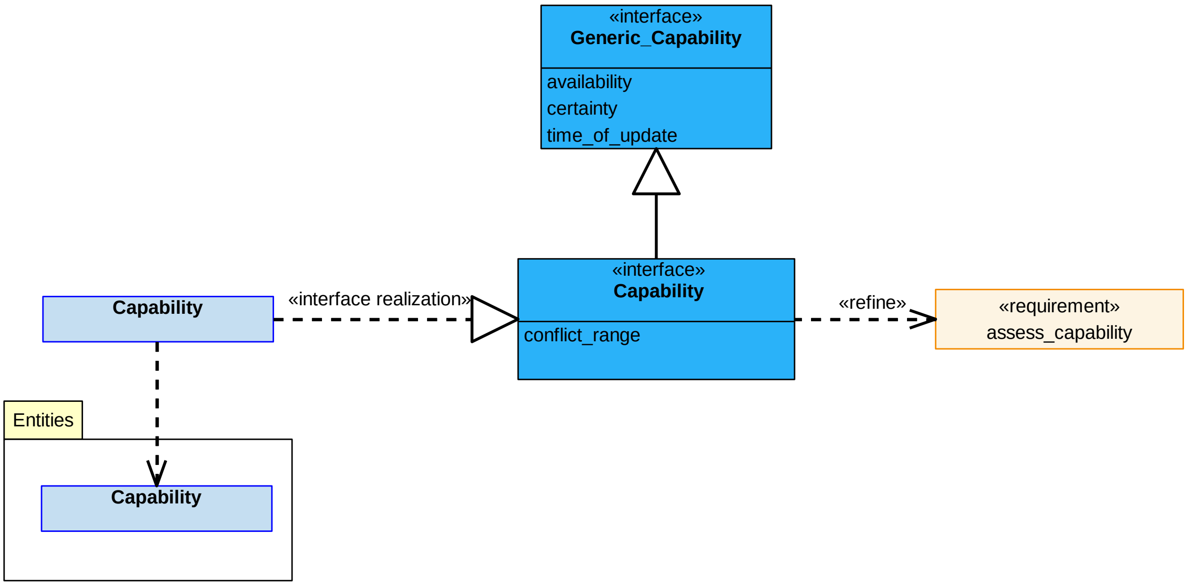


Figure 158: Capability Service Definition

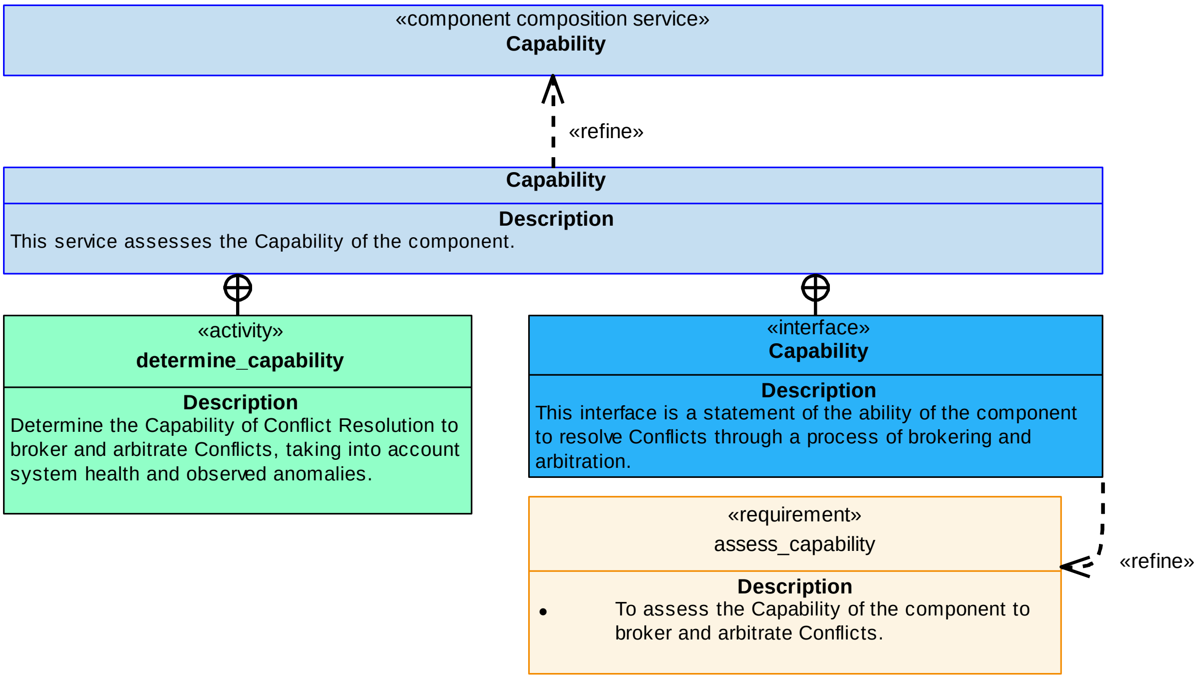

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the Capability of the component.
«activity»
determine_capability
Description
Determine the Capability of Conflict Resolution to broker and arbitrate Conflicts, taking into account system health and observed anomalies.
«interface»
Capability
Description
This interface is a statement of the ability of the component to resolve Conflicts through a process of brokering and arbitration.
«requirement»
assess_capability
Description
To assess the Capability of the component to broker and arbitrate Conflicts.
«refine»

Figure 159: Capability Service Policy

Capability

This service assesses the Capability of the component.

Interface

Capability

This interface is a statement of the ability of the component to resolve Conflicts through a process of brokering and arbitration.

Attribute

conflict_range	The range of Conflicts to which the Capability statement applies (in terms of the Recipients involved, the required Arbitration_Authority and other relevant factors).

Activity

determine_capability

Determine the Capability of Conflict Resolution to broker and arbitrate Conflicts, taking into account system health and observed anomalies.

5.4.2.8.7.1.6 Capability_Evidence

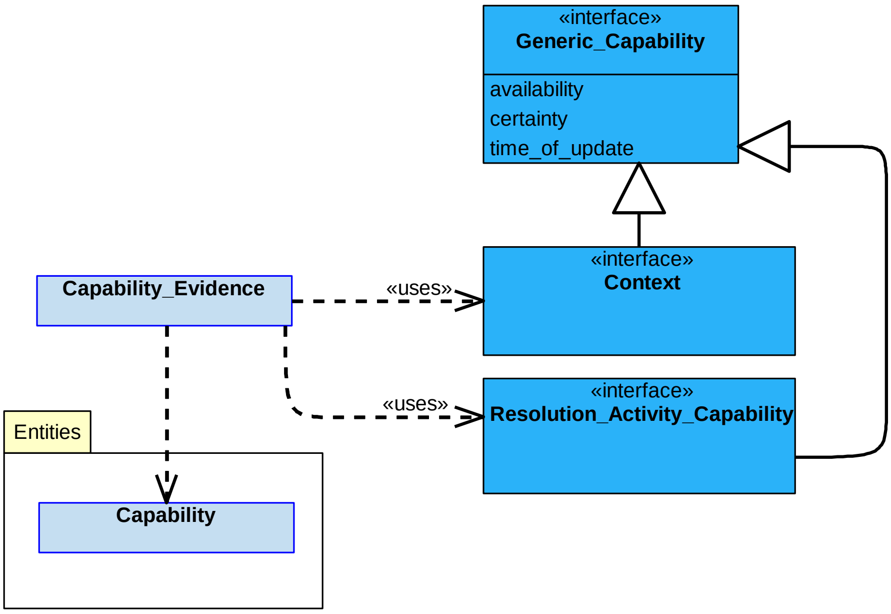


Figure 160: Capability_Evidence Service Definition

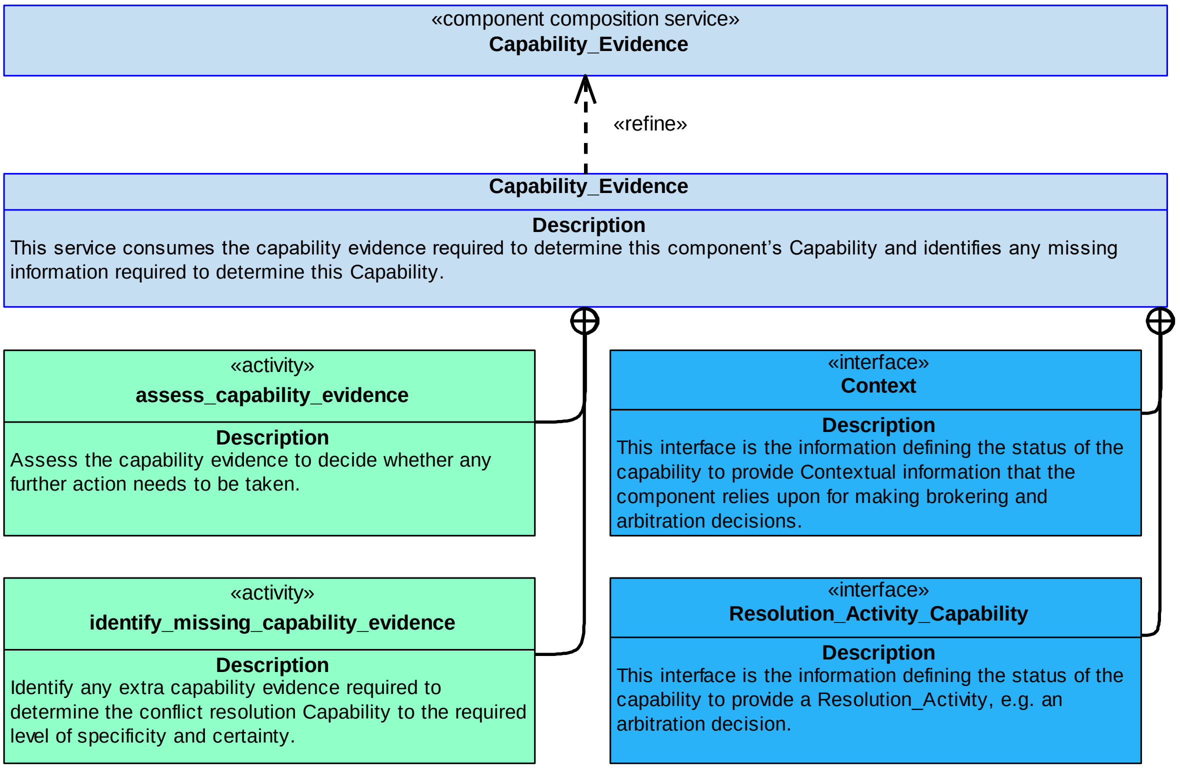

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the capability evidence required to determine this component's Capability and identifies any missing information required to determine this Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«interface»
Context
Description
This interface is the information defining the status of the capability to provide Contextual information that the component relies upon for making brokering and arbitration decisions.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the conflict resolution Capability to the required level of specificity and certainty.
«interface»
Resolution_Activity_Capability
Description
This interface is the information defining the status of the capability to provide a Resolution_Activity, e.g. an arbitration decision.

Figure 161: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the capability evidence required to determine this component's Capability and identifies any missing information required to determine this Capability.

Interfaces

Context

This interface is the information defining the status of the capability to provide Contextual information that the component relies upon for making brokering and arbitration decisions.

Resolution_Activity_Capability

This interface is the information defining the status of the capability to provide a Resolution_Activity, e.g. an arbitration decision.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the conflict resolution Capability to the required level of specificity and certainty.

5.4.2.8.7.2 Service Dependencies

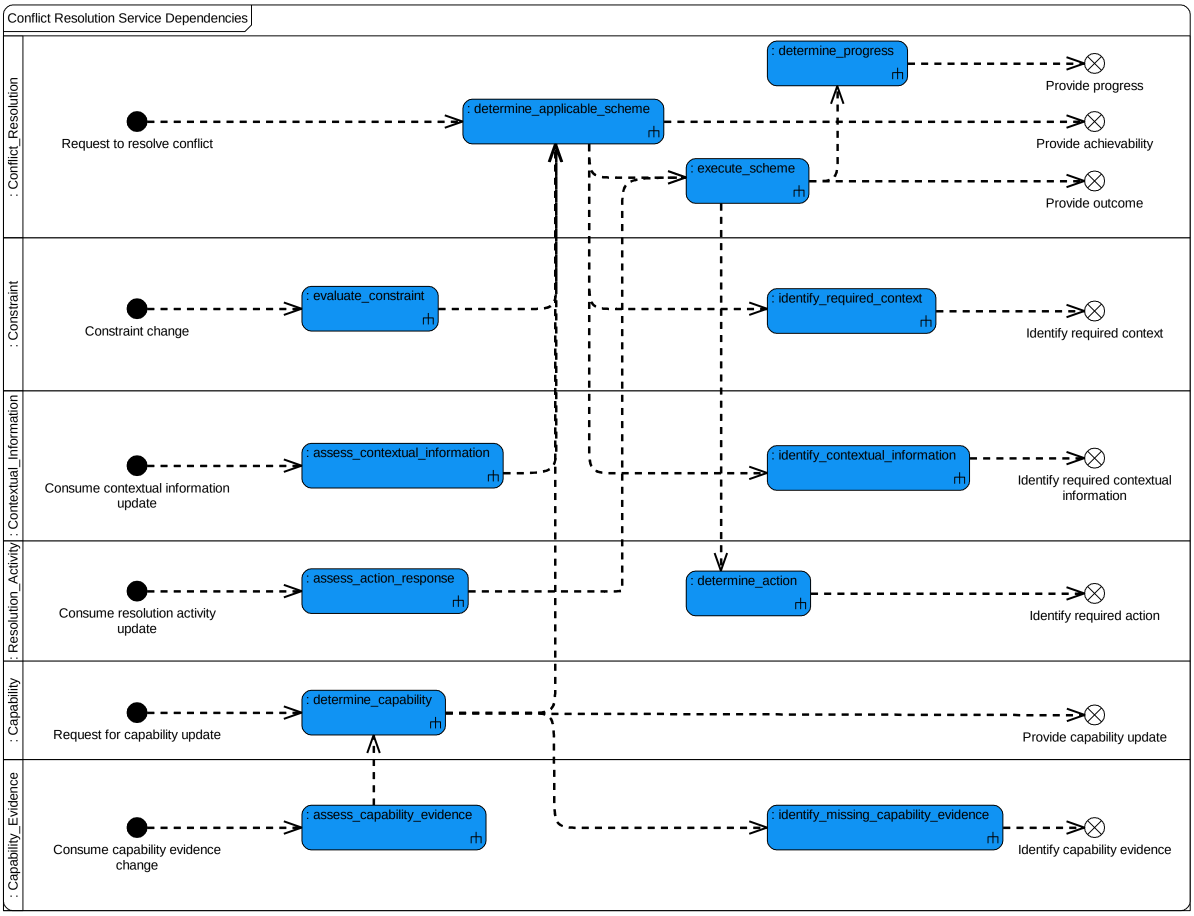

Conflict Resolution Service Dependencies
: Conflict_Resolution
: Constraint
: Contextual_Information
: Resolution_Activity
: Capability
: Capability_Evidence
Request to resolve conflict
: determine_applicable_scheme
: determine_progress
: execute_scheme
Provide progress
Provide achievability
Provide outcome
Constraint change
: evaluate_constraint
: identify_required_context
Identify required context
Consume contextual information update
: assess_contextual_information
: identify_contextual_information
Identify required contextual information
Consume resolution activity update
: assess_action_response
: determine_action
Identify required action
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 162: Conflict Resolution Service Dependencies

5.4.2.9 Countermeasures

5.4.2.9.1 Role

The role of Countermeasures is to counteract immediate threats to an air vehicle or multiple air vehicles.

5.4.2.9.2 Overview

Control Architecture

Countermeasures is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Countermeasures has been pre-authorised to enact a Countermeasure_Strategy in response to particular threats. When notified of a threat that has exceeded the threat level threshold, Countermeasures enacts its Countermeasure_Strategy to lower this threat to an acceptable level. Countermeasures enacts the Countermeasure_Strategy utilising Countermeasure_Resources, taking into account Capability and Constraints and monitors the Countermeasure_Strategy to determine the effectiveness until the Deliverable has been met.

Countermeasures can also receive a requirement to enact a Countermeasure_Strategy without the presence or detection of a threat. This will still utilise Countermeasure_Resources, taking into account Capability and Constraints, however the effectiveness of the strategy will not be monitored.

Examples of Use

The Countermeasures component can be used for:

- Deploying defensive Countermeasure_Resources (e.g. flares, defensive manoeuvres, chaff deployment, or jamming).
- Coordinating a Countermeasure_Strategy across a formation of Exploiting Platforms.

5.4.2.9.3 Service Summary

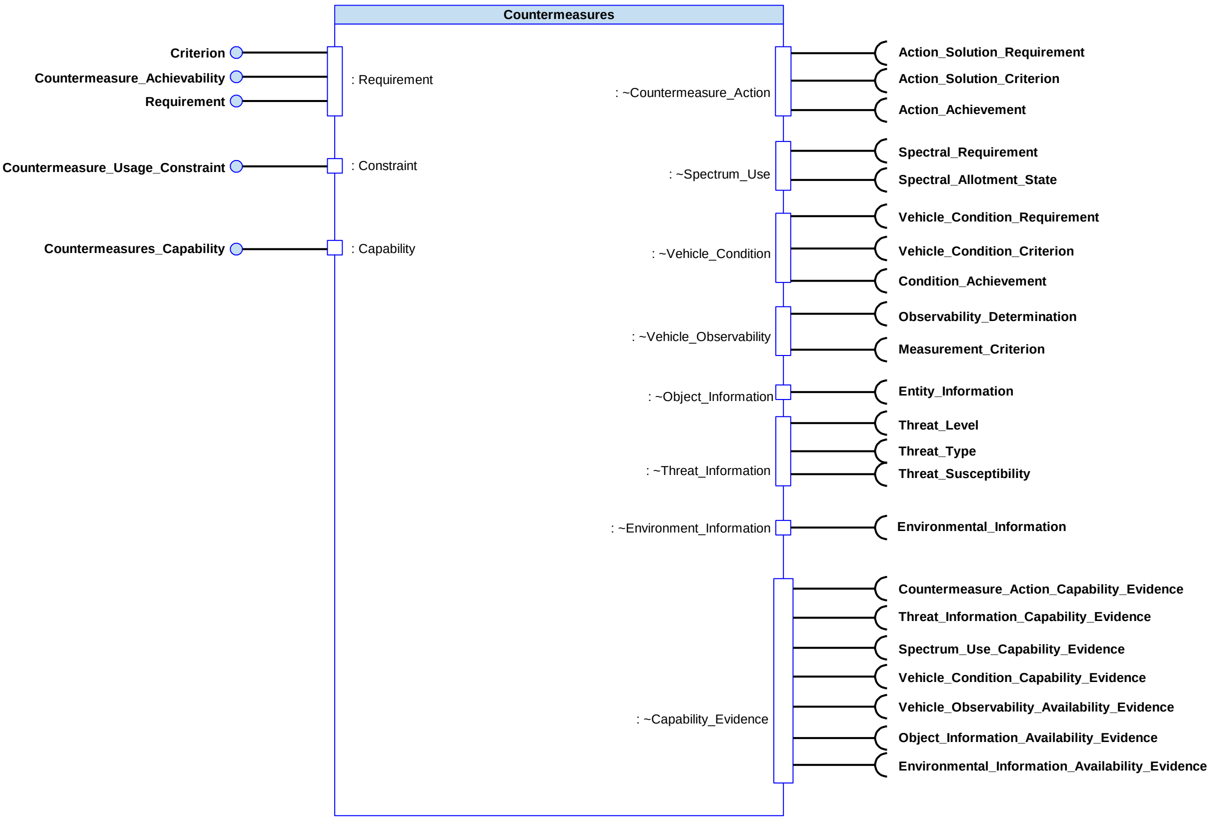

Countermeasures
Criterion
Countermeasure_Achievability
Requirement
: Requirement
Countermeasure_Usage_Constraint
: Constraint
Countermeasures_Capability
: Capability
: ~Countermeasure_Action
Action_Solution_Requirement
Action_Solution_Criterion
Action_Achievement
: ~Spectrum_Use
Spectral_Requirement
Spectral_Allotment_State
: ~Vehicle_Condition
Vehicle_Condition_Requirement
Vehicle_Condition_Criterion
Condition_Achievement
: ~Vehicle_Observability
Observability_Determination
Measurement_Criterion
: ~Object_Information
Entity_Information
: ~Threat_Information
Threat_Level
Threat_Type
Threat_Susceptibility
: ~Environment_Information
Environmental_Information
: ~Capability_Evidence
Countermeasure_Action_Capability_Evidence
Threat_Information_Capability_Evidence
Spectrum_Use_Capability_Evidence
Vehicle_Condition_Capability_Evidence
Vehicle_Observability_Availability_Evidence
Object_Information_Availability_Evidence
Environmental_Information_Availability_Evidence

Figure 163: Countermeasures Service Summary

5.4.2.9.4 Responsibilities

capture_countermeasure_requirements

- To capture given Countermeasure_Requirements (e.g. reduce the threat risk to an acceptable level).

capture_measurement_criteria

- To capture given Measurement_Criterion/criteria (e.g. reduction in threat level) for countermeasure solutions.

capture_countermeasure_constraints

- To capture given countermeasure Constraints (e.g. transmission restrictions or allowable level of response).

identify_if_countermeasure_requirement_remains_achievable

- To identify whether a Countermeasure_Requirement is still achievable given current or predicted capability and conditions.

determine_countermeasure_strategy

- To determine a Countermeasure_Strategy that meets the Countermeasure_Requirements with available Countermeasure_Resources, and within given Constraints.

determine_quality_of_countermeasure_strategy

- To determine the quality of a proposed Countermeasure_Strategy against given Measurement_Criterion/criteria.

identify_countermeasure_strategy_in_progress_remains_feasible

- To identify if a Countermeasure_Strategy in progress remains feasible given current Countermeasure_Resources, Constraints and external factors (e.g. changes in environmental conditions).

identify_pre-conditions

- To identify the Pre-conditions required to support a Countermeasure_Strategy.

co-ordinate_countermeasure_strategy

- To execute the selected Countermeasure_Strategy by commanding Countermeasure_Resources.

identify_progress_of_countermeasure_strategy

- To identify the progress of the Countermeasure_Strategy against the Countermeasure_Requirement(s) and external factors (e.g. changes in the threat level and environmental conditions).

determine_quality_of_deliverables

- To determine the quality of the Deliverables provided by a Countermeasure_Strategy, measured against given Countermeasure_Requirements and Measurement_Criterion/criteria.

assess_countermeasure_capability

- To assess the Capability to carry out a Countermeasure_Strategy or strategies using available Countermeasure_Resources, taking into account observed anomalies.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Countermeasure Capability assessment.

predict_capability_progression

- To predict the progression of countermeasure Capability over time and with use.

5.4.2.9.5 Subject Matter Semantics

The subject matter of Countermeasures is the defensive Countermeasure_Strategy or strategies to counteract immediate threats.

Exclusions

The subject matter of Countermeasures does not include:

- The authorisation of deployment and/or activation of Countermeasure_Resources.
- Conflict resolution for EMCON.

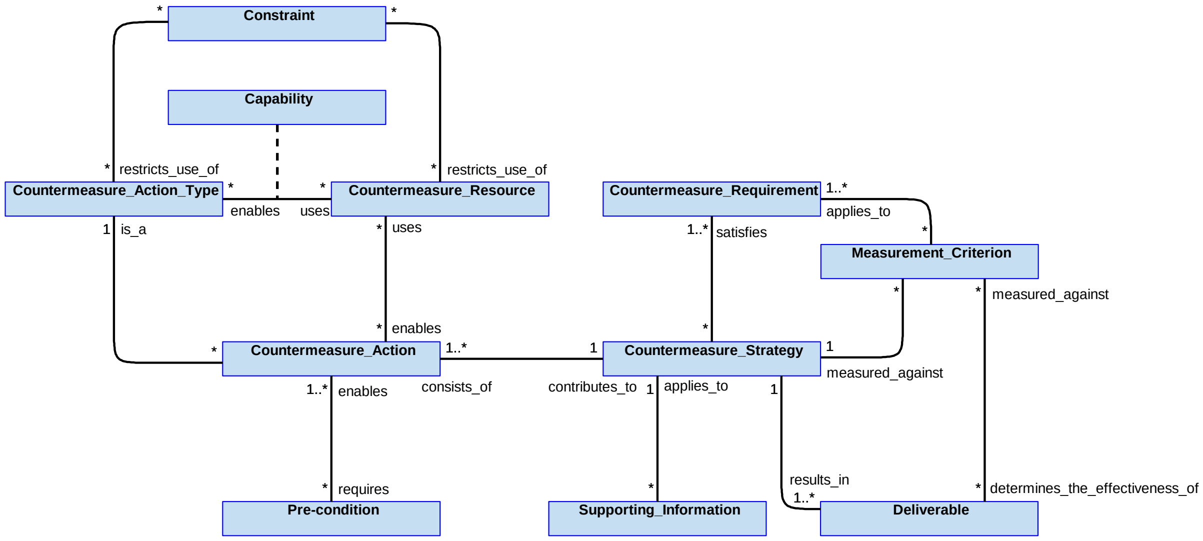


Figure 164: Countermeasures Semantics

5.4.2.9.5.1 Entities

Capability

The range of Countermeasure_Action_Types that the component is able to perform with its available Countermeasure_Resources.

Constraint

An externally imposed restriction that limits when or how a Countermeasure_Action_Type or Countermeasure_Resource can be used.

Countermeasure_Action

The activation and/or deployment of Countermeasure_Resources or other vehicle actions, e.g. manoeuvre.

Countermeasure_Action_Type

A type of Countermeasure_Action. Examples of Countermeasure_Action_Types include jamming, chaff deployment, and flare deployment.

Countermeasure_Requirement

A requirement to mitigate an identified threat.

Countermeasure_Resource

A resource that can be instructed to carry out a Countermeasure_Action_Type, e.g. a dispenser or a jammer, or other objects capable of carrying out the necessary actions to perform a manoeuvre.

Countermeasure_Strategy

A sequence of Countermeasure_Actions that will satisfy one or more Countermeasure_Requirements, this includes countermeasure strategies across formations of Exploiting Platforms.

Deliverable

An outcome (e.g. a lowered threat level) that results from executing the planned Countermeasure_Strategy.

Measurement_Criterion

A criterion that the quality of a Countermeasure_Strategy and its Deliverable will be measured against; e.g. jamming effectiveness.

Pre-condition

Items that must be true before a Countermeasure_Action can take place, e.g. authorisation.

Supporting_Information

Information that supports the strategic use of countermeasures.

5.4.2.9.6 Design Rationale

5.4.2.9.6.1 Assumptions

- The types of countermeasure (e.g. manoeuvre or release of chaff) will be updated rarely (e.g. would be common to multiple variants).
- The types of threatening element (the external entities that are threatening to the air vehicle) will be updated regularly.
- The Countermeasure_Actions used (e.g. pull this manoeuvre then release chaff) to mitigate particular threat types may change between missions and Exploiting Platforms.

5.4.2.9.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Countermeasures:

- Multi-Vehicle Coordination - This PYRAMID concept is applicable in scenarios where defensive actions would need co-ordination between vehicles and/or platforms.
- Data Driving - This PYRAMID concept is applicable to cope with the change in countermeasure types and techniques used against particular threatening elements, with the countermeasures defined as build time data. The data would also need to be defined for runtime. This allows the component to be reusable between multiple Exploiting Programmes and maintainable as behaviours change and resources are replaced.
- Recording and Logging - This PYRAMID concept is applicable to cover logging of data relating to authorisations and release actions including for audit and non-repudiation purposes.

Extensions

- The responsibility determine_countermeasure_strategy could be developed as an extension component to capture different a Countermeasure_Strategy or strategies to ensure the component is flexible.
- The responsibility determine_quality_of_countermeasure_strategy could be developed as an extension component to capture different Measurement_Criterion/criteria by which to measure the effectiveness of a Countermeasure_Strategy. The Countermeasures component will likely provide a default effectiveness model, however in many cases it will be appropriate to implement alternative effectiveness models as component extensions. This will facilitate the flexibility to develop and use different models in different contexts and allows for model development evolution and competition over time without affecting the parent Countermeasures component.

Exploitation Considerations

- There could be a single instance or multiple instances of Countermeasures for multiple vehicles (in accordance with Multi-Vehicle Coordination).
- There could be a single instance or multiple instances of Countermeasures for activating and/or deploying different Countermeasure_Resources; e.g. one instance for flare deployment and another for jamming activation, although this is left as an exploitation decision.

5.4.2.9.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

Failure of this component could result in:

- Selection of countermeasures (e.g. the release of expendables, the use of ECM, or a vehicle manoeuvre) that may cause damage to the air vehicle (e.g. if the air vehicle is not within the safe envelope for release of expendables) or harm to people (ground crew or third parties). However, it is expected that other components (e.g. Interlocks and Authorisation) will be relied upon to prevent countermeasures being enacted when not safe, independently of this component. Therefore, DAL C is appropriate for this component as the likelihood of any catastrophic accidents is reduced to an acceptable level by other components.
- Failure to defeat a threat due to either selecting an ineffective countermeasure or not selecting any countermeasure. Whilst this could result in loss of the air vehicle or crew fatality, failure to defeat external physical threats are not considered within the scope of safety analysis.

5.4.2.9.6.4 Security Considerations

The indicative security classification is SNEO.

This component is responsible for the determination of countermeasures to be applied against threats to the Exploiting Platform, knowledge of the capabilities of the countermeasures, countermeasure strategies and performance are considered SNEO, with the possibility of TS for certain electronic warfare applications. Where there are instances in different security domains, it is likely these will need to communicate with each other in order to coordinate the countermeasures to be deployed. Any separation will be performed by a boundary protection function outside these components. The integrity and availability of both input and output data for the countermeasures functions will need protection in order to ensure the correct countermeasure is deployed when (and only when) required.

The component is expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** to ensure only permitted and trustable sources will trigger requirements for countermeasures to be deployed.
- **Logging of Security Data** relating to authorisation successes and failures.
- **Maintaining Audit Records** of authorisations and decisions to deploy and actual countermeasures deployed during a mission. This is a key audit point.

The component is considered unlikely to directly implement security enforcing functions, although it is dependent on the integrity of threat information that indicates countermeasures are required.

5.4.2.9.7 Services

5.4.2.9.7.1 Service Definitions

5.4.2.9.7.1.1 Requirement

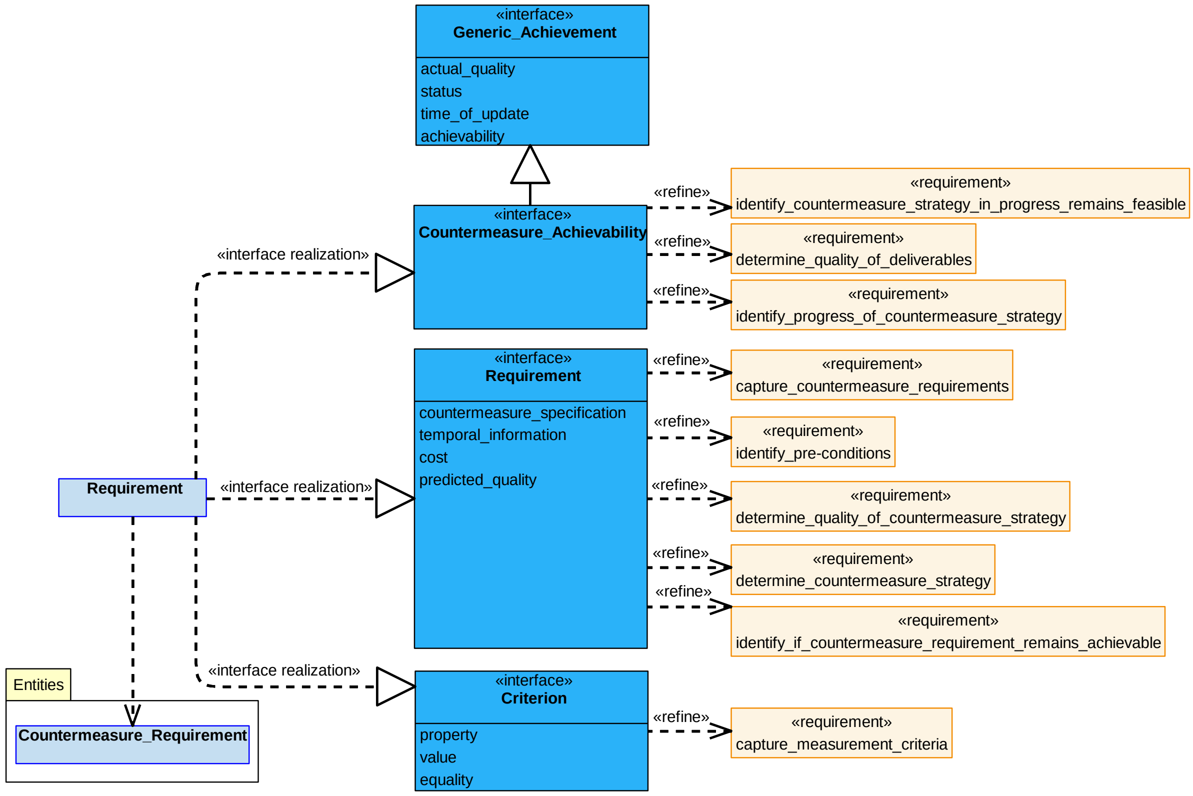


Figure 165: Requirement Service Definition

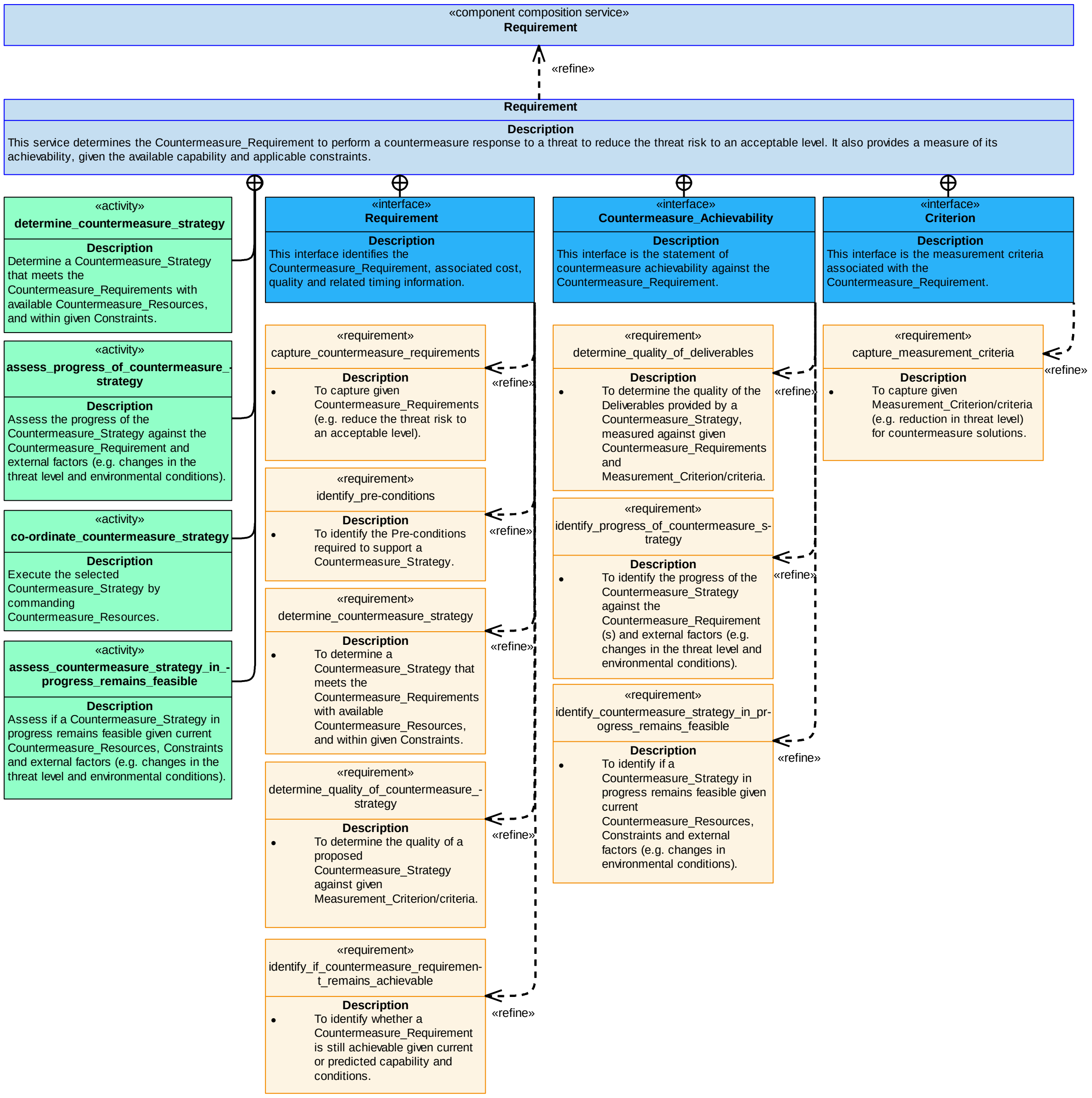

«component composition service»
Requirement
«refine»
Requirement
Description
This service determines the Countermeasure_Requirement to perform a countermeasure response to a threat to reduce the threat risk to an acceptable level. It also provides a measure of its achievability, given the available capability and applicable constraints.
«activity»
determine_countermeasure_strategy
Description
Determine a Countermeasure_Strategy that meets the Countermeasure_Requirements with available Countermeasure_Resources, and within given Constraints.
«activity»
assess_progress_of_countermeasure_strategy
Description
Assess the progress of the Countermeasure_Strategy against the Countermeasure_Requirement and external factors (e.g. changes in the threat level and environmental conditions).
«activity»
co-ordinate_countermeasure_strategy
Description
Execute the selected Countermeasure_Strategy by commanding Countermeasure_Resources.
«activity»
assess_countermeasure_strategy_in_progress_remains_feasible
Description
Assess if a Countermeasure_Strategy in progress remains feasible given current Countermeasure_Resources, Constraints and external factors (e.g. changes in the threat level and environmental conditions).
«interface»
Requirement
Description
This interface identifies the Countermeasure_Requirement, associated cost, quality and related timing information.
«interface»
Countermeasure_Achievability
Description
This interface is the statement of countermeasure achievability against the Countermeasure_Requirement.
«interface»
Criterion
Description
This interface is the measurement criteria associated with the Countermeasure_Requirement.
«requirement»
capture_countermeasure_requirements
Description
To capture given Countermeasure_Requirements (e.g. reduce the threat risk to an acceptable level).
«refine»
«requirement»
identify_pre-conditions
Description
To identify the Pre-conditions required to support a Countermeasure_Strategy.
«refine»
«requirement»
determine_countermeasure_strategy
Description
To determine a Countermeasure_Strategy that meets the Countermeasure_Requirements with available Countermeasure_Resources, and within given Constraints.
«refine»
«requirement»
determine_quality_of_countermeasure_strategy
Description
To determine the quality of a proposed Countermeasure_Strategy against given Measurement_Criterion/criteria.
«refine»
«requirement»
identify_if_countermeasure_requirement_remains_achievable
Description
To identify whether a Countermeasure_Requirement is still achievable given current or predicted capability and conditions.
«refine»
«requirement»
determine_quality_of_deliverables
Description
To determine the quality of the Deliverables provided by a Countermeasure_Strategy, measured against given Countermeasure_Requirements and Measurement_Criterion/criteria.
«refine»
«requirement»
identify_progress_of_countermeasure_strategy
Description
To identify the progress of the Countermeasure_Strategy against the Countermeasure_Requirement (s) and external factors (e.g. changes in the threat level and environmental conditions).
«refine»
«requirement»
identify_countermeasure_strategy_in_progress_remains_feasible
Description
To identify if a Countermeasure_Strategy in progress remains feasible given current Countermeasure_Resources, Constraints and external factors (e.g. changes in environmental conditions).
«refine»
«requirement»
capture_measurement_criteria
Description
To capture given Measurement_Criterion/criteria (e.g. reduction in threat level) for countermeasure solutions.
«refine»

Figure 166: Requirement Service Policy

Requirement

This service determines the Countermeasure_Requirement to perform a countermeasure response to a threat to reduce the threat risk to an acceptable level. It also provides a measure of its achievability, given the available capability and applicable constraints.

Interfaces

Criterion

This interface is the measurement criteria associated with the Countermeasure_Requirement.

Attributes

property	The criterion property to be measured, e.g. breaking lock from an identified threat.

value	The amount related to the property to be measured, e.g. emitter lock has been broken.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Requirement

This interface identifies the Countermeasure_Requirement, associated cost, quality and related timing information.

Attributes

countermeasure_specification	A requirement to mitigate an identified threat by constructing a Countermeasure_Strategy as a sequence of Countermeasure_Actions.
temporal_information	Information covering timing, such as start and end times for when a Countermeasure_Strategy is to be executed.
cost	The cost of executing the Countermeasure_Strategy as a result of using specific assets, equipment or resources (e.g. the resources depleted, power used, or time taken).
predicted_quality	How well the planned Countermeasure_Strategy is predicted to satisfy the Countermeasure_Requirement.

Countermeasure_Achievability

This interface is the statement of countermeasure achievability against the Countermeasure_Requirement.

Activities

determine_countermeasure_strategy

Determine a Countermeasure_Strategy that meets the Countermeasure_Requirements with available Countermeasure_Resources, and within given Constraints.

assess_progress_of_countermeasure_strategy

Assess the progress of the Countermeasure_Strategy against the Countermeasure_Requirement and external factors (e.g. changes in the threat level and environmental conditions).

co-ordinate_countermeasure_strategy

Execute the selected Countermeasure_Strategy by commanding Countermeasure_Resources.

assess_countermeasure_strategy_in_progress_remains_feasible

Assess if a Countermeasure_Strategy in progress remains feasible given current Countermeasure_Resources, Constraints and external factors (e.g. changes in the threat level and environmental conditions).

5.4.2.9.7.1.2 Countermeasure_Action

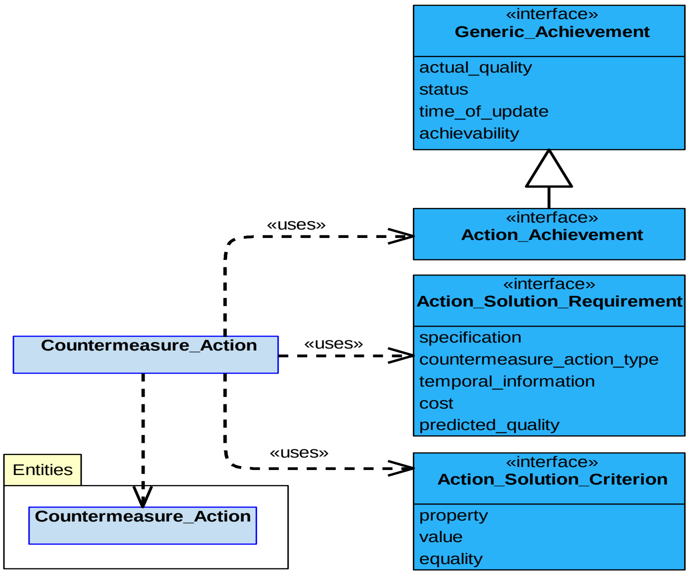


Figure 167: Countermeasure Service Definition

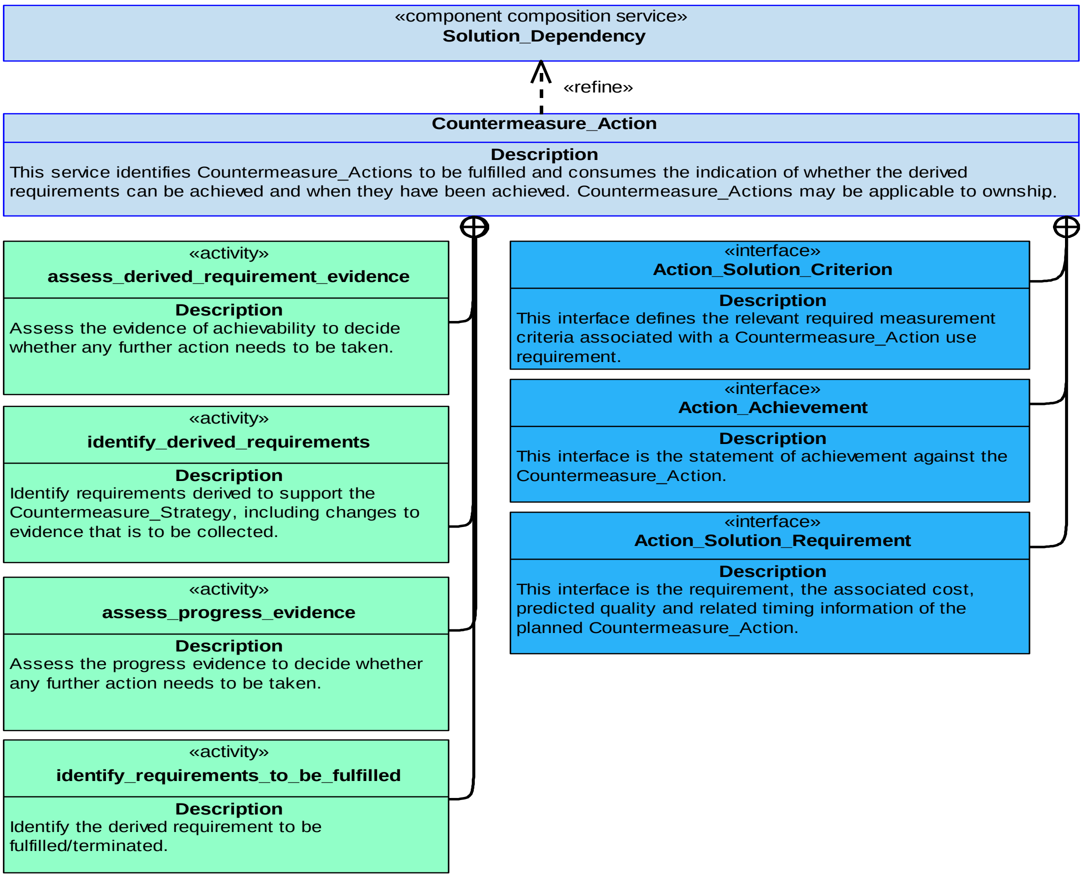

«component composition service»
Solution_Dependency
«refine»
Countermeasure_Action
Description
This service identifies Countermeasure_Actions to be fulfilled and consumes the indication of whether the derived requirements can be achieved and when they have been achieved. Countermeasure_Actions may be applicable to ownship.
«activity»
assess_derived_requirement_evidence
Description
Assess the evidence of achievability to decide whether any further action needs to be taken.
«activity»
identify_derived_requirements
Description
Identify requirements derived to support the Countermeasure_Strategy, including changes to evidence that is to be collected.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
identify_requirements_to_be_fulfilled
Description
Identify the derived requirement to be fulfilled/terminated.
«interface»
Action_Solution_Criterion
Description
This interface defines the relevant required measurement criteria associated with a Countermeasure_Action use requirement.
«interface»
Action_Achievement
Description
This interface is the statement of achievement against the Countermeasure_Action.
«interface»
Action_Solution_Requirement
Description
This interface is the requirement, the associated cost, predicted quality and related timing information of the planned Countermeasure_Action.

Figure 168: Countermeasure Service Policy

Countermeasure_Action

This service identifies Countermeasure_Actions to be fulfilled and consumes the indication of whether the derived requirements can be achieved and when they have been achieved. Countermeasure_Actions may be applicable to ownship or cooperating vehicles (e.g. a change of formation as part of a Countermeasure_Strategy).

Interfaces

Action_Solution_Criterion

This interface defines the relevant required measurement criteria associated with a Countermeasure_Action use requirement.

Attributes

property	The property to be measured, e.g. effectiveness of the jammer in terms of coverage or power level.
value	The amount related to the property to be measured.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Action_Solution_Requirement

This interface is the requirement, the associated cost, predicted quality and related timing information of the planned Countermeasure_Action.

Attributes

specification	The definition of the derived requirement, e.g. decoy release pattern.
countermeasure_action_type	The type of Countermeasure_Action required (e.g. chaff, flare, manoeuvre or pre-emptive attack).
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used or time taken.
predicted_quality	How well the proposed Countermeasure_Action is predicted to satisfy the requirement.

Action_Achievement

This interface is the statement of achievement against the Countermeasure_Action.

Activities

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_derived_requirements

Identify requirements derived to support the Countermeasure_Strategy, including changes to evidence that is to be collected.

identify_requirements_to_be_fulfilled

Identify the derived requirement to be fulfilled/terminated.

assess_derived_requirement_evidence

Assess the evidence of achievability to decide whether any further action needs to be taken.

5.4.2.9.7.1.3 Vehicle_Condition

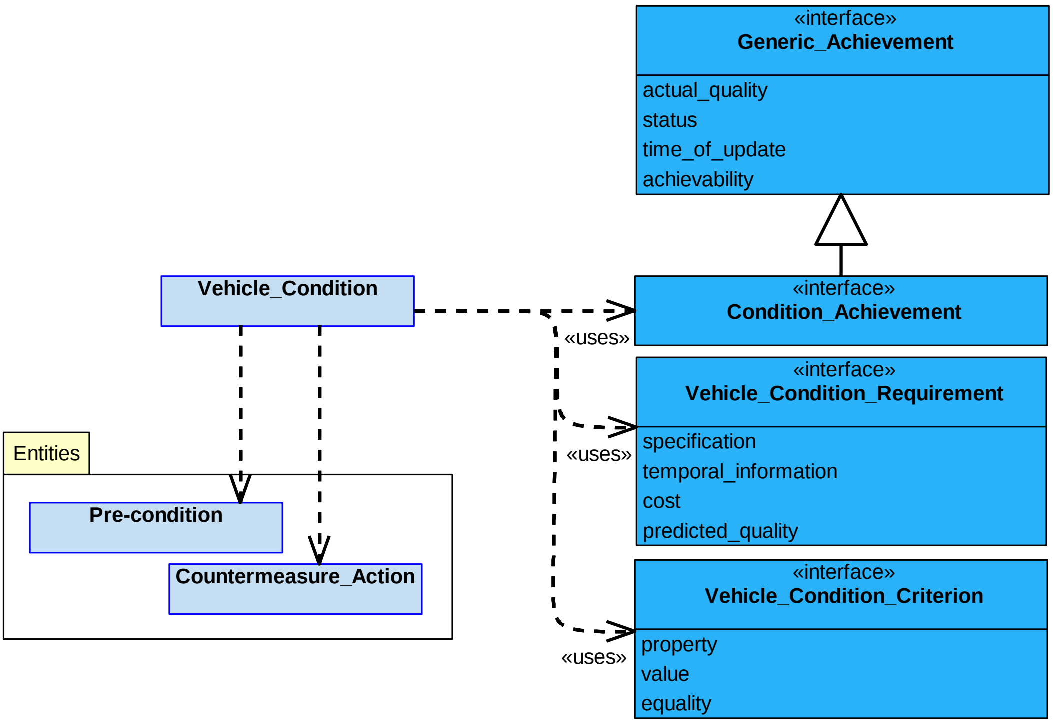


Figure 169: Vehicle_Condition Service Definition

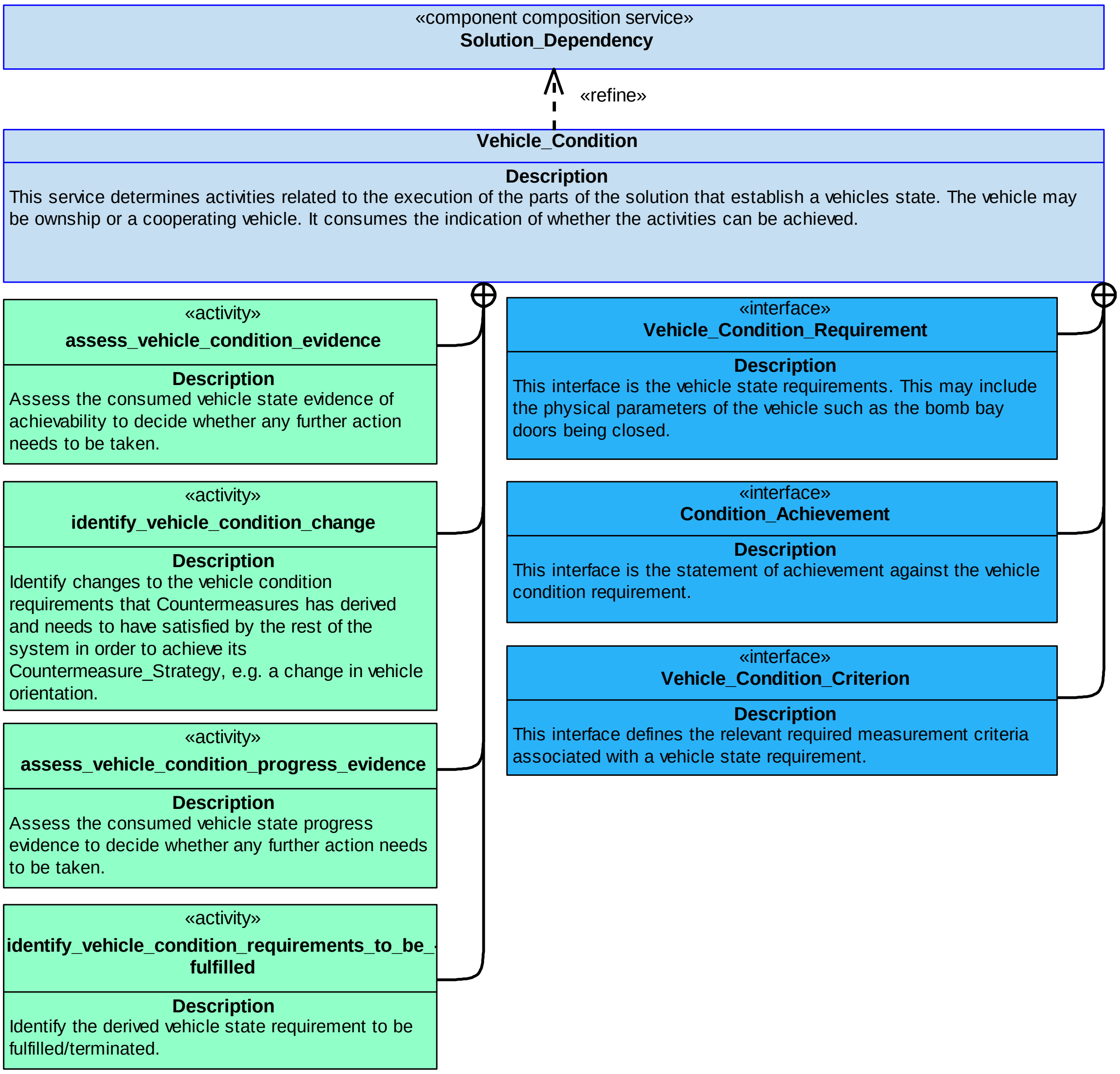

«component composition service»
Solution_Dependency
«refine»
Vehicle_Condition
Description
This service determines activities related to the execution of the parts of the solution that establish a vehicles state. The vehicle may be ownship or a cooperating vehicle. It consumes the indication of whether the activities can be achieved.
«activity»
assess_vehicle_condition_evidence
Description
Assess the consumed vehicle state evidence of achievability to decide whether any further action needs to be taken.
«activity»
identify_vehicle_condition_change
Description
Identify changes to the vehicle condition requirements that Countermeasures has derived and needs to have satisfied by the rest of the system in order to achieve its Countermeasure_Strategy, e.g. a change in vehicle orientation.
«activity»
assess_vehicle_condition_progress_evidence
Description
Assess the consumed vehicle state progress evidence to decide whether any further action needs to be taken.
«activity»
identify_vehicle_condition_requirements_to_be_-fulfilled
Description
Identify the derived vehicle state requirement to be fulfilled/terminated.
«interface»
Vehicle_Condition_Requirement
Description
This interface is the vehicle state requirements. This may include the physical parameters of the vehicle such as the bomb bay doors being closed.
«interface»
Condition_Achievement
Description
This interface is the statement of achievement against the vehicle condition requirement.
«interface»
Vehicle_Condition_Criterion
Description
This interface defines the relevant required measurement criteria associated with a vehicle state requirement.

Figure 170: Vehicle_Condition Service Policy

Vehicle_Condition

This service determines activities related to the execution of the parts of the solution that establish a vehicles state. The vehicle may be ownship or a cooperating vehicle. It consumes the indication of whether the activities can be achieved.

Interfaces

Vehicle_Condition_Requirement

This interface is the vehicle state requirements. This may include the physical parameters of the vehicle such as the bomb bay doors being closed.

Attributes

specification	The definition of the derived requirement, e.g. close weapons bay doors.
temporal_information	Information covering timing, such as start and end times.

cost	The cost of executing the solution, for example: resources used or time taken.
predicted_quality	How well the proposed vehicle state solution is predicted to satisfy the requirement.

Vehicle_Condition_Criterion

This interface defines the relevant required measurement criteria associated with a vehicle state requirement.

Attributes

property	The property to be measured, e.g. vehicle orientation.
value	The amount related to the property to be measured.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Condition_Achievement

This interface is the statement of achievement against the vehicle condition requirement.

Activities

assess_vehicle_condition_evidence

Assess the consumed vehicle state evidence of achievability to decide whether any further action needs to be taken.

assess_vehicle_condition_progress_evidence

Assess the consumed vehicle state progress evidence to decide whether any further action needs to be taken.

identify_vehicle_condition_change

Identify changes to the vehicle condition requirements that Countermeasures has derived and needs to have satisfied by the rest of the system in order to achieve its Countermeasure_Strategy, e.g. a change in vehicle orientation.

identify_vehicle_condition_requirements_to_be_fulfilled

Identify the derived vehicle state requirement to be fulfilled/terminated.

5.4.2.9.7.1.4 Spectrum_Use

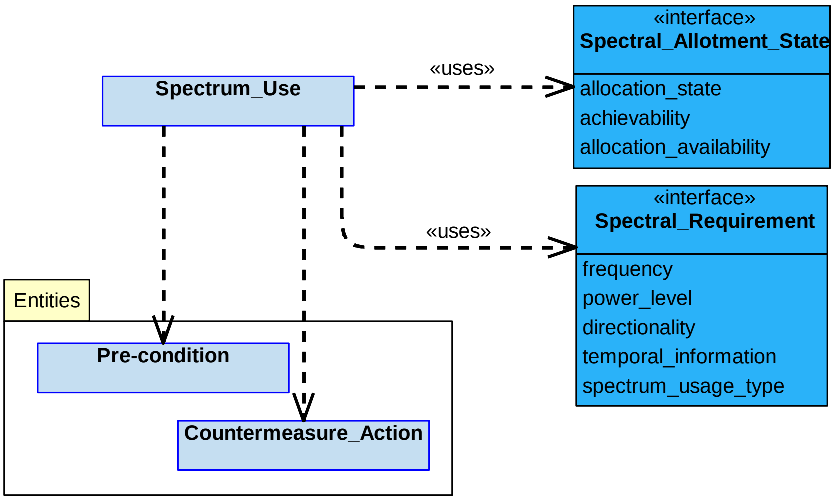


Figure 171: Spectrum_Use Service Definition

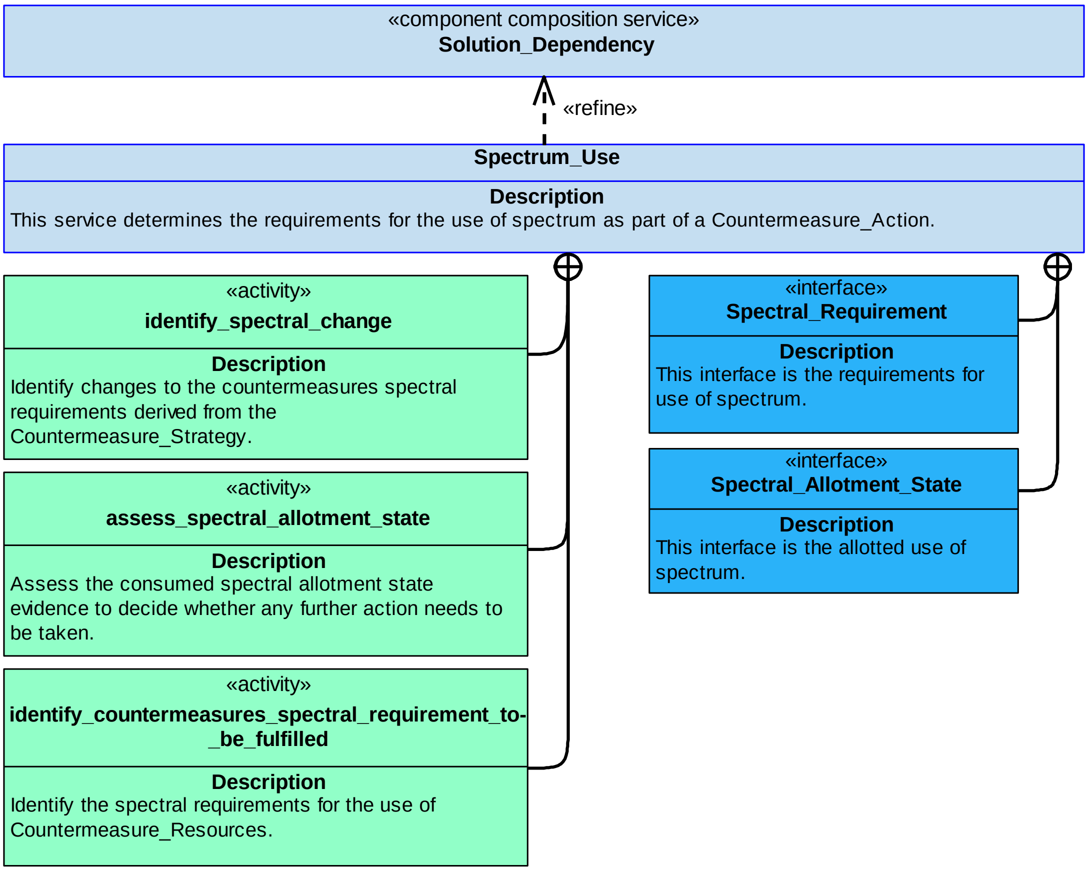

«component composition service»
Solution_Dependency
«refine»
Spectrum_Use
Description
This service determines the requirements for the use of spectrum as part of a Countermeasure_Action.
«activity»
identify_spectral_change
Description
Identify changes to the countermeasures spectral requirements derived from the Countermeasure_Strategy.
«activity»
assess_spectral_allotment_state
Description
Assess the consumed spectral allotment state evidence to decide whether any further action needs to be taken.
«activity»
identify_countermeasures_spectral_requirement_to-_be_fulfilled
Description
Identify the spectral requirements for the use of Countermeasure_Resources.
«interface»
Spectral_Requirement
Description
This interface is the requirements for use of spectrum.
«interface»
Spectral_Allotment_State
Description
This interface is the allotted use of spectrum.

Figure 172: Spectrum_Use Service Policy

Spectrum_Use

This service determines the requirements for the use of spectrum as part of a Countermeasure_Action.

Interfaces

Spectral_Requirement

This interface is the requirements for use of spectrum.

Attributes

frequency	The spectrum of interest, i.e. frequency, frequency range and tolerance.
power_level	The preferred power level.
directionality	The direction and spread (e.g. to direct the effects of jamming towards a threat).
temporal_information	Timing for the requested spectrum, such as start and end times. This might include segments of a requested time window that must not be interrupted etc.
spectrum_usage_type	Whether spectrum use is for transmitting, receiving or both.

Spectral_Allotment_State

This interface is the allotted use of spectrum.

Attributes

allocation_state	The current state of the service within its own lifecycle, e.g. raised, requested, acknowledged, allocated, rejected, claimed or released.
achievability	The achievability of a particular requirement (e.g. cannot find a resolution).
allocation_availability	The state of the allocation, from an availability point of view, e.g. it is assigned to a requirement and the window for use is open.

Activities

identify_countermeasures_spectral_requirement_to_be_fulfilled

Identify the spectral requirements for the use of Countermeasure_Resources.

identify_spectral_change

Identify changes to the countermeasures spectral requirements derived from the Countermeasure_Strategy.

assess_spectral_allotment_state

Assess the consumed spectral allotment state evidence to decide whether any further action needs to be taken.

5.4.2.9.7.1.5 Constraint

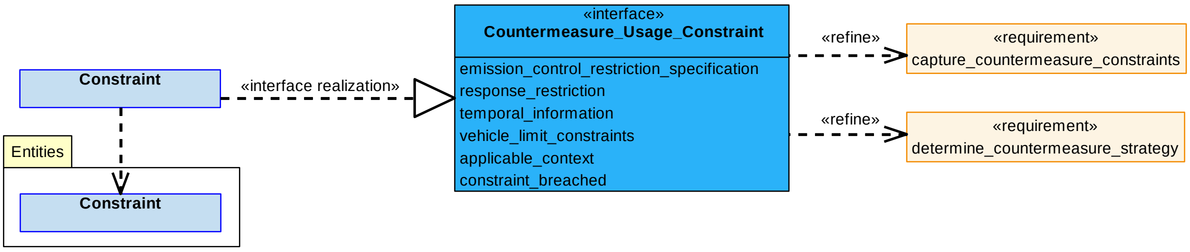


Figure 173: Constraint Service Definition

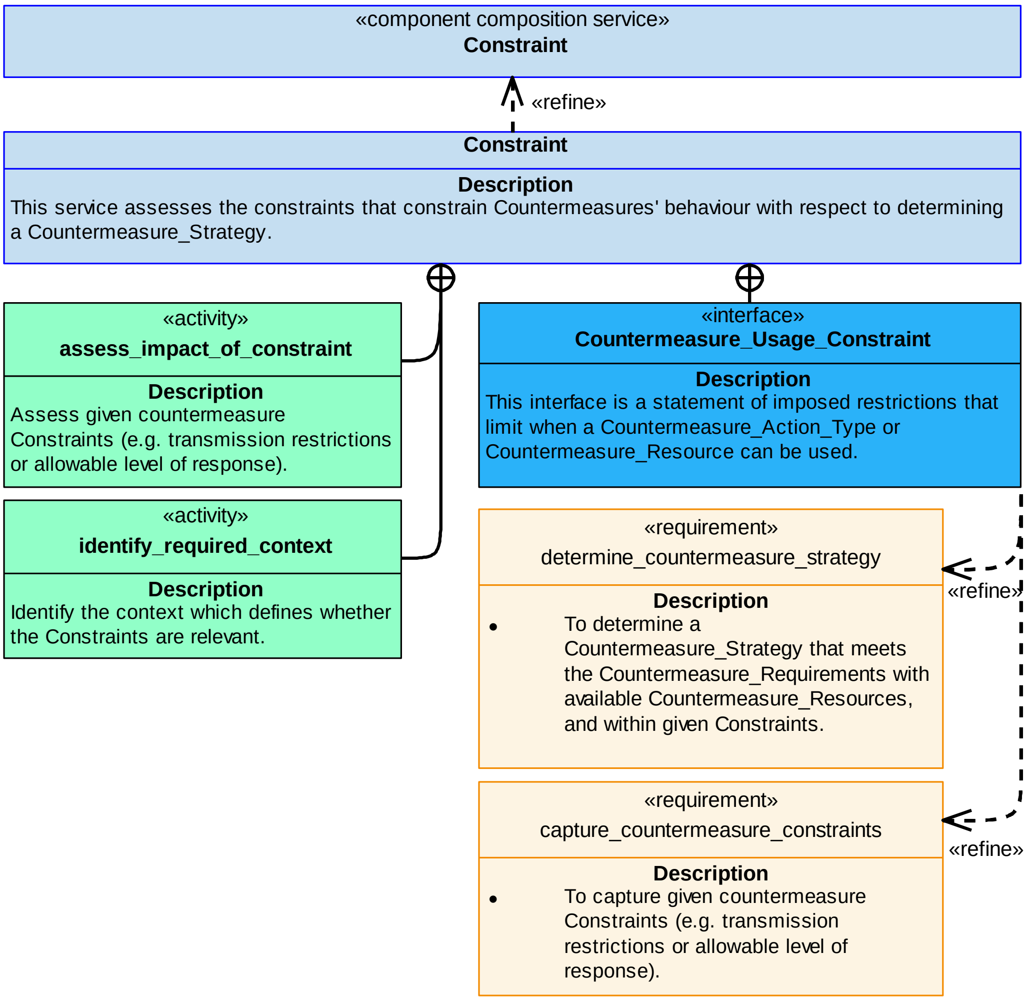

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the constraints that constrain Countermeasures' behaviour with respect to determining a Countermeasure_Strategy.
«activity»
assess_impact_of_constraint
Description
Assess given countermeasure Constraints (e.g. transmission restrictions or allowable level of response).
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Countermeasure_Usage_Constraint
Description
This interface is a statement of imposed restrictions that limit when a Countermeasure_Action_Type or Countermeasure_Resource can be used.
«requirement»
determine_countermeasure_strategy
Description
• To determine a Countermeasure_Strategy that meets the Countermeasure_Requirements with available Countermeasure_Resources, and within given Constraints.
«refine»
«requirement»
capture_countermeasure_constraints
Description
• To capture given countermeasure Constraints (e.g. transmission restrictions or allowable level of response).
«refine»

Figure 174: Constraint Service Policy

Constraint

This service assesses the constraints that constrain Countermeasures' behaviour with respect to determining a Countermeasure_Strategy.

Interface

Countermeasure_Usage_Constraint

This interface is a statement of imposed restrictions that limit when a Countermeasure_Action_Type or Countermeasure_Resource can be used.

Attributes

emission_control_restriction_specification	EMCON restrictions applied to the Countermeasure_Strategy, e.g. characteristics of the EM spectrum that the vehicle is not permitted to use.
response_restriction	A response restriction that is applied to the Countermeasure_Strategy, e.g. a limitation on the level of response.
temporal_information	Timing information pertaining to the periods of time when the Countermeasures constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
vehicle_limit_constraints	Constraints on the Countermeasure_Strategy imposed by Exploiting Platform performance, e.g. altitude limits, minimum and maximum speed.
applicable_context	The context in which the constraint is applicable.
constraint_breached	Whether the Countermeasures constraint has been inadvertently breached due to external factors.

Activities

assess_impact_of_constraint

Assess given countermeasure Constraints (e.g. transmission restrictions or allowable level of response).

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.9.7.1.6 Vehicle_Observability

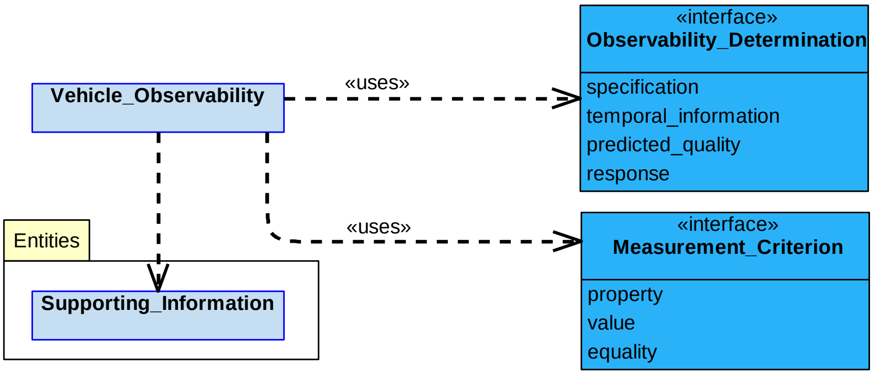


Figure 175: Vehicle_Observability Service Definition

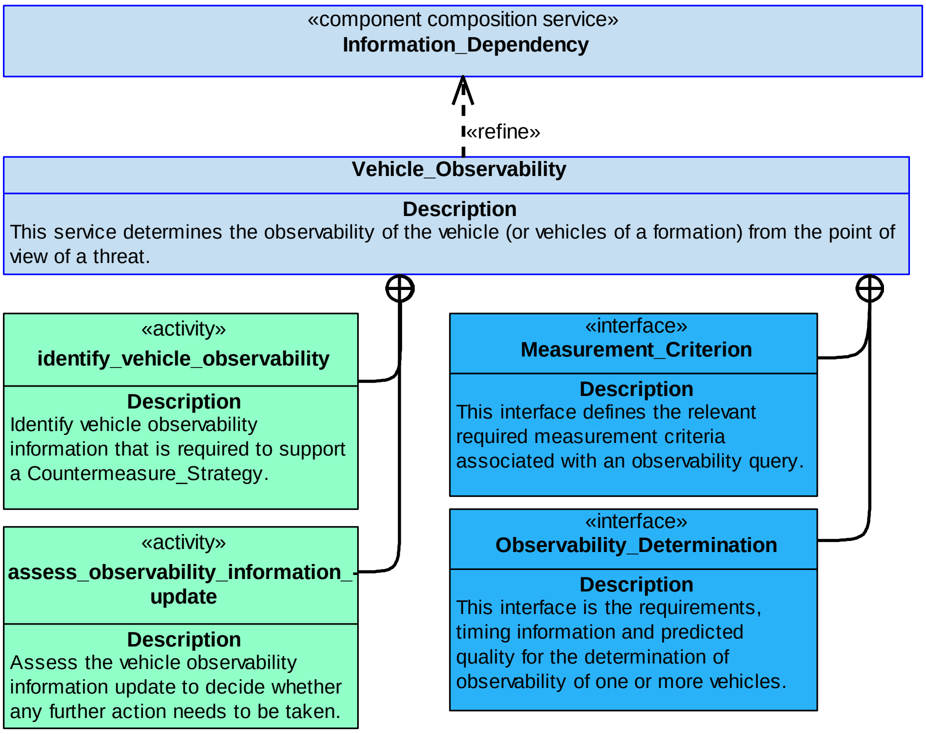

«component composition service»
Information_Dependency
«refine»
Vehicle_Observability
Description
This service determines the observability of the vehicle (or vehicles of a formation) from the point of view of a threat.
«activity»
identify_vehicle_observability
Description
Identify vehicle observability information that is required to support a Countermeasure_Strategy.
«activity»
assess_observability_information_update
Description
Assess the vehicle observability information update to decide whether any further action needs to be taken.
«interface»
Measurement_Criterion
Description
This interface defines the relevant required measurement criteria associated with an observability query.
«interface»
Observability_Determination
Description
This interface is the requirements, timing information and predicted quality for the determination of observability of one or more vehicles.

Figure 176: Vehicle_Observability Service Policy

Vehicle_Observability

This service determines the observability of the vehicle (or vehicles of a formation) from the point of view of a threat.

Interfaces

Measurement_Criterion

This interface defines the relevant required measurement criteria associated with an observability query.

Attributes

property	The property to be measured, e.g. calculated observability compared to a threshold.
value	The amount related to the property to be measured.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Observability_Determination

This interface is the requirements, timing information and predicted quality for the determination of observability of one or more vehicles.

Attributes

specification	The definition of the derived requirement, e.g. a query requesting observability data for a vehicle.
temporal_information	Information covering timing, such as start and end times.

predicted_quality	The quality of a query response against the defined Measurement_Criterion.
response	The response to the requirement specification.

Activities

identify_vehicle_observability

Identify vehicle observability information that is required to support a Countermeasure_Strategy.

assess_observability_information_update

Assess the vehicle observability information update to decide whether any further action needs to be taken.

5.4.2.9.7.1.7 Environment_Information

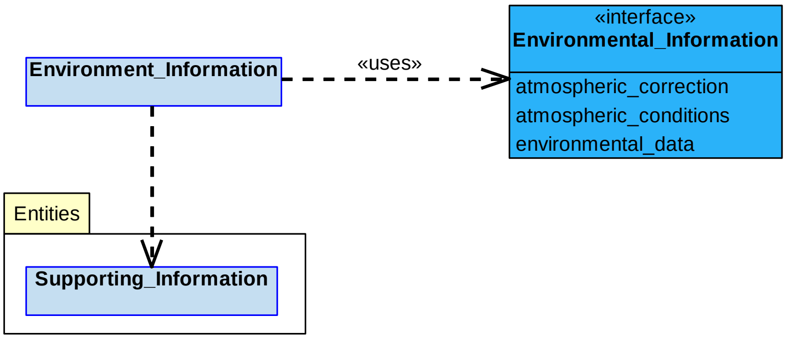
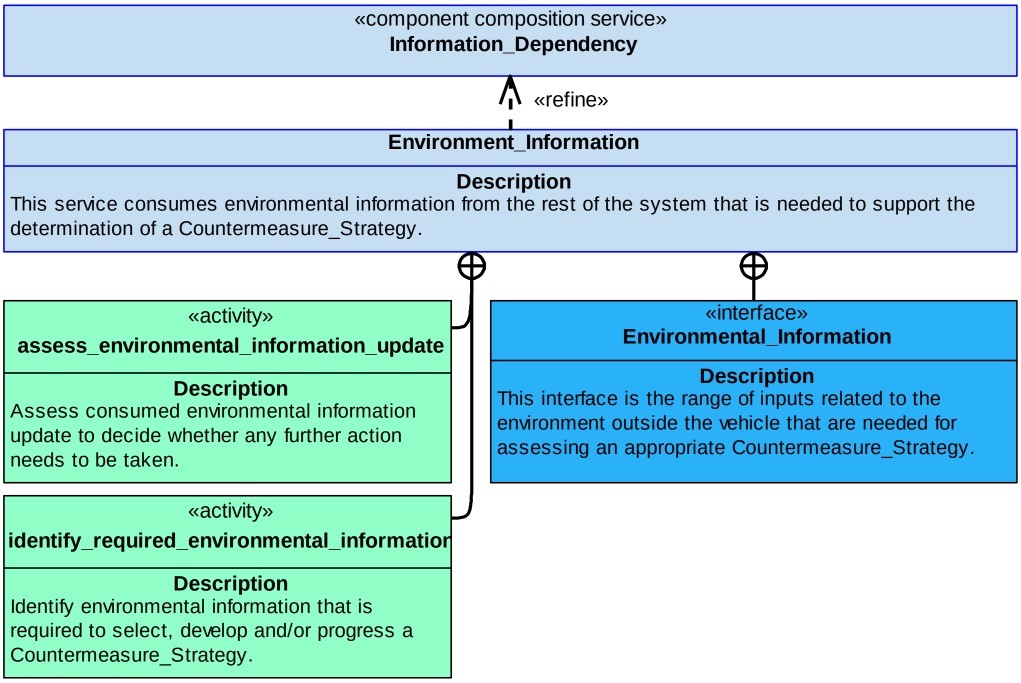


Figure 177: Environment_Information Service Definition


«component composition service»
Information_Dependency
«refine»
Environment_Information
Description
«activity»
assess_environmental_information_update
Description
«interface»
Environmental_Information
Description
«activity»
identify_required_environmental_informatior
Description

This service consumes environmental information from the rest of the system that is needed to support the determination of a Countermeasure_Strategy.

Assess consumed environmental information update to decide whether any further action needs to be taken.

This interface is the range of inputs related to the environment outside the vehicle that are needed for assessing an appropriate Countermeasure_Strategy.

Identify environmental information that is required to select, develop and/or progress a Countermeasure_Strategy.

Figure 178: Environment_Information Service Policy

Environment_Information

This service consumes environmental information from the rest of the system that is needed to support the determination of a Countermeasure_Strategy.

Interface

Environmental_Information

This interface is the range of inputs related to the environment outside the vehicle that are needed for assessing an appropriate Countermeasure_Strategy.

Attributes

atmospheric_correction	Spatial correction for sensors related to changes in atmospheric conditions that affects sensing processes and performance.
atmospheric_conditions	Current and predicted weather conditions and features.
environmental_data	Information describing surfaces and features in the environment that may affect sensing or effector processing and performance. This may include land terrain or other environments.

Activities

assess_environmental_information_update

Assess consumed environmental information update to decide whether any further action needs to be taken.

identify_required_environmental_information

Identify environmental information that is required to select, develop and/or progress a Countermeasure_Strategy.

5.4.2.9.7.1.8 Threat_Information

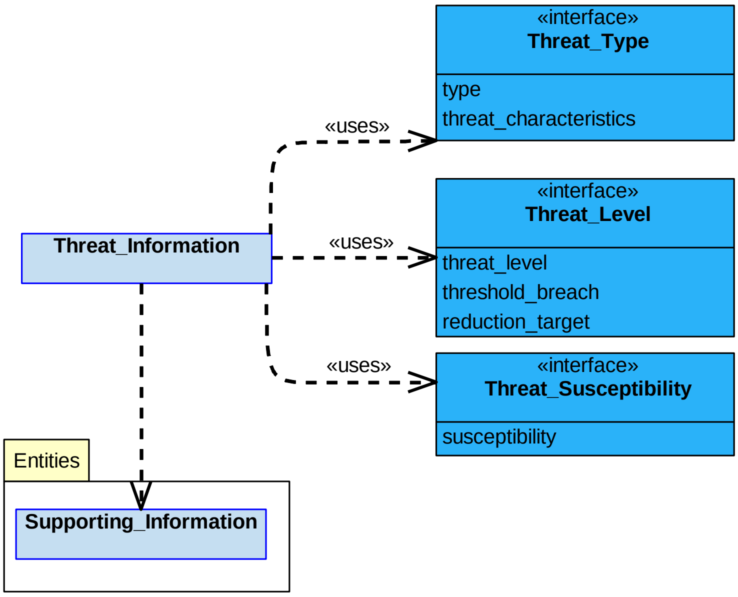


Figure 179: Threat_Information Service Definition

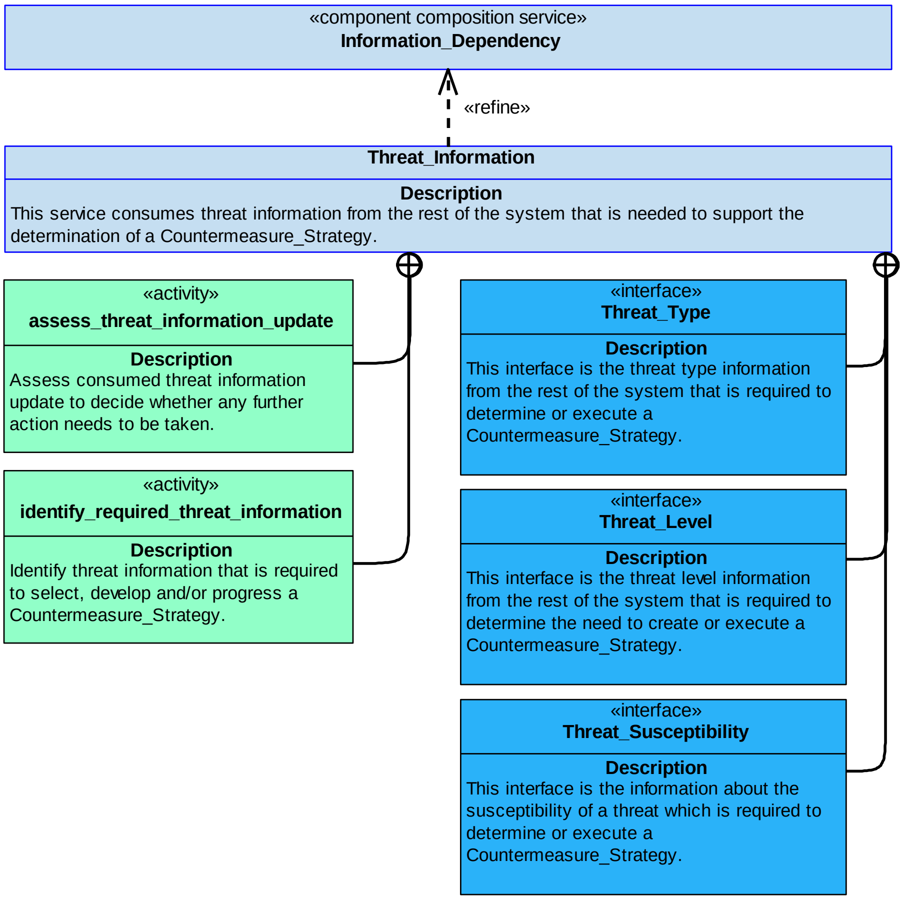

«component composition service»
Information_Dependency
«refine»
Threat_Information
Description
This service consumes threat information from the rest of the system that is needed to support the determination of a Countermeasure_Strategy.
«activity»
assess_threat_information_update
Description
Assess consumed threat information update to decide whether any further action needs to be taken.
«activity»
identify_required_threat_information
Description
Identify threat information that is required to select, develop and/or progress a Countermeasure_Strategy.
«interface»
Threat_Type
Description
This interface is the threat type information from the rest of the system that is required to determine or execute a Countermeasure_Strategy.
«interface»
Threat_Level
Description
This interface is the threat level information from the rest of the system that is required to determine the need to create or execute a Countermeasure_Strategy.
«interface»
Threat_Susceptibility
Description
This interface is the information about the susceptibility of a threat which is required to determine or execute a Countermeasure_Strategy.

Figure 180: Threat_Information Service Policy

Threat_Information

This service consumes threat information from the rest of the system that is needed to support the determination of a Countermeasure_Strategy.

Interfaces

Threat_Level

This interface is the threat level information from the rest of the system that is required to determine the need to create or execute a Countermeasure_Strategy.

Attributes

threat_level	The level of risk posed by a threat in the battlespace.
threshold_breach	An indication that the threat risk threshold has been breached, e.g. the proximity to threat has changed.
reduction_target	The target reduction to be achieved, in order to return the threat to an acceptable level.

Threat_Type

This interface is the threat type information from the rest of the system that is required to determine or execute a Countermeasure_Strategy.

Attributes

type	The type of threat, e.g. a missile, laser or radar.
threat_characteristics	The characteristics of a threat, e.g. radar frequency.

Threat_Susceptibility

This interface is the information about the susceptibility of a threat which is required to determine or execute a Countermeasure_Strategy.

Attribute

susceptibility	The susceptibility of a threat.

Activities

assess_threat_information_update

Assess consumed threat information update to decide whether any further action needs to be taken.

identify_required_threat_information

Identify threat information that is required to select, develop and/or progress a Countermeasure_Strategy.

5.4.2.9.7.1.9 Object_Information

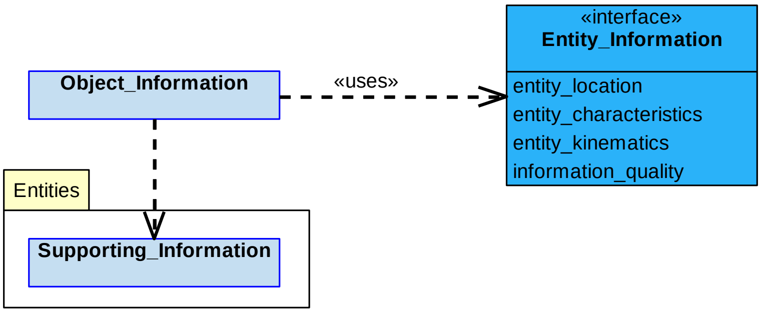


Figure 181: Object_Information Service Definition

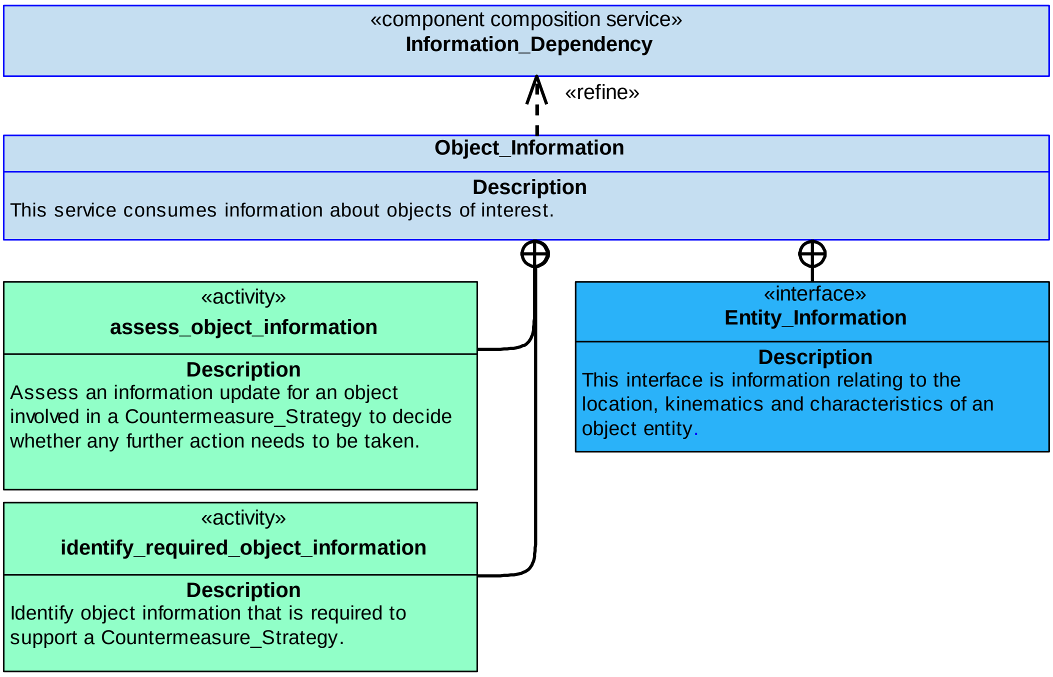

«component composition service»
Information_Dependency
«refine»
Object_Information
Description
This service consumes information about objects of interest.
«activity»
assess_object_information
Description
Assess an information update for an object involved in a Countermeasure_Strategy to decide whether any further action needs to be taken.
«activity»
identify_required_object_information
Description
Identify object information that is required to support a Countermeasure_Strategy.
«interface»
Entity_Information
Description
This interface is information relating to the location, kinematics and characteristics of an object entity.

Figure 182: Object_Information Service Policy

Object_Information

This service consumes information about objects of interest.

Interface

Entity_Information

This interface is information relating to the location, kinematics and characteristics of an object entity.

Attributes

entity_location	The relative position of an entity (e.g. range and bearing).
entity_characteristics	The characteristics of an entity, e.g. type, behaviour or allegiance.
entity_kinematics	Information relating to an entity's motion which may include predicted trajectory, speed, accelerations (x/y/z), altitude, maximum speed, etc.
information_quality	The quality of entity information.

Activities

assess_object_information

Assess an information update for an object involved in a Countermeasure_Strategy to decide whether any further action needs to be taken.

identify_required_object_information

Identify object information that is required to support a Countermeasure_Strategy.

5.4.2.9.7.1.10 Capability

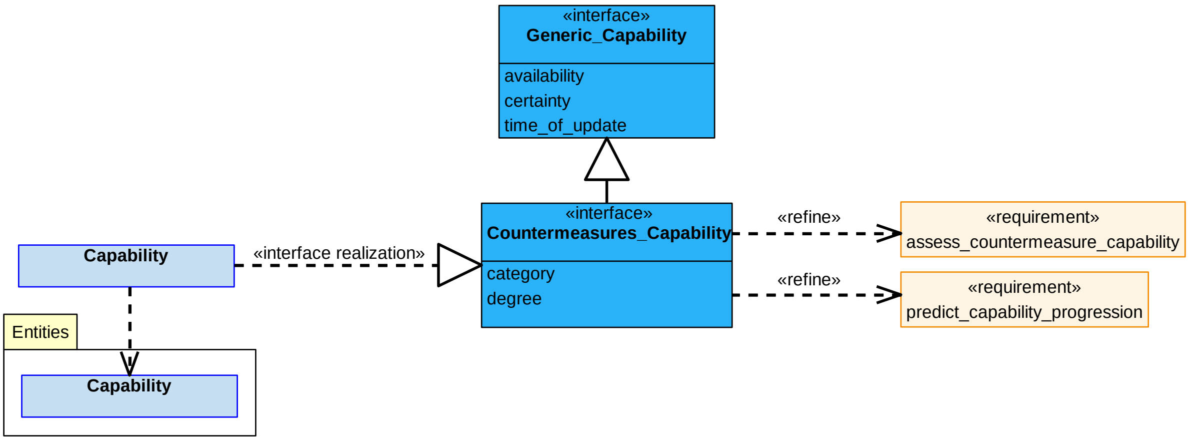


Figure 183: Capability Service Definition

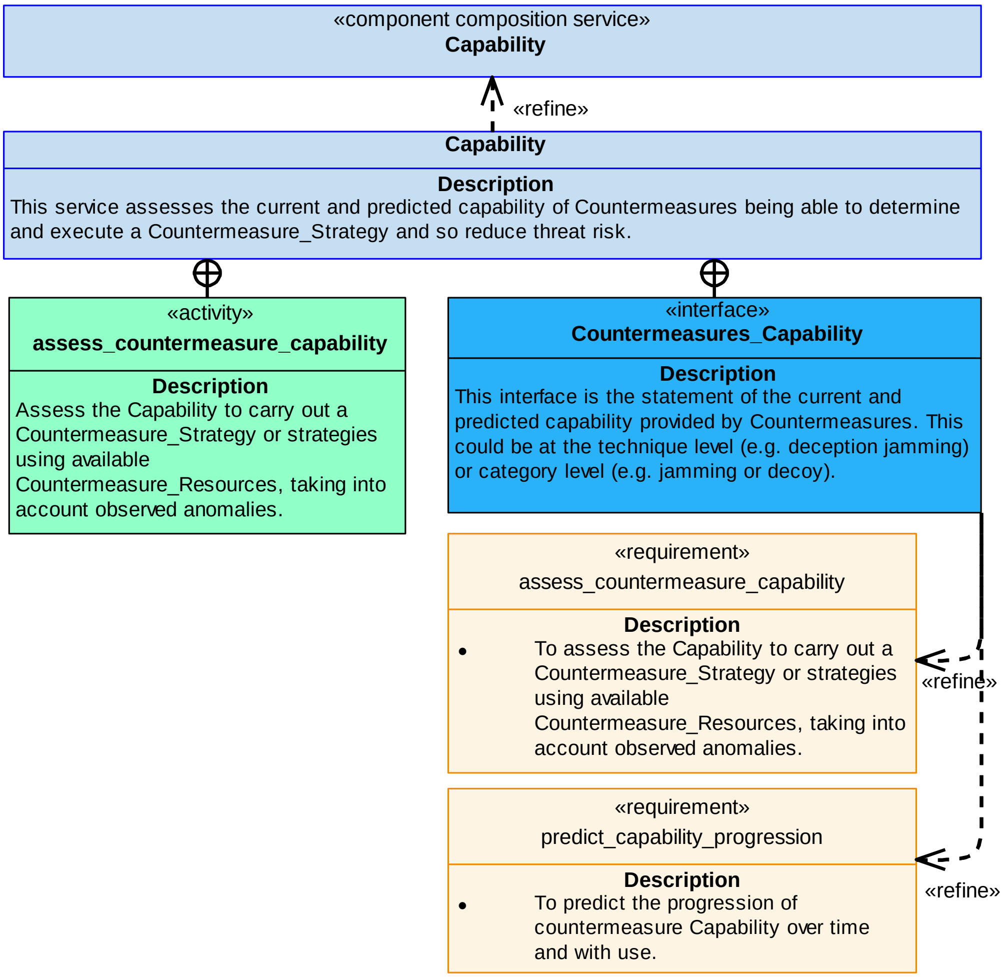

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability of Countermeasures being able to determine and execute a Countermeasure_Strategy and so reduce threat risk.
«activity»
assess_countermeasure_capability
Description
Assess the Capability to carry out a Countermeasure_Strategy or strategies using available Countermeasure_Resources, taking into account observed anomalies.
«interface»
Countermeasures_Capability
Description
This interface is the statement of the current and predicted capability provided by Countermeasures. This could be at the technique level (e.g. deception jamming) or category level (e.g. jamming or decoy).
«requirement»
assess_countermeasure_capability
Description
• To assess the Capability to carry out a Countermeasure_Strategy or strategies using available Countermeasure_Resources, taking into account observed anomalies.
«refine»
«requirement»
predict_capability_progression
Description
• To predict the progression of countermeasure Capability over time and with use.
«refine»

Figure 184: Capability Service Policy

Capability

This service assesses the current and predicted capability of Countermeasures being able to determine and execute a Countermeasure_Strategy and so reduce threat risk.

Interface

Countermeasures_Capability

This interface is the statement of the current and predicted capability provided by Countermeasures. This could be at the technique level (e.g. deception jamming) or category level (e.g. jamming or decoy).

Attributes

category	The type or category of Capability that is being provided.
degree	The level of performance or effectiveness that can be achieved for this Capability.

Activity

assess_countermeasure_capability

Assess the Capability to carry out a Countermeasure_Strategy or strategies using available Countermeasure_Resources, taking into account observed anomalies.

5.4.2.9.7.1.11 Capability_Evidence

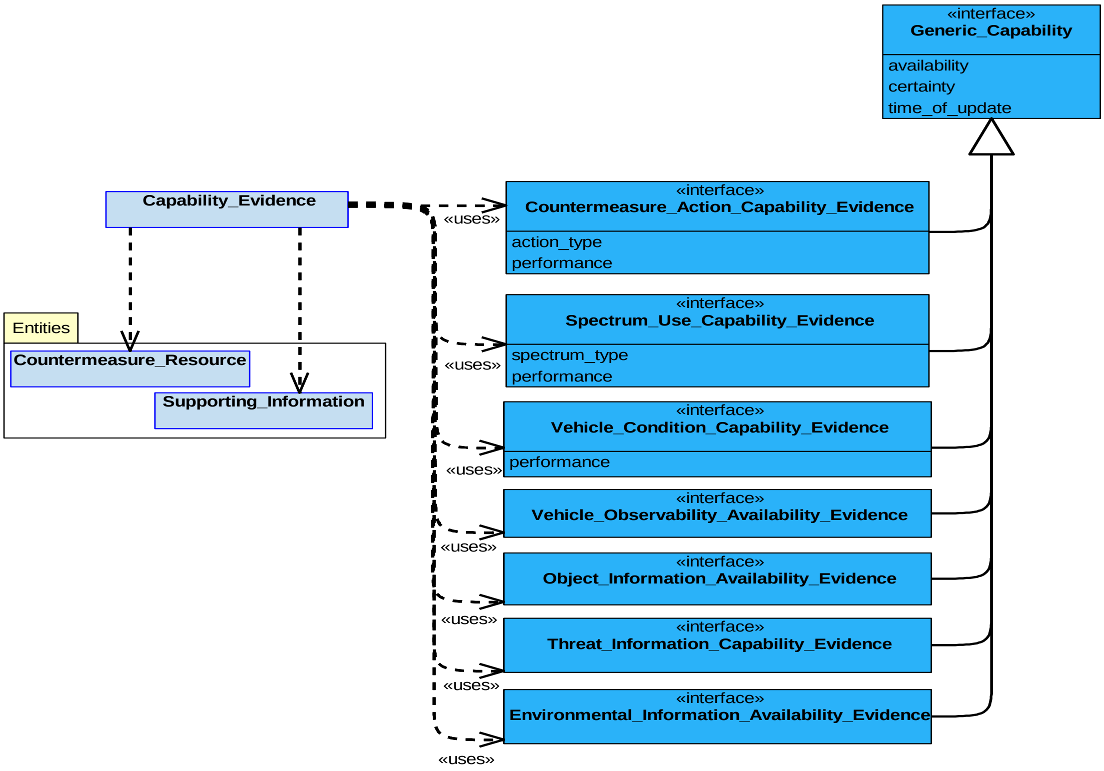
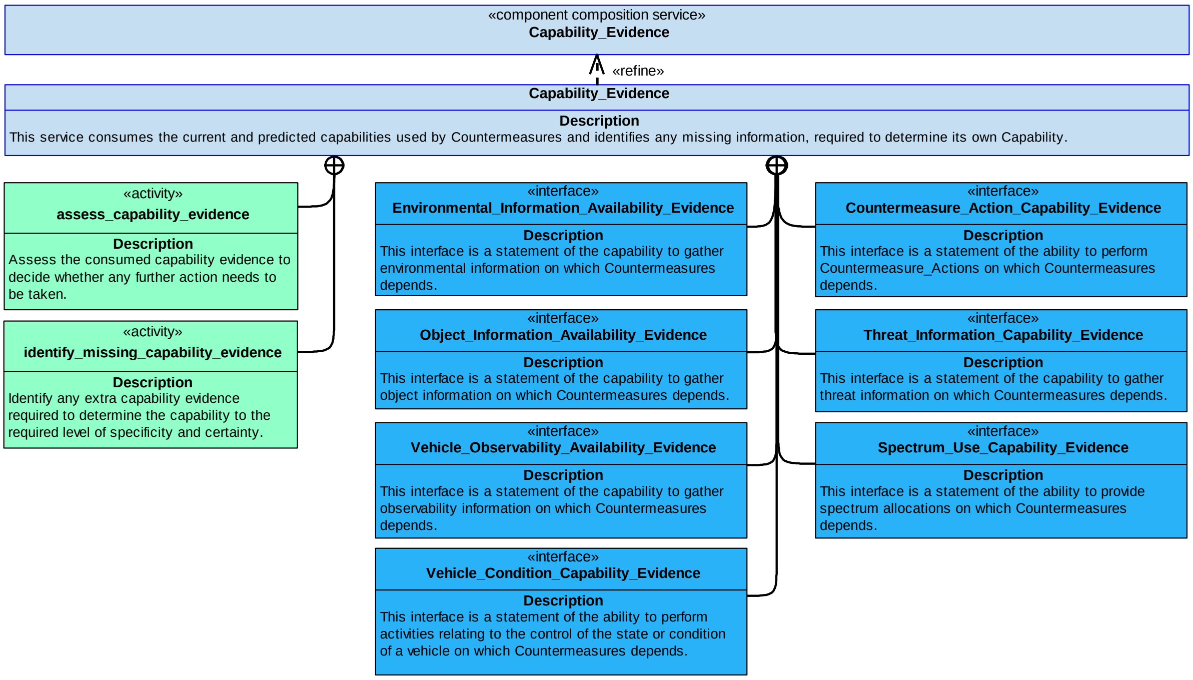

«interface»
Generic_Capability
availability
certainty
time_of_update
Capability_Evidence
Entities
Countermeasure_Resource
Supporting_Information
«uses»
«interface»
Countermeasure_Action_Capability_Evidence
action_type
performance
«interface»
Spectrum_Use_Capability_Evidence
spectrum_type
performance
«interface»
Vehicle_Condition_Capability_Evidence
performance
«interface»
Vehicle_Observability_Availability_Evidence
«interface»
Object_Information_Availability_Evidence
«interface»
Threat_Information_Capability_Evidence
«interface»
Environmental_Information_Availability_Evidence

Figure 185: Capability_Evidence Service Definition


«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
«activity»
assess_capability_evidence
Description
«activity»
identify_missing_capability_evidence
Description
«interface»
Environmental_Information_Availability_Evidence
Description
«interface»
Object_Information_Availability_Evidence
Description
«interface»
Vehicle_Observability_Availability_Evidence
Description
«interface»
Vehicle_Condition_Capability_Evidence
Description
«interface»
Countermeasure_Action_Capability_Evidence
Description
«interface»
Threat_Information_Capability_Evidence
Description
«interface»
Spectrum_Use_Capability_Evidence
Description

This service consumes the current and predicted capabilities used by Countermeasures and identifies any missing information, required to determine its own Capability.

Assess the consumed capability evidence to decide whether any further action needs to be taken.

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

This interface is a statement of the capability to gather environmental information on which Countermeasures depends.

This interface is a statement of the capability to gather object information on which Countermeasures depends.

This interface is a statement of the capability to gather observability information on which Countermeasures depends.

This interface is a statement of the ability to perform activities relating to the control of the state or condition of a vehicle on which Countermeasures depends.

This interface is a statement of the ability to perform Countermeasure_Actions on which Countermeasures depends.

This interface is a statement of the capability to gather threat information on which Countermeasures depends.

This interface is a statement of the ability to provide spectrum allocations on which Countermeasures depends.

Figure 186: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capabilities used by Countermeasures and identifies any missing information, required to determine its own Capability.

<u>Interfaces</u>

Environmental_Information_Availability_Evidence

This interface is a statement of the capability to gather environmental information on which Countermeasures depends.

Object_Information_Availability_Evidence

This interface is a statement of the capability to gather object information on which Countermeasures depends.

Vehicle_Observability_Availability_Evidence

This interface is a statement of the capability to gather observability information on which Countermeasures depends.

Vehicle_Condition_Capability_Evidence

This interface is a statement of the ability to perform activities relating to the control of the state or condition of a vehicle on which Countermeasures depends.

<u>Attribute</u>

performance	The level or degree of capability available for achieving a particular vehicle state or condition.

Spectrum_Use_Capability_Evidence

This interface is a statement of the ability to provide spectrum allocations on which Countermeasures depends.

Attributes

spectrum_type	The type of spectrum to which the capability statement applies.
performance	The level or degree of capability available for the use of spectrum.

Threat_Information_Capability_Evidence

This interface is a statement of the capability to gather threat information on which Countermeasures depends.

Countermeasure_Action_Capability_Evidence

This interface is a statement of the ability to perform Countermeasure_Actions on which Countermeasures depends.

Attributes

action_type	The Countermeasure_Action_Type to which the capability statement applies.
performance	The level or degree of capability available for a specific use of a countermeasure action.

Activities

assess_capability_evidence

Assess the consumed capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.9.7.2 Service Dependencies

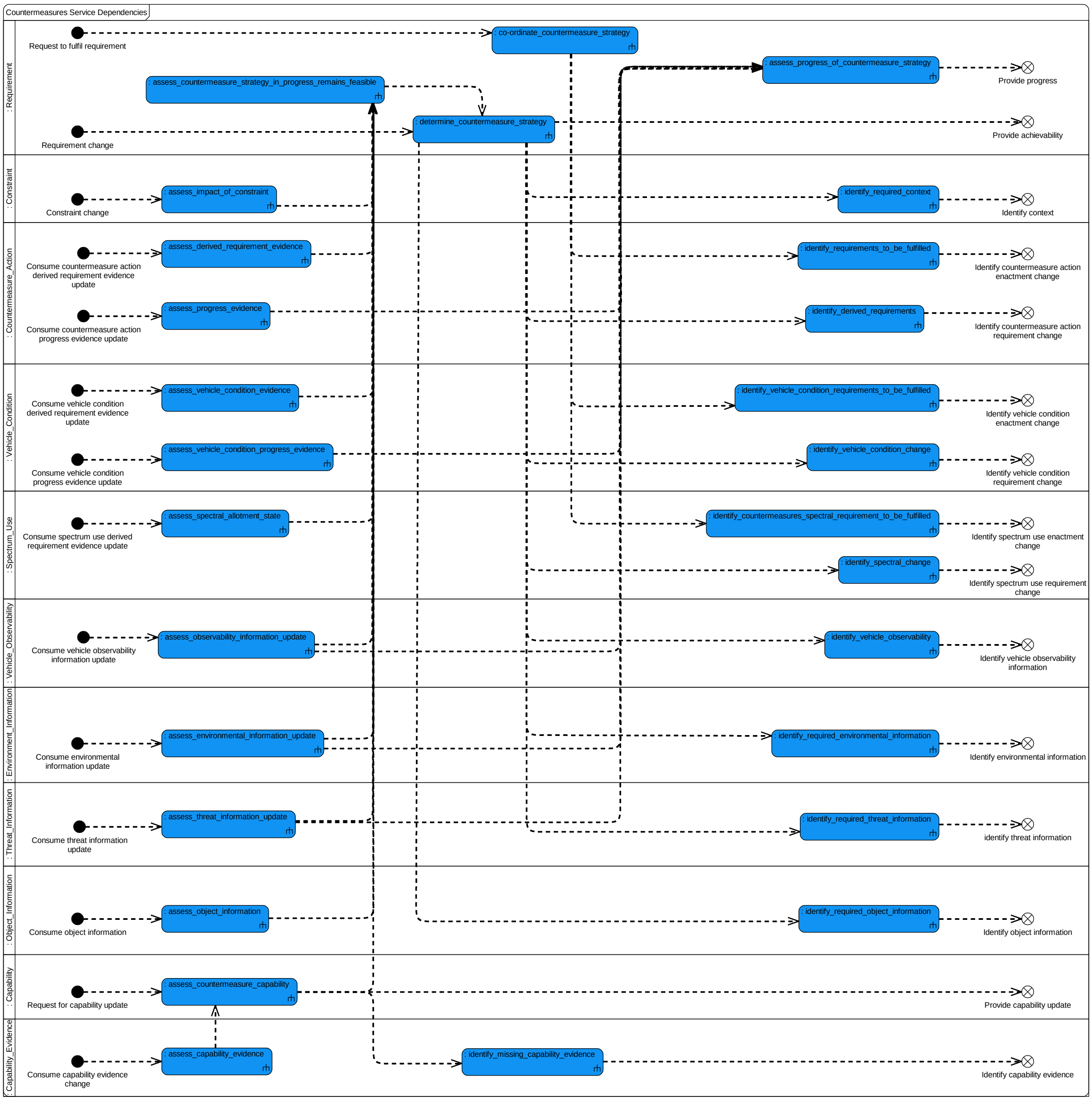

Countermeasures Service Dependencies
: Requirement
: Constraint
: Countermeasure_Action
: Vehicle_Condition
: Spectrum_Use
: Vehicle_Observability
: Environment_Information
: Threat_Information
: Object_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
: co-ordinate_countermeasure_strategy
: assess_progress_of_countermeasure_strategy
Provide progress
: assess_countermeasure_strategy_in_progress_remains_feasible
: determine_countermeasure_strategy
Requirement change
Provide achievability
Constraint change
: assess_impact_of_constraint
: identify_required_context
Identify context
Consume countermeasure action derived requirement evidence update
: assess_derived_requirement_evidence
: identify_requirements_to_be_fulfilled
Identify countermeasure action enactment change
Consume countermeasure action progress evidence update
: assess_progress_evidence
: identify_derived_requirements
Identify countermeasure action requirement change
Consume vehicle condition derived requirement evidence update
: assess_vehicle_condition_evidence
: identify_vehicle_condition_requirements_to_be_fulfilled
Identify vehicle condition enactment change
Consume vehicle condition progress evidence update
: assess_vehicle_condition_progress_evidence
: identify_vehicle_condition_change
Identify vehicle condition requirement change
Consume spectrum use derived requirement evidence update
: assess_spectral_allotment_state
: identify_countermeasures_spectral_requirement_to_be_fulfilled
Identify spectrum use enactment change
: identify_spectral_change
Identify spectrum use requirement change
Consume vehicle observability information update
: assess_observability_information_update
: identify_vehicle_observability
Identify vehicle observability information
Consume environmental information update
: assess_environmental_information_update
: identify_required_environmental_information
Identify environmental information
Consume threat information update
: assess_threat_information_update
: identify_required_threat_information
identify threat information
Consume object information
: assess_object_information
: identify_required_object_information
Identify object information
Request for capability update
: assess_countermeasure_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 187: Countermeasures Service Dependencies

5.4.2.10 Cryptographic Materials

5.4.2.10.1 Role

The role of Cryptographic Materials is to manage and distribute cryptographic keys, algorithms and certificates.

5.4.2.10.2 Overview

Control Architecture

Cryptographic Materials is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

A Requirement for encryption or decryption will trigger Cryptographic Materials into making Cryptographic_Material available to support the cryptographic activity that is appropriate for the Protection_Level of the data (cryptography is not performed by this component). The update of Cryptographic_Material will be coordinated in accordance with the Material_Plan, e.g. rolled at a defined point of time or sanitised if reported compromised.

Examples of Use

- Cryptographic Materials will be used where management of cryptographic keys, algorithms and certificates is required.

5.4.2.10.3 Service Summary

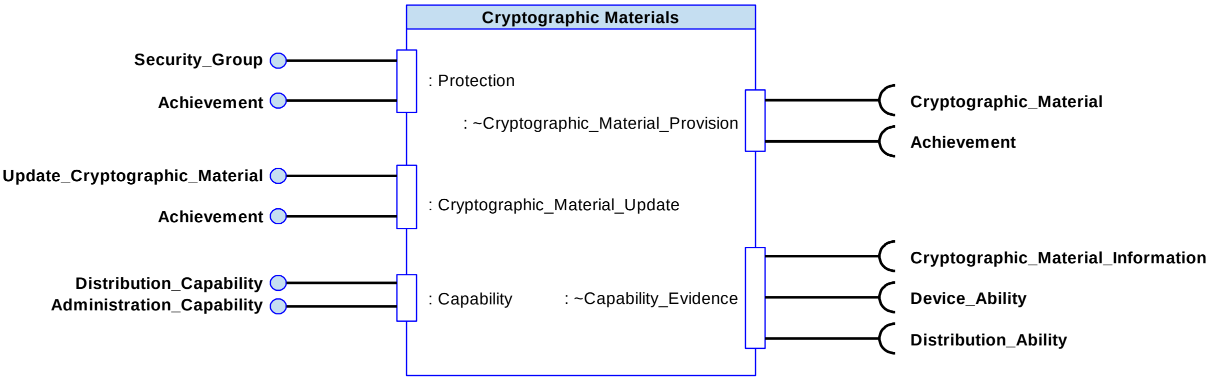


Figure 188: Cryptographic Materials Service Summary

5.4.2.10.4 Responsibilities

capture_crypto_material_requirements

- To capture Requirements for the use of Cryptographic_Material.

determine_material_plan

- To determine a Material_Plan that complies with the data Segregation_Policy (e.g. for specific security domains, types or classifications of data).

determine_crypto_material_usage

- To determine when and where particular Cryptographic_Material needs to be used.

coordinate_sanitisation

- To coordinate the sanitisation of Cryptographic_Material, including emergency sanitisation, response to compromised key lists and certificate revocation list (CRL) checking.

coordinate_material_change

- To coordinate the change of Cryptographic_Material, e.g. rollover of cryptographic keys and certificates to maintain its validity.

distribute_crypto_material

- To distribute the required Cryptographic_Material.

identify_material_solution_progress

- To identify what progress has been made against the Requirement.

assess_capability

- To assess the ability of the component to provide appropriate Cryptographic_Material, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

5.4.2.10.5 Subject Matter Semantics

The subject matter of Cryptographic Materials is which cryptographic keys, algorithms and certificates are to be used for encryption and decryption on the Exploiting Platform, including where they are and the level of protection they provide.

Exclusions

The subject matter of Cryptographic Materials does not include:

- The implementation details by which Cryptographic_Material is utilised, only how it is managed and made available.

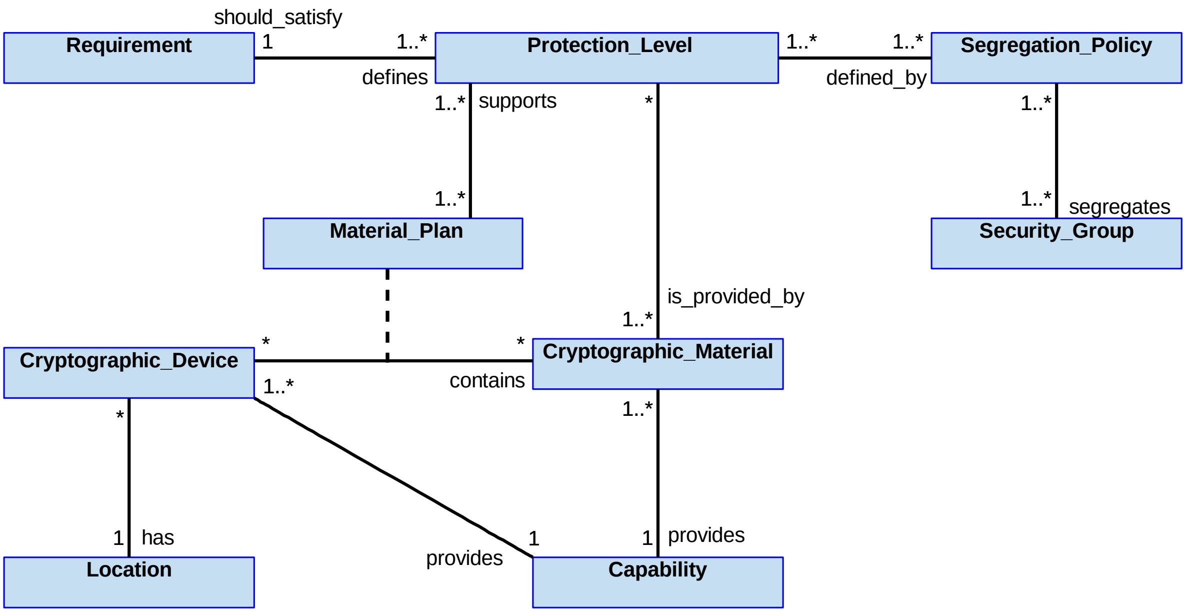


Figure 189: Cryptographic Materials Semantics

5.4.2.10.5.1 Entities

Capability

The capability of the component to update and make cryptographic material available for use.

Cryptographic_Device

A set of hardware, software and firmware that performs one or more cryptographic functions, such as being the secure key store or key generator.

Cryptographic_Material

An item used in the process of encryption or decryption, e.g. a key, algorithm (including for hash functions) or certificate.

Location

A defined logical position where encryption/decryption occurs, e.g. the location in a network or link, the storage location or other crypto device location.

Material_Plan

The relationship between Cryptographic_Material and Cryptographic_Devices that defines conditions of use, e.g. requested and granted distribution, installation, compatibility of, and destruction of Cryptographic_Material.

Protection_Level

The level of protection provided by cryptography against loss of confidentiality, integrity and/or availability (or other security attributes).

Requirement

A requirement to manage or apply a cryptographic protection.

Security_Group

A group of items that have similar segregation requirements, e.g. security domains, types and classification of data.

Segregation_Policy

The definition of the specific security domains, their segregation and use.

5.4.2.10.6 Design Rationale

5.4.2.10.6.1 Assumptions

- Cryptographic_Material is frequently stored in an encrypted form in a Cryptographic_Device, so it usually has an associated encryption key set. Keys available will include the Cryptographic Ignition Key (CIK), Algorithm Encryption Key (AEK), Key Encryption Key (KEK) and Data Encryption Key (DEK).
- Legacy, coalition and sovereign cryptographic implementations (which may not be PYRAMID compliant) will need to be supported.
- Military and commercial cryptography will need to be supported.
- If a specific key needs to be removed from a known named Cryptographic_Device, that will be done directly and Cryptographic Materials doesn't need to get involved other than being informed of the intended removal.
- Cryptographic_Devices will be responsible for sanitisation of their own Cryptographic_Material; the Cryptographic Materials component coordinates the activity.
- Over The Air Rekeying (OTAR) can be handled like any other rekeying and any extra protection required in transit will be provided by other components.
- Cryptographic Materials will be responsible for making Cryptographic_Material available for an encryption or decryption task, not the fulfilment of that task.
- Cryptographic_Devices will be allocated to security domains at build time.

5.4.2.10.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Cryptographic Materials:

- Recording and Logging - the crypto-related security logging for Cryptographic_Devices will also be done by Cryptographic Materials.
- Storage - encryption of storage media is expected, with the possibility of data being made inaccessible by revocation of the Cryptographic_Material.

Exploitation Considerations

- This component will capture the bulk of information expected within the cryptographic plan.

- Cryptographic_Material will always be planned to be used as part of a set, the whole set being needed to conduct an encryption/decryption activity. However different storage devices may be approved for different security levels so different parts of a set may be distributed to different devices, in accordance with the applicable Segregation_Policy.
- Cryptographic Materials may request to sanitise data. Complying with the request will be managed by the Cryptographic_Device. However Cryptographic Materials will report any store/device that fails to confirm that a request to sanitise data has been completed.
- Keys will generally be stored in a dedicated key store with an emergency sanitisation facility.
- Emergency sanitisation may need to be implemented as a dedicated service interface due to timeliness concerns.

5.4.2.10.6.3 Safety Considerations

The indicative IDAL is DAL A*.*

The rationale behind this is:

Failure of this component could lead to:

- Loss of availability of transfers of encrypted and/or hashed data by failure to provide the correct Cryptographic_Material. For example, communications between a ground-based control station and the air vehicle, which is primarily a concern for UAVs, but may apply to manned air vehicles where some functions are controlled by external users. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) then the air vehicle will have been designed to mitigate a loss of communications. For UAS this would by rely on pre-determined automated or autonomous behaviour. For this failure mode it is concluded that failure of this component may result a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL for this aspect is C.
- If Cryptographic_Material is required to access data critical to flight, inadvertent sanitisation could lead to an uncontrolled crash of the air vehicle and fatalities, i.e. a catastrophic hazard. Unless an Exploiting Platform can include a protection mechanism downstream of this component that prevents deletion of material at the wrong time then failure of this component could be catastrophic. Therefore the indicative DAL for this component is A.

5.4.2.10.6.4 Security Considerations

The indicative security classification is SNEO.

This component is central to the confidentiality, integrity and authenticity of system data; it is responsible for the distribution of Cryptographic_Material that could be up to TS, however it is expected such material would be handled separately from any other secure data and the component itself will likely have an indicative classification of SNEO. The confidentiality of the Cryptographic_Material is paramount to that of all data handled by the Exploiting Platform, with additional handling methods being required due to the nature of the different material, e.g. CIK and DEK in different stores.

This component provides security related functions through:

- **Logging of Security Data** for the component and its associated Cryptographic_Devices, including location of Cryptographic_Material and its uses.
- **Maintaining Audit Records** relating to authorisation of key changes, key rollovers and sanitisation/device purges, etc.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may need to be protected to assure continued airworthiness.
- **System Status and Monitoring** for the Cryptographic_Material and Cryptographic_Devices, including for the enacting of a demand for sanitisation.

It fundamentally implements security enforcing functions by:

- Managing the Cryptographic_Material used for **Encrypting Data** in order to protect that data.
- Protecting the confidentiality, integrity and authenticity of encrypted system data, therefore **Preventing Cyber Attacks and Malware**.
- **Rendering Sensitive Data Inaccessible** by coordinating the sanitisation of Cryptographic_Material, adhering to compromised key lists and certificate revocation list (CRL) checking, etc.
- **Restricting Access to Data** that is encrypted. Access to the Cryptographic_Material itself is also strictly controlled to ensure keys are not compromised. This component is cognisant of the separation and CIA requirements of data within different security domains.

5.4.2.10.7 Services

5.4.2.10.7.1 Service Definitions

5.4.2.10.7.1.1 Protection

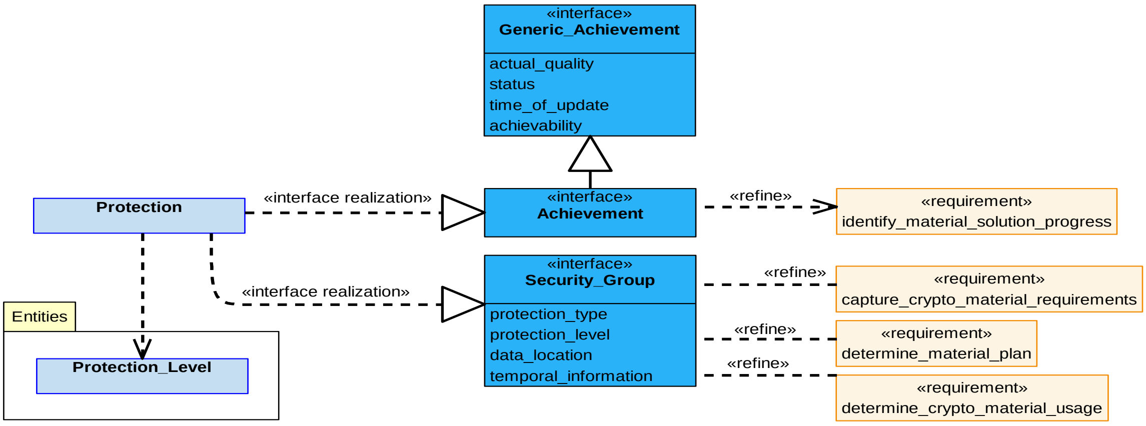


Figure 190: Protection Service Definition

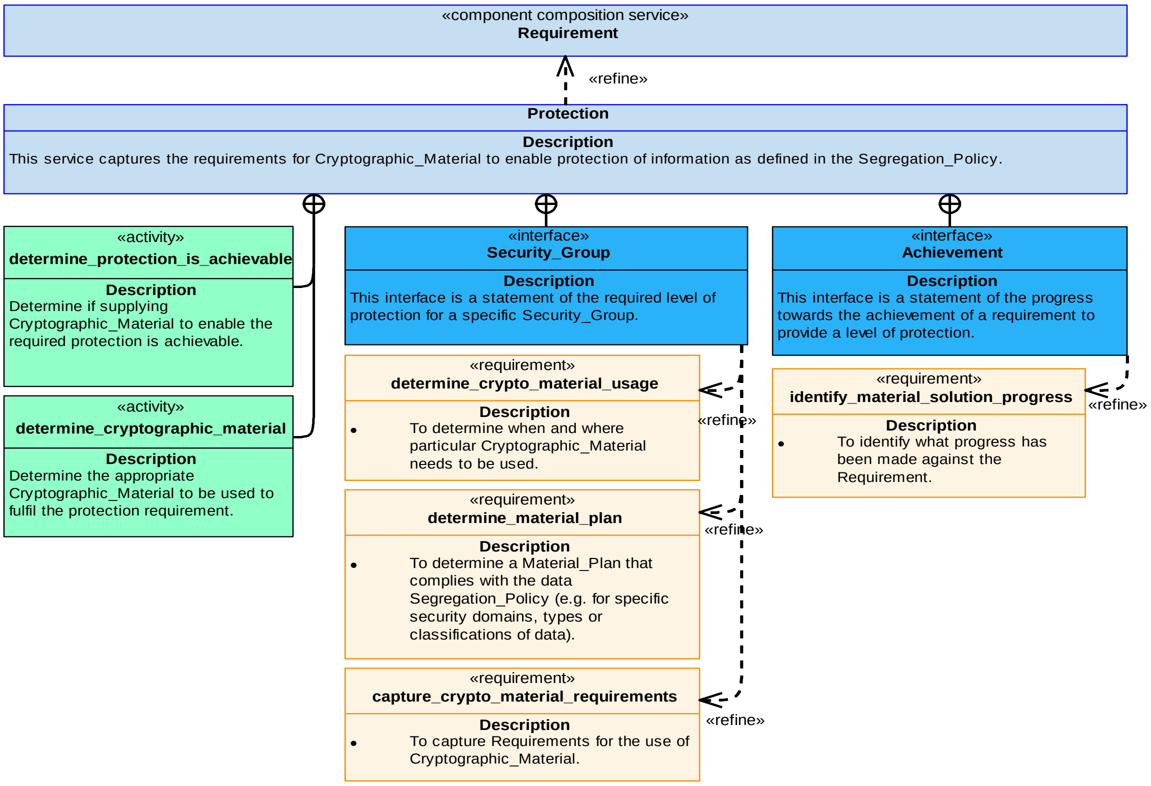

«component composition service»
Requirement
«refine»
Protection
Description
This service captures the requirements for Cryptographic_Material to enable protection of information as defined in the Segregation_Policy.
«activity»
determine_protection_is_achievable
Description
Determine if supplying Cryptographic_Material to enable the required protection is achievable.
«activity»
determine_cryptographic_material
Description
Determine the appropriate Cryptographic_Material to be used to fulfil the protection requirement.
«interface»
Security_Group
Description
This interface is a statement of the required level of protection for a specific Security_Group.
«interface»
Achievement
Description
This interface is a statement of the progress towards the achievement of a requirement to provide a level of protection.
«requirement»
determine_crypto_material_usage
Description
To determine when and where particular Cryptographic_Material needs to be used.
«refine»
«requirement»
determine_material_plan
Description
To determine a Material_Plan that complies with the data Segregation_Policy (e.g. for specific security domains, types or classifications of data).
«refine»
«requirement»
capture_crypto_material_requirements
Description
To capture Requirements for the use of Cryptographic_Material.
«refine»
«requirement»
identify_material_solution_progress
Description
To identify what progress has been made against the Requirement.
«refine»

Figure 191: Protection Service Policy

Protection

This service captures the requirements for Cryptographic_Material to enable protection of information as defined in the Segregation_Policy.

Interfaces

Achievement

This interface is a statement of the progress towards the achievement of a requirement to provide a level of protection.

Security_Group

This interface is a statement of the required level of protection for a specific Security_Group.

Attributes

protection_type	The type of protection required, e.g. for data in transit or at rest.
protection_level	The level of protection needed to ensure confidentiality, integrity, availability, etc. or a combination of these.
data_location	The location (e.g. security domain) of the data to be cryptographically transformed.
temporal_information	Information covering timing for the requested protection, such as start and end times.

Activities

determine_protection_is_achievable

Determine if supplying Cryptographic_Material to enable the required protection is achievable.

determine_cryptographic_material

Determine the appropriate Cryptographic_Material to be used to fulfil the protection requirement.

5.4.2.10.7.1.2 Cryptographic_Material_Provision

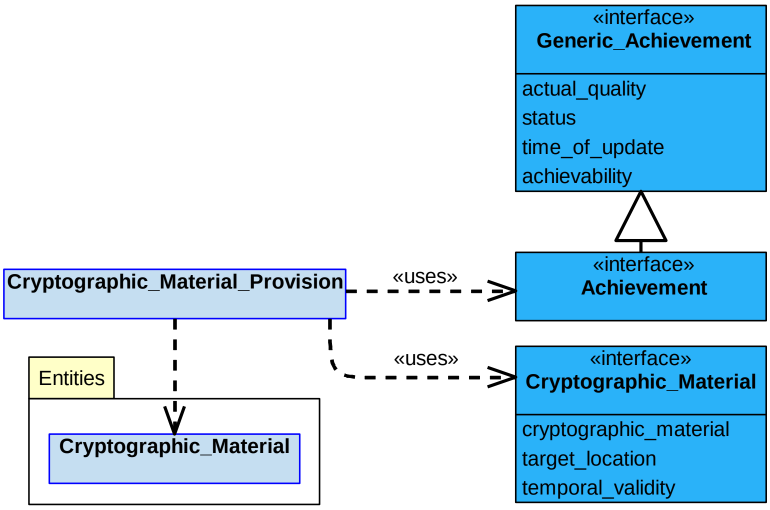
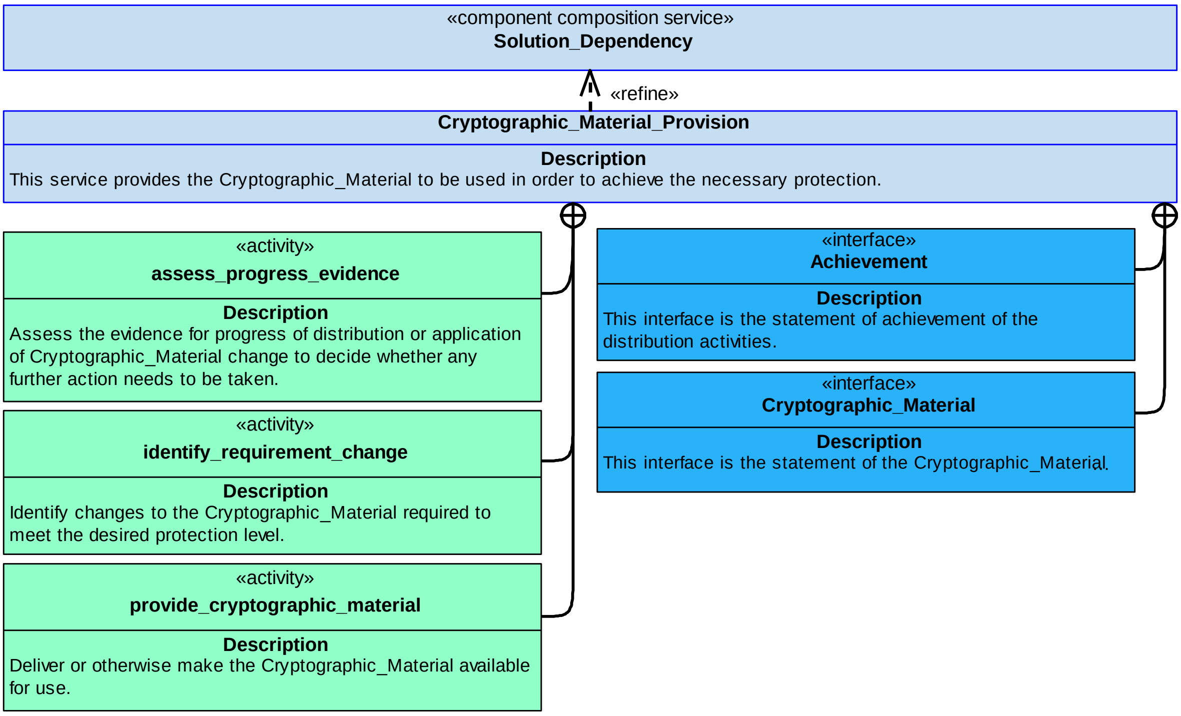


Figure 192: Cryptographic_Material_Provision Service Definition



This service provides the Cryptographic_Material to be used in order to achieve the necessary protection.

Assess the evidence for progress of distribution or application of Cryptographic_Material change to decide whether any further action needs to be taken.

Identify changes to the Cryptographic_Material required to meet the desired protection level.

Deliver or otherwise make the Cryptographic_Material available for use.

This interface is the statement of achievement of the distribution activities.

This interface is the statement of the Cryptographic_Material.

Figure 193: Cryptographic_Material_Provision Service Policy

Cryptographic_Material_Provision

This service provides the Cryptographic_Material to be used in order to achieve the necessary protection.

Interfaces

Cryptographic_Material

This interface is the statement of the Cryptographic_Material for distribution.

Attributes

cryptographic_material	The identified Cryptographic_Material.
target_location	The location where the Cryptographic_Material is needed (e.g. the Cryptographic_Device or Location).
temporal_validity	Information covering timing, such as when or how long the Cryptographic_Material is valid for use.

Achievement

This interface is the statement of achievement of the distribution activities.

Activities

assess_progress_evidence

Assess the evidence for progress of distribution or application of Cryptographic_Material change to decide whether any further action needs to be taken.

identify_requirement_change

Identify changes to the Cryptographic_Material required to meet the desired protection level.

provide_cryptographic_material

Deliver or otherwise make the Cryptographic_Material available for use.

5.4.2.10.7.1.3 Cryptographic_Material_Update

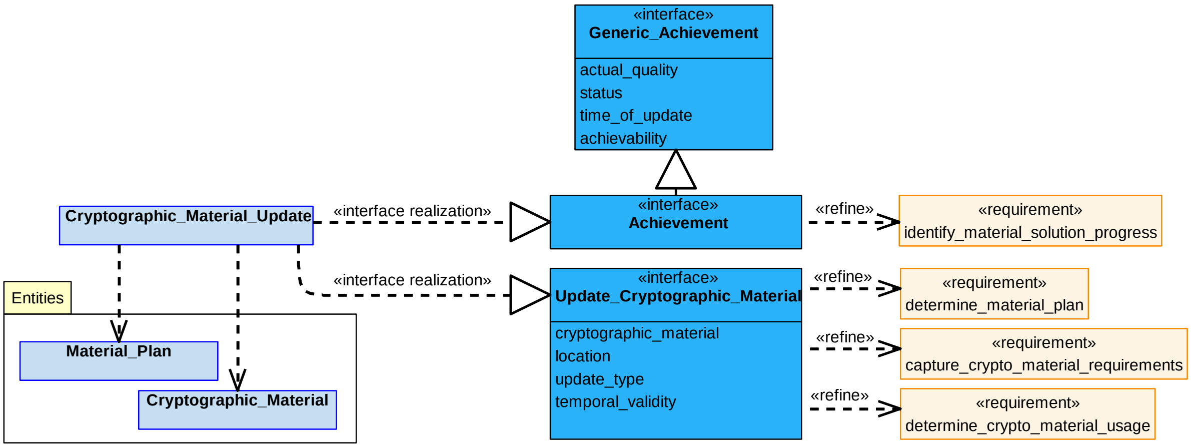


Figure 194: Cryptographic_Material_Update Service Definition

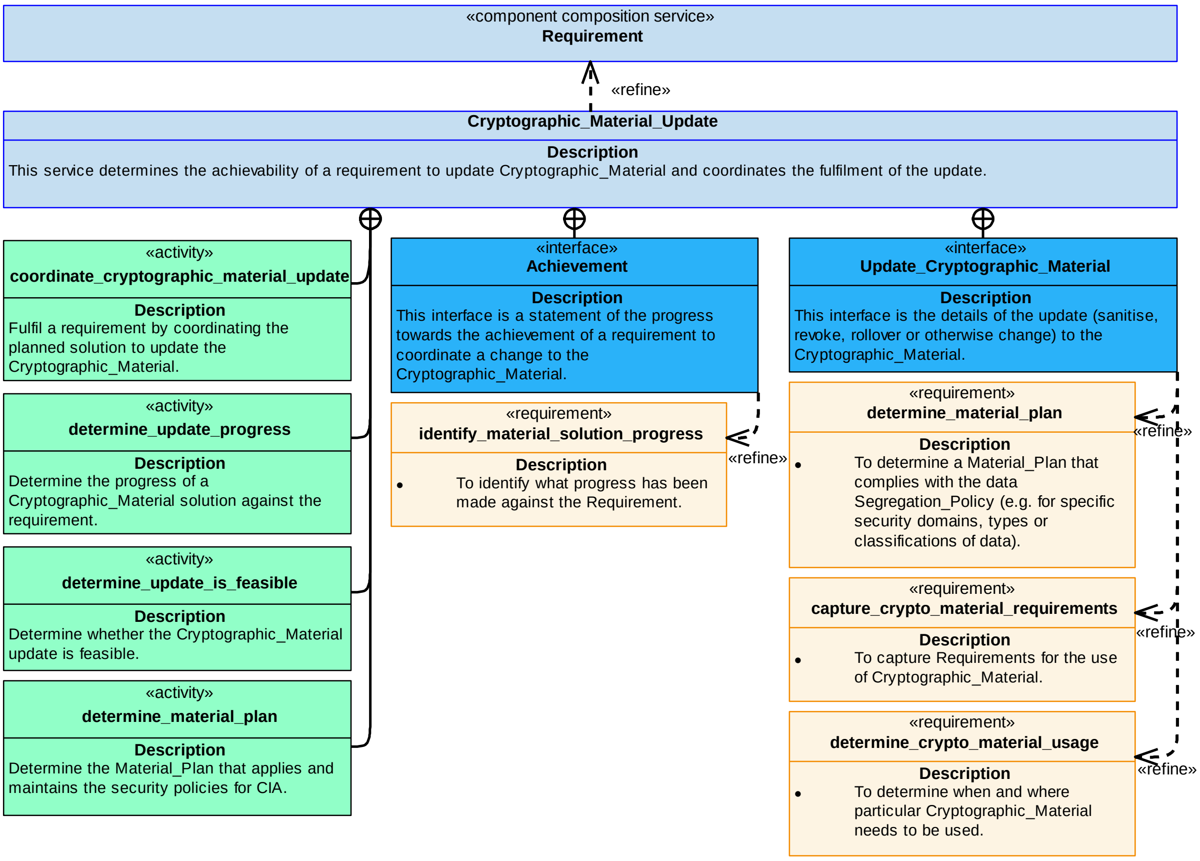

«component composition service»
Requirement
«refine»
Cryptographic_Material_Update
Description
This service determines the achievability of a requirement to update Cryptographic_Material and coordinates the fulfilment of the update.
«activity»
coordinate_cryptographic_material_update
Description
Fulfil a requirement by coordinating the planned solution to update the Cryptographic_Material.
«activity»
determine_update_progress
Description
Determine the progress of a Cryptographic_Material solution against the requirement.
«activity»
determine_update_is_feasible
Description
Determine whether the Cryptographic_Material update is feasible.
«activity»
determine_material_plan
Description
Determine the Material_Plan that applies and maintains the security policies for CIA.
«interface»
Achievement
Description
This interface is a statement of the progress towards the achievement of a requirement to coordinate a change to the Cryptographic_Material.
«requirement»
identify_material_solution_progress
Description
To identify what progress has been made against the Requirement.
«refine»
«interface»
Update_Cryptographic_Material
Description
This interface is the details of the update (sanitise, revoke, rollover or otherwise change) to the Cryptographic_Material.
«requirement»
determine_material_plan
Description
To determine a Material_Plan that complies with the data Segregation_Policy (e.g. for specific security domains, types or classifications of data).
«refine»
«requirement»
capture_crypto_material_requirements
Description
To capture Requirements for the use of Cryptographic_Material.
«refine»
«requirement»
determine_crypto_material_usage
Description
To determine when and where particular Cryptographic_Material needs to be used.
«refine»

Figure 195: Cryptographic_Material_Update Service Policy

Cryptographic_Material_Update

This service determines the achievability of a requirement to update Cryptographic_Material and coordinates the fulfilment of the update.

Interfaces

Update_Cryptographic_Material

This interface is the details of the update (sanitise, revoke, rollover or otherwise change) to the Cryptographic_Material.

Attributes

cryptographic_material	The Cryptographic_Material to be updated.
location	The location of the Cryptographic_Material to be updated.
update_type	Whether the update is to create, sanitise, revoke, rollover or otherwise change the Cryptographic_Material.
temporal_validity	Information covering the validity timing of the Cryptographic_Material, e.g. its start or expiration time.

Achievement

This interface is a statement of the progress towards the achievement of a requirement to coordinate a change to the Cryptographic_Material.

Activities

coordinate_cryptographic_material_update

Fulfil a requirement by coordinating the planned solution to update the Cryptographic_Material.

determine_update_is_feasible

Determine whether the Cryptographic_Material update is feasible.

determine_material_plan

Determine the Material_Plan that applies and maintains the security policies for CIA.

determine_update_progress

Determine the progress of a Cryptographic_Material solution against the requirement.

5.4.2.10.7.1.4 Capability

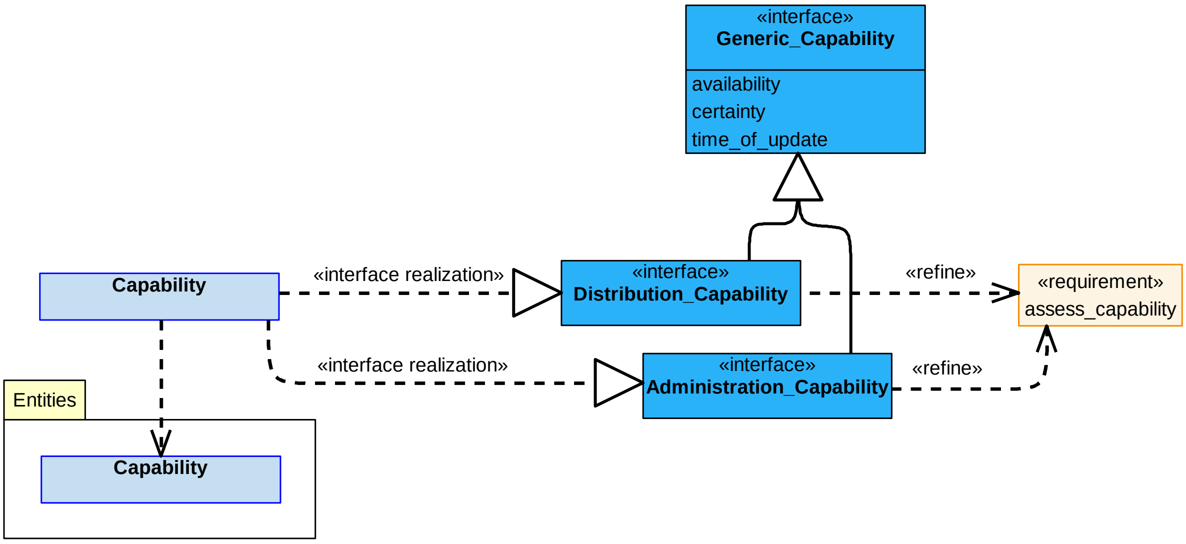


Figure 196: Capability Service Definition

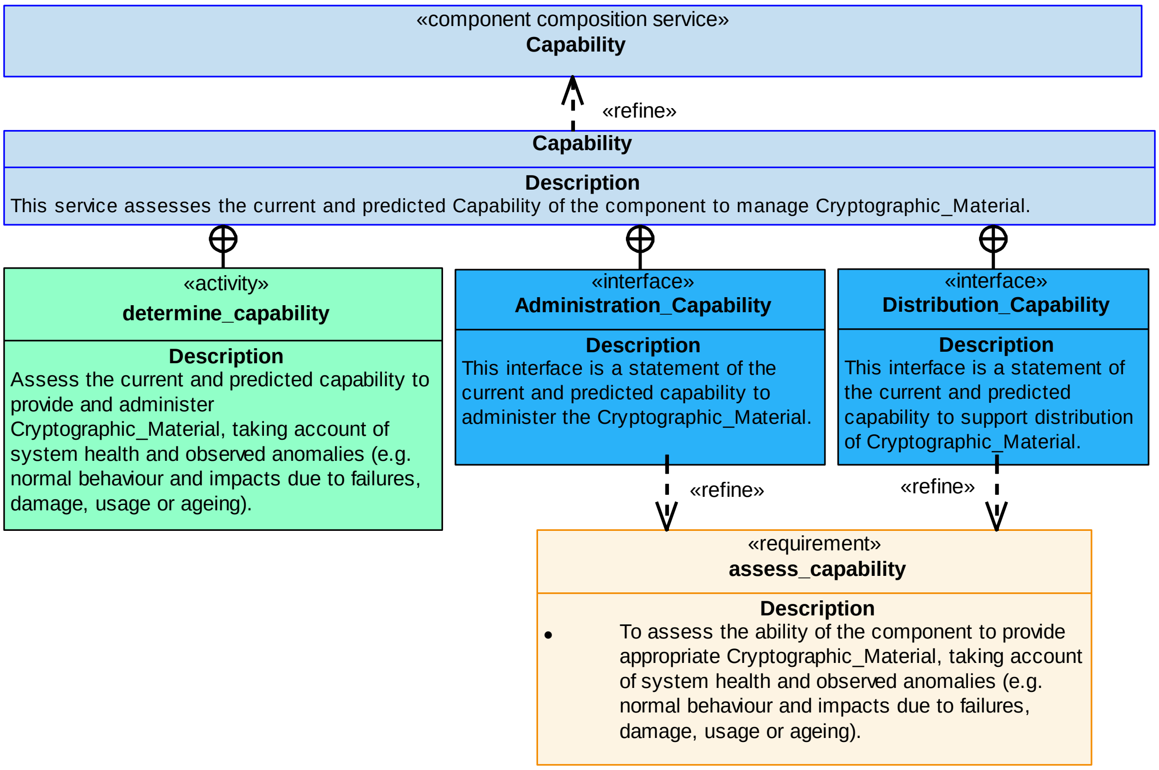

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability of the component to manage Cryptographic_Material.
«activity»
determine_capability
Description
Assess the current and predicted capability to provide and administer Cryptographic_Material, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Administration_Capability
Description
This interface is a statement of the current and predicted capability to administer the Cryptographic_Material.
«interface»
Distribution_Capability
Description
This interface is a statement of the current and predicted capability to support distribution of Cryptographic_Material.
«refine»
«refine»
«requirement»
assess_capability
Description
To assess the ability of the component to provide appropriate Cryptographic_Material, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Figure 197: Capability Service Policy

Capability

This service assesses the current and predicted Capability of the component to manage Cryptographic_Material.

Interfaces

Administration_Capability

This interface is a statement of the current and predicted capability to administer the Cryptographic_Material.

Distribution_Capability

This interface is a statement of the current and predicted capability to support distribution of Cryptographic_Material.

Activity

determine_capability

Assess the current and predicted capability to provide and administer Cryptographic_Material, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.10.7.1.5 Capability_Evidence

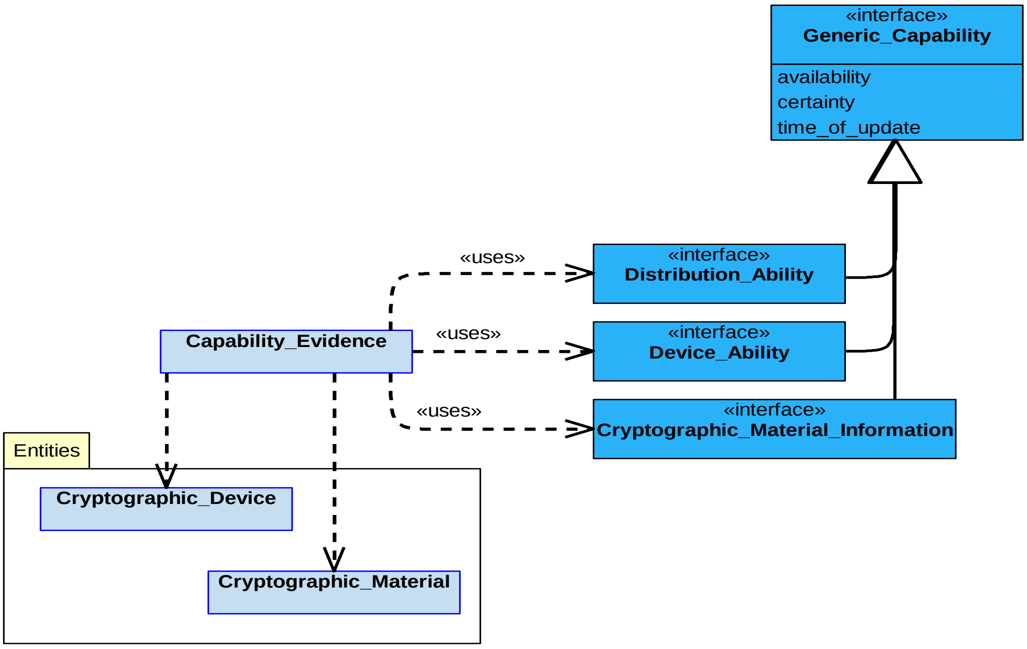
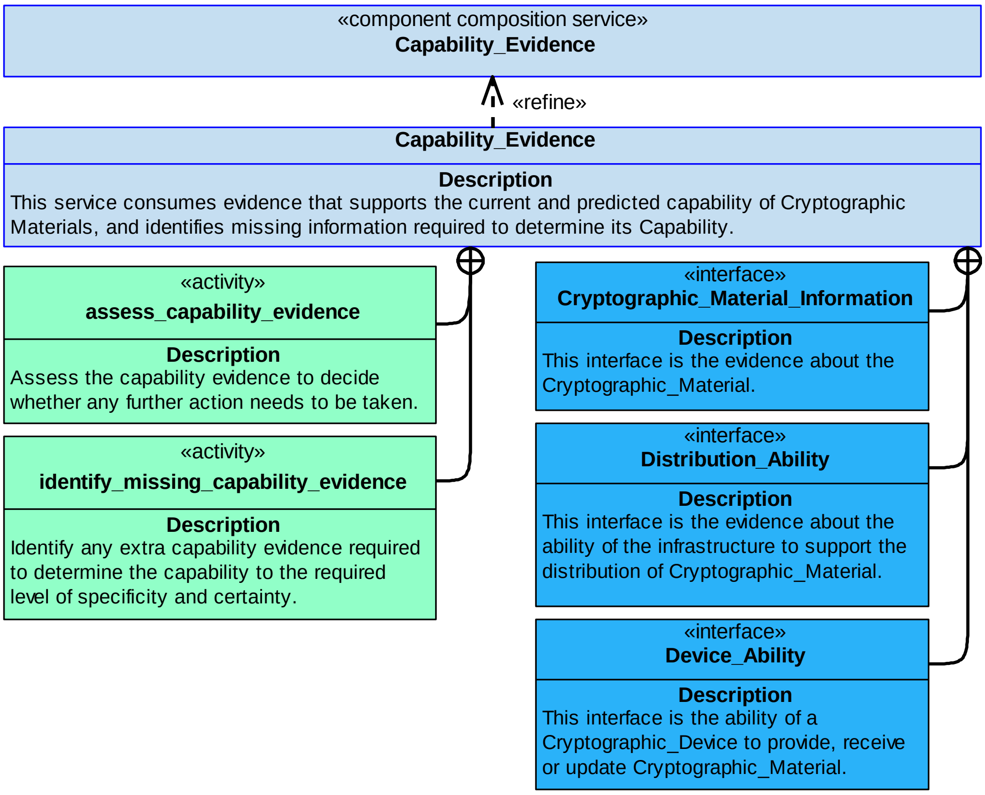


Figure 198: Capability_Evidence Service Definition



This service consumes evidence that supports the current and predicted capability of Cryptographic Materials, and identifies missing information required to determine its Capability.

Assess the capability evidence to decide whether any further action needs to be taken.

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

This interface is the evidence about the Cryptographic_Material.

This interface is the evidence about the ability of the infrastructure to support the distribution of Cryptographic_Material.

This interface is the ability of a Cryptographic_Device to provide, receive or update Cryptographic_Material.

Figure 199: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence that supports the current and predicted capability of Cryptographic Materials, and identifies missing information required to determine its Capability.

Interfaces

Device_Ability

This interface is the ability of a Cryptographic_Device to provide, receive or update Cryptographic_Material.

Distribution_Ability

This interface is the evidence about the ability of the infrastructure to support the distribution of Cryptographic_Material.

Cryptographic_Material_Information

This interface is the evidence about the Cryptographic_Material.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.10.7.2 Service Dependencies

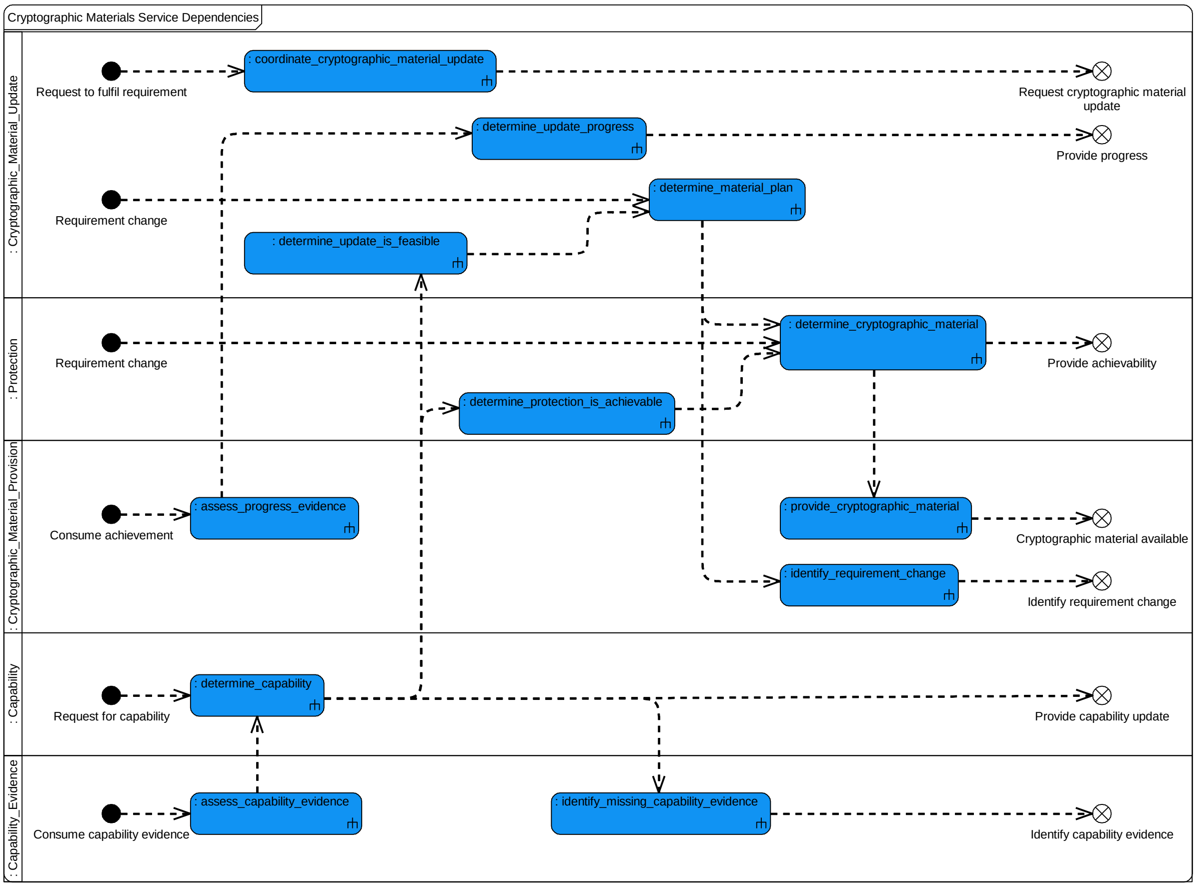

Cryptographic Materials Service Dependencies
: Cryptographic_Material_Update
: Protection
: Cryptographic_Material_Provision
: Capability
: Capability_Evidence
Request to fulfil requirement
: coordinate_cryptographic_material_update
Request cryptographic material update
: determine_update_progress
Provide progress
Requirement change
: determine_material_plan
: determine_update_is_feasible
Requirement change
: determine_cryptographic_material
Provide achievability
: determine_protection_is_achievable
Consume achievement
: assess_progress_evidence
: provide_cryptographic_material
Cryptographic material available
: identify_requirement_change
Identify requirement change
Request for capability
: determine_capability
Provide capability update
Consume capability evidence
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 200: Cryptographic Materials Service Dependencies

5.4.2.11 Cryptographic Methods

5.4.2.11.1 Role

The role of Cryptographic Methods is to perform cryptographic transformations.

5.4.2.11.2 Overview

Control Architecture

Cryptographic Methods is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

A Requirement will trigger Cryptographic Methods to perform a Cryptographic_Action using the provided or pre-loaded Cryptographic_Materials.

Examples of Use

Cryptographic Methods is used when data is required to be cryptographically transformed, examples of this can include:

- **Link Encryption** - Data-in-transit on a single hop (point-to-point).
- **Traffic Encryption** - Data-in-transit for an end-to-end link (e.g. PRIME IPSec).
- **Secure Data at Rest** - Disk encryption and file encryption.
- **Payload Encryption** - Encryption of part of a message for data-in-transit (e.g. MIKEY-SAKKE or TLS).
- **Analogue Encryption** - Frequency based encryption of a radio signal over the air (scramble).

5.4.2.11.3 Service Summary

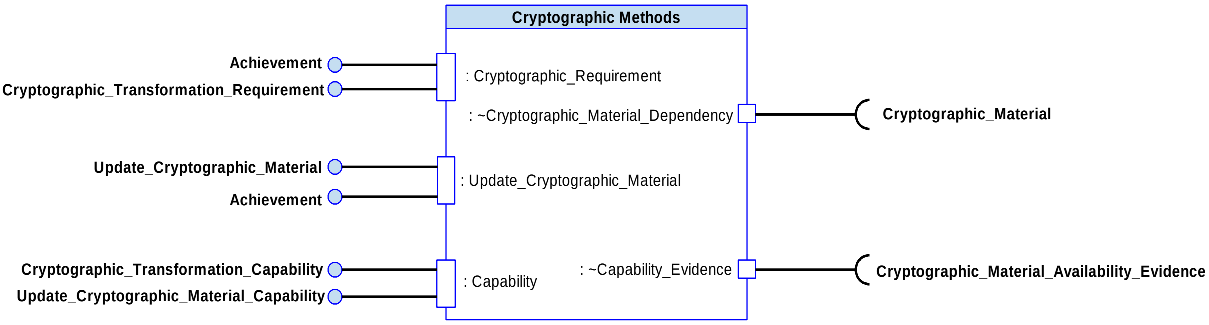

Cryptographic Methods
Achievement
Cryptographic_Transformation_Requirement
: Cryptographic_Requirement
: ~Cryptographic_Material_Dependency
Cryptographic_Material
Update_Cryptographic_Material
Achievement
: Update_Cryptographic_Material
Cryptographic_Transformation_Capability
Update_Cryptographic_Material_Capability
: Capability
: ~Capability_Evidence
Cryptographic_Material_Availability_Evidence

Figure 201: Cryptographic Methods Service Summary

5.4.2.11.4 Responsibilities

capture_cryptographic_requirement

- To capture Requirements for a Cryptographic_Function (e.g. encryption, decryption, or hashing).

determine_cryptographic_material_for_use

- To determine the Cryptographic_Material to be used for any particular Cryptographic_Action.

identify_if_cryptographic_transformation_solution_remains_feasible

- To identify if a cryptographic transformation in progress remains feasible given current resources.

determine_cryptographic_state

- To determine the current state of the cryptography, e.g. encryption is available, complete, or failed.

encrypt_data

- To encrypt data.

decrypt_data

- To decrypt data.

provide_hashing_function

- To hash data.

capture_cryptographic_material

- To capture provided Cryptographic_Material.

assess_cryptographic_capability

- To assess the Capability of the component taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

sanitise_cryptographic_material

- To sanitise provided Cryptographic_Material.

5.4.2.11.5 Subject Matter Semantics

The subject matter of Cryptographic Methods is the use of Cryptographic_Material to perform Cryptographic_Functions such as encryption, decryption and hashing.

Exclusions

The subject matter of Cryptographic Methods does not include:

- Cryptanalysis (breaking of encrypted data).
- The management and distribution of Cryptographic_Material, only its use.
- Why cryptography is required.
- The security implications presented by threats to encrypted data.

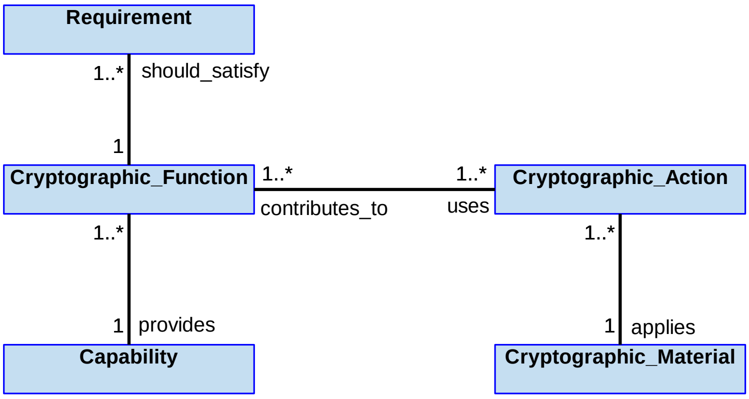


Figure 202: Cryptographic Methods Semantics

5.4.2.11.5.1 Entities

Capability

The capability of the component to perform cryptographic transformations.

Cryptographic_Action

A discrete cryptographic step. This could be either the steps in a discrete cryptographic delivery (e.g. individual steps to encrypt a discrete message) or the steps in a continuous cryptographic process (e.g. apply the encryption to each discrete message, in turn).

Cryptographic_Function

A cryptographic process. The processes have a crypto usage state, e.g. encryption is available, crypto channel open, and decryption complete.

Cryptographic_Material

The cryptographic material or material set, e.g. keys, algorithms, or certificates.

Requirement

A requirement to perform a cryptographic transformation.

5.4.2.11.6 Design Rationale

5.4.2.11.6.1 Assumptions

- Cryptographic Methods will be used to cryptographically protect confidentiality and integrity when data is at rest and during transit, including when crossing security domain boundaries.
- The Cryptographic_Material utilised may be received, or generated by this component as appropriate to its use.

5.4.2.11.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Cryptographic Methods:

- Cyber Defence - This component is involved with cyber defence activities.
- Use of Communications - Communications are expected to be encrypted and decrypted.
- Recording and Logging - This PYRAMID concept will carry out logging of cryptography events.

Extensions

- Different cryptographic methods can be accommodated by extensions.

5.4.2.11.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

Failure of this component could lead to:

- Loss of availability of transfers of encrypted and/or hashed data. For example, communications between a ground-based control station and the air vehicle, which is primarily a concern for UAVs, but may apply to manned air vehicles where some functions are controlled by crew external to the air vehicle. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) then the air vehicle will have been designed to mitigate a loss of communications. For UAS this would by relying on pre-determined automatic or autonomous behaviour. For this failure mode it is concluded that failure of this component may result a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.
- Failure to detect corruption of data transfers. The transfers would include those between safety critical software items within the air vehicle, between a ground based control station and the air vehicle and from external systems. In the worst case, the air system may erroneously perform an activity with catastrophic consequences (e.g. unintended weapon release). The data protections applied by this component can enable the transfer of safety critical data by non-safety critical systems, such as by making data corruption/manipulation identifiable when transferred in the external environment. As some commands may be simple, no credit is taken for the corruption resulting in data not considered "believable" by the receiving component. Therefore, the indicative DAL is conservatively assessed as DAL A.

Failures of encryption resulting in compromise of sensitive data or allowing control of the air vehicle by unauthorised users is covered by the Cyber Defence PYRAMID concept.

5.4.2.11.6.4 Security Considerations

The indicative security classification is O-S, however the data that it is used to protect will be a significant factor.

This component is central to protecting the confidentiality, integrity and authenticity of system data, both at rest and when crossing security domain boundaries; it is responsible for the cryptographic transformation of data appropriate to the requirements of that data. Cryptography may be required for all classifications of data, therefore there may be instances of this component in different security domains, potentially using different algorithms, etc. It is not expected that these instances will need to communicate with each other.

Additional protection may be required due to the nature of the component and its role in the security of the Exploiting Platform and its data, this component may be segregated from other components.

This component provides security related functions through:

- **Logging of Security Data** relating to cryptographic events.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may need to be protected to assure continued airworthiness.
- **System Status and Monitoring** of cryptography capability. Loss of this capability will undermine the security of the Exploiting Platform and therefore its operational advantage.

It fundamentally implements security enforcing functions by:

- **Encrypting Data** (including data hashing) as its primary task.
- Protecting the confidentiality, integrity and authenticity of encrypted system data, therefore **Preventing Cyber Attacks and Malware**.
- **Securing Communications** through encryption of data prior to being communicated.
- **Verifying Integrity of Data,** providing hashing functions that confirm data is accurate and complete.

5.4.2.11.7 Services

5.4.2.11.7.1 Service Definitions

5.4.2.11.7.1.1 Cryptographic_Requirement

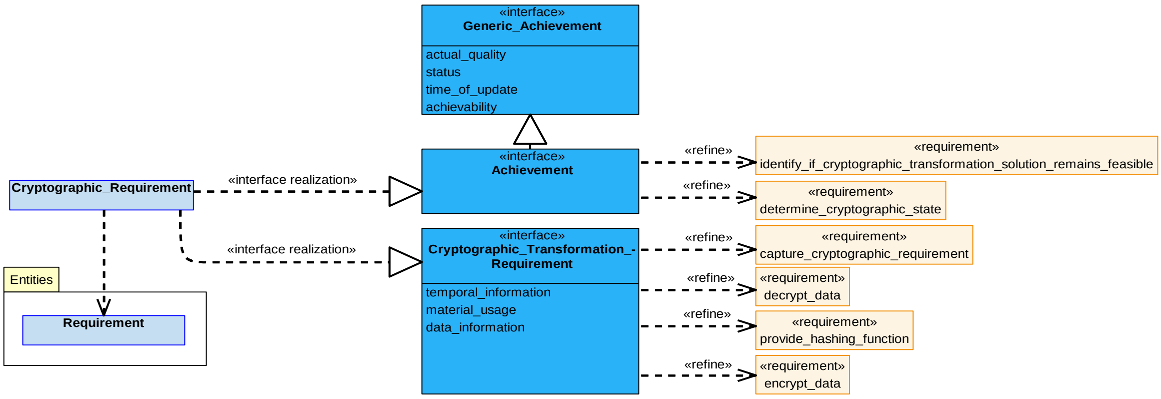


Figure 203: Cryptographic_Requirement Service Definition

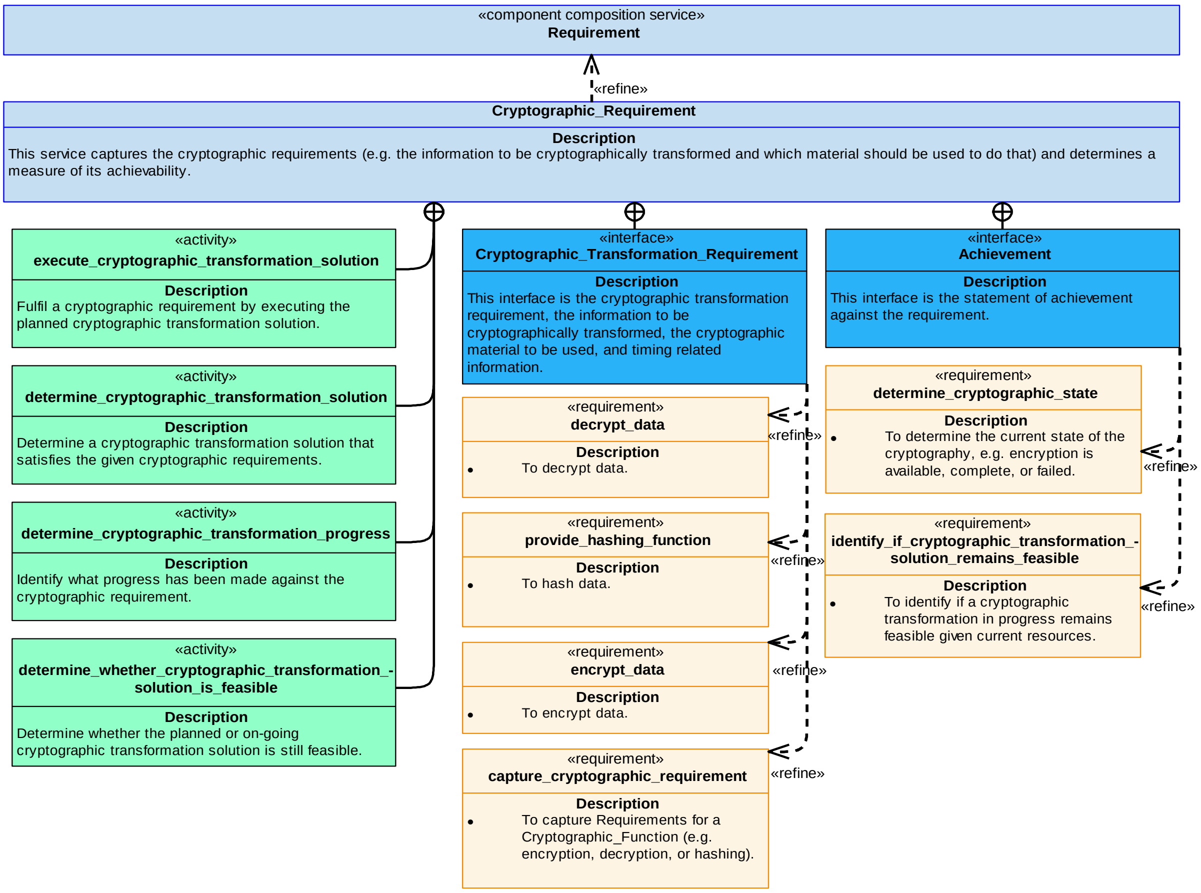

«component composition service»
Requirement
«refine»
Cryptographic_Requirement
Description
This service captures the cryptographic requirements (e.g. the information to be cryptographically transformed and which material should be used to do that) and determines a measure of its achievability.
«activity»
execute_cryptographic_transformation_solution
Description
Fulfil a cryptographic requirement by executing the planned cryptographic transformation solution.
«activity»
determine_cryptographic_transformation_solution
Description
Determine a cryptographic transformation solution that satisfies the given cryptographic requirements.
«activity»
determine_cryptographic_transformation_progress
Description
Identify what progress has been made against the cryptographic requirement.
«activity»
determine_whether_cryptographic_transformation_-solution_is_feasible
Description
Determine whether the planned or on-going cryptographic transformation solution is still feasible.
«interface»
Cryptographic_Transformation_Requirement
Description
This interface is the cryptographic transformation requirement, the information to be cryptographically transformed, the cryptographic material to be used, and timing related information.
«requirement»
decrypt_data
Description
To decrypt data.
«refine»
«requirement»
provide_hashing_function
Description
To hash data.
«refine»
«requirement»
encrypt_data
Description
To encrypt data.
«refine»
«requirement»
capture_cryptographic_requirement
Description
To capture Requirements for a Cryptographic_Function (e.g. encryption, decryption, or hashing).
«refine»
«interface»
Achievement
Description
This interface is the statement of achievement against the requirement.
«requirement»
determine_cryptographic_state
Description
To determine the current state of the cryptography, e.g. encryption is available, complete, or failed.
«refine»
«requirement»
identify_if_cryptographic_transformation_-solution_remains_feasible
Description
To identify if a cryptographic transformation in progress remains feasible given current resources.
«refine»

Figure 204: Cryptographic_Requirement Service Policy

Cryptographic_Requirement

This service captures the cryptographic requirements (e.g. the information to be cryptographically transformed and which material should be used to do that) and determines a measure of its achievability.

Interfaces

Cryptographic_Transformation_Requirement

This interface is the cryptographic transformation requirement, the information to be cryptographically transformed, the cryptographic material to be used, and timing related information.

Attributes

temporal_information	Timing information about the requirement, such as time to complete or time to start and finish.
material_usage	The usage of the cryptographic transformation, e.g. the type of transformation to be performed for data-in-transit or for data-at-rest.
data_information	The information which is required to be cryptographically transformed.

Achievement

This interface is the statement of achievement against the requirement.

Activities

determine_cryptographic_transformation_solution

Determine a cryptographic transformation solution that satisfies the given cryptographic requirements.

determine_whether_cryptographic_transformation_solution_is_feasible

Determine whether the planned or on-going cryptographic transformation solution is still feasible.

execute_cryptographic_transformation_solution

Fulfil a cryptographic requirement by executing the planned cryptographic transformation solution.

determine_cryptographic_transformation_progress

Identify what progress has been made against the cryptographic requirement.

5.4.2.11.7.1.2 Update_Cryptographic_Material

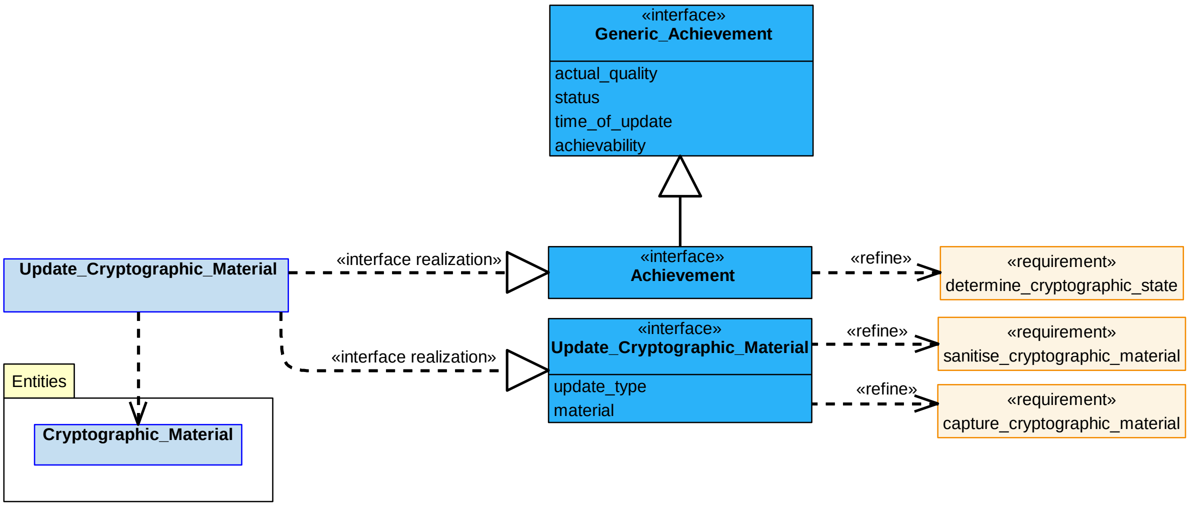


Figure 205: Update_Cryptographic_Material Service Definition

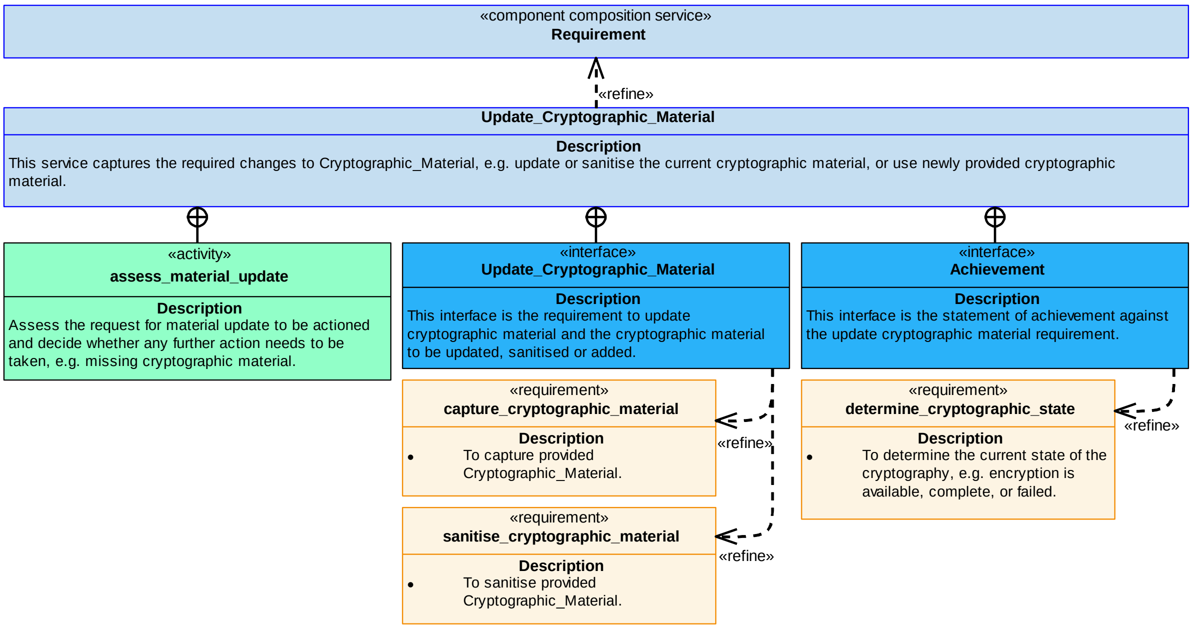

«component composition service»
Requirement
«refine»
Update_Cryptographic_Material
Description
This service captures the required changes to Cryptographic_Material, e.g. update or sanitise the current cryptographic material, or use newly provided cryptographic material.
«activity»
assess_material_update
Description
Assess the request for material update to be actioned and decide whether any further action needs to be taken, e.g. missing cryptographic material.
«interface»
Update_Cryptographic_Material
Description
This interface is the requirement to update cryptographic material and the cryptographic material to be updated, sanitised or added.
«interface»
Achievement
Description
This interface is the statement of achievement against the update cryptographic material requirement.
«requirement»
capture_cryptographic_material
Description
To capture provided Cryptographic_Material.
«refine»
«requirement»
sanitise_cryptographic_material
Description
To sanitise provided Cryptographic_Material.
«refine»
«requirement»
determine_cryptographic_state
Description
To determine the current state of the cryptography, e.g. encryption is available, complete, or failed.
«refine»

Figure 206: Update_Cryptographic_Material Service Policy

Update_Cryptographic_Material

This service captures the required changes to Cryptographic_Material, e.g. update or sanitise the current cryptographic material, or use newly provided cryptographic material.

Interfaces

Update_Cryptographic_Material

This interface is the requirement to update cryptographic material and the cryptographic material to be updated, sanitised or added.

Attributes

update_type	The update type of the cryptographic material, e.g. update, delete, sanitise cryptographic material.
material	The cryptographic material to be updated, e.g. key set, or device certificate.

Achievement

This interface is the statement of achievement against the update cryptographic material requirement.

Activity

assess_material_update

Assess the request for material update to be actioned and decide whether any further action needs to be taken, e.g. missing cryptographic material.

5.4.2.11.7.1.3 Cryptographic_Material_Dependency

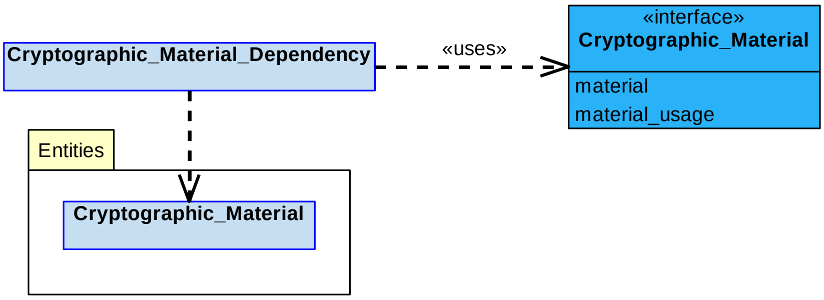


Figure 207: Cryptographic_Material_Dependency Service Definition

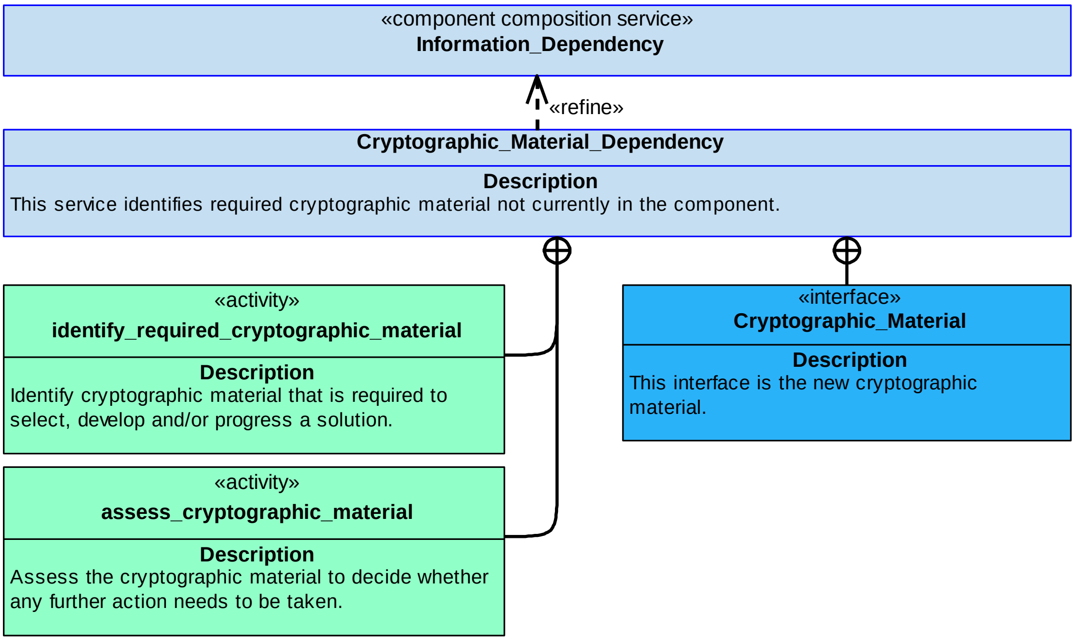

«component composition service»
Information_Dependency
«refine»
Cryptographic_Material_Dependency
Description
This service identifies required cryptographic material not currently in the component.
«activity»
identify_required_cryptographic_material
Description
Identify cryptographic material that is required to select, develop and/or progress a solution.
«activity»
assess_cryptographic_material
Description
Assess the cryptographic material to decide whether any further action needs to be taken.
«interface»
Cryptographic_Material
Description
This interface is the new cryptographic material.

Figure 208: Cryptographic_Material_Dependency Service Policy

Cryptographic_Material_Dependency

This service identifies required cryptographic material not currently in the component.

Interface

Cryptographic_Material

This interface is the new cryptographic material.

Attributes

material	The new cryptographic material.
material_usage	The intended usage of the cryptographic material, e.g. destination for data-in-transit or store id for data-at-rest.

Activities

assess_cryptographic_material

Assess the cryptographic material to decide whether any further action needs to be taken.

identify_required_cryptographic_material

Identify cryptographic material that is required to select, develop and/or progress a solution.

5.4.2.11.7.1.4 Capability

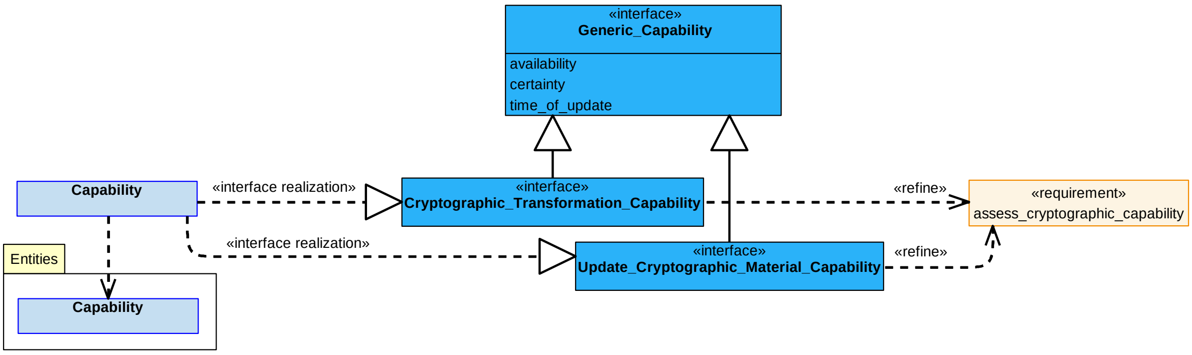
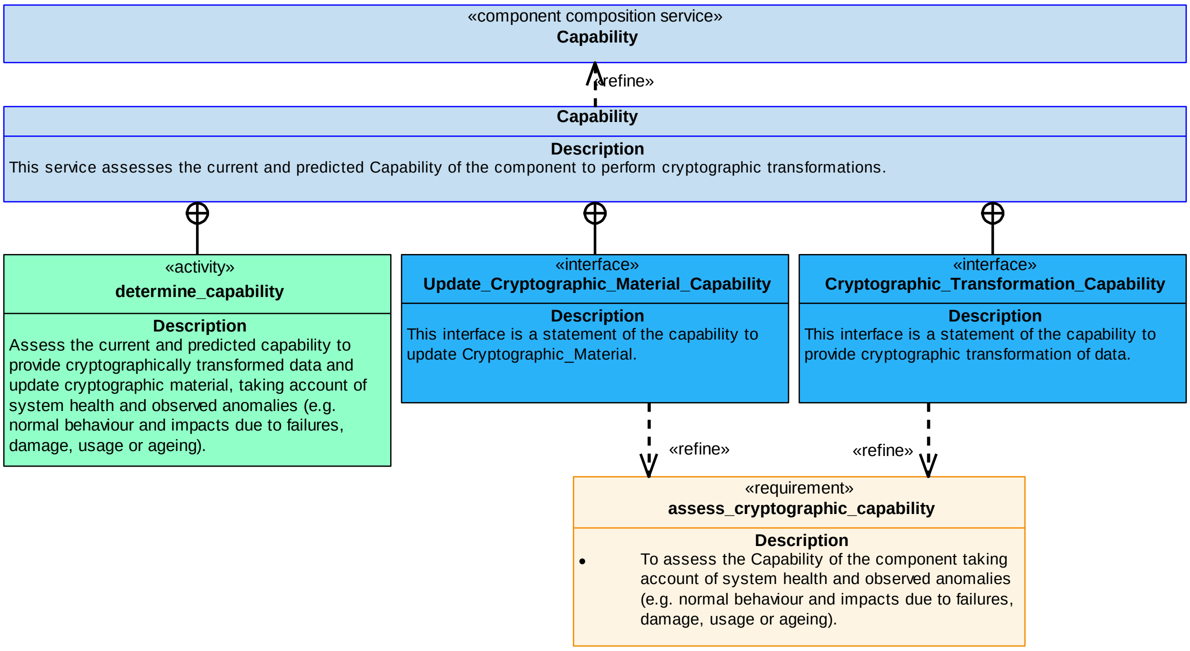


Figure 209: Capability Service Definition


«component composition service»
Capability
«refine»
Capability
«activity»
determine_capability
«interface»
Update_Cryptographic_Material_Capability
«interface»
Cryptographic_Transformation_Capability
«refine»
«refine»
«requirement»
assess_cryptographic_capability

Description
This service assesses the current and predicted Capability of the component to perform cryptographic transformations.

Description
Assess the current and predicted capability to provide cryptographically transformed data and update cryptographic material, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Description
This interface is a statement of the capability to update Cryptographic_Material.

Description
This interface is a statement of the capability to provide cryptographic transformation of data.

Description
- To assess the Capability of the component taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Figure 210: Capability Service Policy

Capability

This service assesses the current and predicted Capability of the component to perform cryptographic transformations.

Interfaces

Cryptographic_Transformation_Capability

This interface is a statement of the capability to provide cryptographic transformation of data.

Update_Cryptographic_Material_Capability

This interface is a statement of the capability to update Cryptographic_Material.

Activity

determine_capability

Assess the current and predicted capability to provide cryptographically transformed data and update cryptographic material, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.11.7.1.5 Capability_Evidence

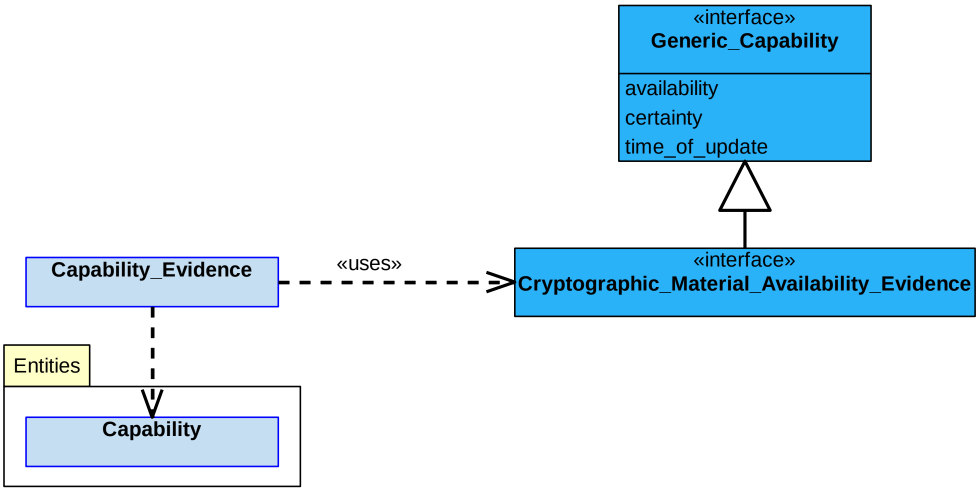


Figure 211: Capability_Evidence Service Definition

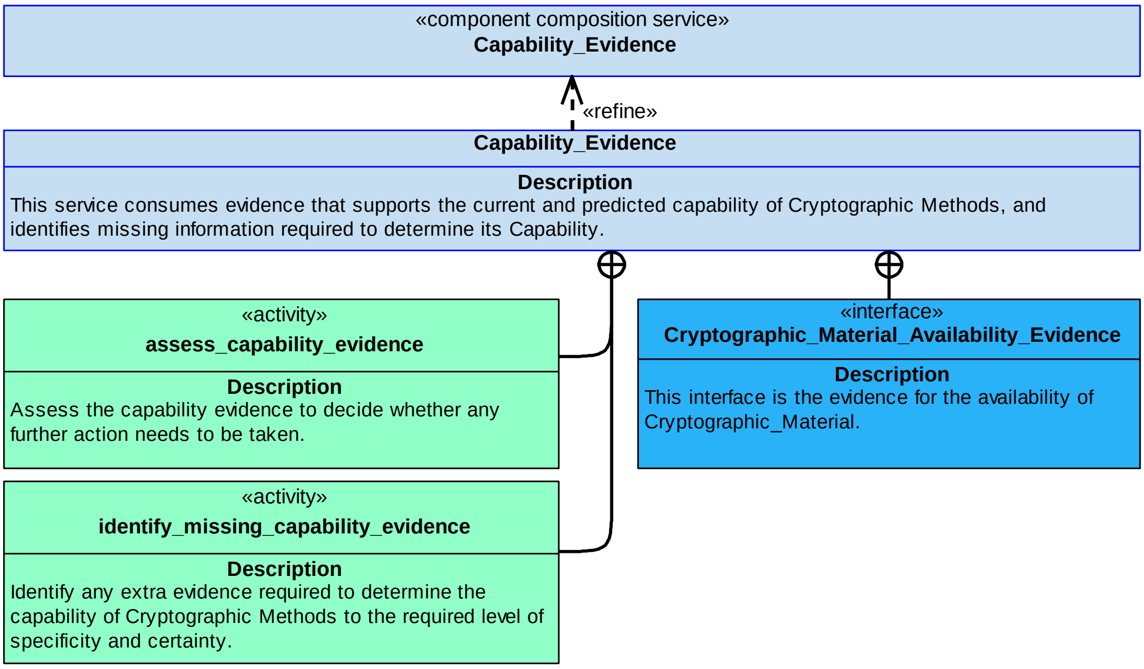

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes evidence that supports the current and predicted capability of Cryptographic Methods, and identifies missing information required to determine its Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«interface»
Cryptographic_Material_Availability_Evidence
Description
This interface is the evidence for the availability of Cryptographic_Material.
«activity»
identify_missing_capability_evidence
Description
Identify any extra evidence required to determine the capability of Cryptographic Methods to the required level of specificity and certainty.

Figure 212: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence that supports the current and predicted capability of Cryptographic Methods, and identifies missing information required to determine its Capability.

<u>Interface</u>

Cryptographic_Material_Availability_Evidence

This interface is the evidence for the availability of Cryptographic_Material.

<u>Activities</u>

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra evidence required to determine the capability of Cryptographic Methods to the required level of specificity and certainty.

5.4.2.11.7.2 Service Dependencies

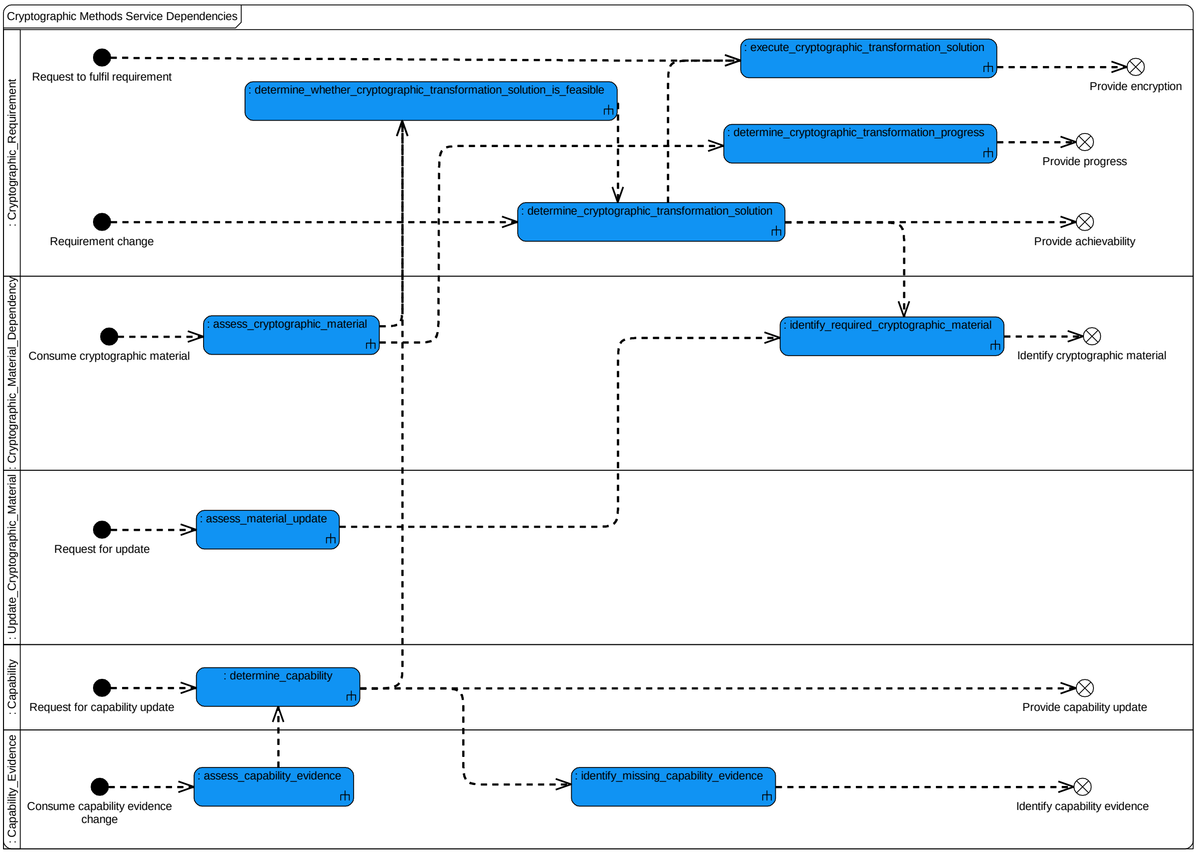

Cryptographic Methods Service Dependencies
: Cryptographic_Requirement
: Cryptographic_Material_Dependency
: Update_Cryptographic_Material
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_cryptographic_transformation_solution
Provide encryption
: determine_whether_cryptographic_transformation_solution_is_feasible
: determine_cryptographic_transformation_progress
Provide progress
Requirement change
: determine_cryptographic_transformation_solution
Provide achievability
Consume cryptographic material
: assess_cryptographic_material
: identify_required_cryptographic_material
Identify cryptographic material
Request for update
: assess_material_update
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 213: Cryptographic Methods Service Dependencies

5.4.2.12 Cyber Defence

5.4.2.12.1 Role

The role of Cyber Defence is to identify when system elements have been affected by a suspected cyber attack and to determine how to respond to a suspected cyber attack.

5.4.2.12.2 Overview

Control Architecture

Cyber Defence is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

With an anomaly detected, Cyber Defence determines if the anomaly may be the result of a cyber attack. If a cyber attack is suspected, Cyber Defence determines the System_Elements that are likely to be affected and identifies possible Responses to counter the effects of the attack.

Examples of Use

Cyber Defence will be used where a system may be vulnerable to cyber attacks and:

- It is necessary to determine if anomalous behaviour may be the result of a cyber attack.
- It is necessary to minimise the effects of a cyber attack by actions of the system.

5.4.2.12.3 Service Summary

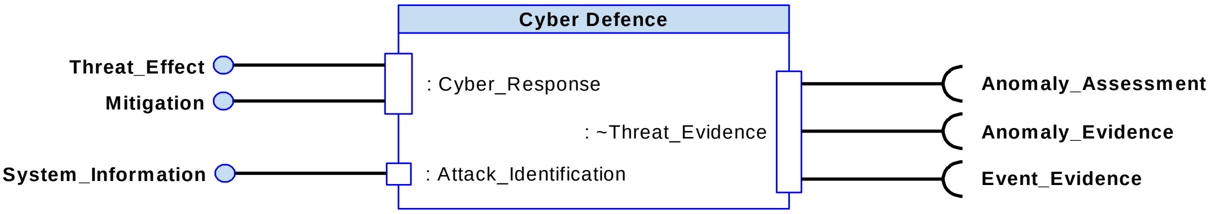


Figure 214: Cyber Defence Service Summary

5.4.2.12.4 Responsibilities

determine_possible_actions

- To identify possible actions to counteract a suspected cyber attack.

determine_anomaly_cause

- To determine that anomalous system behaviour may be the result of a cyber attack.

identify_affected_system_elements

- To identify System_Elements that have been affected by a suspected cyber attack.

predict_cyber_attack_progression

- To predict the progression of a cyber attack through the system (i.e. the expected sequence in which System_Elements are likely to be affected).

determine_quality_of_identification

- To determine the quality of a cyber attack determination, against given Measurement_Criterion/criteria.

determine_quality_of_response

- To determine the quality of a Response to a cyber attack, against given Measurement_Criterion/criteria.

identify_additional_evidence_to_improve_identification

- To identify additional Evidence that could improve the certainty or specificity of a cyber attack determination.

5.4.2.12.5 Subject Matter Semantics

The subject matter of Cyber Defence is cyber attacks to which a system may be vulnerable.

Exclusions

The subject matter of Cyber Defence does not include:

- The implementation details of any mitigating actions.

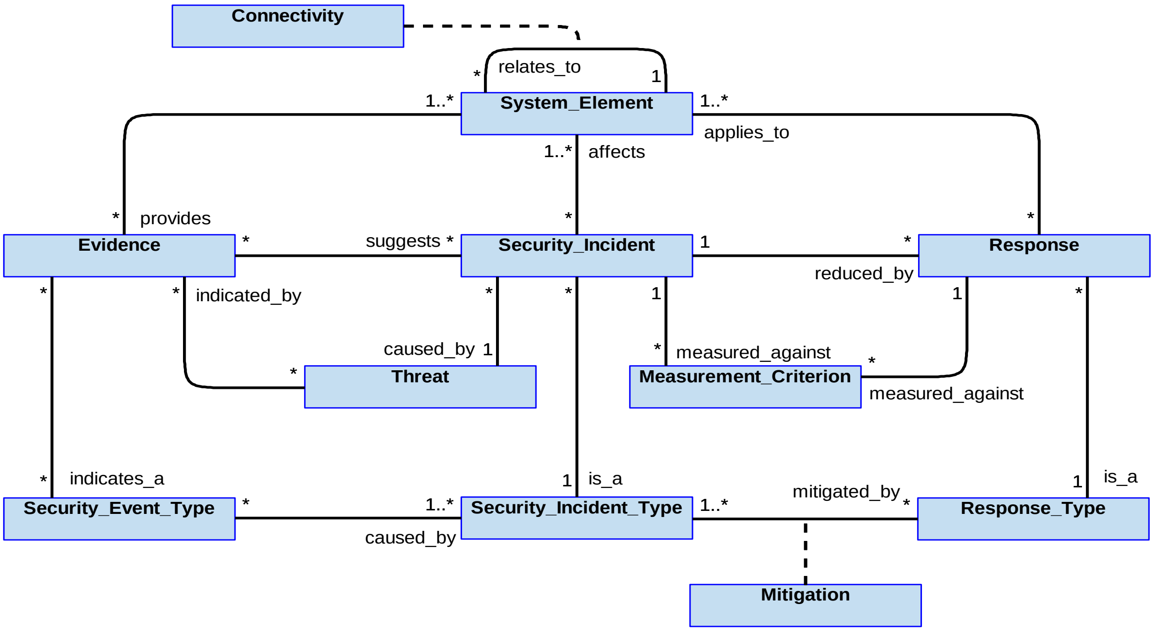


Figure 215: Cyber Defence Semantics

5.4.2.12.5.1 Entities

Connectivity

The type of relationship between System_Elements. This can impact how behaviour exhibited by a number of system elements may, when considered together, indicate anomalous behaviour or a security event. It can also indicate how a cyber attack may propagate through a system.

Evidence

Information about anomalous system behaviour or a security event.

Measurement_Criterion

A criterion that the quality of an assessment will be measured against; e.g. confidence with which a cyber attack has been determined, or confidence of effectiveness of Response.

Mitigation

A Response_Type appropriate to a Security_Incident_Type.

Response

A possible action that may reduce the effectiveness of a cyber attack, or may limit the effects of the attack from promulgating through the system.

Response_Type

A kind of action that may reduce the effectiveness of a cyber attack, or may limit the effects of the attack from promulgating through the system (e.g. re-routing traffic or quarantining a system element).

Security_Event_Type

A type of occurrence that may point towards a cyber attack (e.g. escalation of user privileges or increased network traffic).

Security_Incident

A specific security incident for assessment of whether the system is the subject of a cyber attack.

Security_Incident_Type

The kind of security incident that a system may be subject to. For example, loss of confidentiality or denial of service.

System_Element

Part of the system that either provides evidence of anomalous behaviour, or is affected or likely to be affected by a cyber attack.

Threat

A known threat (vulnerability, attack vector, exploit, etc.) that might affect the system.

5.4.2.12.6 Design Rationale

5.4.2.12.6.1 Assumptions

- Cyber Defence will be at the core of any Security Information & Event Management (SIEM) solution, able to compare events from multiple components that may not be considered a fault by themselves (e.g. an elevation of user privilege combined with an increase in data being transferred to that user).
- Cyber Defence will interpret software integrity failures, viruses and other attacks directed at the execution platform (i.e. IT and information processing infrastructure).

5.4.2.12.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Cyber Defence:

- Data Driving - This PYRAMID concept is applicable as:
    - The known threats (vulnerabilities, exploits, signatures, attack vectors, malware, viruses, etc.) to the system used to assess whether anomalies represent a possible cyber attack should be data-driven to allow them to be kept up-to-date.
    - This component is likely to need to be highly specific to the software and middleware being assessed. It should know the functions, vulnerabilities and failure modes of each software element, and how the elements are related. Data driving of the component with this information should be considered.
- Capability Management - Cyber Defence does not provide an evolving view of its own capabilities but it supports the capability assessment of other components.

Exploitation Considerations

- The possible actions that Cyber Defence may identify to counteract a security incident will be highly dependent on the requirements of the Exploiting Programme, and it will be up to the Exploiting Programme to determine, e.g. suggesting quarantining untrusted data.
- Cyber Defence will need to know under what conditions the system should identify a suspected security incident, e.g. the confidence threshold. This will depend on the requirements of the Exploiting Programme.
- Cyber Defence may adjust the conditions under which a suspected security incident is identified, so attacks are not repeatedly reported erroneously. For example, if failure of equipment results in increased processor time, the processor time threshold above which a suspected security incident may be identified, should be increased. This will depend on the requirements of the Exploiting Programme.
- Cyber Defence will need to know the rate at which it should provide notification of a security incident in order that notifications do not themselves restrict the legitimate availability of a communications channel. This will depend on the requirements of the Exploiting Programme.

5.4.2.12.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- This component could identify a cyber attack when none exists or incorrectly identify the System_Elements likely to be affected. This may cause other components to perform unnecessary actions to mitigate against an attack. In the worst case this could result in the Controlled-Trajectory Termination (CTT) of a UAV in a location that minimises the risk of third party fatalities, resulting in loss of the air vehicle (critical severity).

Failures of this component to detect a cyber attack resulting in compromise of sensitive data or allowing control of the air vehicle by unauthorised users is covered by the Cyber Defence PYRAMID concept.

Where instances of this component are used to prevent hazards that are less severe, the Exploiting Platform may require a less onerous DAL.

5.4.2.12.6.4 Security Considerations

The indicative security classification is SNEO.

This component requires information about the system, including its vulnerabilities and connectivity, etc. in order to be able to determine the attack vectors and progression of cyber attacks. It is therefore considered SNEO. The confidentiality of information that might divulge additional vulnerabilities to an adversary should be adequately protected.

The component is expected to satisfy security related functions relating to:

- **Logging of Security Data** for subsequent forensic examination of incidents and events, which might then point to the presence of a cyber attack or other breach.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **System Status and Monitoring** for cyber attacks.
- **Warnings and Notifications** for potential or confirmed cyber attacks.

The component is a cornerstone in the detection of cyber attacks, and is directly involved in satisfying security enforcing functions relating to:

- **Detecting Security Breaches** caused by cyber attacks. It may therefore have a level of awareness of security domains.
- **Preventing Cyber Attacks and Malware** as its primary function.

5.4.2.12.7 Services

5.4.2.12.7.1 Service Definitions

5.4.2.12.7.1.1 Cyber_Response

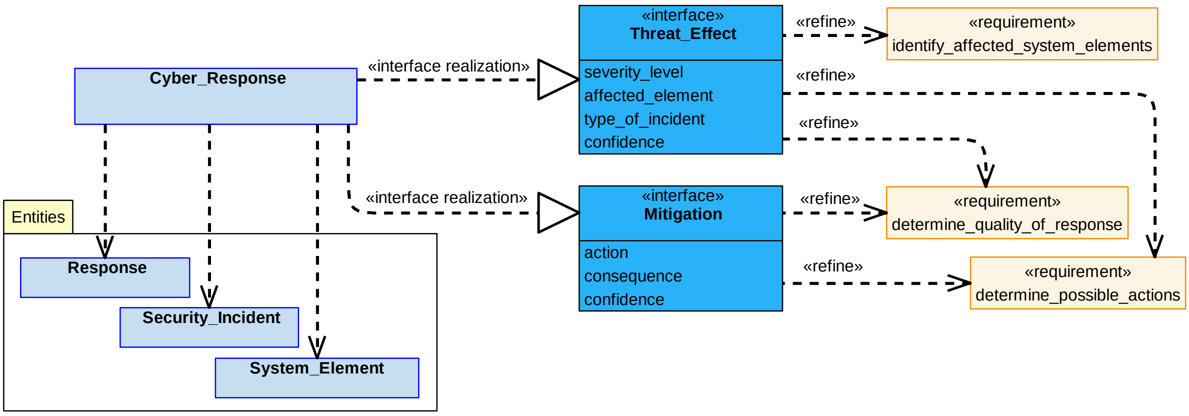
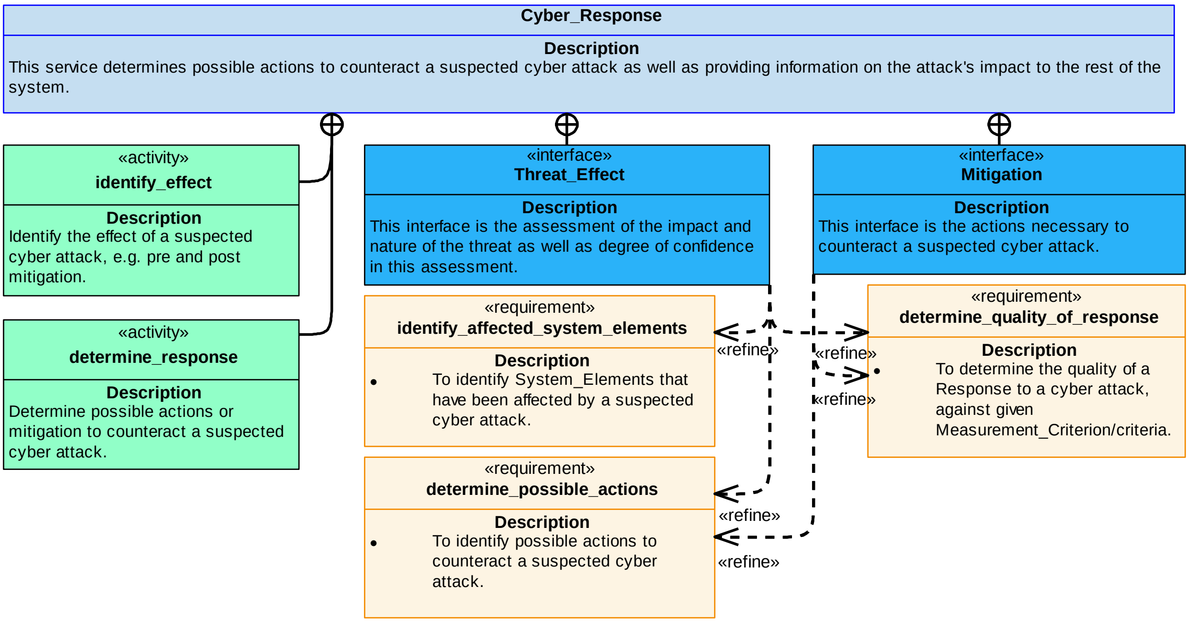


Figure 216: Cyber_Response Service Definition



This service determines possible actions to counteract a suspected cyber attack as well as providing information on the attack's impact to the rest of the system.

Identify the effect of a suspected cyber attack, e.g. pre and post mitigation.

Determine possible actions or mitigation to counteract a suspected cyber attack.

This interface is the assessment of the impact and nature of the threat as well as degree of confidence in this assessment.

This interface is the actions necessary to counteract a suspected cyber attack.

- To identify System_Elements that have been affected by a suspected cyber attack.

- To identify possible actions to counteract a suspected cyber attack.

- To determine the quality of a Response to a cyber attack, against given Measurement_Criterion/criteria.

Figure 217: Cyber Response Service Policy

Cyber_Response

This service determines possible actions to counteract a suspected cyber attack as well as providing information on the attack's impact to the rest of the system.

<u>Interfaces</u>

Threat_Effect

This interface is the assessment of the impact and nature of the threat as well as degree of confidence in this assessment.

Attributes

severity_level	The measure of severity of a suspected cyber attack.
affected_element	The System_Element affected by the suspected cyber attack.
type_of_incident	The nature of the Security_Incident the system has been affected by.
confidence	The confidence that the cyber attack and its effect have been correctly identified.

Mitigation

This interface is the actions necessary to counteract a suspected cyber attack.

Attributes

action	The action to be taken in mitigation of a cyber attack (e.g. blacklist a network port).
consequence	The expected consequence of not observing the mitigation so that trade-off can be assessed (e.g. mission objectives vs security, safety vs security).
confidence	The expected quality of cyber attack response.

Activities

determine_response

Determine possible actions or mitigation to counteract a suspected cyber attack.

identify_effect

Identify the effect of a suspected cyber attack, e.g. pre and post mitigation.

5.4.2.12.7.1.2 Attack_Identification

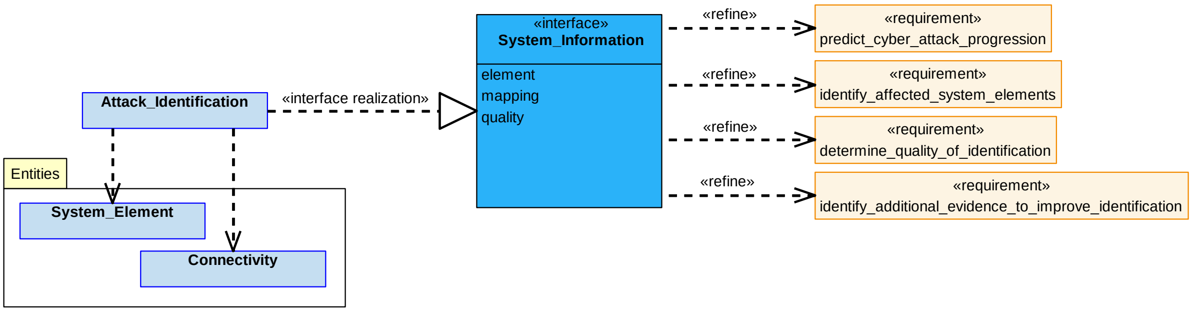


Figure 218: Attack_Identification Service Definition

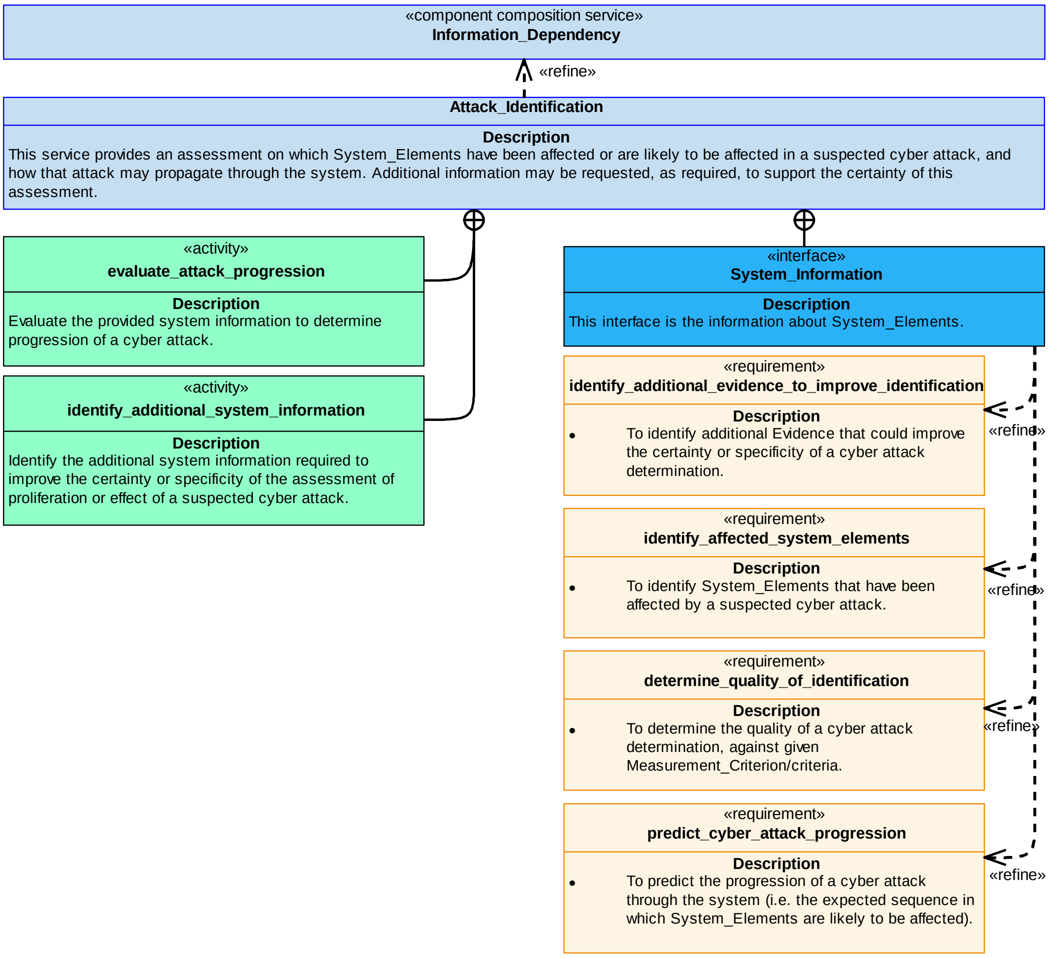

«component composition service»
Information_Dependency
«refine»
Attack_Identification
Description
This service provides an assessment on which System_Elements have been affected or are likely to be affected in a suspected cyber attack, and how that attack may propagate through the system. Additional information may be requested, as required, to support the certainty of this assessment.
«activity»
evaluate_attack_progression
Description
Evaluate the provided system information to determine progression of a cyber attack.
«activity»
identify_additional_system_information
Description
Identify the additional system information required to improve the certainty or specificity of the assessment of proliferation or effect of a suspected cyber attack.
«interface»
System_Information
Description
This interface is the information about System_Elements.
«requirement»
identify_additional_evidence_to_improve_identification
Description
To identify additional Evidence that could improve the certainty or specificity of a cyber attack determination.
«refine»
«requirement»
identify_affected_system_elements
Description
To identify System_Elements that have been affected by a suspected cyber attack.
«refine»
«requirement»
determine_quality_of_identification
Description
To determine the quality of a cyber attack determination, against given Measurement_Criterion/criteria.
«refine»
«requirement»
predict_cyber_attack_progression
Description
To predict the progression of a cyber attack through the system (i.e. the expected sequence in which System_Elements are likely to be affected).
«refine»

Figure 219: Attack Identification Service Policy

Attack_Identification

This service provides an assessment on which System_Elements have been affected or are likely to be affected in a suspected cyber attack, and how that attack may propagate through the system. Additional information may be requested, as required, to support the certainty of this assessment.

Interface

System_Information

This interface is the information about System_Elements.

Attributes

element	The System_Element the information is about.
mapping	The Connectivity between elements, e.g. physical, available or allowable (whitelist) connections.
quality	The quality of cyber attack identification.

Activities

evaluate_attack_progression

Evaluate the provided system information to determine progression of a cyber attack.

identify_additional_system_information

Identify the additional system information required to improve the certainty or specificity of the assessment of proliferation or effect of a suspected cyber attack.

5.4.2.12.7.1.3 Threat_Evidence

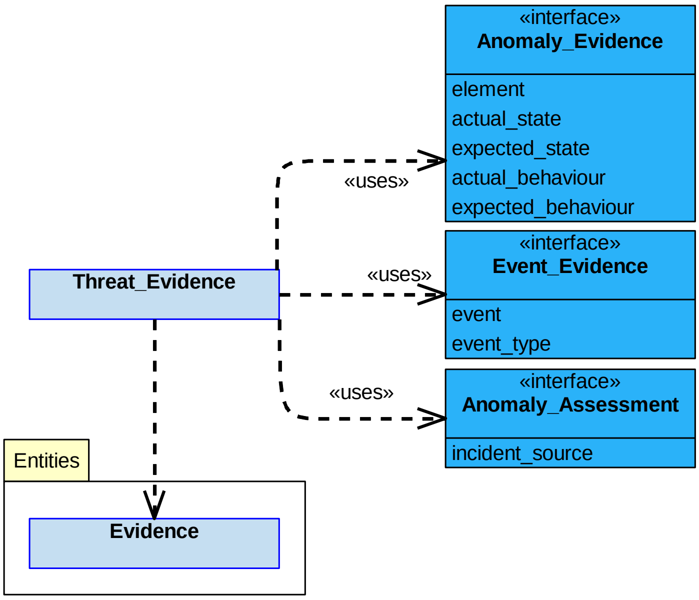
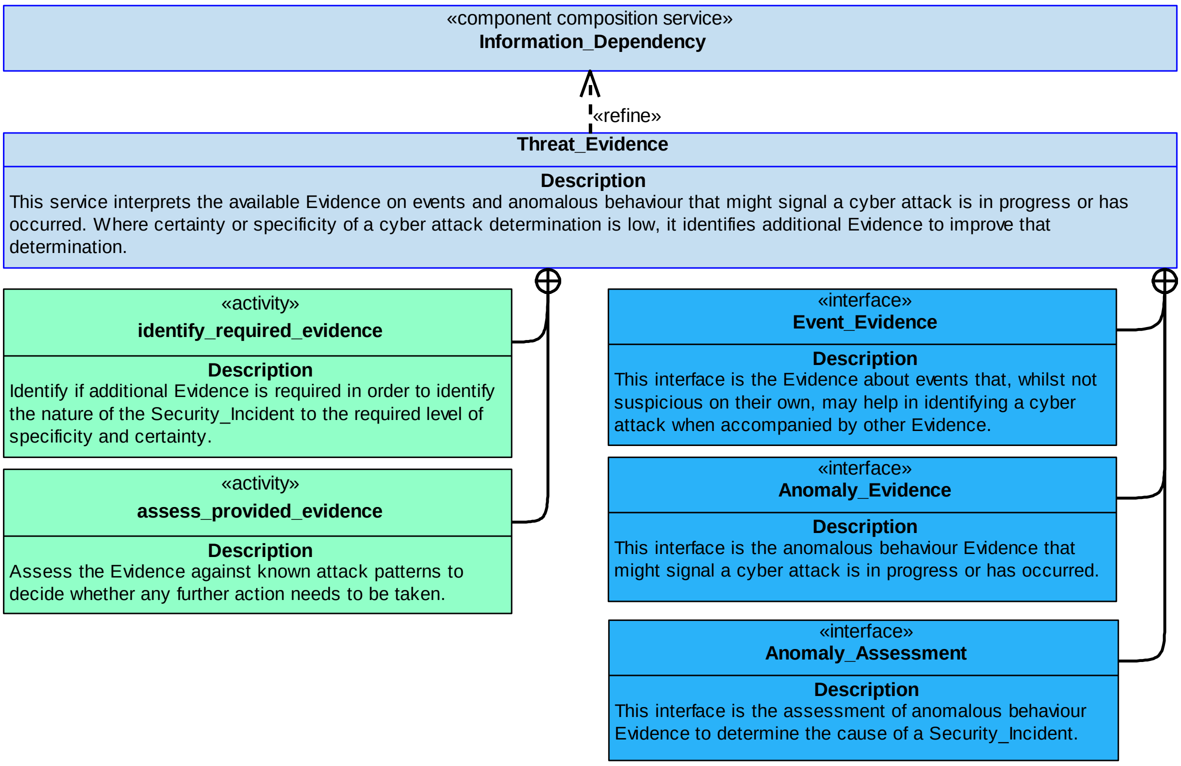


Figure 220: Threat_Evidence Service Definition



This service interprets the available Evidence on events and anomalous behaviour that might signal a cyber attack is in progress or has occurred. Where certainty or specificity of a cyber attack determination is low, it identifies additional Evidence to improve that determination.

Identify if additional Evidence is required in order to identify the nature of the Security_Incident to the required level of specificity and certainty.

Assess the Evidence against known attack patterns to decide whether any further action needs to be taken.

This interface is the Evidence about events that, whilst not suspicious on their own, may help in identifying a cyber attack when accompanied by other Evidence.

This interface is the anomalous behaviour Evidence that might signal a cyber attack is in progress or has occurred.

This interface is the assessment of anomalous behaviour Evidence to determine the cause of a Security_Incident.

Figure 221: Threat Evidence Service Policy

Threat_Evidence

This service interprets the available Evidence on events and anomalous behaviour that might signal a cyber attack is in progress or has occurred. Where certainty or specificity of a cyber attack determination is low, it identifies additional Evidence to improve that determination.

Interfaces

Anomaly_Evidence

This interface is the anomalous behaviour Evidence that might signal a cyber attack is in progress or has occurred.

Attributes

element	The anomalous System_Element the evidence is about.
actual_state	The actual state of the System_Element, e.g. active.
expected_state	The expected state of the System_Element, e.g. inactive.
actual_behaviour	The actual behaviour of the System_Element, e.g. data flow is unexpectedly sluggish.
expected_behaviour	The expected behaviour of the System_Element, e.g. data flow falls within an expected range.

Event_Evidence

This interface is the Evidence about events that, whilst not suspicious on their own, may help in identifying a cyber attack when accompanied by other Evidence.

Attributes

event	The specific event being reported (e.g. that user X's privileges have been increased).
event_type	The type of event being reported (e.g. elevation of privileges or increase in network traffic).

Anomaly_Assessment

This interface is the assessment of anomalous behaviour Evidence to determine the cause of a Security_Incident.

Attribute

incident_source	The origin of a Security_Incident (e.g. a system fault if it was determined to be non-hostile or deliberate cyber attack if determined to be hostile)

Activities

assess_provided_evidence

Assess the Evidence against known attack patterns to decide whether any further action needs to be taken.

identify_required_evidence

Identify if additional Evidence is required in order to identify the nature of the Security_Incident to the required level of specificity and certainty.

5.4.2.12.7.2 Service Dependencies

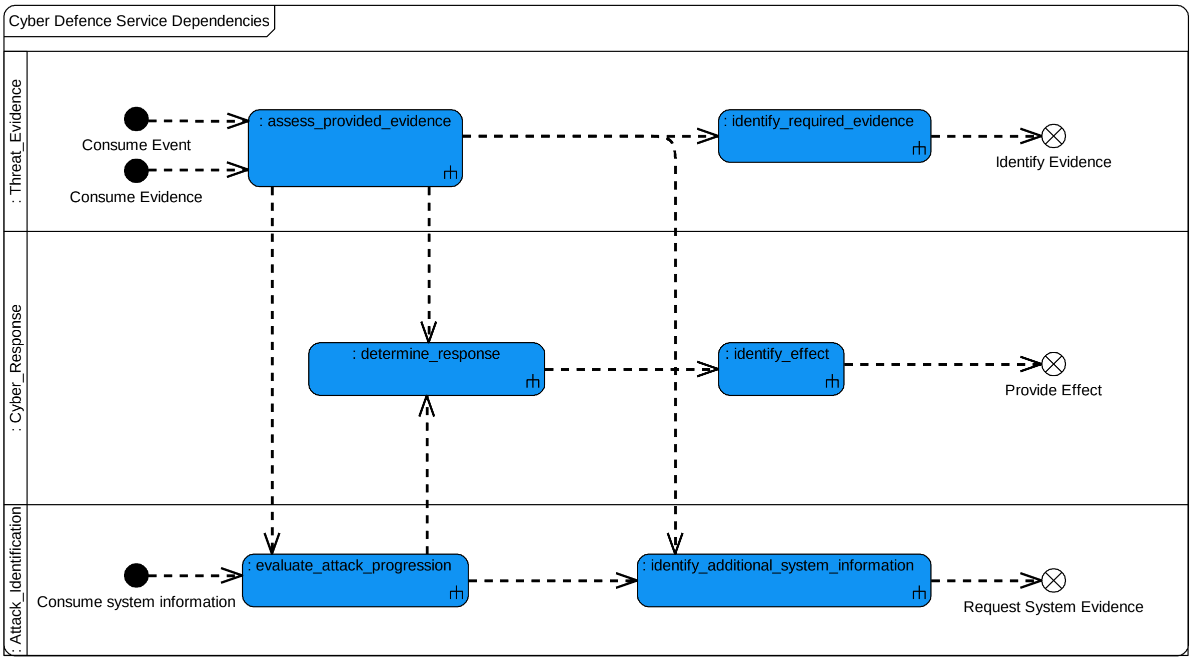

Cyber Defence Service Dependencies
: Threat_Evidence
: Cyber_Response
: Attack_Identification
Consume Event
Consume Evidence
: assess_provided_evidence
: identify_required_evidence
Identify Evidence
: determine_response
: identify_effect
Provide Effect
Consume system information
: evaluate_attack_progression
: identify_additional_system_information
Request System Evidence

Figure 222: Cyber Defence Service Dependencies

5.4.2.13 Data Distribution

5.4.2.13.1 Role

The role of Data Distribution is to prepare data for delivery (including preparing received data for delivery to an internal user), and instigate the delivery of data.

5.4.2.13.2 Overview

Control Architecture

Data Distribution is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

The Data Distribution component receives a Distribution_Requirement to distribute data to a Participant. The component identifies the available data Delivery_Resources and determines a delivery solution, given applicable Distribution_Constraints, including the required integrity checking and protection. Data Distribution gathers Data_Items, formats and protects them to create Delivery_Item(s) in accordance with Formatting_Rules and the agreed data distribution solution. The Data Distribution component also performs the reverse of this process to extract Data_Items from received Delivery_Items.

Examples of Use

Data Distribution will be used:

- For the delivery of data to and from a Participant external to the PYRAMID deployment.
- For the delivery of data between Participants that are internal to a PYRAMID deployment but located on separate nodes.
- For formatting data for delivery (e.g. by adding metadata and 'packaging' overheads)
- To automatically produce reports.

5.4.2.13.3 Service Summary

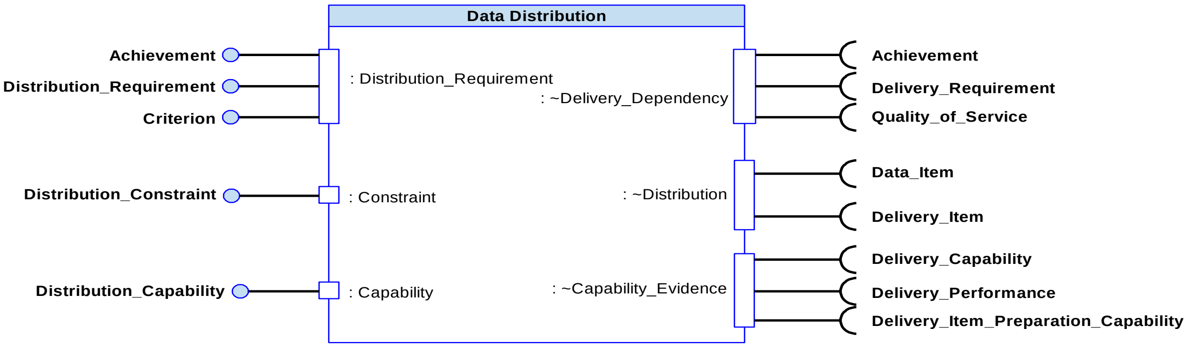


Figure 223: Data Distribution Service Summary

5.4.2.13.4 Responsibilities

capture_requirements_for_data_distribution

- To capture Distribution_Requirements for distribution of Delivery_Items.

capture_measurement_criteria_for_data_distribution

- To capture Measurement_Criteria for distribution of a Delivery_Item.

capture_constraints_for_data_distribution

- To capture Distribution_Constraints for the distribution of Delivery_Items.

determine_data_distribution_solution

- To determine a data distribution solution (e.g. transport method and protocol or report formatting) for use of Delivery_Resources that will meet given Distribution_Requirements, Distribution_Constraints and Measurement_Criteria.

gather_data_for_distribution

- To gather a Data_Item for distribution to a Participant in accordance with an agreed data distribution solution using Delivery_Resources.

format_data_for_distribution

- To format a Delivery_Item using the specified Formatting_Rules for distribution to a Participant in accordance with an agreed data distribution solution.

protect_data_for_distribution

- To protect a Delivery_Item using the specified Protections for distribution to a Participant in accordance with an agreed data distribution solution.

identify_data_distribution_solution_in_progress_remains_feasible

- To identify whether a data distribution solution currently in progress remains feasible.

distribute_data

- To distribute a Delivery_Item in accordance with an agreed data distribution solution using Delivery_Resources.

identify_progress_of_data_distribution_solution

- To identify the progress of a data distribution solution against the captured Distribution_Requirements.

determine_quality_of_data_distribution

- To determine the quality of a Delivery_Interaction, measured against given Distribution_Requirements and Measurement_Criteria (e.g. for data delivery this could be loss of packets or corrupted data).

assess_data_distribution_capability

- To assess the Distribution_Capability to distribute Delivery_Items taking account of system health and observed anomalies.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Distribution_Capability assessment.

predict_capability_progression

- To predict the progression of the Distribution_Capability over time and with use.

5.4.2.13.5 Subject Matter Semantics

The subject matter of Data Distribution is the distribution and reception of data, including collation and formatting for delivery (e.g. adding 'packaging' overheads to data that contributes to assurance of protection and integrity for delivery) and decomposition of delivered data on receipt.

Exclusions

The subject matter of Data Distribution does not include:

- Formatting of data items beyond what is required by Formatting_Rules for distribution; e.g. translation of Data_Items in accordance with the internal rules of another system or manipulating the format of information within Data_Items (e.g. presenting a value as kg or grammes).

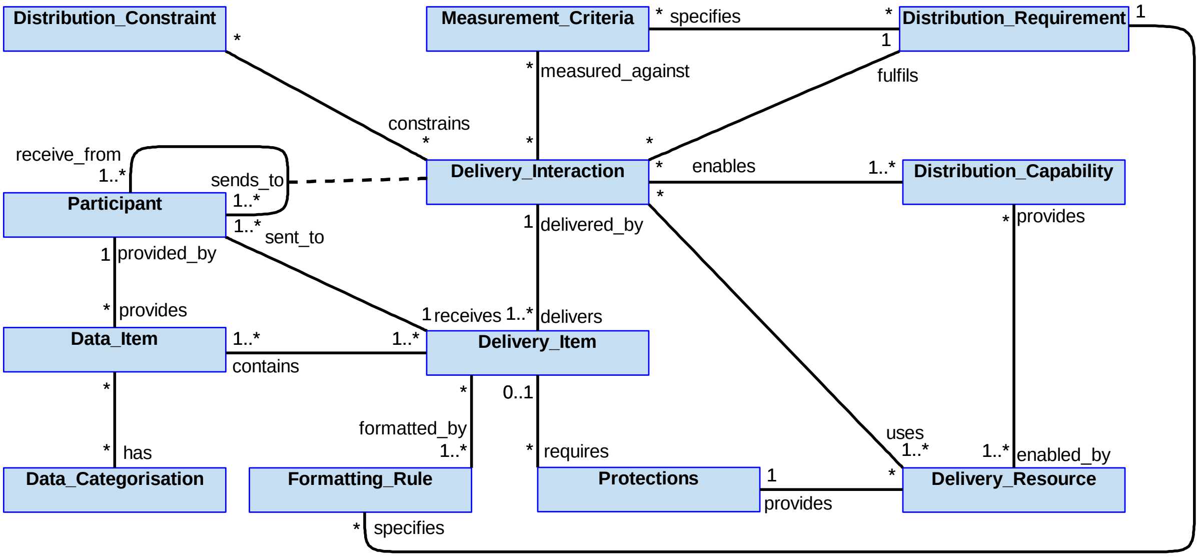


Figure 224: Data Distribution Semantics

5.4.2.13.5.1 Entities

Data_Categorisation

Information about a Data_Item, e.g. the classification, priority, or whether it is safety critical.

Data_Item

A specific item of data to be distributed, e.g. a sensor measurement or a TDL message.

Delivery_Interaction

An interaction to between Participants to deliver data.

Delivery_Item

The collated data and packaging to be delivered to the participant(s) or received from an external source, e.g. mission report, release of mission data or Internet Protocol packet. This is inclusive of any metadata or packaging overhead to be included, including that due to security and integrity protections being applied.

Delivery_Resource

The resources available to the component to perform a data exchange, e.g. communication and network resources, or cryptographic devices.

Distribution_Capability

The capability of this component to deliver data, e.g. the range of Delivery_Interactions which the component can support given available Delivery_Resources.

Distribution_Constraint

An externally placed limit on how data can be distributed, e.g. limits on Delivery_Resources that can be used or Delivery_Interactions that can occur.

Distribution_Requirement

The requirement defining the needs to deliver data.

Formatting_Rule

The rules for configuring the required Data_Item(s) into a specific Delivery_Item format for delivery and any metadata or overhead 'packaging' which needs to be added (e.g. the construction rules for a mission report or the structure of specific IP packet). This does not cover any formatting changes not required for the delivery of data (e.g. presenting a value as kg or grammes).

Measurement_Criteria

A measurable parameter that can be used to evaluate performance, quality and progress, e.g. quality of service.

Participant

The providers and receivers of data in an exchange. Multiple participants can provide Data_Items and Delivery_Items can be received by multiple participants.

Protections

The methods applied in order to protect a data exchange, e.g. the use of secure Delivery_Resources, or encryption of Delivery_Items.

5.4.2.13.6 Design Rationale

5.4.2.13.6.1 Assumptions

- This component is likely to interface with hardware responsible for data distribution activities.
- This component will support inter-nodal data distribution and data distribution where data is passing out of the system.
- This component may make use of communications resources to distribute data to recipients that are not physically co-located.
- This component will not be responsible for determining the specific data required or its required recipient, only determining a solution (including protection requirements, etc.) to distribute data appropriately.
- Cryptographic protection will be provided by another component.
- Gathering and formatting data either for transmission as part of a Delivery_Interaction or for the generation of a report are functionally identical processes from the perspective of the Data Distribution component, with the only differences being the formatting rules applied and the intended recipient.
- This component will typically not be used for component to component distribution that does not cross a system boundary.

5.4.2.13.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Data Distribution:

- Use of Communications - Allowing the component to be a ‘communications capability’ component as described in the Use of Communications PYRAMID concept in order to communicate with another instance of Data Distribution on another platform.
- Data Exchange - This component is at the core of the data exchange solution.
- Data Driving - The understanding of the communications capability available for use in Delivery_Interaction and the required formatting rules for reports is expected to be data-driven.
- Component Extensions - To allow more detailed formatting to be allowed for by specific tailored extension components.

Extensions

- Data Distribution can be developed to support different types of communication. This can be achieved by using extension components. The assumption is that different data exchange protocols will be implemented by the use of different protocol specific extensions; examples may include J series messages for communicating with Link16 systems, Ref. [47], STANAG 4586 common messaging for use with STANAG 4817 systems, Ref. [48], or DDS for communications with land systems, Ref. [49].

Exploitation Considerations

- Data Distribution is not required to have any understanding of what the data is that it will be distributing and receiving. It needs to understand the method of distribution of the data, and any information pertaining to that, but it does not understand the data being delivered (e.g. the payload).

5.4.2.13.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component may result in the inability to distribute data, between, for example, a ground based control station and the air vehicle. This is primarily a concern for UAVs, but may apply to manned air vehicles where some functions are controlled by external users. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) therefore the air vehicle will have been designed to mitigate a loss of communications. For a UAS this would be achieved by relying on pre-determined automatic or autonomous behaviour. However the failure of this component could also lead to the inability to distribute data between different system nodes on an Exploiting Platform (e.g. between the PYRAMID mission system and a non-PYRAMID sensor). For this failure mode it is concluded that failure of this component may result a "large reduction in safety margins", which has a critical severity.
- Failure of this component may also corrupt the data being distributed, which could result in the incorrect operation of the air vehicle, potentially resulting in hazardous consequences. However, where safety critical data is being distributed it is expected that the source application would have data integrity protection functionality provided outside of this component. The receiving application would only use the data if this functionality indicated the data was not corrupted. Corruption of the data transfer would therefore result in loss of "useable" data, as above.

Therefore, the indicative DAL is B.

5.4.2.13.6.4 Security Considerations

The indicative security classification is O.

This component is at the centre of an Exploiting Platform’s ability to exchange Data_Items, including with external entities (civil and military); preparing and instigating output data for delivery and received data for consumption. The classification of the component is dependent on that of the data it handles, meaning more stringent confidentiality requirements will apply in some cases. Instances of the component may therefore be necessary in different security domains.

It understands where the data is going, understands the protocols involved and has knowledge of message structures, etc. Based upon the rules imposed on it, it determines the required integrity checking and payload encryption of data as it crosses security domain boundaries and ensures data is not sent to the wrong recipient.

The component is responsible for adding checksums to data and as such plays a key role in maintaining system integrity. It’s involved in data prioritisation, potentially affecting availability of data, although this is not in a decision-making role. Loss of integrity or availability of this component would result a reduction in, or unreliable, data exchange capability.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Information** relating to changes made to the metadata of the data being distributed, protocols used, etc.

The component satisfies security enforcing functions by:

- **Protecting Integrity of Data** by adding checksums to data prior to distribution.
- **Ensuring Separation of Security Domains** by performing validity checks on data as it crosses domain boundaries.
- **Restricting Access to Data** through ensuring different classification of communications remain separated and are not directed inappropriately.

5.4.2.13.7 Services

5.4.2.13.7.1 Service Definitions

5.4.2.13.7.1.1 Distribution_Requirement

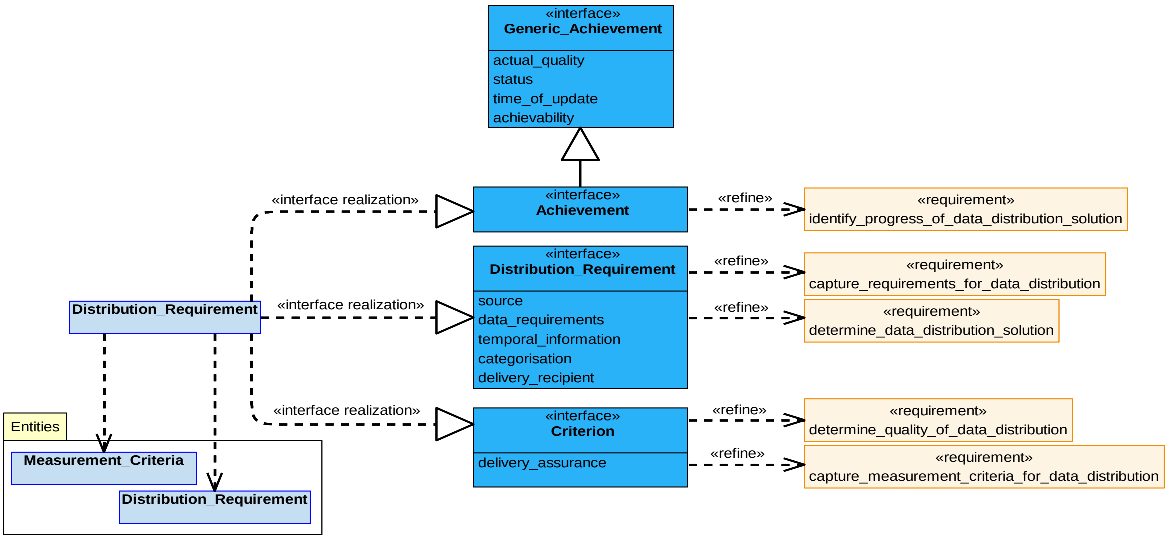


Figure 225: Distribution_Requirement Service Definition

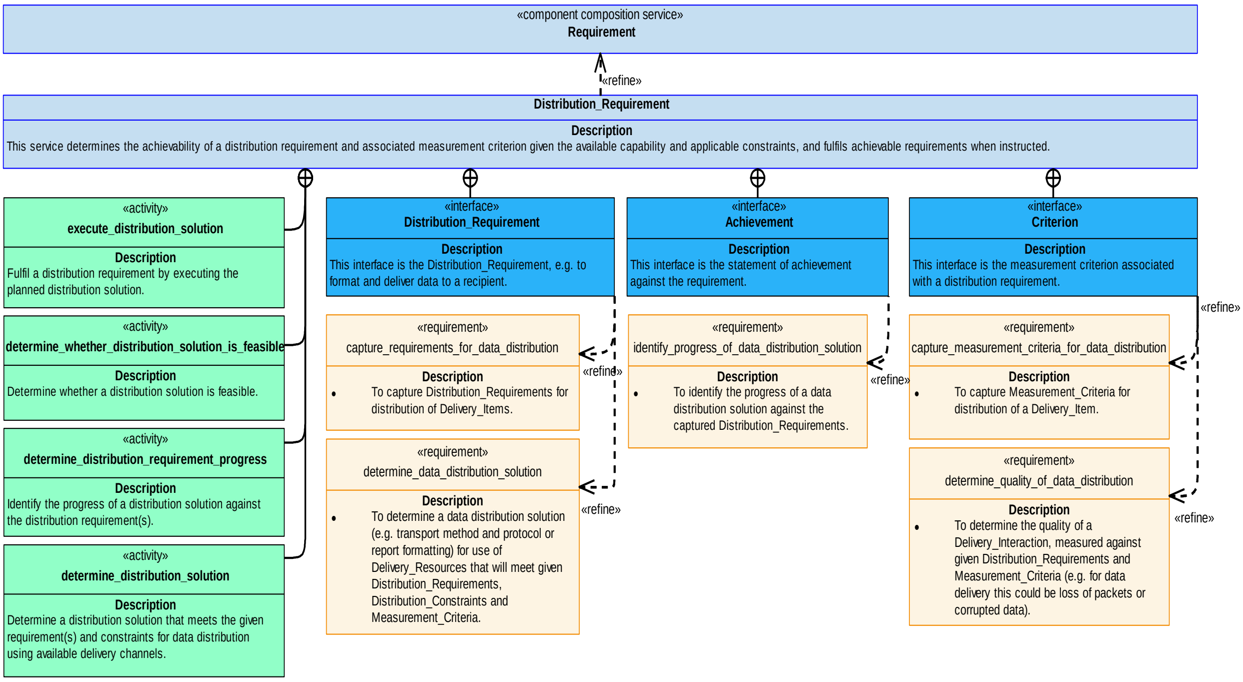

«component composition service»
Requirement
«refine»
Distribution_Requirement
Description
This service determines the achievability of a distribution requirement and associated measurement criterion given the available capability and applicable constraints, and fulfils achievable requirements when instructed.
«activity»
execute_distribution_solution
Description
Fulfil a distribution requirement by executing the planned distribution solution.
«activity»
determine_whether_distribution_solution_is_feasible
Description
Determine whether a distribution solution is feasible.
«activity»
determine_distribution_requirement_progress
Description
Identify the progress of a distribution solution against the distribution requirement(s).
«activity»
determine_distribution_solution
Description
Determine a distribution solution that meets the given requirement(s) and constraints for data distribution using available delivery channels.
«interface»
Distribution_Requirement
Description
This interface is the Distribution_Requirement, e.g. to format and deliver data to a recipient.
«interface»
Achievement
Description
This interface is the statement of achievement against the requirement.
«interface»
Criterion
Description
This interface is the measurement criterion associated with a distribution requirement.
«requirement»
capture_requirements_for_data_distribution
Description
To capture Distribution_Requirements for distribution of Delivery_Items.
«refine»
«requirement»
determine_data_distribution_solution
Description
To determine a data distribution solution (e.g. transport method and protocol or report formatting) for use of Delivery_Resources that will meet given Distribution_Requirements, Distribution_Constraints and Measurement_Criteria.
«refine»
«requirement»
identify_progress_of_data_distribution_solution
Description
To identify the progress of a data distribution solution against the captured Distribution_Requirements.
«refine»
«requirement»
capture_measurement_criteria_for_data_distribution
Description
To capture Measurement_Criteria for distribution of a Delivery_Item.
«refine»
«requirement»
determine_quality_of_data_distribution
Description
To determine the quality of a Delivery_Interaction, measured against given Distribution_Requirements and Measurement_Criteria (e.g. for data delivery this could be loss of packets or corrupted data).
«refine»

Figure 226: Distribution_Requirement Service Policy

Distribution_Requirement

This service determines the achievability of a distribution requirement and associated measurement criterion given the available capability and applicable constraints, and fulfils achievable requirements when instructed.

Interfaces

Achievement

This interface is the statement of achievement against the requirement.

Distribution_Requirement

This interface is the Distribution_Requirement, e.g. to format and deliver data to a recipient.

Attributes

source	The source(s) of the data.
data_requirements	The requirements for the data, including volume, format criteria, and type of data that is required to be distributed.
temporal_information	Timing information related to a distribution requirement, e.g. time frame for the data to be distributed.
categorisation	Information about the data being distributed, e.g. the classification, priority, or whether it is safety critical.
delivery_recipient	Who and where the data is to be distributed to.

Criterion

This interface is the measurement criterion associated with a distribution requirement.

Attribute

delivery_assurance	The likelihood that a delivery will be made, e.g. timeliness, data loss, and amount of errors and corruption.

Activities

determine_distribution_solution

Determine a distribution solution that meets the given requirement(s) and constraints for data distribution using available delivery channels.

determine_whether_distribution_solution_is_feasible

Determine whether a distribution solution is feasible.

execute_distribution_solution

Fulfil a distribution requirement by executing the planned distribution solution.

determine_distribution_requirement_progress

Identify the progress of a distribution solution against the distribution requirement(s).

5.4.2.13.7.1.2 Delivery_Dependency

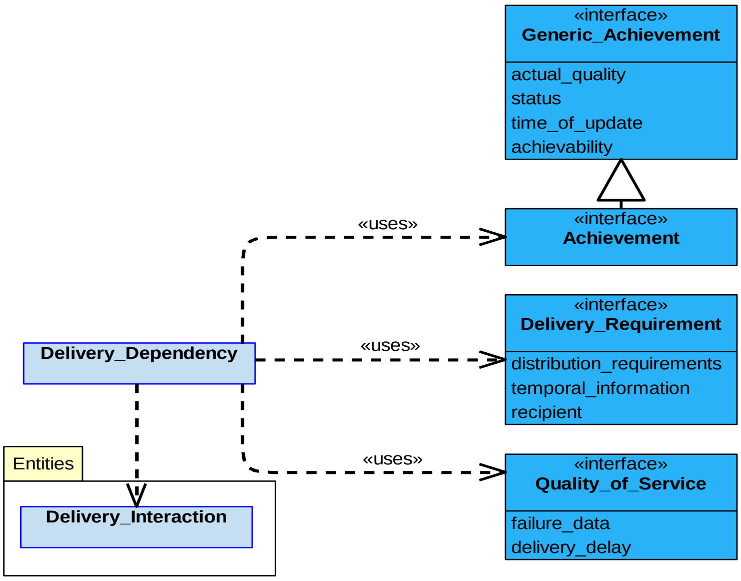


Figure 227: Delivery_Dependency Service Definition

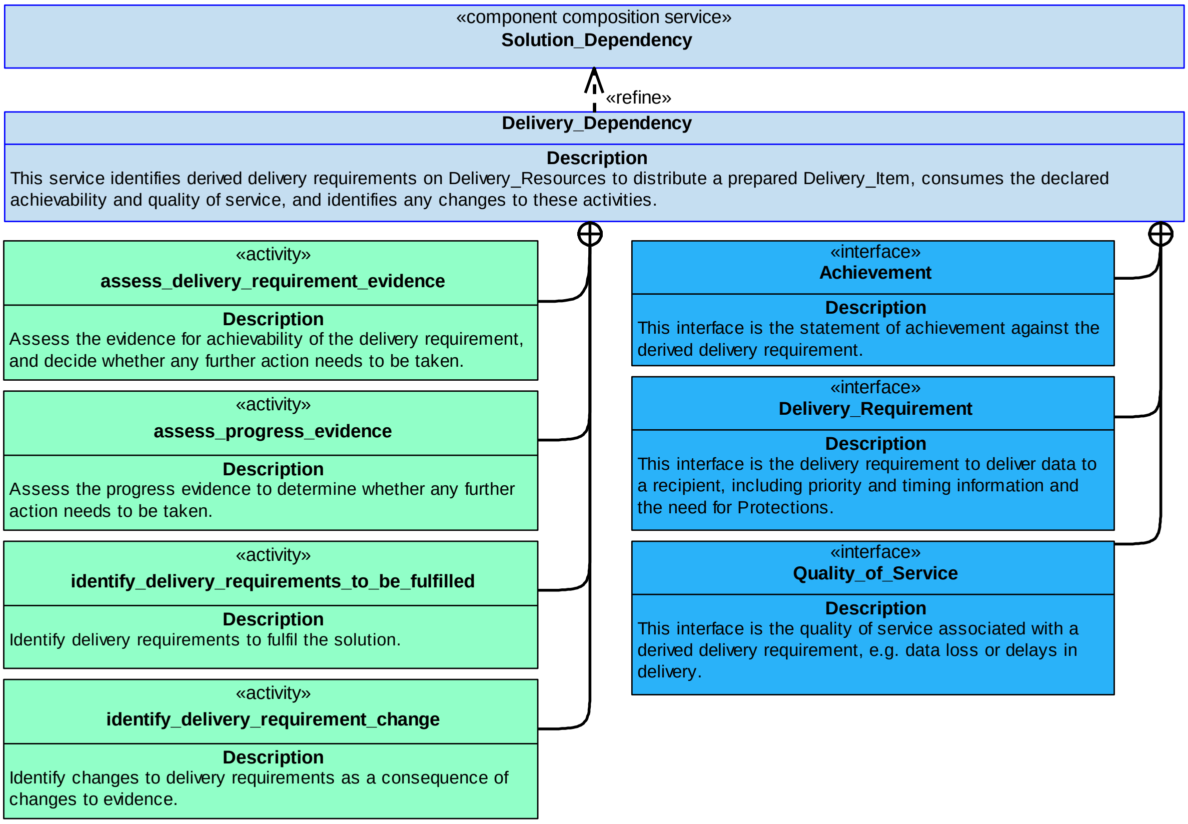

«component composition service»
Solution_Dependency
«refine»
Delivery_Dependency
Description
This service identifies derived delivery requirements on Delivery_Resources to distribute a prepared Delivery_Item, consumes the declared achievability and quality of service, and identifies any changes to these activities.
«activity»
assess_delivery_requirement_evidence
Description
Assess the evidence for achievability of the delivery requirement, and decide whether any further action needs to be taken.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to determine whether any further action needs to be taken.
«activity»
identify_delivery_requirements_to_be_fulfilled
Description
Identify delivery requirements to fulfil the solution.
«activity»
identify_delivery_requirement_change
Description
Identify changes to delivery requirements as a consequence of changes to evidence.
«interface»
Achievement
Description
This interface is the statement of achievement against the derived delivery requirement.
«interface»
Delivery_Requirement
Description
This interface is the delivery requirement to deliver data to a recipient, including priority and timing information and the need for Protections.
«interface»
Quality_of_Service
Description
This interface is the quality of service associated with a derived delivery requirement, e.g. data loss or delays in delivery.

Figure 228: Delivery_Dependency Service Policy

Delivery_Dependency

This service identifies derived delivery requirements on Delivery_Resources to distribute a prepared Delivery_Item, consumes the declared achievability and quality of service, and identifies any changes to these activities.

Interfaces

Achievement

This interface is the statement of achievement against the derived delivery requirement.

Delivery_Requirement

This interface is the delivery requirement to deliver data to a recipient, including priority and timing information and the need for Protections.

Attributes

distribution_requirements	The requirements for the data to be delivered, including volume, protections, priority, and type of data.
temporal_information	Timing information related to a delivery requirement, e.g. time to deliver data.
recipient	The target to which data will be delivered.

Quality_of_Service

This interface is the quality of service associated with a derived delivery requirement, e.g. data loss or delays in delivery.

Attributes

failure_data	Information about the data that is failing to be delivered or the rate at which data is failing to be delivered.
delivery_delay	The time taken to deliver the data.

Activities

assess_delivery_requirement_evidence

Assess the evidence for achievability of the delivery requirement, and decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to determine whether any further action needs to be taken.

identify_delivery_requirements_to_be_fulfilled

Identify delivery requirements to fulfil the solution.

identify_delivery_requirement_change

Identify changes to delivery requirements as a consequence of changes to evidence.

5.4.2.13.7.1.3 Distribution

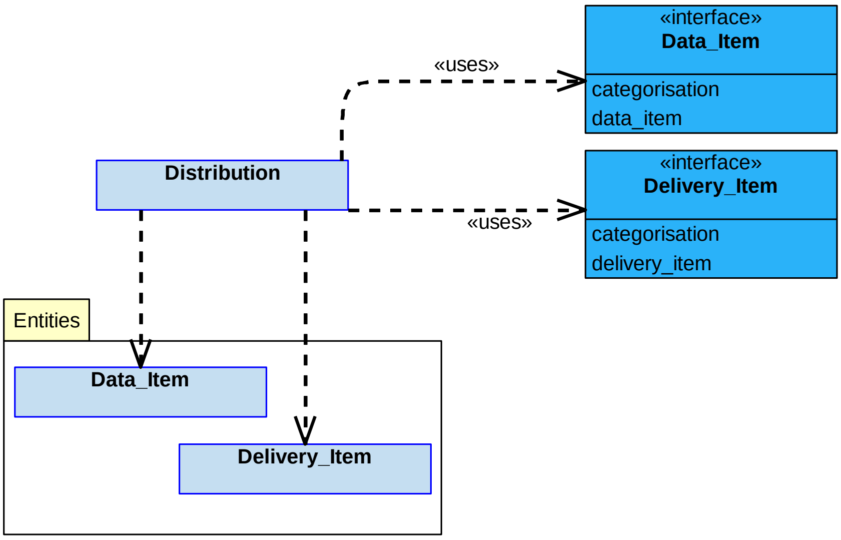


Figure 229: Distribution Service Definition

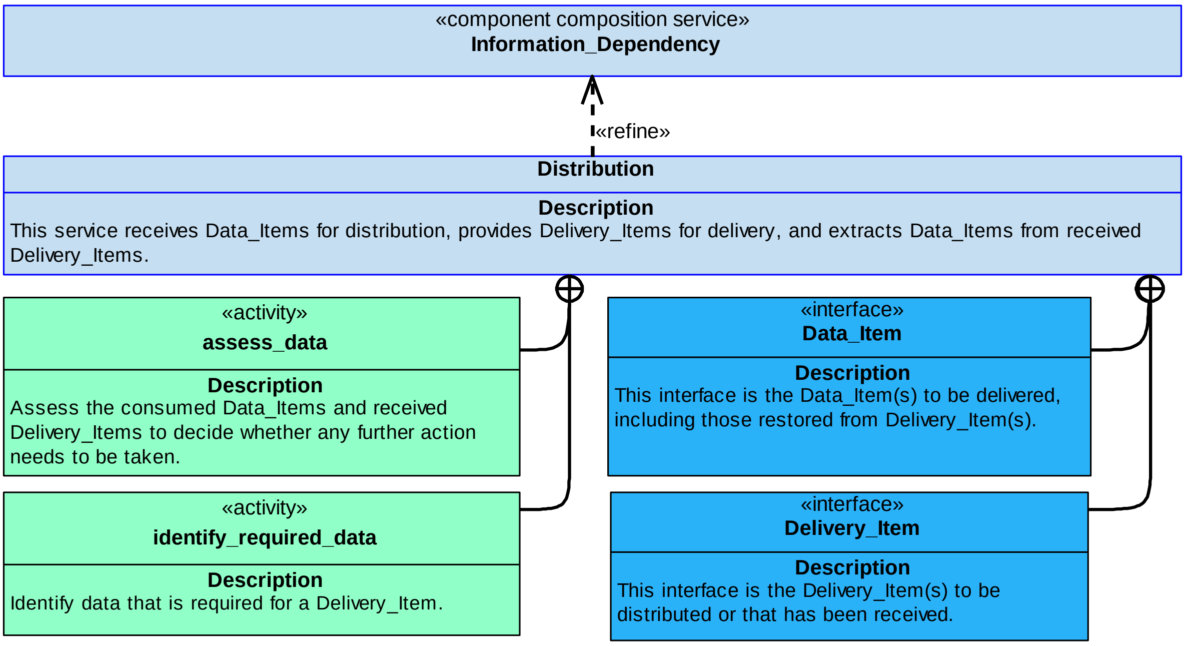

«component composition service»
Information_Dependency
«refine»
Distribution
Description
This service receives Data_Items for distribution, provides Delivery_Items for delivery, and extracts Data_Items from received Delivery_Items.
«activity»
assess_data
Description
Assess the consumed Data_Items and received Delivery_Items to decide whether any further action needs to be taken.
«activity»
identify_required_data
Description
Identify data that is required for a Delivery_Item.
«interface»
Data_Item
Description
This interface is the Data_Item(s) to be delivered, including those restored from Delivery_Item(s).
«interface»
Delivery_Item
Description
This interface is the Delivery_Item(s) to be distributed or that has been received.

Figure 230: Distribution Service Policy

Distribution

This service receives Data_Items for distribution, provides Delivery_Items for delivery, and extracts Data_Items from received Delivery_Items.

Interfaces

Data_Item

This interface is the Data_Item(s) to be delivered, including those restored from Delivery_Item(s).

Attributes

categorisation	Information about the Data_Item, e.g. the classification, priority, or criticality.
data_item	The specific item of data that is the subject of a Distribution_Requirement.

Delivery_Item

This interface is the Delivery_Item(s) to be distributed or that has been received.

Attributes

categorisation	Information about the Delivery_Item, e.g. the classification, priority, or criticality.
delivery_item	The specific item of data to be distributed or that has been received.

Activities

assess_data

Assess the consumed Data_Items and received Delivery_Items to decide whether any further action needs to be taken.

identify_required_data

Identify data that is required for a Delivery_Item.

5.4.2.13.7.1.4 Constraint

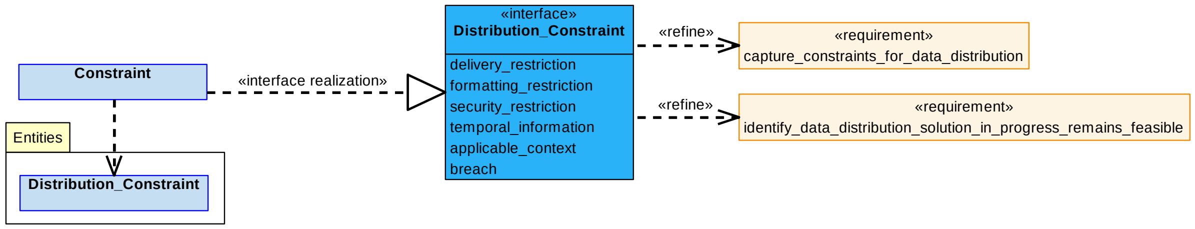
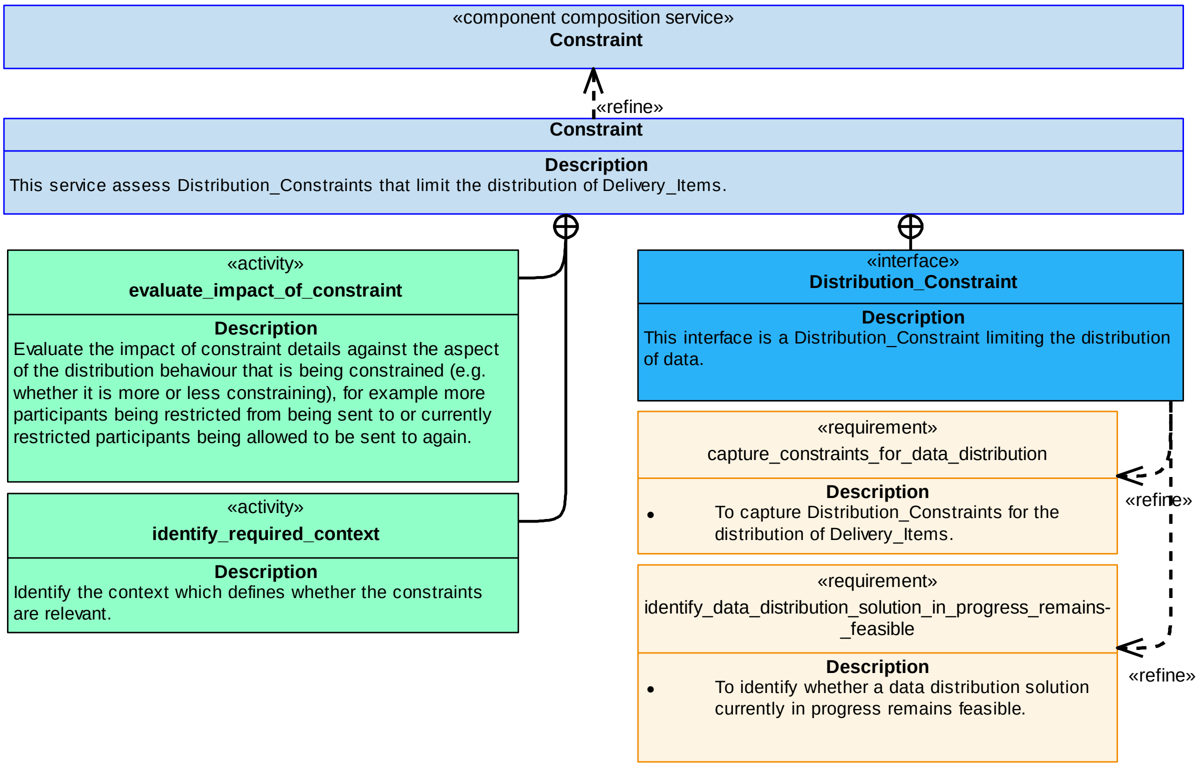


Figure 231: Constraint Service Definition


«component composition service»
Constraint
«refine»
Constraint
Description
«activity»
evaluate_impact_of_constraint
Description
«activity»
identify_required_context
Description
«interface»
Distribution_Constraint
Description
«requirement»
capture_constraints_for_data_distribution
Description
«refine»
«requirement»
identify_data_distribution_solution_in_progress_remains-_feasible
Description
«refine»

This service assess Distribution_Constraints that limit the distribution of Delivery_Items.

Evaluate the impact of constraint details against the aspect of the distribution behaviour that is being constrained (e.g. whether it is more or less constraining), for example more participants being restricted from being sent to or currently restricted participants being allowed to be sent to again.

Identify the context which defines whether the constraints are relevant.

This interface is a Distribution_Constraint limiting the distribution of data.

- To capture Distribution_Constraints for the distribution of Delivery_Items.

- To identify whether a data distribution solution currently in progress remains feasible.

Figure 232: Constraint Service Policy

Constraint

This service assess Distribution_Constraints that limit the distribution of Delivery_Items.

<u>Interface</u>

Distribution_Constraint

This interface is a Distribution_Constraint limiting the distribution of data.

<u>Attributes</u>

delivery_restriction	How or when a delivery is allowed, e.g. which participants are allowed to be involved, or how much data is allowed to be delivered.

formatting_restriction	What formatting types are allowed to be used, e.g. which data delivery packaging methods, or mission report types can be used.
security_restriction	The minimum permissible protection level to be used for the distribution of a Delivery_Item.
temporal_information	Timing information related to when a constraint is applicable, e.g. start and end time or applicable for 30 minutes.
applicable_context	The context in which the constraint is applicable.
breach	A statement that a Distribution_Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of constraint details against the aspect of the distribution behaviour that is being constrained (e.g. whether it is more or less constraining), for example more participants being restricted from being sent to or currently restricted participants being allowed to be sent to again.

identify_required_context

Identify the context which defines whether the constraints are relevant.

5.4.2.13.7.1.5 Capability

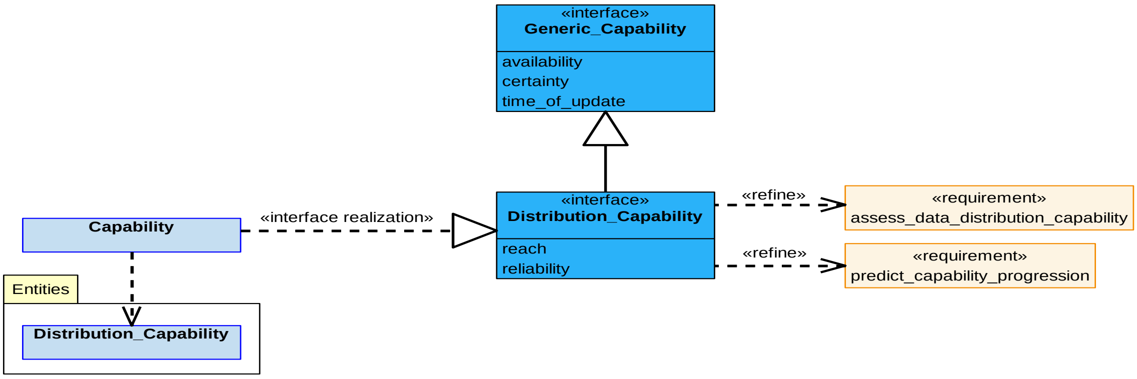


Figure 233: Capability Service Definition

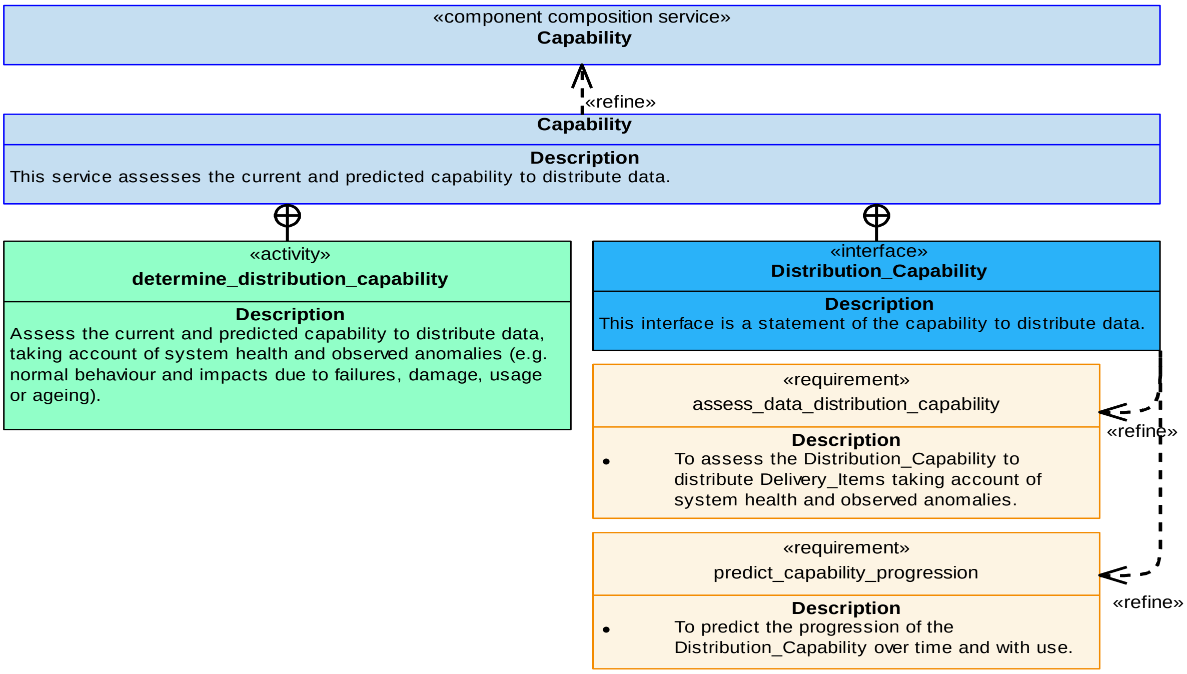

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability to distribute data.
«activity»
determine_distribution_capability
Description
Assess the current and predicted capability to distribute data, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Distribution_Capability
Description
This interface is a statement of the capability to distribute data.
«requirement»
assess_data_distribution_capability
Description
To assess the Distribution_Capability to distribute Delivery_Items taking account of system health and observed anomalies.
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the Distribution_Capability over time and with use.
«refine»

Figure 234: Capability Service Policy

Capability

This service assesses the current and predicted capability to distribute data.

Interface

Distribution_Capability

This interface is a statement of the capability to distribute data.

Attributes

reach	Where and who the component is able to receive from and deliver to, e.g. the endpoint data user.
reliability	The reliability of a delivery, e.g. whether a delivery will not arrive at the intended recipient.

Activity

determine_distribution_capability

Assess the current and predicted capability to distribute data, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.13.7.1.6 Capability_Evidence

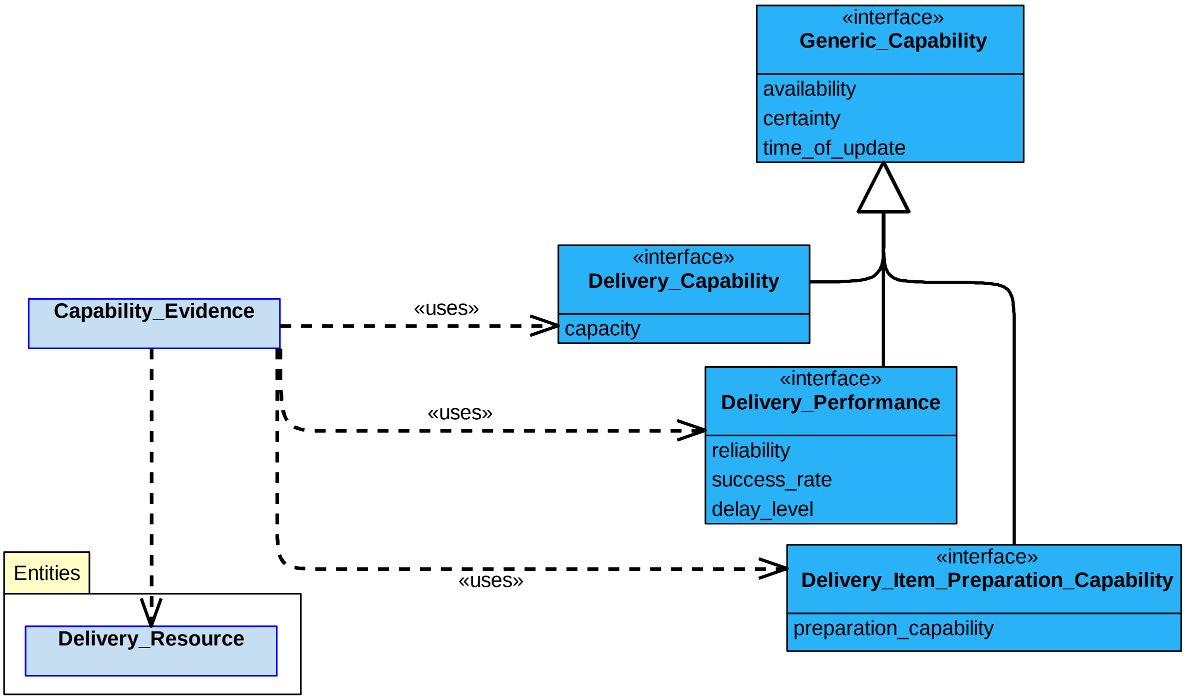
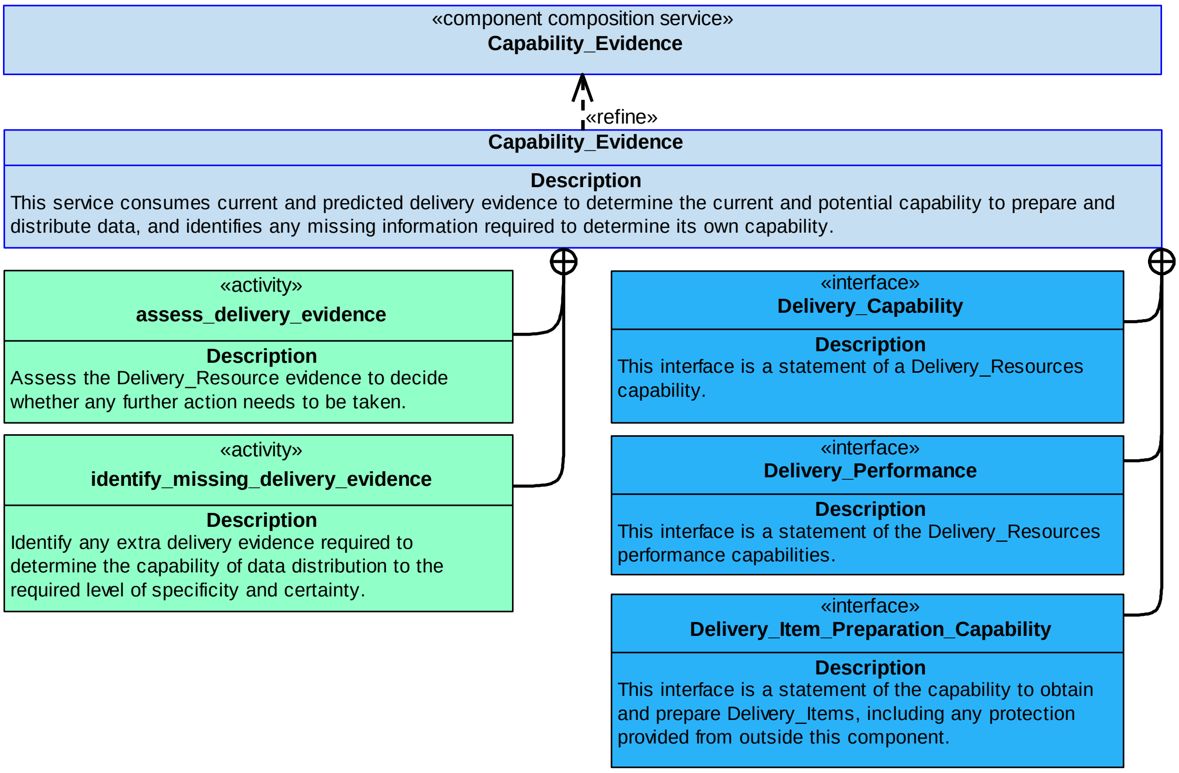


Figure 235: Capability_Evidence Service Definition



Description
This service consumes current and predicted delivery evidence to determine the current and potential capability to prepare and distribute data, and identifies any missing information required to determine its own capability.

Description
Assess the Delivery_Resource evidence to decide whether any further action needs to be taken.

Description
Identify any extra delivery evidence required to determine the capability of data distribution to the required level of specificity and certainty.

Description
This interface is a statement of a Delivery_Resources capability.

Description
This interface is a statement of the Delivery_Resources performance capabilities.

Description
This interface is a statement of the capability to obtain and prepare Delivery_Items, including any protection provided from outside this component.

Figure 236: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted delivery evidence to determine the current and potential capability to prepare and distribute data, and identifies any missing information required to determine its own capability.

Interfaces

Delivery_Capability

This interface is a statement of a Delivery_Resources capability.

Attribute

capacity	The amount of available capacity to be used for a delivery, e.g. throughput.

Delivery_Performance

This interface is a statement of the Delivery_Resources performance capabilities.

Attributes

reliability	The reliability of a Delivery_Resource, e.g. the capability of a Delivery_Resource to consistently deliver as intended.
success_rate	The rate that data is being successfully delivered.
delay_level	The level of delay to complete a delivery, i.e. the time taken to deliver data.

Delivery_Item_Preparation_Capability

This interface is a statement of the capability to obtain and prepare Delivery_Items, including any protection provided from outside this component.

Attribute

preparation_capability	The specific capability (e.g. the ability to source cryptographic protection) about which the capability statement is being made

Activities

assess_delivery_evidence

Assess the Delivery_Resource evidence to decide whether any further action needs to be taken.

identify_missing_delivery_evidence

Identify any extra delivery evidence required to determine the capability of data distribution to the required level of specificity and certainty.

5.4.2.13.7.2 Service Dependencies

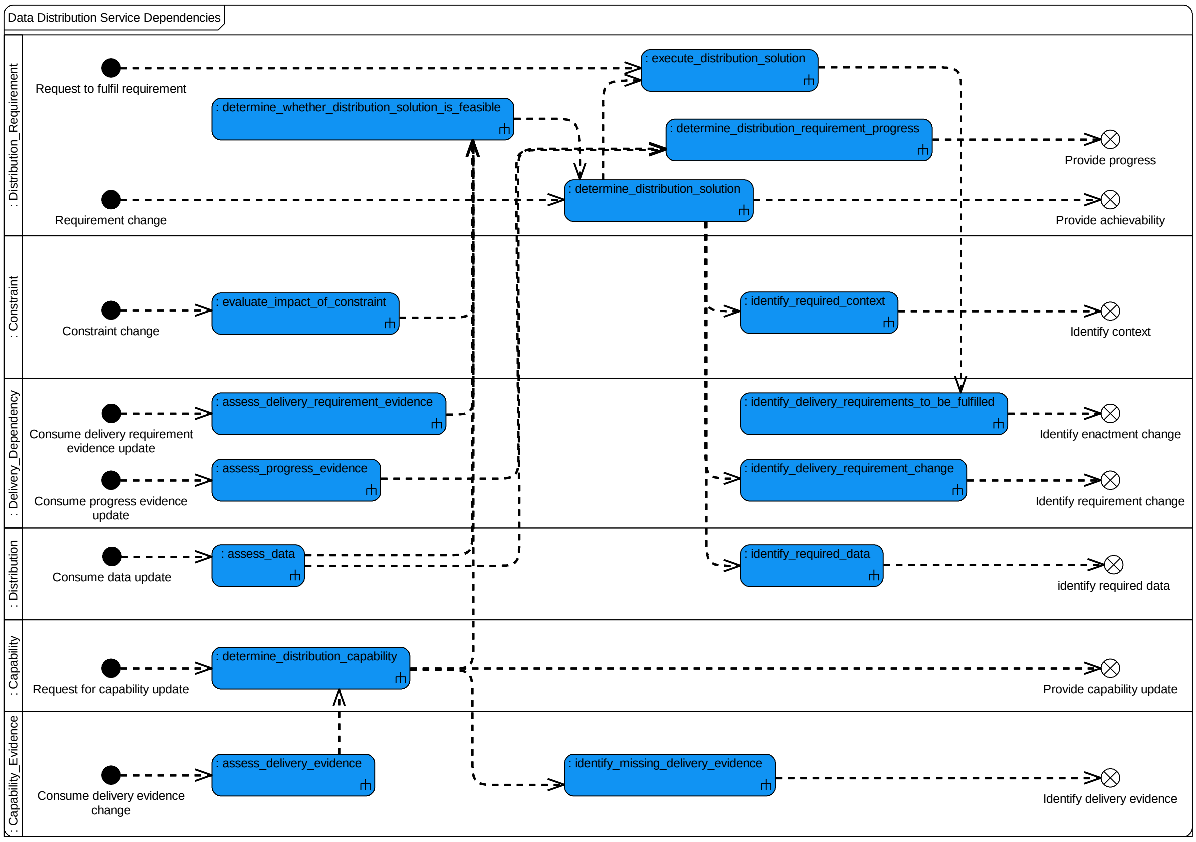

Data Distribution Service Dependencies
: Distribution_Requirement
: Constraint
: Delivery_Dependency
: Distribution
: Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
Constraint change
Consume delivery requirement evidence update
Consume progress evidence update
Consume data update
Request for capability update
Consume delivery evidence change
: execute_distribution_solution
: determine_whether_distribution_solution_is_feasible
: determine_distribution_requirement_progress
: determine_distribution_solution
: evaluate_impact_of_constraint
: identify_required_context
: assess_delivery_requirement_evidence
: identify_delivery_requirements_to_be_fulfilled
: assess_progress_evidence
: identify_delivery_requirement_change
: assess_data
: identify_required_data
: determine_distribution_capability
: assess_delivery_evidence
: identify_missing_delivery_evidence
Provide progress
Provide achievability
Identify context
Identify enactment change
Identify requirement change
identify required data
Provide capability update
Identify delivery evidence

Figure 237: Data Distribution Service Dependencies

5.4.2.14 Data Fusion

5.4.2.14.1 Role

The role of Data Fusion is to evaluate and combine evidence of potential objects to provide an understanding of real-world entities.

5.4.2.14.2 Overview

Control Architecture

Data Fusion is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

A requirement to interpret and combine Evidence of objects will be received. Data Fusion will then generate Fused_Objects by:

- Evaluating the information that can be extracted from the Evidence.
- Determining which pieces of Evidence relate to the same real-world entity.
- Determining and amalgamating the best sources of information from the available Evidence.

Examples of Use

Data Fusion can be used for interpreting Evidence of real-world objects, including the generation of Fused_Objects that are more accurate and reliable than the standalone sources of information from which data was fused. Examples include:

- The position, velocity and identification of an object, by evaluating Evidence from a single source (e.g. provided as the characterisation of a pattern identified in an image).
- The position, velocity and identification of an object, by combining multiple pieces of Evidence (e.g. provided as the characterisation of patterns identified in multiple images).
- The identification of an object by combining Evidence of a vehicle (provided as the characterisation of a pattern identified in an image) with Evidence of an emitter (provided as the characterisation of a pattern within an RF waveform).
- The generation of an object track based on positional Evidence from multiple sources.

5.4.2.14.3 Service Summary

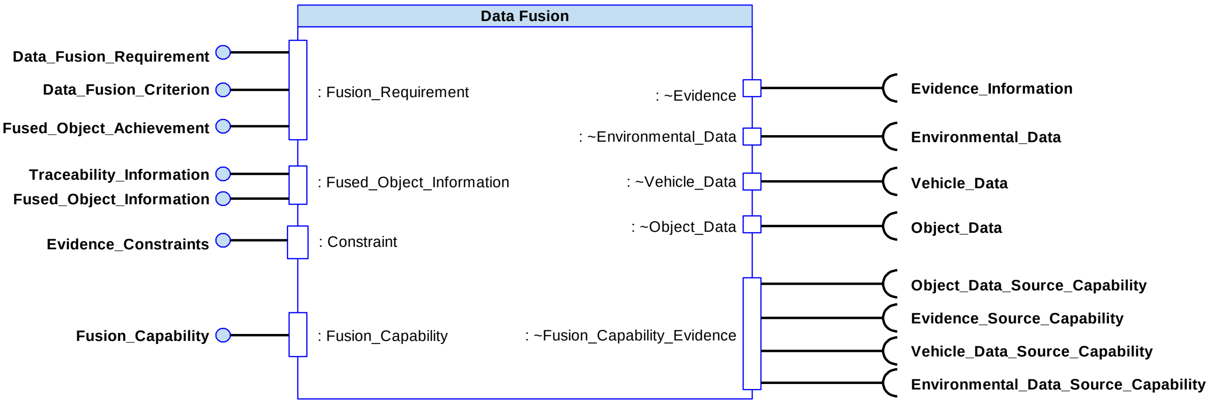


Figure 238: Data Fusion Service Summary

5.4.2.14.4 Responsibilities

capture_fusion_requirements

- To capture Data_Fusion_Requirements for interpreting Evidence and generating Fused_Objects.

capture_measurement_criteria_for_fusion

- To capture provided Measurement_Criterion (e.g. timeliness, confidence, completeness or accuracy) for data fusion.

capture_fusion_constraints

- To capture Fusion_Constraints for data fusion (e.g. restriction on sources of Evidence).

determine_if_fusion_requirement_is_achievable

- To determine if a Data_Fusion_Requirement is achievable, given current Fusion_Constraints and resources.

determine_fusion_solution

- To determine how to meet the given Data_Fusion_Requirements, within applicable Fusion_Constraints.

determine_predicted_quality_of_fusion_solution

- To determine the predicted quality of a data fusion solution against given Measurement_Criterion.

generate_fused_objects

- To generate Fused_Objects based on interpretation of the available Evidence.

maintain_fused_object_lineage

- To maintain the lineage of Fused_Objects (e.g. lineage between Evidence and Fused_Objects, and the merging and splitting Fused_Objects).

capture_evidence

- To capture provided Evidence along with its lineage.

capture_supporting_information

- To capture provided Supporting_Information (e.g. platform data, weather condition or terrain data).

determine_quality_of_fused_objects

- To determine the quality of the Fused_Objects, measured against given Data_Fusion_Requirements and Measurement_Criterion.

assess_capability

- To assess the capability to generate Fused_Objects, taking into account observed anomalies.

identify_missing_capability_information

- To identify missing information that could improve the certainty or specificity of the Data Fusion capability assessment.

predict_capability_progression

- To predict the progression of Fusion_Capability over time and with use, e.g. Data Fusion determines that its Fusion_Capability is downgraded due to the intermittent availability of an Evidence_Type.

5.4.2.14.5 Subject Matter

The subject matter of Data Fusion is the identification and characterisation of real-world entities of tactical significance, based on Evidence from one or more sources, and the lineage of that Evidence.

The subject matter of Data Fusion does not include:

- The combining or amalgamation of data outside of that required to process evidence of real-world entities.
- The combining or amalgamation of data where such processing falls within the subject matter of other components. For example, Data Fusion excludes the combining of navigation data and the combining of sensor product data.

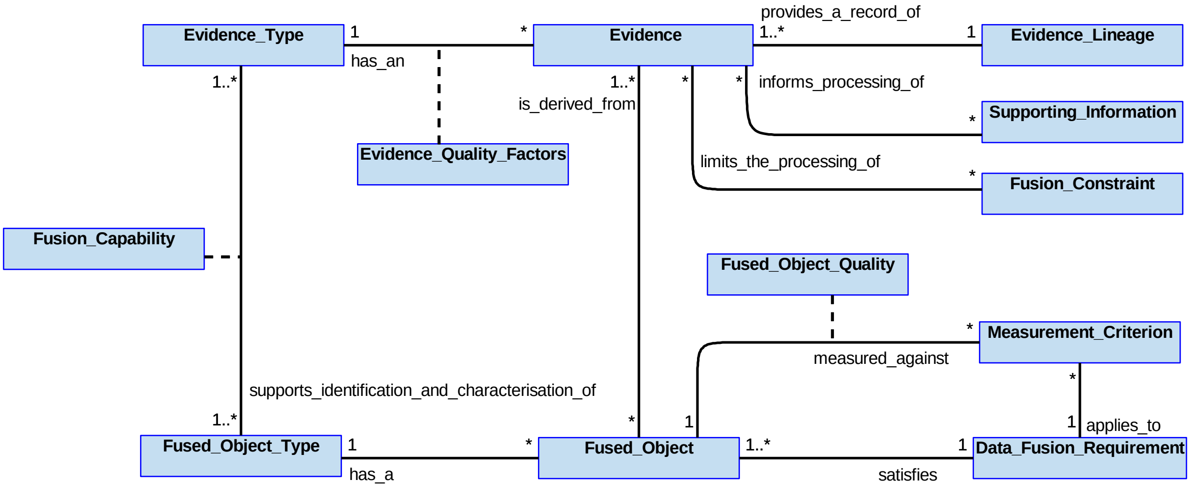


Figure 239: Data Fusion Semantics

5.4.2.14.5.1 Entities

Data_Fusion_Requirement

A requirement to identify and characterise real-world entities by interpreting and amalgamating Evidence.

Evidence

Information used in the process of identifying and characterising real-world entities (e.g. a pattern identified in imagery or RF waveforms).

Evidence_Lineage

A record of the Evidence from which object interpretations are made, including the sources of any previous amalgamation.

Fused_Object_Quality

A measure of the effectiveness or adequacy of the characterisation of a real-world entity (e.g. confidence or accuracy).

Evidence_Quality_Factors

A set of measures that define the effectiveness or adequacy of evidence used as the basis for object interpretation (e.g. confidence, tolerance levels, or accuracy).

Evidence_Type

The type of Evidence (e.g. the nature of the evidence, such as an object or emission pattern, and the type of source from which it is derived, such as imagery or RF waveform).

Fusion_Capability

The range of Evidence_Types that can be processed and the range of Fused_Object_Types that can be generated from such Evidence_Types.

Fusion_Constraint

A restriction on the interpretation and amalgamation of Evidence (e.g. time restriction, processing restriction, Evidence_Type or source restriction).

Fused_Object

The characterisation of a real-world entity, based on Evidence from one or more sources.

Fused_Object_Type

The type of real-world entity (e.g. vehicle type or weapon type).

Measurement_Criterion

A measure against which achievement of the fusion requirement can be assessed (e.g. timeliness, reliability, processing completeness or object identification).

Supporting_Information

Information that dynamically influences the planning or enactment of fusion processing in order to satisfy the Data_Fusion_Requirement. For example, the current weather conditions may result in the component selecting an alternative Evidence_Type in order to satisfy the Data_Fusion_Requirement.

5.4.2.14.6 Design Rationale

5.4.2.14.6.1 Assumptions

- New and updated processing algorithms are expected to be defined frequently over the lifetime of an Exploiting Platform based upon the operating environment and continuous improvements to the algorithms.
- Types of Evidence are expected to be updated as new sensing technologies and techniques emerge.

5.4.2.14.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Data Fusion:

- Data Driving - this component can be data-driven to cater for different Evidence_Types, Evidence_Quality_Factors, Fused_Object_Types and algorithms with varying characteristics.
- Recording and Logging - this component will carry out logging for audit purposes.
- Tactical Information - This PYRAMID concept is applicable because Data Fusion is classified within the PYRAMID concept as the component responsible for interpreting evidence and identifying and characterising objects, based on that interpretation.

Extensions

- Algorithms for generating Fused_Objects are likely to vary in terms of their behaviour and the data involved. As such it is suggested that Data Fusion is extended (see the Component Extensions PYRAMID concept) to cater for different algorithms.

5.4.2.14.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component could result in incorrect geolocation of targets and so result in weapons impacting locations not intended by the crew (e.g. if the locations of a Fused_Object was corrupted), resulting in unintended harm to third parties. This drives a DAL B indicative IDAL.

5.4.2.14.6.4 Security Considerations

The indicative security classification is SNEO.

This component executes fusion algorithms to improve the awareness of entities in the battlefield; both the algorithms and sources of evidence lead to a security classification of SNEO. The confidentiality of the algorithms and Evidence will need appropriate protection. Loss of integrity of source and fused information may lead to loss of data precedence, creation of "false" tracks and a confusing and degraded tactical picture, leading to a significant degradation of platform capability.

The component is expected to at least partially satisfy security related functions by:

- Retaining the highest **Classification of Information** fused to ensure it is handled appropriately. Where additional reclassification of fused data is a possibility, it is assumed that reclassification will require operator intervention and appropriate security measures will be in place.
- **Identifying Data Sources** and the Evidence they can provide.
- **Logging of Security Data** relating to classification changes, etc. for later examination.
- **Maintaining Audit Records** of fused data, especially for non-repudiation relating to assignation of allegiance (e.g. marking a track as hostile as a result of fusion).
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is considered unlikely to directly implement security enforcing functions, although it is dependent on the integrity of Evidence provided for fusing.

5.4.2.14.7 Services

5.4.2.14.7.1 Service Definitions

5.4.2.14.7.1.1 Fusion_Requirement

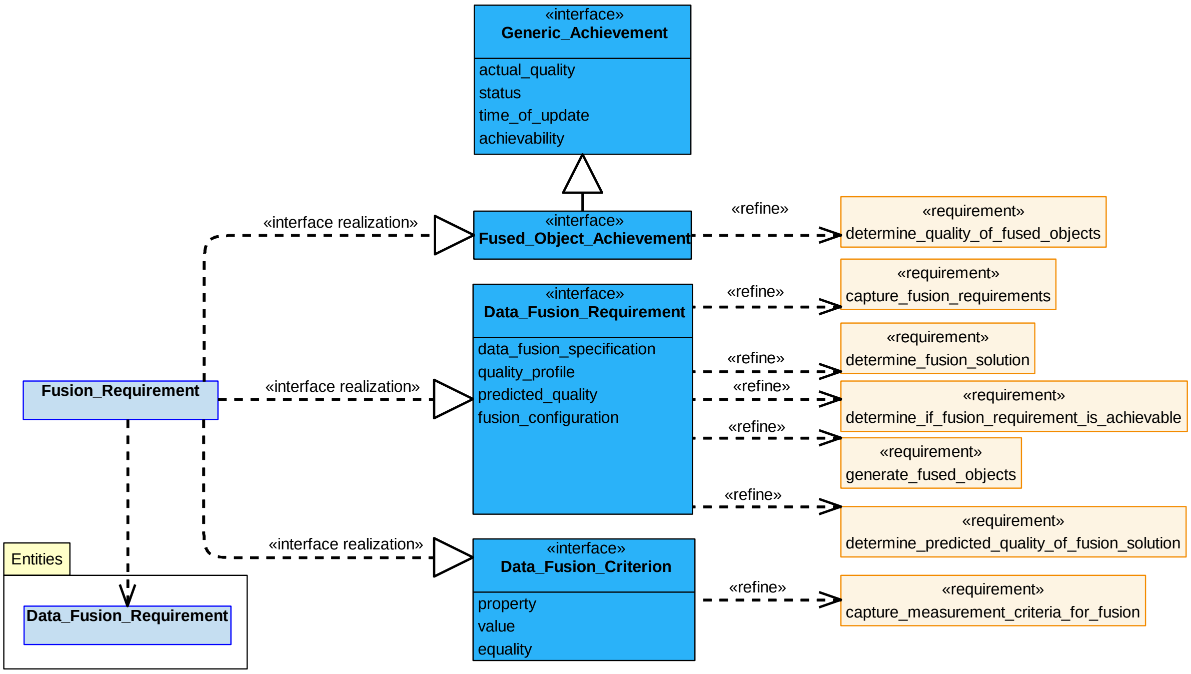


Figure 240: Fusion_Requirement Service Definition

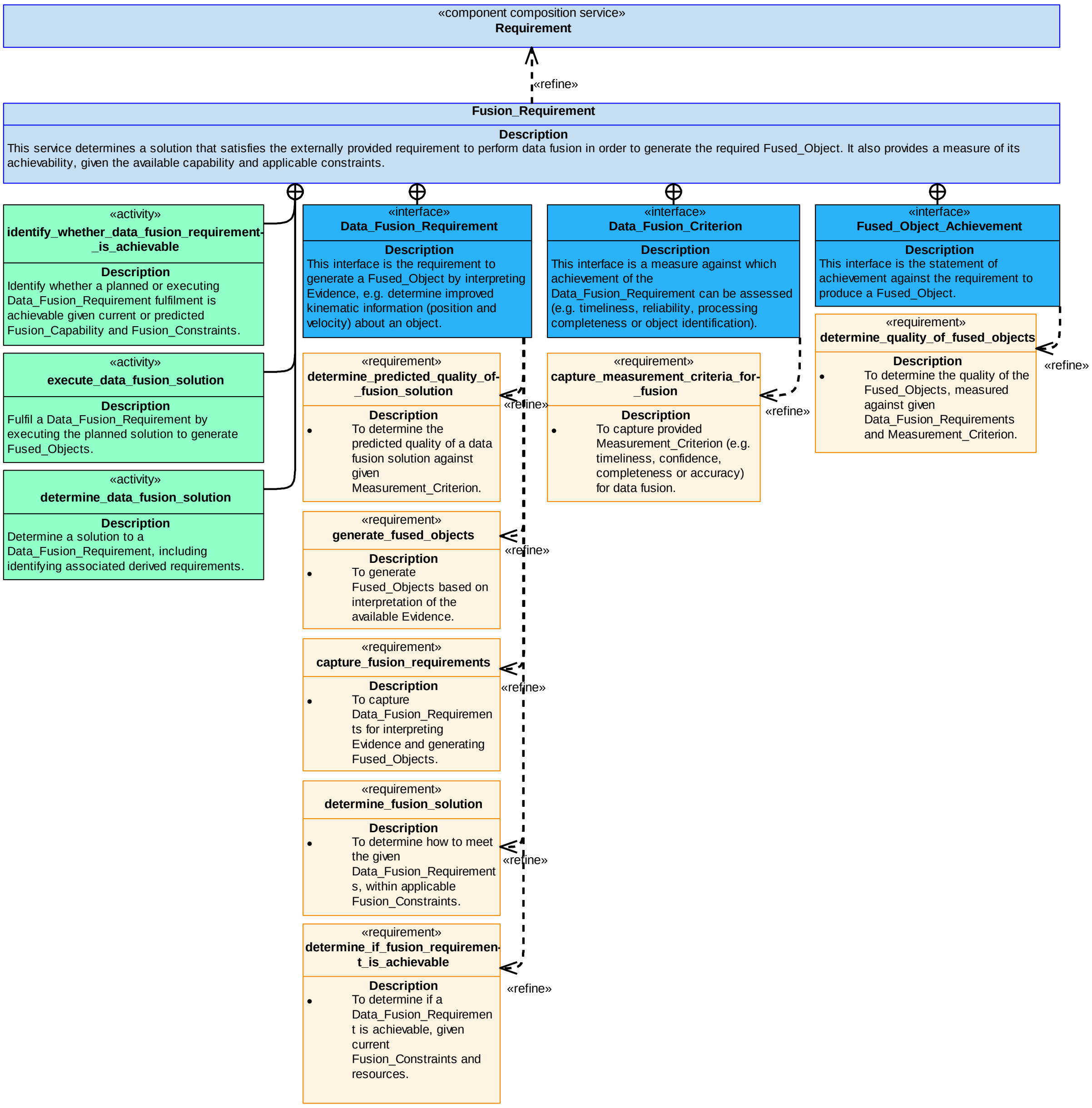

«component composition service»
Requirement
«refine»
Fusion_Requirement
Description
This service determines a solution that satisfies the externally provided requirement to perform data fusion in order to generate the required Fused_Object. It also provides a measure of its achievability, given the available capability and applicable constraints.
«activity»
identify_whether_data_fusion_requirement_is_achievable
Description
Identify whether a planned or executing Data_Fusion_Requirement fulfilment is achievable given current or predicted Fusion_Capability and Fusion_Constraints.
«activity»
execute_data_fusion_solution
Description
Fulfil a Data_Fusion_Requirement by executing the planned solution to generate Fused_Objects.
«activity»
determine_data_fusion_solution
Description
Determine a solution to a Data_Fusion_Requirement, including identifying associated derived requirements.
«interface»
Data_Fusion_Requirement
Description
This interface is the requirement to generate a Fused_Object by interpreting Evidence, e.g. determine improved kinematic information (position and velocity) about an object.
«interface»
Data_Fusion_Criterion
Description
This interface is a measure against which achievement of the Data_Fusion_Requirement can be assessed (e.g. timeliness, reliability, processing completeness or object identification).
«interface»
Fused_Object_Achievement
Description
This interface is the statement of achievement against the requirement to produce a Fused_Object.
«requirement»
determine_quality_of_fused_objects
Description
To determine the quality of the Fused_Objects, measured against given Data_Fusion_Requirements and Measurement_Criterion.
«refine»
«requirement»
determine_predicted_quality_of_fusion_solution
Description
To determine the predicted quality of a data fusion solution against given Measurement_Criterion.
«refine»
«requirement»
capture_measurement_criteria_for_fusion
Description
To capture provided Measurement_Criterion (e.g. timeliness, confidence, completeness or accuracy) for data fusion.
«refine»
«requirement»
generate_fused_objects
Description
To generate Fused_Objects based on interpretation of the available Evidence.
«refine»
«requirement»
capture_fusion_requirements
Description
To capture Data_Fusion_Requirements for interpreting Evidence and generating Fused_Objects.
«refine»
«requirement»
determine_fusion_solution
Description
To determine how to meet the given Data_Fusion_Requirements, within applicable Fusion_Constraints.
«refine»
«requirement»
determine_if_fusion_requirement_is_achievable
Description
To determine if a Data_Fusion_Requirement is achievable, given current Fusion_Constraints and resources.
«refine»

Figure 241: Fusion_Requirement Service Policy

Fusion_Requirement

This service determines a solution that satisfies the externally provided requirement to perform data fusion in order to generate the required Fused_Object. It also provides a measure of its achievability, given the available capability and applicable constraints.

Interfaces

Data_Fusion_Requirement

This interface is the requirement to generate a Fused_Object by interpreting Evidence, e.g. determine improved kinematic information (position and velocity) about an object.

Attributes

data_fusion_specification	This specifies the requirement to generate Fused_Object(s). For example, a requirement could be placed on the component to generate a Fused_Object_Type such as a fused ESM track.
quality_profile	Acceptable quality thresholds (i.e. minimum and ideal) to be obtained by the Evidence processing.
predicted_quality	How well the proposed Fused_Object to be generated is predicted to satisfy the Data_Fusion_Requirement.
fusion_configuration	Specified control and threshold parameter values used to adjust the behaviour of the Evidence processing in order to meet the Data_Fusion_Requirement.

Data_Fusion_Criterion

This interface is a measure against which achievement of the Data_Fusion_Requirement can be assessed (e.g. timeliness, reliability, processing completeness or object identification).

Attributes

property	The criterion property to be measured, e.g. a measure of effectiveness such as the required confidence level for Fused_Object generation.
value	The amount related to the property to be measured.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Fused_Object_Achievement

This interface is the statement of achievement against the requirement to produce a Fused_Object.

Activities

determine_data_fusion_solution

Determine a solution to a Data_Fusion_Requirement, including identifying associated derived requirements.

execute_data_fusion_solution

Fulfil a Data_Fusion_Requirement by executing the planned solution to generate Fused_Objects.

identify_whether_data_fusion_requirement_is_achievable

Identify whether a planned or executing Data_Fusion_Requirement fulfilment is achievable given current or predicted Fusion_Capability and Fusion_Constraints.

5.4.2.14.7.1.2 Fused_Object_Information

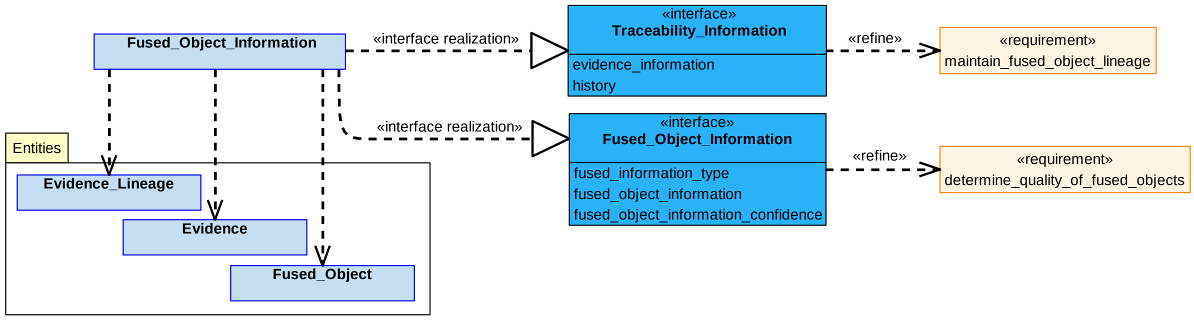


Figure 242: Fused_Object_Information Service Definition

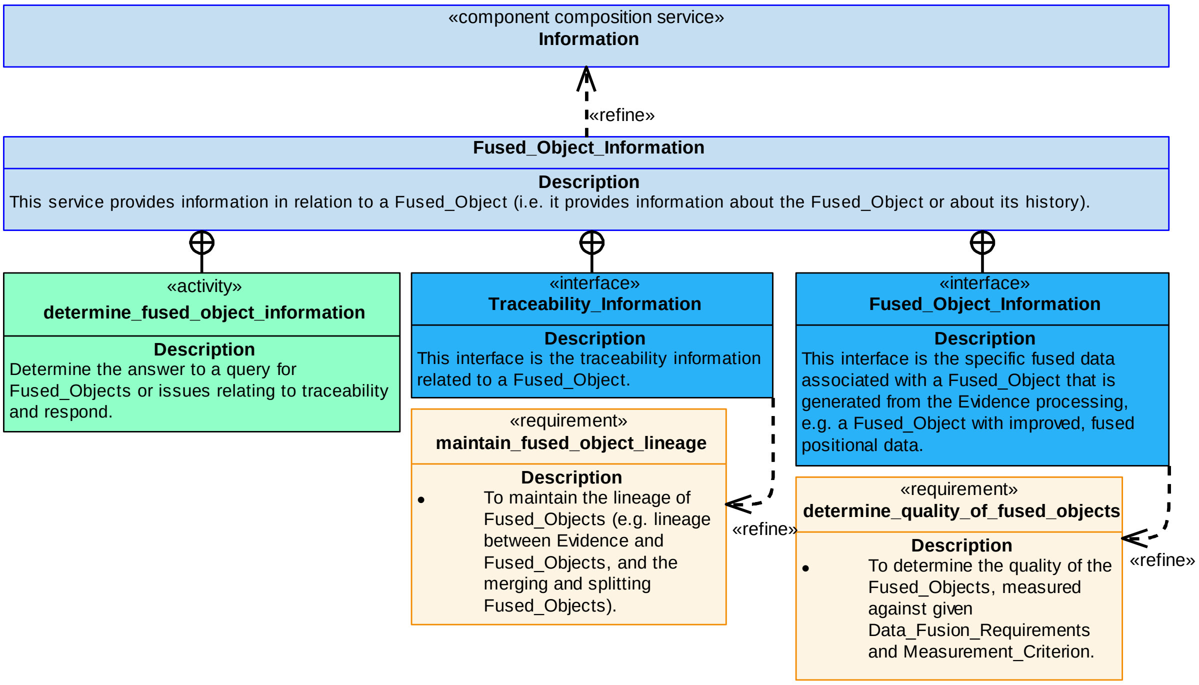

«component composition service»
Information
«refine»
Fused_Object_Information
Description
This service provides information in relation to a Fused_Object (i.e. it provides information about the Fused_Object or about its history).
«activity»
determine_fused_object_information
Description
Determine the answer to a query for Fused_Objects or issues relating to traceability and respond.
«interface»
Traceability_Information
Description
This interface is the traceability information related to a Fused_Object.
«interface»
Fused_Object_Information
Description
This interface is the specific fused data associated with a Fused_Object that is generated from the Evidence processing, e.g. a Fused_Object with improved, fused positional data.
«requirement»
maintain_fused_object_lineage
Description
To maintain the lineage of Fused_Objects (e.g. lineage between Evidence and Fused_Objects, and the merging and splitting Fused_Objects).
«refine»
«requirement»
determine_quality_of_fused_objects
Description
To determine the quality of the Fused_Objects, measured against given Data_Fusion_Requirements and Measurement_Criterion.
«refine»

Figure 243: Fused_Object_Information Service Policy

Fused_Object_Information

This service provides information in relation to a Fused_Object (i.e. it provides information about the Fused_Object or about its history).

Interfaces

Traceability_Information

This interface is the traceability information related to a Fused_Object.

Attributes

evidence_information	The Evidence information which has been used to help generate a given Fused_Object.
history	Historical information relating to the Fused_Object such as identifying when a Fused_Object has been created, merged or split.

Fused_Object_Information

This interface is the specific fused data associated with a Fused_Object that is generated from the Evidence processing, e.g. a Fused_Object with improved, fused positional data.

Attributes

fused_information_type	The category of fusion information that describes the generated Fused_Object, e.g. kinematic level fusion or track fusion.
fused_object_information	The contextually enhanced fused data that is provided as part of the Fused_Object, e.g. the Fused_Object's fused position data.
fused_object_information_confidence	An assessment of the confidence of the information being provided based on knowledge of the information source(s), e.g. for a geo-located Fused_Object, this could be providing the overall estimated error as a result of directional finding measurement errors captured as part of the Evidence.

Activity

determine_fused_object_information

Determine the answer to a query for Fused_Objects or issues relating to traceability and respond.

5.4.2.14.7.1.3 Evidence

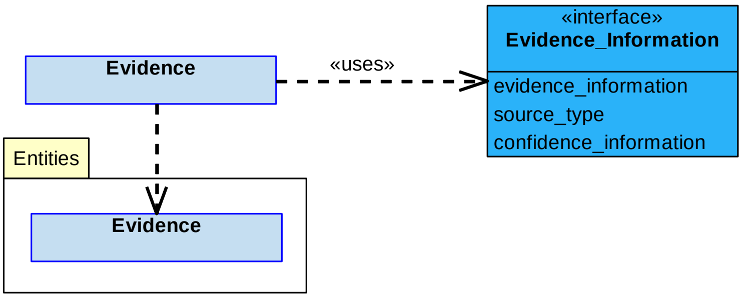


Figure 244: Evidence Service Definition

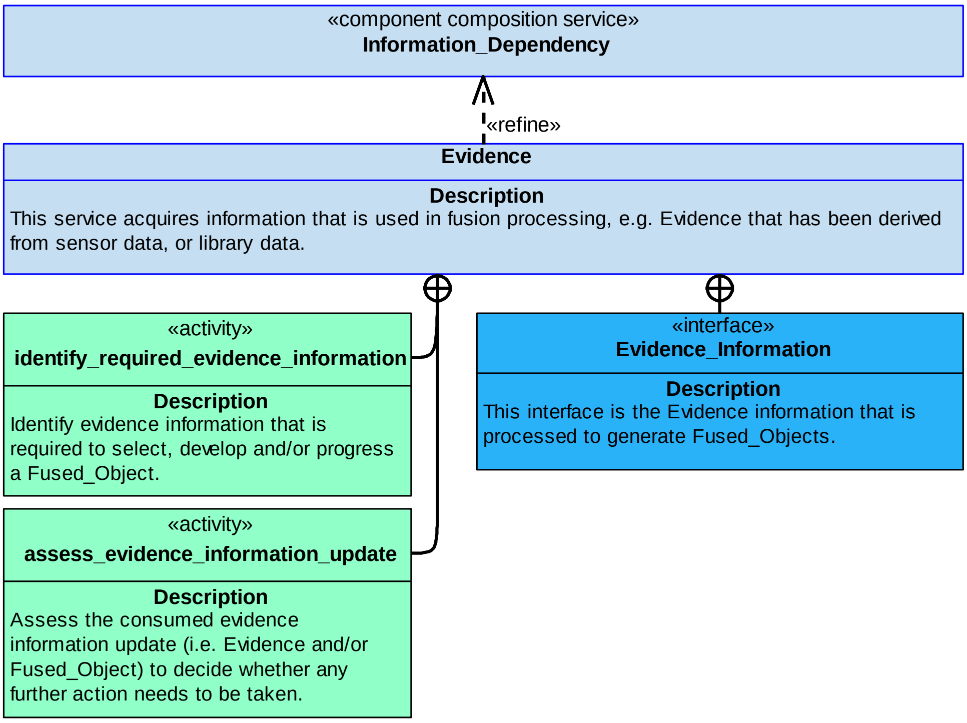

«component composition service»
Information_Dependency
«refine»
Evidence
Description
This service acquires information that is used in fusion processing, e.g. Evidence that has been derived from sensor data, or library data.
«activity»
identify_required_evidence_information
Description
Identify evidence information that is required to select, develop and/or progress a Fused_Object.
«activity»
assess_evidence_information_update
Description
Assess the consumed evidence information update (i.e. Evidence and/or Fused_Object) to decide whether any further action needs to be taken.
«interface»
Evidence_Information
Description
This interface is the Evidence information that is processed to generate Fused_Objects.

Figure 245: Evidence Service Policy

Evidence

This service acquires information that is used in fusion processing, e.g. Evidence that has been derived from sensor data, or library data.

Interface

Evidence_Information

This interface is the Evidence information that is processed to generate Fused_Objects.

Attributes

evidence_information	The Evidence that Data Fusion must account for in its solution (e.g. radar detection plot positions).
source_type	The source type that is providing the Evidence.
confidence_information	Describes the confidence of the Evidence being utilised as part of the fusion process (i.e. estimating the error levels in the information being provided).

Activities

identify_required_evidence_information

Identify evidence information that is required to select, develop and/or progress a Fused_Object.

assess_evidence_information_update

Assess the consumed evidence information update (i.e. Evidence and/or Fused_Object) to decide whether any further action needs to be taken.

5.4.2.14.7.1.4 Environmental_Data

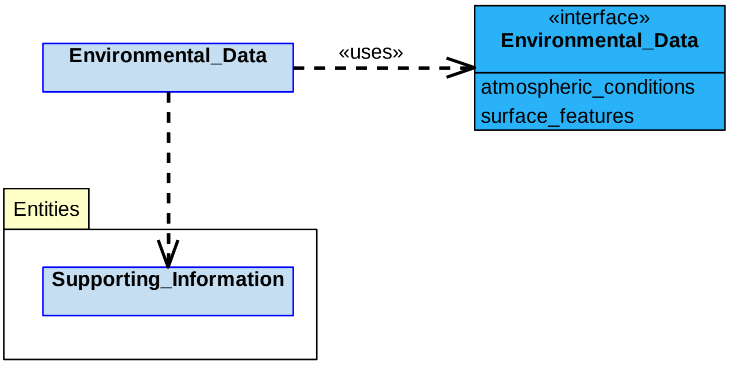
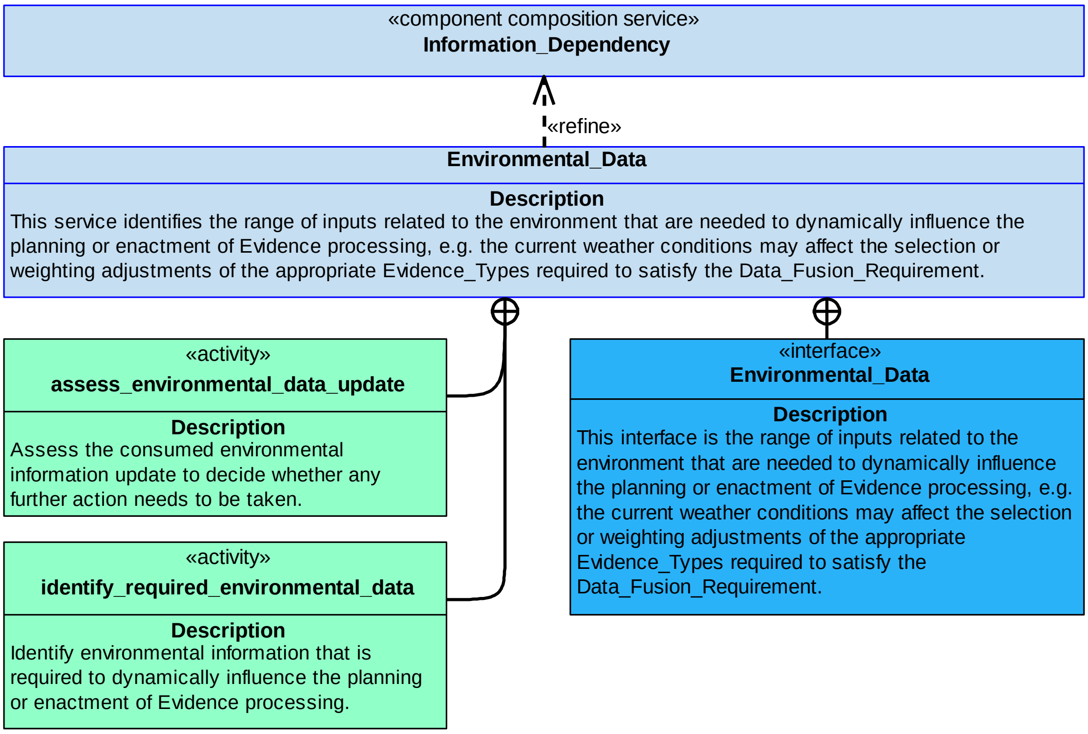


Figure 246: Environmental_Data Service Definition


«component composition service»
Information_Dependency
«refine»
Environmental_Data
«activity»
assess_environmental_data_update
«activity»
identify_required_environmental_data
«interface»
Environmental_Data

Environmental_Data

Description

This service identifies the range of inputs related to the environment that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current weather conditions may affect the selection or weighting adjustments of the appropriate Evidence_Types required to satisfy the Data_Fusion_Requirement.

assess_environmental_data_update

Description

Assess the consumed environmental information update to decide whether any further action needs to be taken.

identify_required_environmental_data

Description

Identify environmental information that is required to dynamically influence the planning or enactment of Evidence processing.

Environmental_Data

Description

This interface is the range of inputs related to the environment that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current weather conditions may affect the selection or weighting adjustments of the appropriate Evidence_Types required to satisfy the Data_Fusion_Requirement.

Figure 247: Environmental_Data Service Policy

Environmental_Data

This service identifies the range of inputs related to the environment that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current weather conditions may affect the selection or weighting adjustments of the appropriate Evidence_Types required to satisfy the Data_Fusion_Requirement.

Interface

Environmental_Data

This interface is the range of inputs related to the environment that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current weather conditions may affect the selection or weighting adjustments of the appropriate Evidence_Types required to satisfy the Data_Fusion_Requirement.

Attributes

atmospheric_conditions	Current weather conditions and features.
surface_features	Information describing surfaces and features in the environment. This may include land terrain or other environments.

Activities

assess_environmental_data_update

Assess the consumed environmental information update to decide whether any further action needs to be taken.

identify_required_environmental_data

Identify environmental information that is required to dynamically influence the planning or enactment of Evidence processing.

5.4.2.14.7.1.5 Vehicle_Data

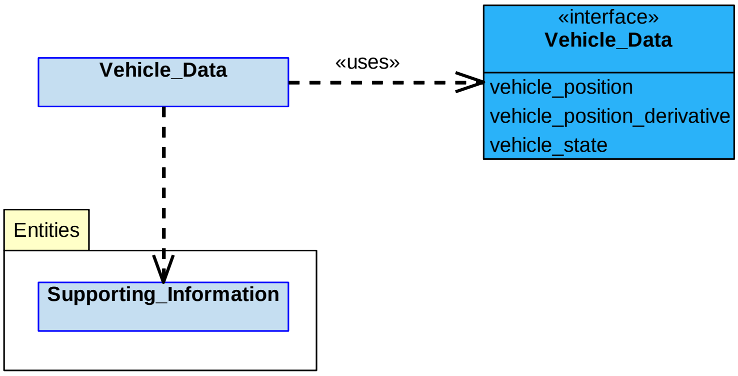


Figure 248: Vehicle_Data Service Definition

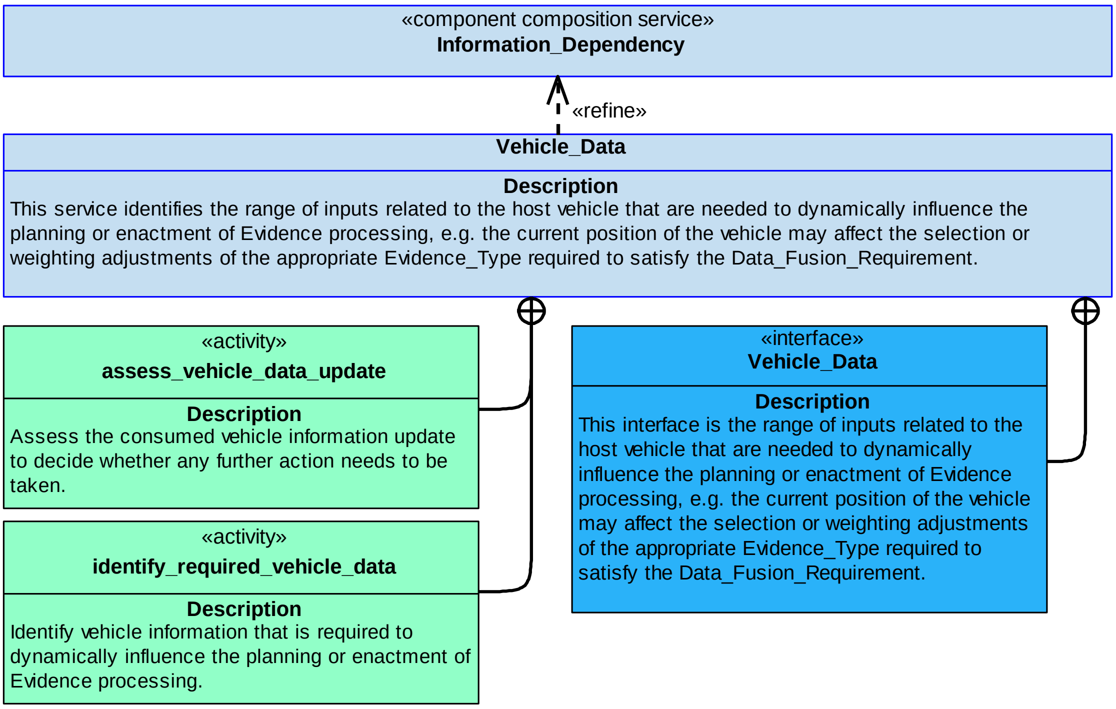

«component composition service»
Information_Dependency
«refine»
Vehicle_Data
Description
This service identifies the range of inputs related to the host vehicle that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of the vehicle may affect the selection or weighting adjustments of the appropriate Evidence_Type required to satisfy the Data_Fusion_Requirement.
«activity»
assess_vehicle_data_update
Description
Assess the consumed vehicle information update to decide whether any further action needs to be taken.
«activity»
identify_required_vehicle_data
Description
Identify vehicle information that is required to dynamically influence the planning or enactment of Evidence processing.
«interface»
Vehicle_Data
Description
This interface is the range of inputs related to the host vehicle that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of the vehicle may affect the selection or weighting adjustments of the appropriate Evidence_Type required to satisfy the Data_Fusion_Requirement.

Figure 249: Vehicle_Data Service Policy

Vehicle_Data

This service identifies the range of inputs related to the host vehicle that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of the vehicle may affect the selection or weighting adjustments of the appropriate Evidence_Type required to satisfy the Data_Fusion_Requirement.

Interface

Vehicle_Data

This interface is the range of inputs related to the host vehicle that are needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of the vehicle may affect the selection or weighting adjustments of the appropriate Evidence_Type required to satisfy the Data_Fusion_Requirement.

Attributes

vehicle_position	The location and orientation of the host vehicle.
vehicle_position_derivative	A derivative of the host vehicle’s location or orientation (e.g. velocity or angular velocity).
vehicle_state	A state of the host vehicle that could affect Evidence processing (e.g. whether the vehicle is transmitting RF energy).

Activities

assess_vehicle_data_update

Assess the consumed vehicle information update to decide whether any further action needs to be taken.

identify_required_vehicle_data

Identify vehicle information that is required to dynamically influence the planning or enactment of Evidence processing.

5.4.2.14.7.1.6 Object_Data

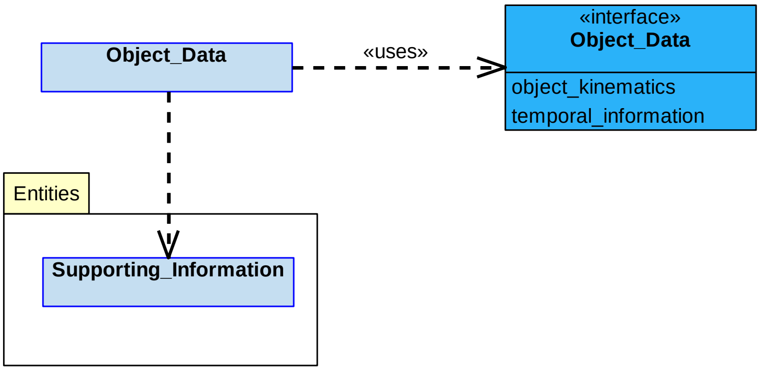


Figure 250: Object_Data Service Definition

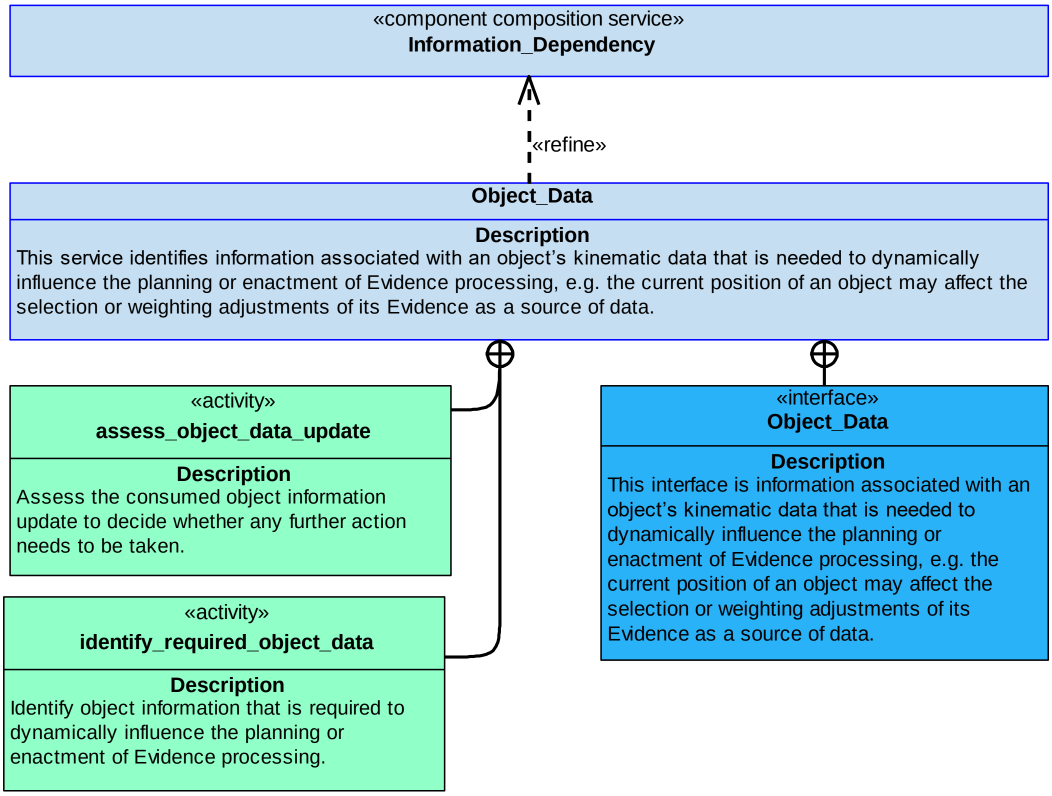

«component composition service»
Information_Dependency
«refine»
Object_Data
Description
This service identifies information associated with an object's kinematic data that is needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of an object may affect the selection or weighting adjustments of its Evidence as a source of data.
«activity»
assess_object_data_update
Description
Assess the consumed object information update to decide whether any further action needs to be taken.
«activity»
identify_required_object_data
Description
Identify object information that is required to dynamically influence the planning or enactment of Evidence processing.
«interface»
Object_Data
Description
This interface is information associated with an object's kinematic data that is needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of an object may affect the selection or weighting adjustments of its Evidence as a source of data.

Figure 251: Object_Data Service Policy

Object_Data

This service identifies information associated with an object's kinematic data that is needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of an object may affect the selection or weighting adjustments of its Evidence as a source of data.

Interface

Object_Data

This interface is information associated with an object's kinematic data that is needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position of an object may affect the selection or weighting adjustments of its Evidence as a source of data.

Attributes

object_kinematics	Information relating to the motion of another object (e.g. heading, speed or acceleration).
temporal_information	Timing of object availability for Evidence updates and low latency synchronization of Evidence.

Activities

assess_object_data_update

Assess the consumed object information update to decide whether any further action needs to be taken.

identify_required_object_data

Identify object information that is required to dynamically influence the planning or enactment of Evidence processing.

5.4.2.14.7.1.7 Constraint

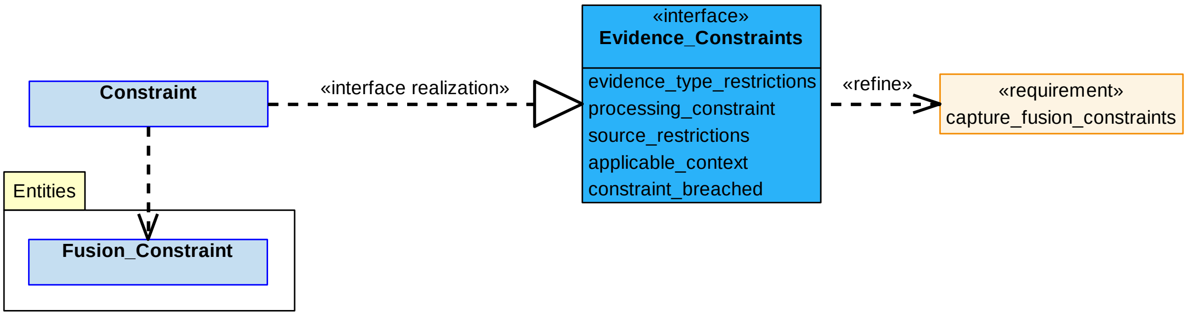


Figure 252: Constraint Service Definition

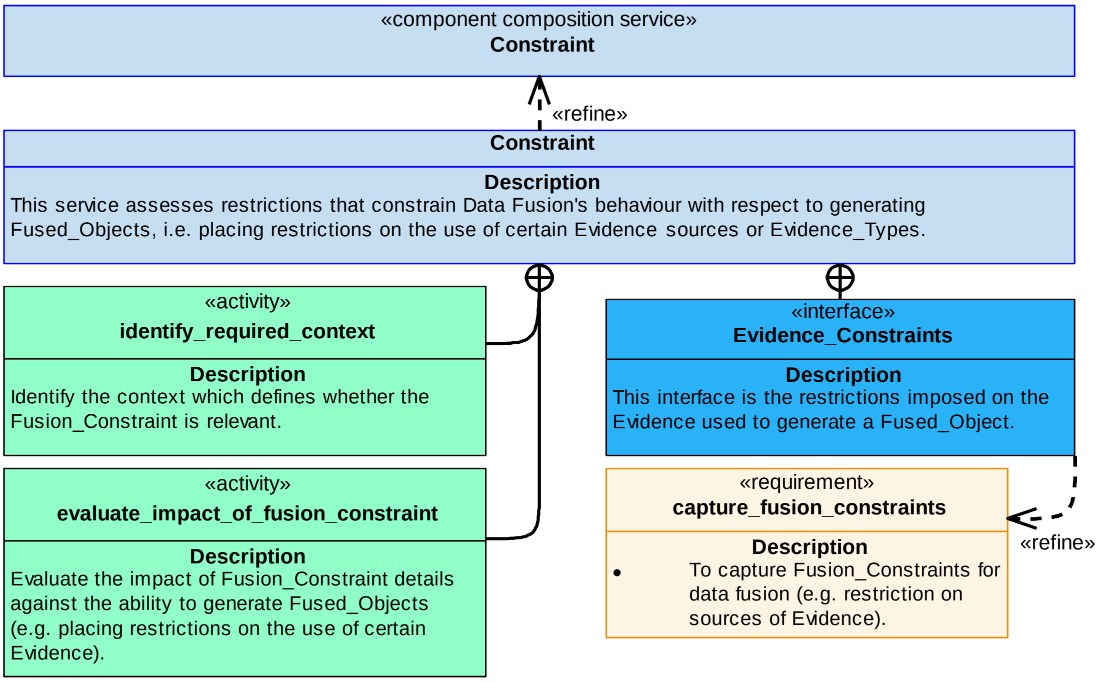

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses restrictions that constrain Data Fusion's behaviour with respect to generating Fused_Objects, i.e. placing restrictions on the use of certain Evidence sources or Evidence_Types.
«activity»
identify_required_context
Description
Identify the context which defines whether the Fusion_Constraint is relevant.
«activity»
evaluate_impact_of_fusion_constraint
Description
Evaluate the impact of Fusion_Constraint details against the ability to generate Fused_Objects (e.g. placing restrictions on the use of certain Evidence).
«interface»
Evidence_Constraints
Description
This interface is the restrictions imposed on the Evidence used to generate a Fused_Object.
«requirement»
capture_fusion_constraints
Description
• To capture Fusion_Constraints for data fusion (e.g. restriction on sources of Evidence).
«refine»

Figure 253: Constraint Service Policy

Constraint

This service assesses restrictions that constrain Data Fusion's behaviour with respect to generating Fused_Objects, i.e. placing restrictions on the use of certain Evidence sources or Evidence_Types.

Interface

Evidence_Constraints

This interface is the restrictions imposed on the Evidence used to generate a Fused_Object.

Attributes

evidence_type_restrictions	A constraint that limits the Evidence_Type that can be used, e.g. due to the evidence being deemed unreliable.
processing_constraint	A constraint limiting the type of processing that can be utilised for a given Evidence_Type.
source_restrictions	A constraint that limits the sources of Evidence that can be used.
applicable_context	The context in which the constraint is applicable.
constraint_breached	Whether a Data Fusion component's constraint has been inadvertently breached due to external factors such as unreliable Evidence.

Activities

evaluate_impact_of_fusion_constraint

Evaluate the impact of Fusion_Constraint details against the ability to generate Fused_Objects (e.g. placing restrictions on the use of certain Evidence).

identify_required_context

Identify the context which defines whether the Fusion_Constraint is relevant.

5.4.2.14.7.1.8 Fusion_Capability

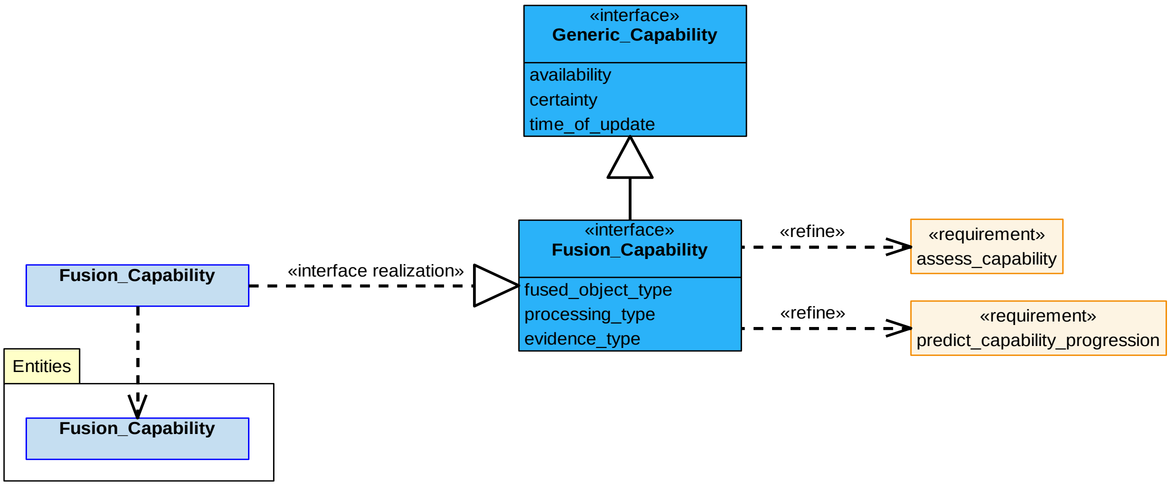


Figure 254: Fusion_Capability Service Definition

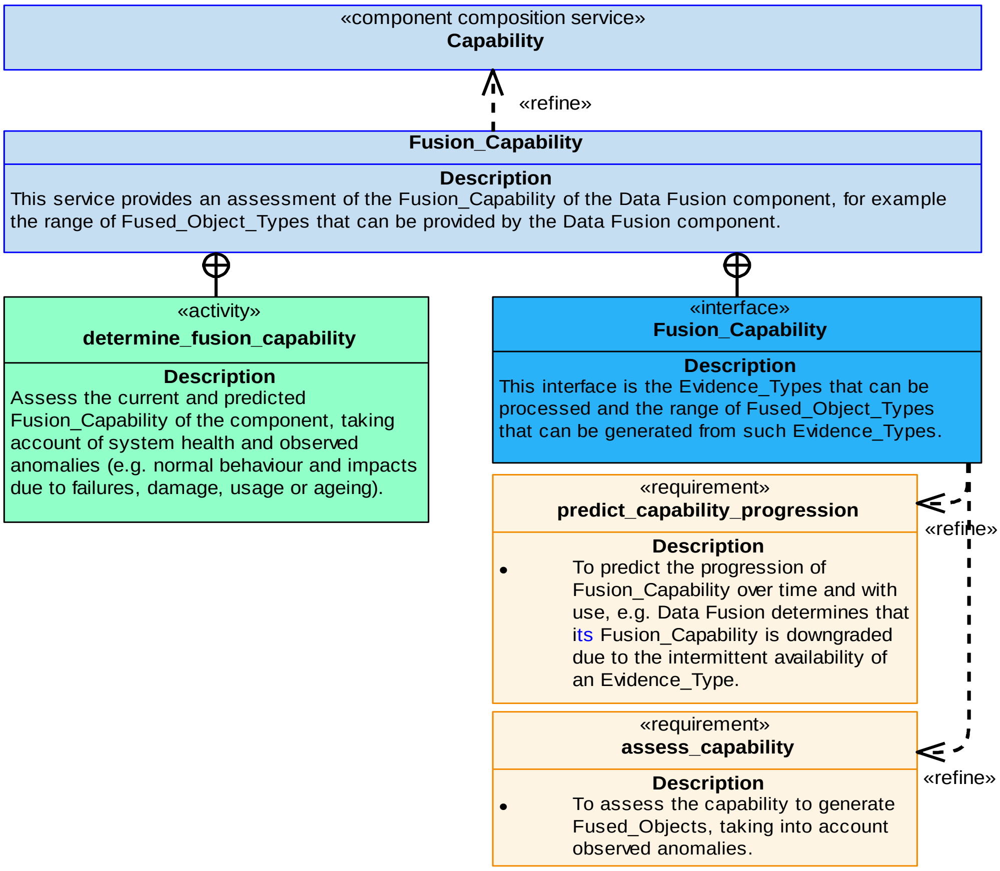

«component composition service»
Capability
«refine»
Fusion_Capability
Description
This service provides an assessment of the Fusion_Capability of the Data Fusion component, for example the range of Fused_Object_Types that can be provided by the Data Fusion component.
«activity»
determine_fusion_capability
Description
Assess the current and predicted Fusion_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Fusion_Capability
Description
This interface is the Evidence_Types that can be processed and the range of Fused_Object_Types that can be generated from such Evidence_Types.
«requirement»
predict_capability_progression
Description
• To predict the progression of Fusion_Capability over time and with use, e.g. Data Fusion determines that its Fusion_Capability is downgraded due to the intermittent availability of an Evidence_Type.
«refine»
«requirement»
assess_capability
Description
• To assess the capability to generate Fused_Objects, taking into account observed anomalies.
«refine»

Figure 255: Fusion_Capability Service Policy

Fusion_Capability

This service provides an assessment of the Fusion_Capability of the Data Fusion component, for example the range of Fused_Object_Types that can be provided by the Data Fusion component.

Interface

Fusion_Capability

This interface is the Evidence_Types that can be processed and the range of Fused_Object_Types that can be generated from such Evidence_Types.

Attributes

fused_object_type	The range of Fused_Object_Types that can be generated.
processing_type	The Evidence processing types that the component provides as part of its Fusion Capability (e.g. association, correlation or state estimation).
evidence_type	The range of Evidence_Types that can be processed.

Activity

determine_fusion_capability

Assess the current and predicted Fusion_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.14.7.1.9 Fusion_Capability_Evidence

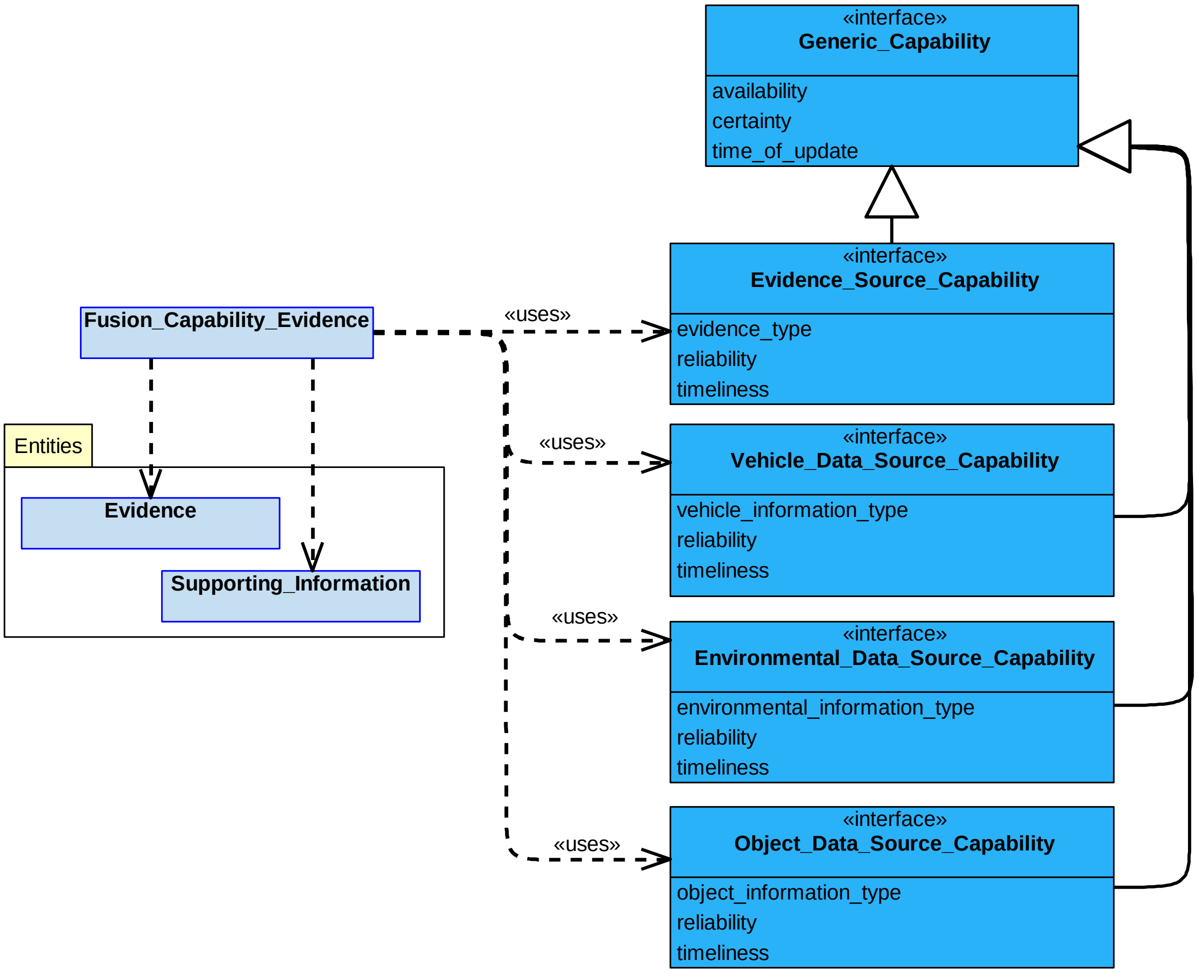


Figure 256: Fusion_Capability_Evidence Service Definition

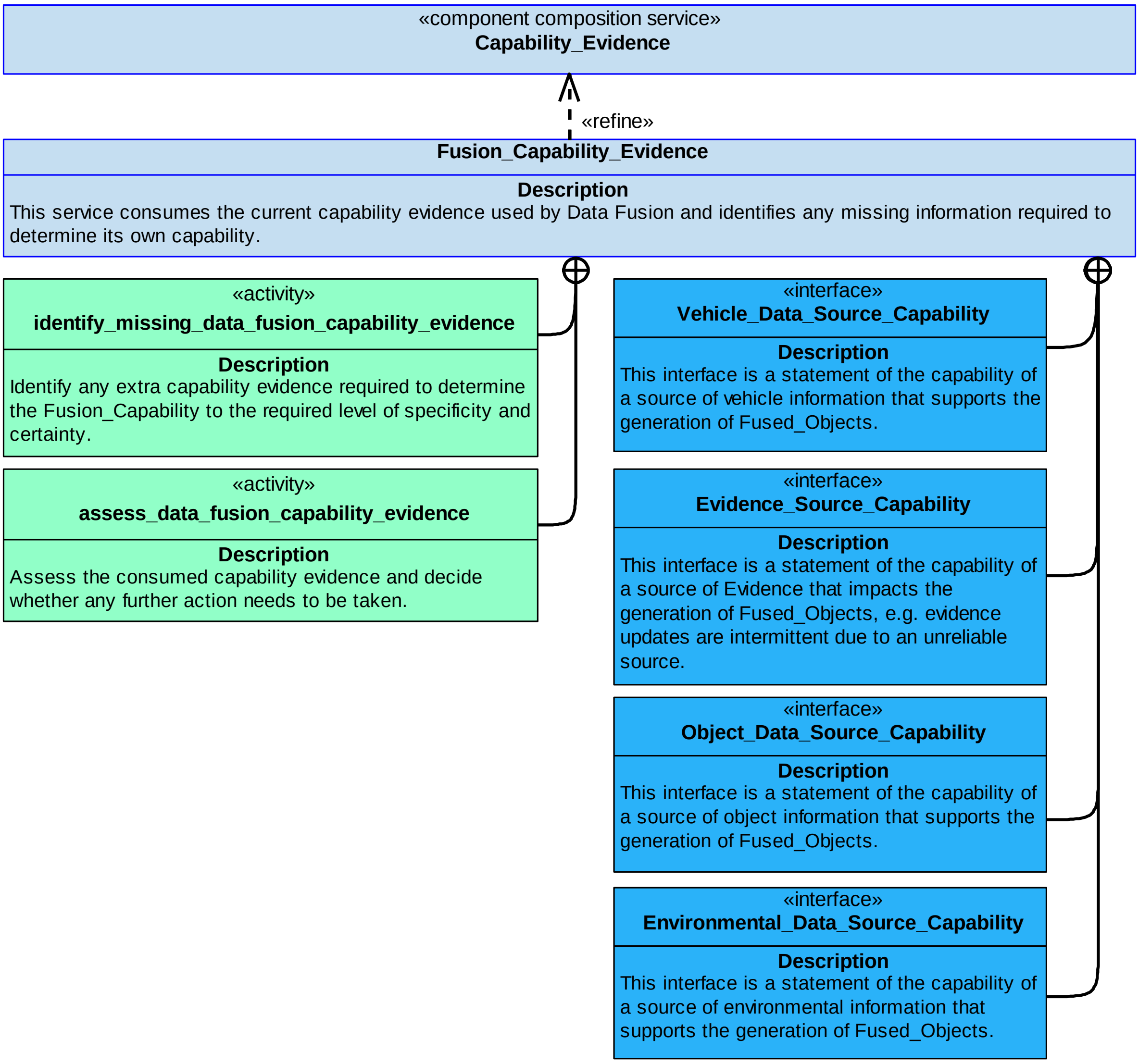

«component composition service»
Capability_Evidence
«refine»
Fusion_Capability_Evidence
Description
This service consumes the current capability evidence used by Data Fusion and identifies any missing information required to determine its own capability.
«activity»
identify_missing_data_fusion_capability_evidence
Description
Identify any extra capability evidence required to determine the Fusion_Capability to the required level of specificity and certainty.
«activity»
assess_data_fusion_capability_evidence
Description
Assess the consumed capability evidence and decide whether any further action needs to be taken.
«interface»
Vehicle_Data_Source_Capability
Description
This interface is a statement of the capability of a source of vehicle information that supports the generation of Fused_Objects.
«interface»
Evidence_Source_Capability
Description
This interface is a statement of the capability of a source of Evidence that impacts the generation of Fused_Objects, e.g. evidence updates are intermittent due to an unreliable source.
«interface»
Object_Data_Source_Capability
Description
This interface is a statement of the capability of a source of object information that supports the generation of Fused_Objects.
«interface»
Environmental_Data_Source_Capability
Description
This interface is a statement of the capability of a source of environmental information that supports the generation of Fused_Objects.

Figure 257: Fusion_Capability_Evidence Service Policy

Fusion_Capability_Evidence

This service consumes the current capability evidence used by Data Fusion and identifies any missing information required to determine its own capability.

Interfaces

Evidence_Source_Capability

This interface is a statement of the capability of a source of Evidence that impacts the generation of Fused_Objects, e.g. evidence updates are intermittent due to an unreliable source.

Attributes

evidence_type	The type of Evidence (e.g. characterisations of the sensor product or any previously generated Fused_Object_Type) that the source is capable of providing.
reliability	A capability measure that indicates whether the Evidence source is able to provide consistent/repeatable Evidence outputs.

timeliness	A capability measure that indicates whether the Evidence source is able to provide accessible and available Evidence in a timely manner.

Vehicle_Data_Source_Capability

This interface is a statement of the capability of a source of vehicle information that supports the generation of Fused_Objects.

Attributes

vehicle_information_type	The type of vehicle information that the source is capable of providing.
reliability	A capability measure that indicates whether the vehicle information source is able to provide consistent/repeatable vehicle information outputs.
timeliness	A capability measure that indicates whether the vehicle information source is able to provide accessible and available vehicle information in a timely manner.

Object_Data_Source_Capability

This interface is a statement of the capability of a source of object information that supports the generation of Fused_Objects.

Attributes

object_information_type	The type of object information that the source is capable of providing.
reliability	A capability measure that indicates whether the object information source is able to provide consistent/repeatable object information outputs.
timeliness	A capability measure that indicates whether the object information source is able to provide accessible and available object information in a timely manner.

Environmental_Data_Source_Capability

This interface is a statement of the capability of a source of environmental information that supports the generation of Fused_Objects.

Attributes

environmental_information_type	The type of environmental information that the source is capable of providing.
reliability	A capability measure that indicates whether the environmental information source is able to provide consistent/repeatable environmental information outputs.
timeliness	A capability measure that indicates whether the environmental information source is able to provide accessible and available environmental information in a timely manner.

Activities

assess_data_fusion_capability_evidence

Assess the consumed capability evidence and decide whether any further action needs to be taken.

identify_missing_data_fusion_capability_evidence

Identify any extra capability evidence required to determine the Fusion_Capability to the required level of specificity and certainty.

5.4.2.14.7.2 Service Dependencies

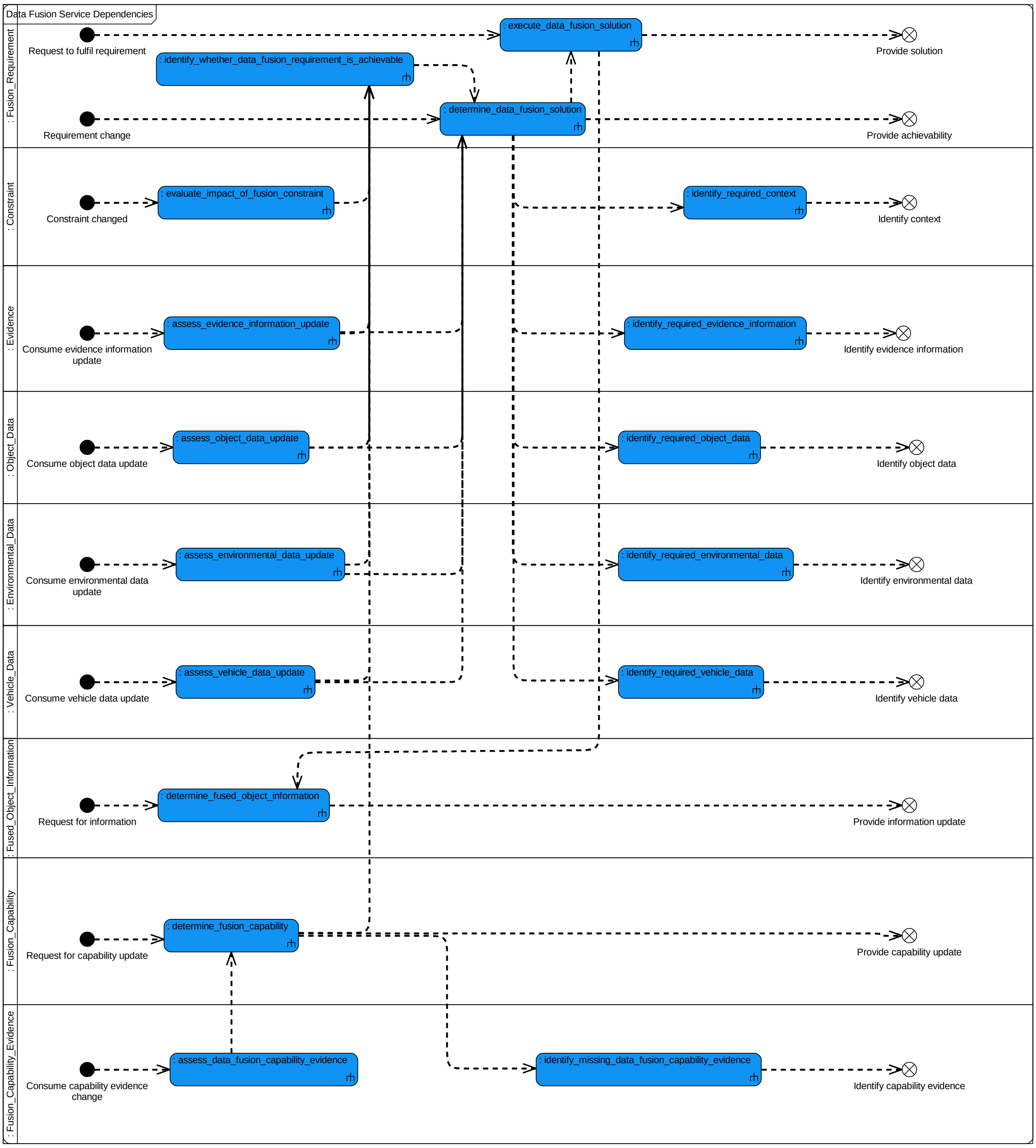

Data Fusion Service Dependencies
: Fusion_Requirement
Request to fulfil requirement
: execute_data_fusion_solution
Provide solution
: identify_whether_data_fusion_requirement_is_achievable
: determine_data_fusion_solution
Requirement change
Provide achievability
: Constraint
: evaluate_impact_of_fusion_constraint
Constraint changed
: identify_required_context
Identify context
: Evidence
: assess_evidence_information_update
Consume evidence information update
: identify_required_evidence_information
Identify evidence information
: Object_Data
: assess_object_data_update
Consume object data update
: identify_required_object_data
Identify object data
: Environmental_Data
: assess_environmental_data_update
Consume environmental data update
: identify_required_environmental_data
Identify environmental data
: Vehicle_Data
: assess_vehicle_data_update
Consume vehicle data update
: identify_required_vehicle_data
Identify vehicle data
: Fused_Object_Information
: determine_fused_object_information
Request for information
Provide information update
: Fusion_Capability
: determine_fusion_capability
Request for capability update
Provide capability update
: Fusion_Capability_Evidence
: assess_data_fusion_capability_evidence
Consume capability evidence change
: identify_missing_data_fusion_capability_evidence
Identify capability evidence

Figure 258: Data Fusion Service Dependencies

5.4.2.15 Destructive Effects

5.4.2.15.1 Role

The role of Destructive Effects is to determine the destructive effects achievable by weapons and the settings necessary to achieve the selected effect.

5.4.2.15.2 Overview

Control Architecture

Destructive Effects is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In response to a Destructive_Effect_Requirement (e.g. determine an appropriate weapon package capable of destroying a hardened aircraft shelter), this component will determine the Weapon_Resource (e.g. a single 1000lb bomb or two 500lb bombs) and Destructive_Effect_Settings (e.g. impact or airburst fusing) that would meet that Destructive_Effect_Requirement. It would then provide the Destructive_Effect_Setting to be applied to the weapon for the selected Destructive_Effect.

Note: This component does not control every aspect of a weapon equipment, only the destructive effect aspects (see exclusions in the Subject Matter Semantics).

Examples of Use

Destructive Effects will be needed as part of a system where a target has been acquired, and Destructive Effects is used to manage the destructive effect, for example:

- Determine which weapons would meet a Destructive_Effect_Requirement.
- Control fusing settings such as airburst or impact.

5.4.2.15.3 Service Summary

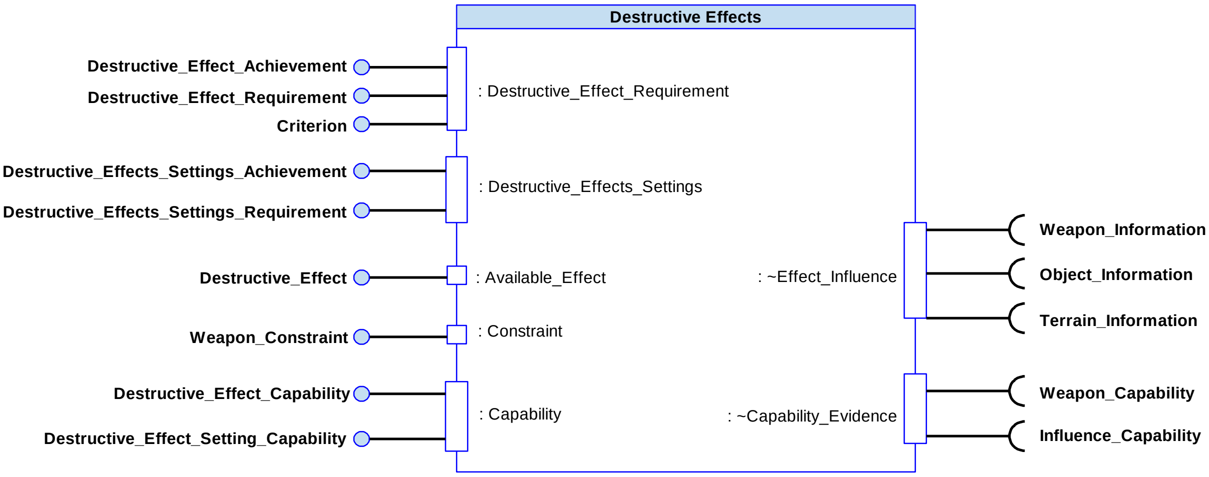

Destructive Effects
Destructive_Effect_Achievement
Destructive_Effect_Requirement
Criterion
: Destructive_Effect_Requirement
Destructive_Effects_Settings_Achievement
Destructive_Effects_Settings_Requirement
: Destructive_Effects_Settings
Destructive_Effect
: Available_Effect
: ~Effect_Influence
Weapon_Information
Object_Information
Terrain_Information
Weapon_Constraint
: Constraint
Destructive_Effect_Capability
Destructive_Effect_Setting_Capability
: Capability
: ~Capability_Evidence
Weapon_Capability
Influence_Capability

Figure 259: Destructive Effects Service Summary

5.4.2.15.4 Responsibilities

capture_requirements_for_destructive_effects

- To capture provided Destructive_Effect_Requirements.

capture_measurement_criteria_for_destructive_effects

- To capture provided Measurement_Criterion for the use of the Destructive_Effects.

capture_constraints_for_destructive_effects

- To capture provided Constraints that apply to the use of Weapon_Resources.

identify_whether_requirement_is_achievable

- To identify whether a Destructive_Effect_Requirement is achievable given current Weapon_Resources, Destructive_Effect_Settings and External_Influences.

determine_destructive_effect

- To determine a Destructive_Effect that will meet given Destructive_Effect_Requirements, and identify the associated Weapon_Type and Destructive_Effect_Settings.

identify_pre-conditions

- To identify Pre-conditions required to achieve a Destructive_Effect.

control_destructive_effect_settings

- To control the Destructive_Effect_Settings, e.g. set fusing mode.

determine_quality_of_destructive_effects_solution

- To determine the quality of a Destructive_Effect against given Measurement_Criterion.

determine_destructive_effects_capability

- To assess the Destructive_Effect_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_required_information

- To identify missing information which could improve the certainty or specificity of the capability assessment for Destructive Effects.

predict_capability_progression

- To predict the progression of the Destructive_Effect_Capability, over time and with use.

5.4.2.15.5 Subject Matter Semantics

The subject matter of Destructive Effects is the destructive effect of a Weapon_Resource.

Exclusions

The subject matter of Destructive Effects does not include:

- The aiming of Weapon_Resources towards an intended target.

- The release of Weapon_Resources.
- Arming (enable or disable) of Weapon_Resources (e.g. activating arming solenoids).
- The flight capability of powered Weapon_Resources.
- The communication capabilities of guided and controlled Weapon_Resources.

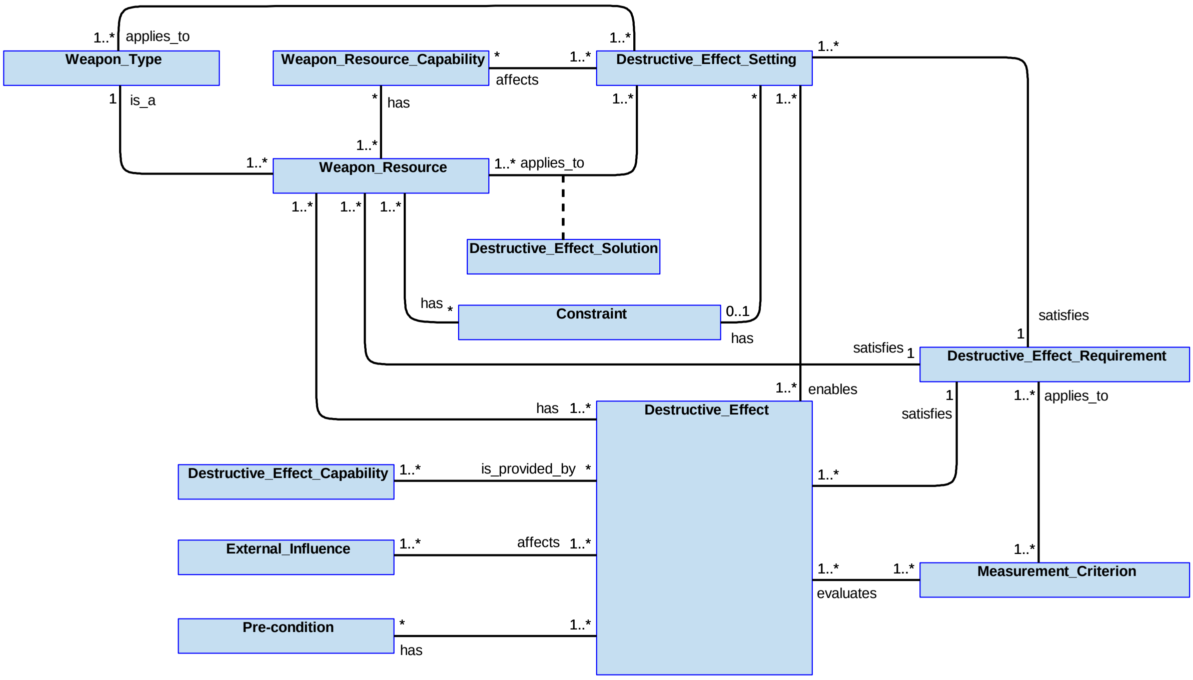


Figure 260: Destructive Effects Semantics

5.4.2.15.5.1 Entities

Constraint

An externally imposed restriction on the use of Weapon_Resources and the applied Destructive_Effect_Settings, e.g. maximum allowable yield limited.

Destructive_Effect

The scale of damage or harm that can be inflicted, e.g. the destruction of an entire building.

Destructive_Effect_Capability

The capability to provide a Destructive_Effect with the range of available Weapon_Resources and Destructive_Effect_Settings e.g. the ability to provide 'bunker busting' penetration with an air-to-surface missile.

Destructive_Effect_Requirement

A requirement to determine which Weapon_Resources and Destructive_Effect_Settings will achieve one or more Destructive_Effects, and to control the Destructive_Effect_Settings on a particular Weapon_Resource.

Destructive_Effect_Setting

A selection of required settings to produce a Destructive_Effect, e.g. setting a bomb to air burst mode.

Destructive_Effect_Solution

A solution to a Destructive_Effect_Requirement determined by selection of Weapon_Resources and the associated Destructive_Effect_Settings.

External_Influence

Something which has an external influence on one or more Destructive_Effects, e.g. a specific type of terrain or target, or information about the Exploiting Platform.

Measurement_Criterion

A criterion by which a Destructive_Effect is assessed, e.g. the amount of damage required.

Pre-condition

A condition which must be met before a Destructive_Effect can be achieved, e.g. prior to providing a weapon a required impact angle, it must be confirmed that the weapon will be released under conditions that allow it to guide itself to the target and achieve the desired impact angle.

Weapon_Resource

A specific instance of a weapon, e.g. an individual Paveway IV guided bomb.

Weapon_Resource_Capability

The capability of the available Weapon_Resources. This includes the different capabilities provided by the type of warhead attached to a particular Weapon_Resource and the serviceability of each Weapon_Resource.

Weapon_Type

A specific class of weapon that can be utilised, e.g. Paveway IV or Stormshadow.

5.4.2.15.6 Design Rationale

5.4.2.15.6.1 Assumptions

None.

5.4.2.15.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Destructive Effects:

- Interaction with Equipment - This is applicable because this component will provide information that is used when interacting with weapons which are external to the PYRAMID deployment.
- Data Driving - This is applicable as the weapon description items should be data driveable. There are numerous Weapon_Types each of which have associated

Destructive_Effect_Settings that can influence the Destructive_Effect; this component could have knowledge of these using data driving.

Extensions

- Extension components may be required. For example, to encapsulate differing approaches to determining a Destructive_Effect. This is because for different target types different aspects of destructive effect may be the element required by the measurement criteria, e.g. penetration or area of effect.

Exploitation Considerations

- Where an Exploiting Programme is using this component to determine a Destructive_Effect it is expected that a single instance of this component (covering multiple weapon types) would be deployed, rather than multiple instances.
- Where an Exploiting Programme is using this component to determine a Weapon_Resource_Capability it is not expected to determine the capability of the weapon control such as battery start, engine start, priming with navigation and target data, cryptographic data or sanitisation of data.

5.4.2.15.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component could cause weapons to be released with fusing settings (e.g. airburst instead of impact) that were not intended by the crew, nor accounted for in collateral damage estimates. This could result in unintended harm to third parties. This drives a DAL B indicative IDAL. N.B. This relies on the enabling / disabling of weapon arming being controlled by the Release Effecting component.

Where instances of this component contribute to hazards that are less severe or more reliance may be placed on other barriers to an accident, then the Exploiting Platform may require a less onerous DAL.

5.4.2.15.6.4 Security Considerations

The indicative security classification is SNEO.

This component manages the capability of the Exploiting Platform to deliver a destructive effect through the most appropriate selection of weapons and their settings for the intended target. Such offensive capability details are SNEO. Some intelligence data may be considered TS, with a corresponding change in confidentiality requirements. Loss of integrity or availability of this component will have a detrimental effect on the continued operational effectiveness of the platform, and are expected to need an appropriate degree of protection.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to changes in definitions of destructive effects, configurability of weapons, etc. for later forensic examination.

- **Maintaining Audit Records** of the weapons and their settings selected for use in order to support non-repudiation and audit, including for investigations into battle or collateral damage, etc.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.

The component is expected to at least partially satisfy security enforcing functions by:

- **Verifying Integrity of Data** for the requests for applying settings to weapons, ensuring they have come from an authorised source.

5.4.2.15.7 Services

5.4.2.15.7.1 Service Definitions

5.4.2.15.7.1.1 Destructive_Effect_Requirement

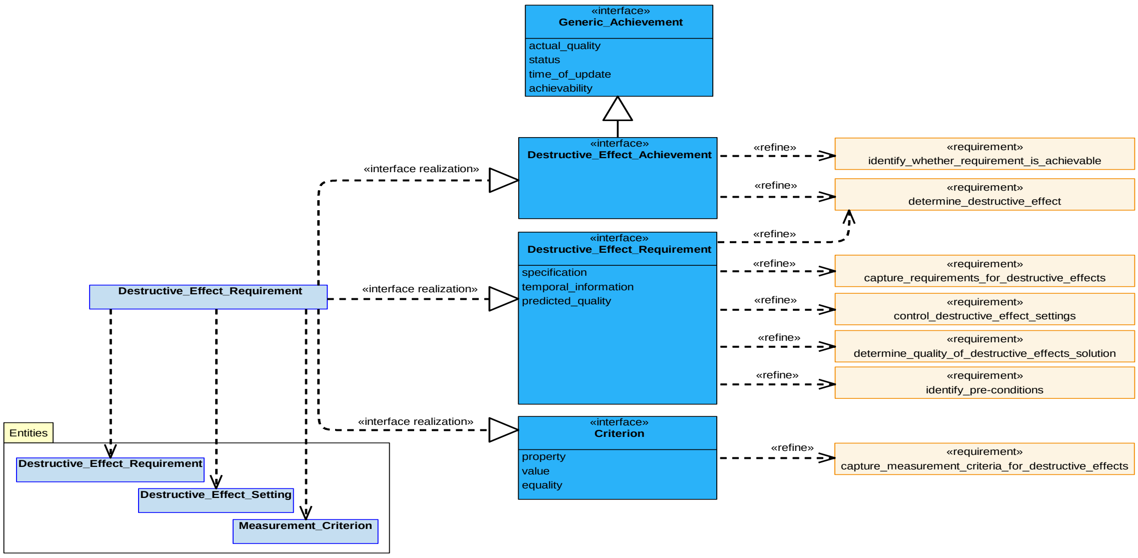
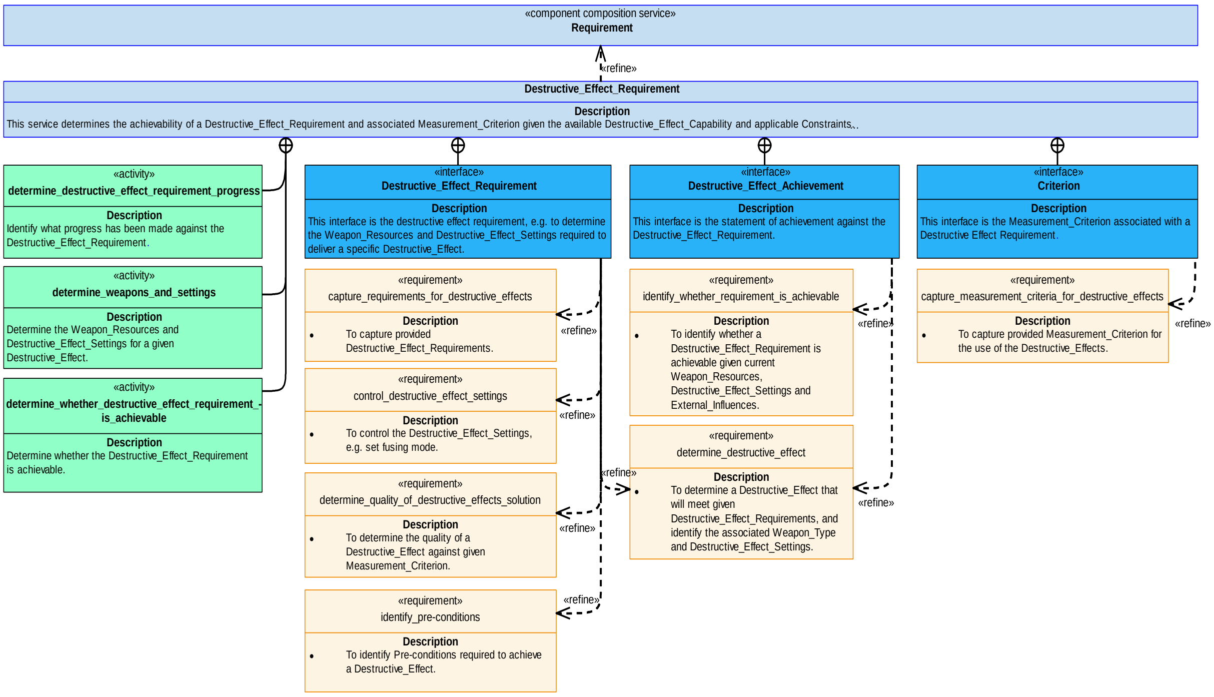


Figure 261: Destructive_Effect_Requirement Service Definition


«component composition service»
Requirement
«refine»
Destructive_Effect_Requirement
Description
This service determines the achievability of a Destructive_Effect_Requirement and associated Measurement_Criterion given the available Destructive_Effect_Capability and applicable Constraints..
«activity»
determine_destructive_effect_requirement_progress
Description
Identify what progress has been made against the Destructive_Effect_Requirement.
«activity»
determine_weapons_and_settings
Description
Determine the Weapon_Resources and Destructive_Effect_Settings for a given Destructive_Effect.
«activity»
determine_whether_destructive_effect_requirement_-is_achievable
Description
Determine whether the Destructive_Effect_Requirement is achievable.

This interface is the destructive effect requirement, e.g. to determine the Weapon_Resources and Destructive_Effect_Settings required to deliver a specific Destructive_Effect.

- To capture provided Destructive_Effect_Requirements.

- To control the Destructive_Effect_Settings, e.g. set fusing mode.

- To determine the quality of a Destructive_Effect against given Measurement_Criterion.

- To identify Pre-conditions required to achieve a Destructive_Effect.

This interface is the statement of achievement against the Destructive_Effect_Requirement.

- To identify whether a Destructive_Effect_Requirement is achievable given current Weapon_Resources, Destructive_Effect_Settings and External_Influences.

- To determine a Destructive_Effect that will meet given Destructive_Effect_Requirements, and identify the associated Weapon_Type and Destructive_Effect_Settings.

This interface is the Measurement_Criterion associated with a Destructive Effect Requirement.

- To capture provided Measurement_Criterion for the use of the Destructive_Effects.

Figure 262: Destructive_Effect_Requirement Service Policy

Destructive_Effect_Requirement

This service determines the achievability of a Destructive_Effect_Requirement and associated Measurement_Criterion given the available Destructive_Effect_Capability and applicable Constraints.

Interfaces

Destructive_Effect_Achievement

This interface is the statement of achievement against the Destructive_Effect_Requirement.

Destructive_Effect_Requirement

This interface is the destructive effect requirement, e.g. to determine the Weapon_Resources and Destructive_Effect_Settings required to deliver a specific Destructive_Effect.

Attributes

specification	The definition of the Destructive Effect Requirement, e.g. to determine the Weapon_Resources and Destructive_Effect_Settings that meet the need for an effect that will penetrate a hardened shelter to x metres.
temporal_information	Information covering timing, such as start and end times.
predicted_quality	How well the Destructive_Effect, Destructive_Effect_Settings and Weapon_Resources are predicted to satisfy the Destructive Effect Requirement.

Criterion

This interface is the Measurement_Criterion associated with a Destructive Effect Requirement.

Attributes

property	The property to be measured, e.g. amount of harm delivered by a Destructive_Effect.
value	The measured value of the property, e.g. the extent to which the Weapon_Resources and Destructive_Effect_Settings selected in response to the specification must meet the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

determine_destructive_effect_requirement_progress

Identify what progress has been made against the Destructive_Effect_Requirement.

determine_weapons_and_settings

Determine the Weapon_Resources and Destructive_Effect_Settings for a given Destructive_Effect.

determine_whether_destructive_effect_requirement_is_achievable

Determine whether the Destructive_Effect_Requirement is achievable.

5.4.2.15.7.1.2 Destructive_Effects_Settings

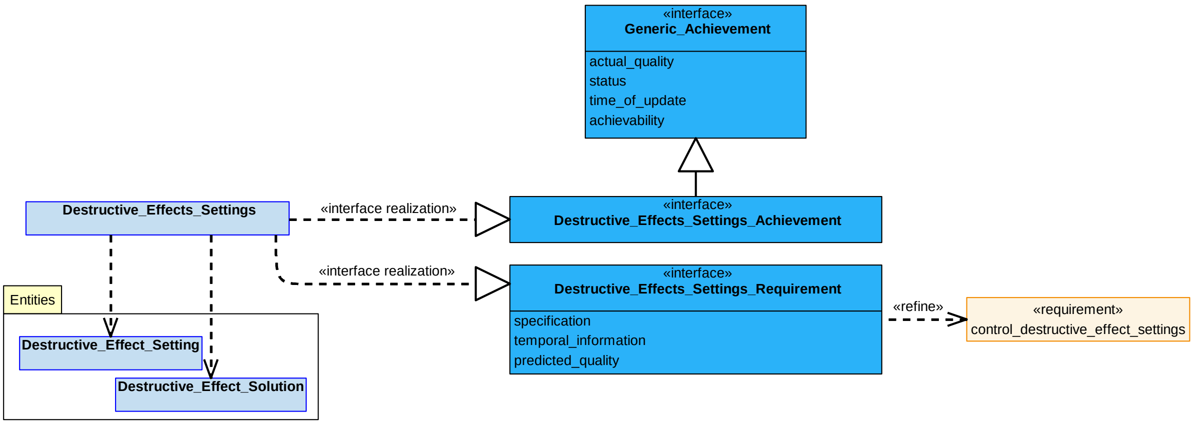


Figure 263: Destructive_Effects_Setting Service Definition

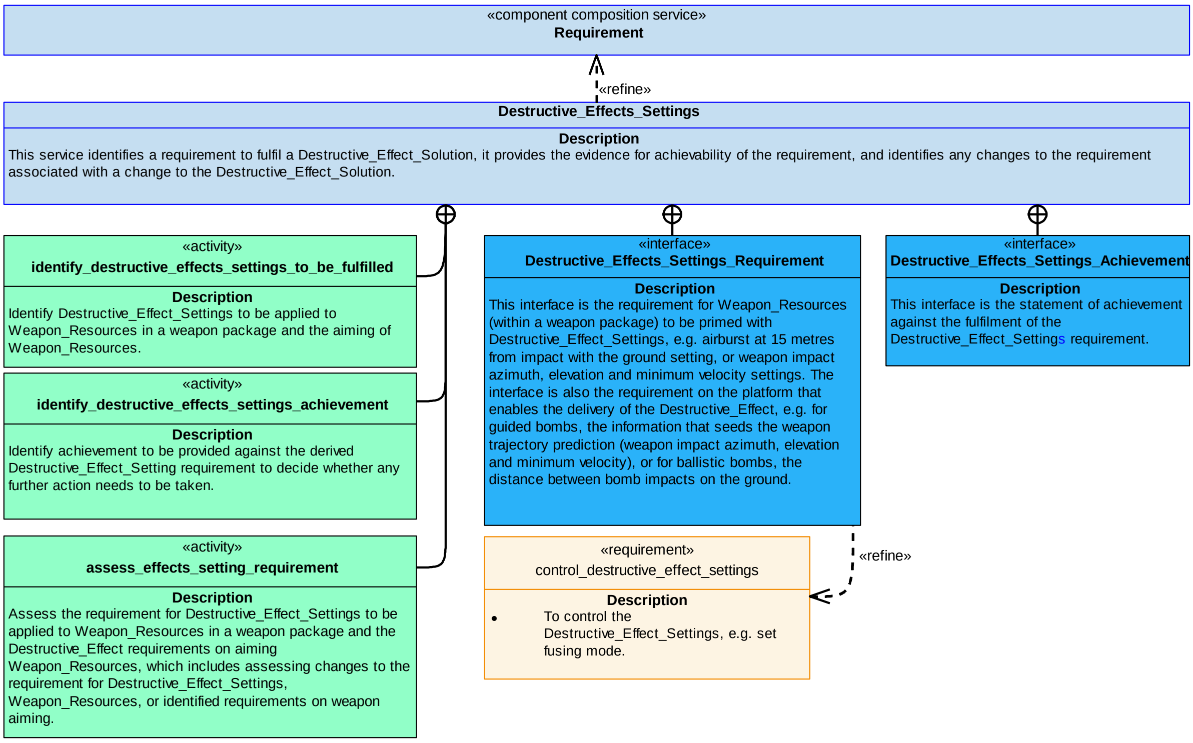

«component composition service»
Requirement
«refine»
Destructive_Effects_Settings
Description
This service identifies a requirement to fulfil a Destructive_Effect_Solution, it provides the evidence for achievability of the requirement, and identifies any changes to the requirement associated with a change to the Destructive_Effect_Solution.
«activity»
identify_destructive_effects_settings_to_be_fulfilled
Description
Identify Destructive_Effect_Settings to be applied to Weapon_Resources in a weapon package and the aiming of Weapon_Resources.
«activity»
identify_destructive_effects_settings_achievement
Description
Identify achievement to be provided against the derived Destructive_Effect_Setting requirement to decide whether any further action needs to be taken.
«activity»
assess_effects_setting_requirement
Description
Assess the requirement for Destructive_Effect_Settings to be applied to Weapon_Resources in a weapon package and the Destructive_Effect requirements on aiming Weapon_Resources, which includes assessing changes to the requirement for Destructive_Effect_Settings, Weapon_Resources, or identified requirements on weapon aiming.
«interface»
Destructive_Effects_Settings_Requirement
Description
This interface is the requirement for Weapon_Resources (within a weapon package) to be primed with Destructive_Effect_Settings, e.g. airburst at 15 metres from impact with the ground setting, or weapon impact azimuth, elevation and minimum velocity settings. The interface is also the requirement on the platform that enables the delivery of the Destructive_Effect, e.g. for guided bombs, the information that seeds the weapon trajectory prediction (weapon impact azimuth, elevation and minimum velocity), or for ballistic bombs, the distance between bomb impacts on the ground.
«interface»
Destructive_Effects_Settings_Achievement
Description
This interface is the statement of achievement against the fulfilment of the Destructive_Effect_Settings requirement.
«requirement»
control_destructive_effect_settings
Description
• To control the Destructive_Effect_Settings, e.g. set fusing mode.
«refine»

Figure 264: Destructive_Effects_Settings Service Policy

Destructive_Effects_Settings

This service identifies a requirement to fulfil a Destructive_Effect_Solution, it provides the evidence for achievability of the requirement, and identifies any changes to the requirement associated with a change to the Destructive_Effect_Solution.

Interfaces

Destructive_Effects_Settings_Achievement

This interface is the statement of achievement against the fulfilment of the Destructive_Effect_Settings requirement.

Destructive_Effects_Settings_Requirement

This interface is the requirement for Weapon_Resources (within a weapon package) to be primed with Destructive_Effect_Settings, e.g. airburst at 15 metres from impact with the ground setting, or weapon impact azimuth, elevation and minimum velocity settings. The interface is also the requirement on the platform that enables the delivery of the Destructive_Effect, e.g. for guided bombs, the information that seeds the weapon trajectory prediction (weapon impact azimuth, elevation and minimum velocity), or for ballistic bombs, the distance between bomb impacts on the ground.

Attributes

specification	This is the requirement for application of Destructive_Effect_Settings for each Weapon_Resource in a weapon package, e.g. airburst at 15 metres from impact with the ground setting, or weapon impact azimuth, elevation and minimum velocity settings. This is also requirement information relating to the Destructive_Effect_Settings for the Weapon_Resources in a weapon

	package that may constrain the aiming solution for the weapon package, e.g. for guided bombs, the information that seeds the weapon trajectory prediction (weapon impact azimuth, elevation and minimum velocity), or for ballistic bombs, the distance between bomb impacts on the ground.
temporal_information	Information covering timing, such as start and end times.
predicted_quality	How well the Destructive_Effect_Solution is applied.

Activities

identify_destructive_effects_settings_to_be_fulfilled

Identify Destructive_Effect_Settings to be applied to Weapon_Resources in a weapon package and the aiming of Weapon_Resources.

identify_destructive_effects_settings_achievement

Identify achievement to be provided against the derived Destructive_Effect_Setting requirement to decide whether any further action needs to be taken.

assess_effects_setting_requirement

Assess the requirement for Destructive_Effect_Settings to be applied to Weapon_Resources in a weapon package and the Destructive_Effect requirements on aiming Weapon_Resources, which includes assessing changes to the requirement for Destructive_Effect_Settings, Weapon_Resources, or identified requirements on weapon aiming.

5.4.2.15.7.1.3 Available_Effect

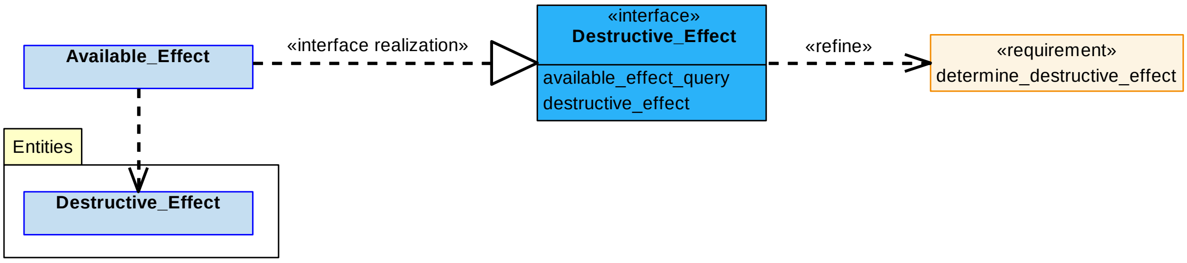


Figure 265: Available_Effect Service Definition

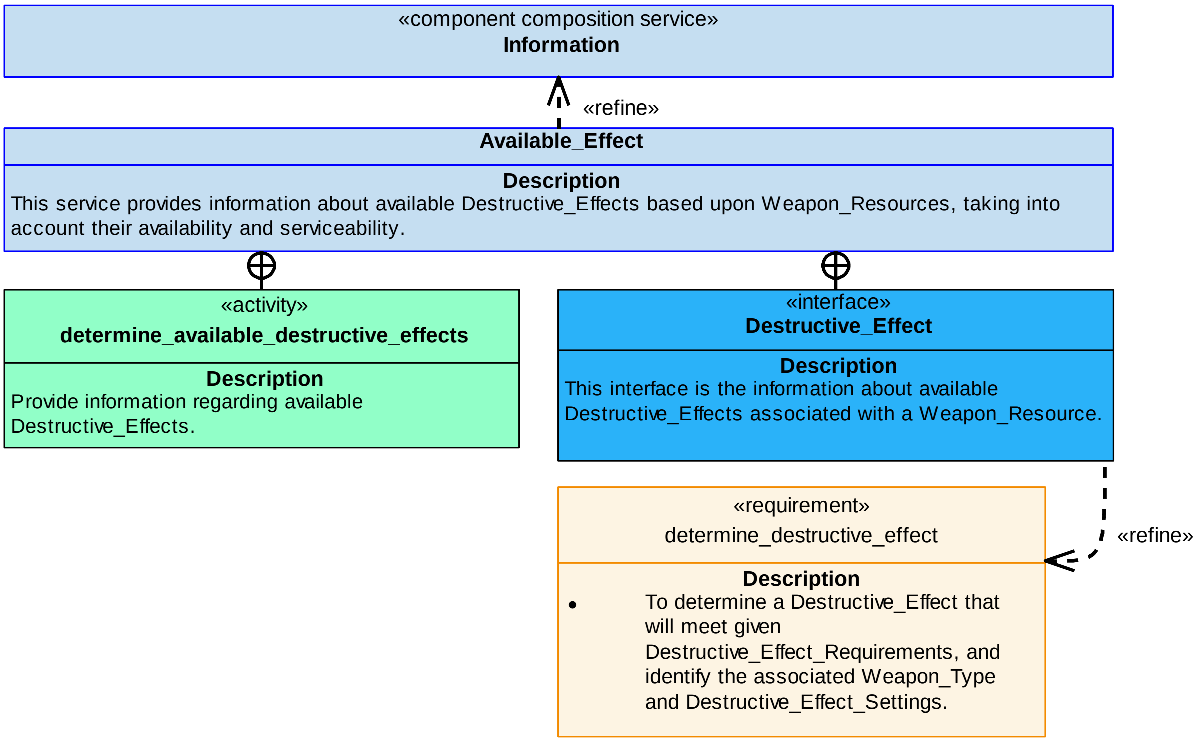

«component composition service»
Information
«refine»
Available_Effect
Description
This service provides information about available Destructive_Effects based upon Weapon_Resources, taking into account their availability and serviceability.
«activity»
determine_available_destructive_effects
Description
Provide information regarding available Destructive_Effects.
«interface»
Destructive_Effect
Description
This interface is the information about available Destructive_Effects associated with a Weapon_Resource.
«requirement»
determine_destructive_effect
«refine»
Description
To determine a Destructive_Effect that will meet given Destructive_Effect_Requirements, and identify the associated Weapon_Type and Destructive_Effect_Settings.

Figure 266: Available_Effect Service Policy

Available_Effect

This service provides information about available Destructive_Effects based upon Weapon_Resources, taking into account their availability and serviceability.

Interface

Destructive_Effect

This interface is the information about available Destructive_Effects associated with a Weapon_Resource.

Attributes

available_effect_query	The definition of the query about which Destructive_Effects are available.
destructive_effect	The details of available Destructive_Effects associated with a Weapon_Resource.

Activity

determine_available_destructive_effects

Provide information regarding available Destructive_Effects.

5.4.2.15.7.1.4 Effect_Influence

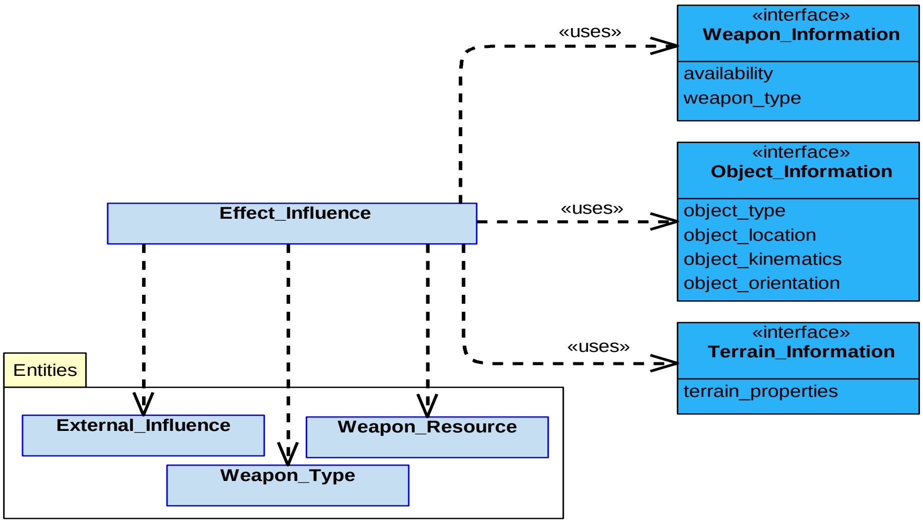


Figure 267: Effect_Influence Service Definition

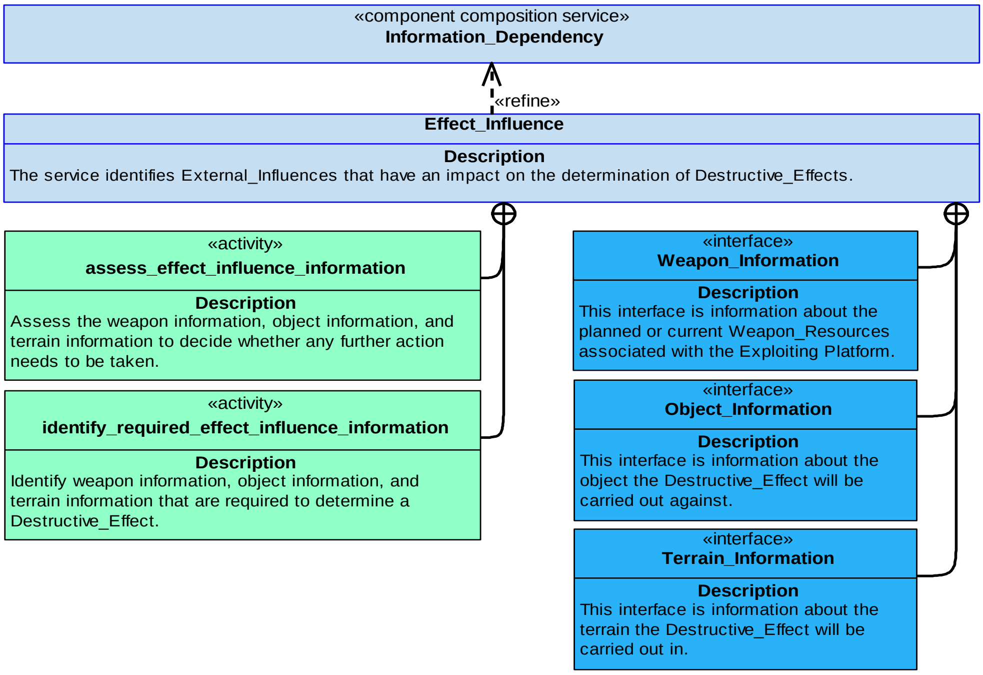

«component composition service»
Information_Dependency
«refine»
Effect_Influence
Description
The service identifies External_Influences that have an impact on the determination of Destructive_Effects.
«activity»
assess_effect_influence_information
Description
Assess the weapon information, object information, and terrain information to decide whether any further action needs to be taken.
«activity»
identify_required_effect_influence_information
Description
Identify weapon information, object information, and terrain information that are required to determine a Destructive_Effect.
«interface»
Weapon_Information
Description
This interface is information about the planned or current Weapon_Resources associated with the Exploiting Platform.
«interface»
Object_Information
Description
This interface is information about the object the Destructive_Effect will be carried out against.
«interface»
Terrain_Information
Description
This interface is information about the terrain the Destructive_Effect will be carried out in.

Figure 268: Effect_Influence Service Policy

Effect_Influence

The service identifies External_Influences that have an impact on the determination of Destructive_Effects.

Interfaces

Weapon_Information

This interface is information about the planned or current Weapon_Resources associated with the Exploiting Platform.

Attributes

availability	The availability of a weapon.
weapon_type	The Weapon_Type, e.g. a missile, or the type of warhead fitted.

Object_Information

This interface is information about the object the Destructive_Effect will be carried out against.

Attributes

object_type	The type of object, e.g. armoured or unarmoured.
object_location	The location of the object, e.g. altitude.
object_kinematics	The kinematics of the object, e.g. velocity.
object_orientation	The orientation of the object, e.g. the direction a target is facing or travelling.

Terrain_Information

This interface is information about the terrain the Destructive_Effect will be carried out in.

Attribute

terrain_properties	Information about the terrain that Destructive_Effect will be carried out in, e.g. a topographical feature that may affect the Destructive_Effect of a Weapon_Resource.

Activities

assess_effect_influence_information

Assess the weapon information, object information, and terrain information to decide whether any further action needs to be taken.

identify_required_effect_influence_information

Identify weapon information, object information, and terrain information that are required to determine a Destructive_Effect.

5.4.2.15.7.1.5 Constraint

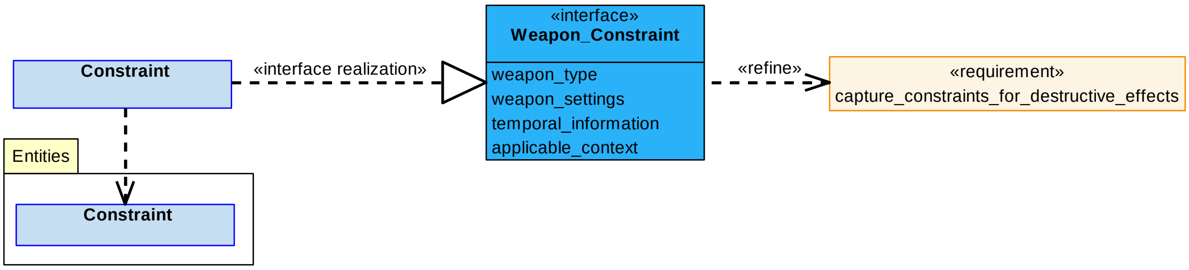


Figure 269: Constraint Service Definition

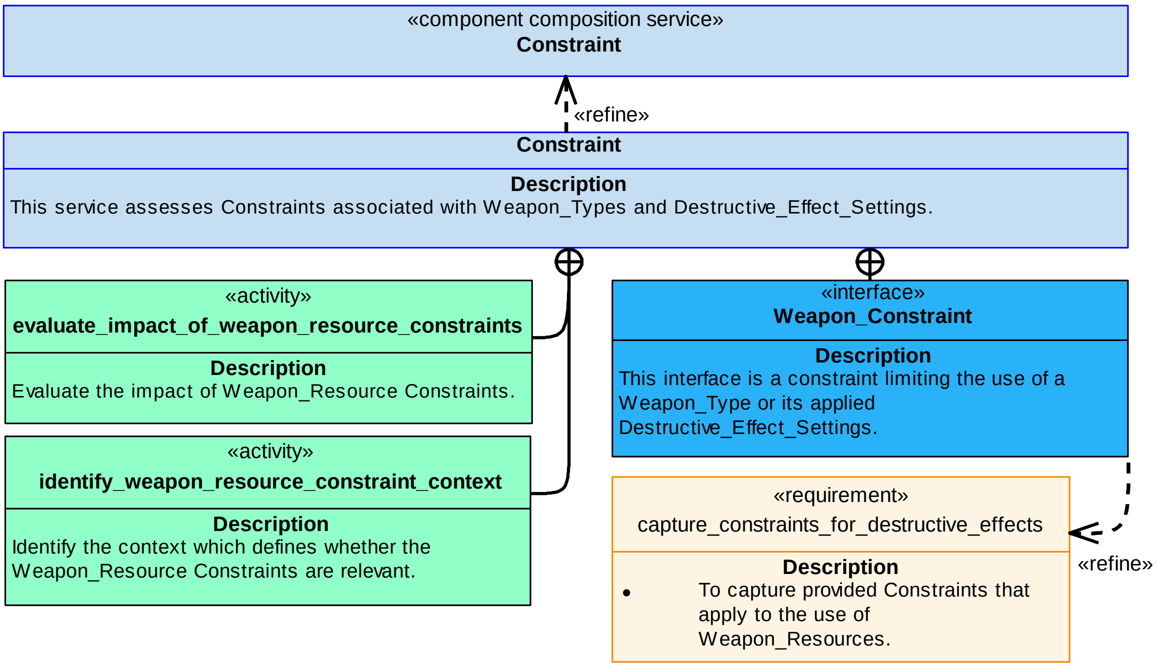

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints associated with Weapon_Types and Destructive_Effect_Settings.
«activity»
evaluate_impact_of_weapon_resource_constraints
Description
Evaluate the impact of Weapon_Resource Constraints.
«activity»
identify_weapon_resource_constraint_context
Description
Identify the context which defines whether the Weapon_Resource Constraints are relevant.
«interface»
Weapon_Constraint
Description
This interface is a constraint limiting the use of a Weapon_Type or its applied Destructive_Effect_Settings.
«requirement»
capture_constraints_for_destructive_effects
Description
• To capture provided Constraints that apply to the use of Weapon_Resources.
«refine»

Figure 270: Constraint Service Policy

Constraint

This service assesses Constraints associated with Weapon_Types and Destructive_Effect_Settings.

Interface

Weapon_Constraint

This interface is a constraint limiting the use of a Weapon_Type or its applied Destructive_Effect_Settings.

Attributes

weapon_type	The Weapon_Type that is restricted for use in providing a Destructive_Effect, e.g. ballistic weapon or direct fire rocket weapon.
weapon_settings	The Destructive_Effect_Setting(s) that are restricted for use in providing a Destructive_Effect, e.g. setting a bomb to air burst mode.
temporal_information	Timing information pertaining to the periods of time when a constraint will be applicable, such as start time and duration, or end time, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the constraint is applicable.

Activities

identify_weapon_resource_constraint_context

Identify the context which defines whether the Weapon_Resource Constraints are relevant.

evaluate_impact_of_weapon_resource_constraints

Evaluate the impact of Weapon_Resource Constraints.

5.4.2.15.7.1.6 Capability

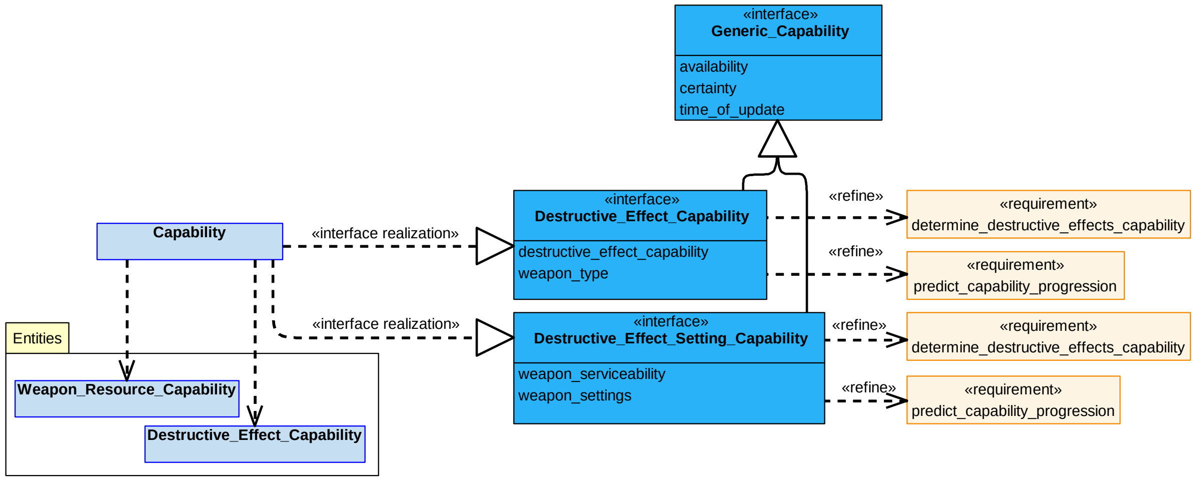
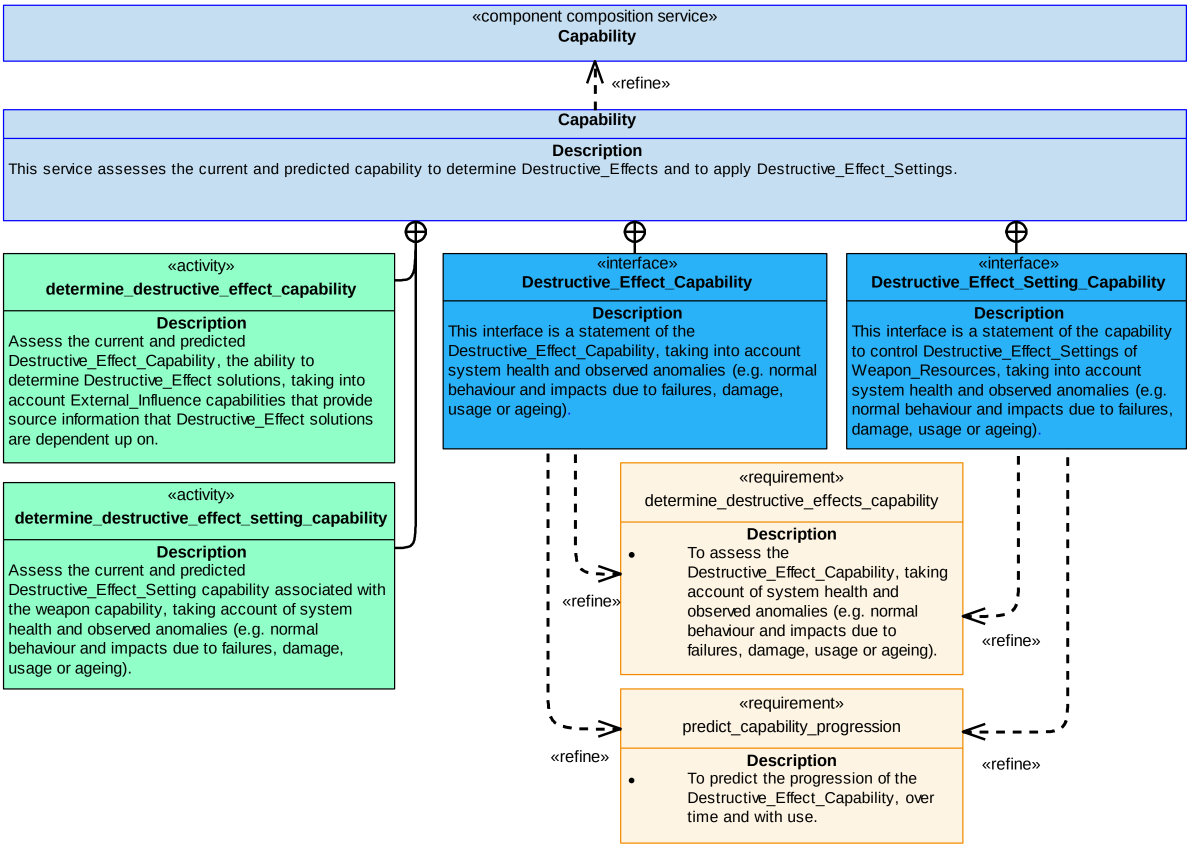

«interface»
Generic_Capability
availability
certainty
time_of_update
Capability
«interface realization»
«interface»
Destructive_Effect_Capability
destructive_effect_capability
weapon_type
«refine»
«requirement»
determine_destructive_effects_capability
«refine»
«requirement»
predict_capability_progression
«interface realization»
«interface»
Destructive_Effect_Setting_Capability
weapon_serviceability
weapon_settings
«refine»
«requirement»
determine_destructive_effects_capability
«refine»
«requirement»
predict_capability_progression
Entities
Weapon_Resource_Capability
Destructive_Effect_Capability

Figure 271: Capability Service Definition


«component composition service»
Capability
«refine»
Capability
Description
«activity»
determine_destructive_effect_capability
Description
«interface»
Destructive_Effect_Capability
Description
«interface»
Destructive_Effect_Setting_Capability
Description
«activity»
determine_destructive_effect_setting_capability
Description
«requirement»
determine_destructive_effects_capability
Description
«refine»
«refine»
«requirement»
predict_capability_progression
Description
«refine»
«refine»

This service assesses the current and predicted capability to determine Destructive_Effects and to apply Destructive_Effect_Settings.

Assess the current and predicted Destructive_Effect_Capability, the ability to determine Destructive_Effect solutions, taking into account External_Influence capabilities that provide source information that Destructive_Effect solutions are dependent up on.

This interface is a statement of the Destructive_Effect_Capability, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

This interface is a statement of the capability to control Destructive_Effect_Settings of Weapon_Resources, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Assess the current and predicted Destructive_Effect_Setting capability associated with the weapon capability, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

- To assess the Destructive_Effect_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

- To predict the progression of the Destructive_Effect_Capability, over time and with use.

Figure 272: Capability Service Policy

Capability

This service assesses the current and predicted capability to determine Destructive_Effects and to apply Destructive_Effect_Settings.

Interfaces

Destructive_Effect_Capability

This interface is a statement of the Destructive_Effect_Capability, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Attributes

destructive_effect_capability	The range of Destructive_Effects that can be provided.
weapon_type	The range of Weapon_Types that can be utilised.

Destructive_Effect_Setting_Capability

This interface is a statement of the capability to control Destructive_Effect_Settings of Weapon_Resources, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Attributes

weapon_serviceability	The serviceability of a Weapon_Resource, e.g. whether certain fusing modes can be applied to it.
weapon_settings	The range of Destructive_Effect_Settings that can be applied to a Weapon_Resource.

Activities

determine_destructive_effect_capability

Assess the current and predicted Destructive_Effect_Capability, the ability to determine Destructive_Effect solutions, taking into account External_Influence capabilities that provide source information that Destructive_Effect solutions are dependent up on.

determine_destructive_effect_setting_capability

Assess the current and predicted Destructive_Effect_Setting capability associated with the weapon capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.15.7.1.7 Capability_Evidence

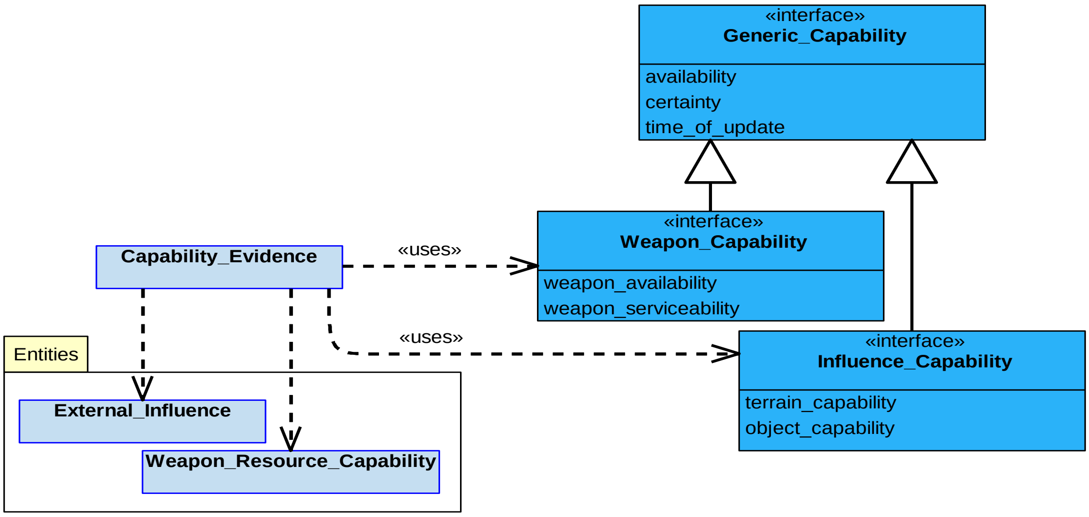
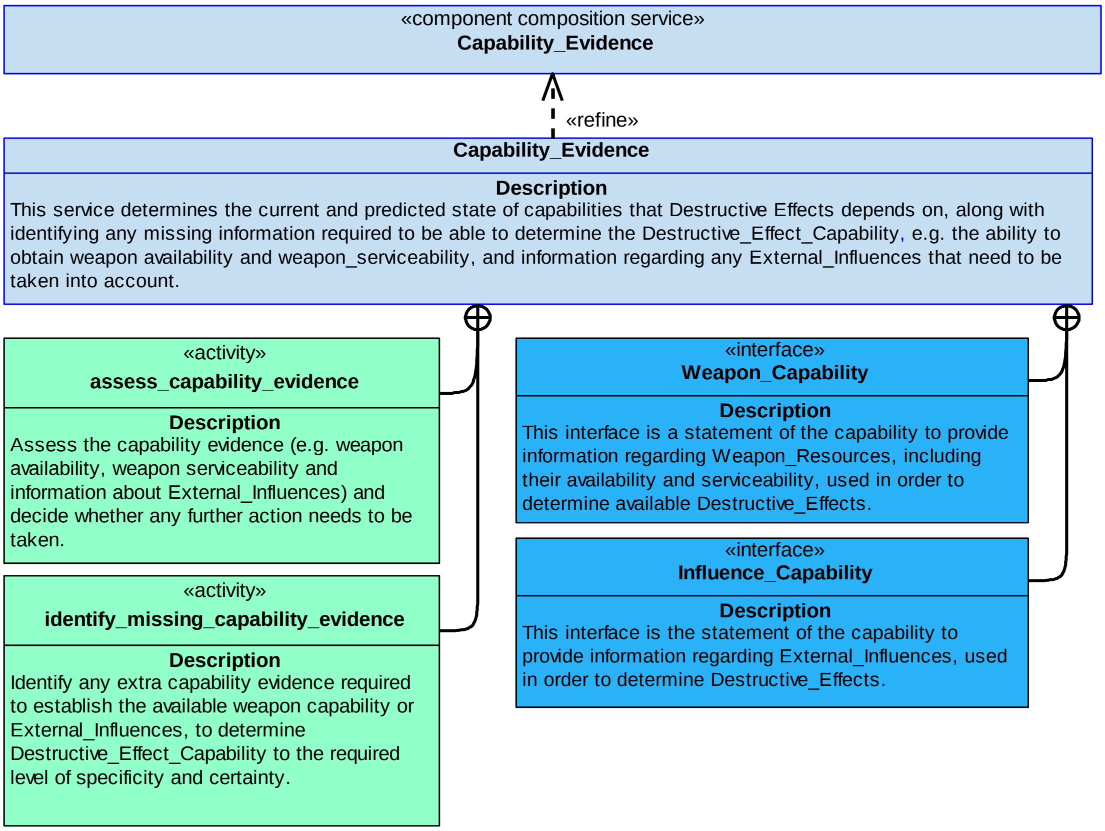


Figure 273: Capability_Evidence Service Definition



Capability_Evidence

Description
This service determines the current and predicted state of capabilities that Destructive Effects depends on, along with identifying any missing information required to be able to determine the Destructive_Effect_Capability, e.g. the ability to obtain weapon availability and weapon_serviceability, and information regarding any External_Influences that need to be taken into account.

«activity» assess_capability_evidence

Description
Assess the capability evidence (e.g. weapon availability, weapon serviceability and information about External_Influences) and decide whether any further action needs to be taken.

«activity» identify_missing_capability_evidence

Description
Identify any extra capability evidence required to establish the available weapon capability or External_Influences, to determine Destructive_Effect_Capability to the required level of specificity and certainty.

«interface» Weapon_Capability

Description
This interface is a statement of the capability to provide information regarding Weapon_Resources, including their availability and serviceability, used in order to determine available Destructive_Effects.

«interface» Influence_Capability

Description
This interface is the statement of the capability to provide information regarding External_Influences, used in order to determine Destructive_Effects.

Figure 274: Capability_Evidence Service Policy

Capability_Evidence

This service determines the current and predicted state of capabilities that Destructive Effects depends on, along with identifying any missing information required to be able to determine the

Destructive_Effect_Capability, e.g. the ability to obtain weapon availability and weapon_serviceability, and information regarding any External_Influences that need to be taken into account.

Interfaces

Weapon_Capability

This interface is a statement of the capability to provide information regarding Weapon_Resources, including their availability and serviceability, used in order to determine available Destructive_Effects.

Attributes

weapon_availability	The availability of Weapon_Resources, e.g. what weapons are planned to be or currently fitted to the Exploiting Platform.
weapon_serviceability	The serviceability of Weapon_Resources, e.g. the ability of a Weapon_Resource to have particular settings applied to it.

Influence_Capability

This interface is the statement of the capability to provide information regarding External_Influences, used in order to determine Destructive_Effects.

Attributes

terrain_capability	The capability to provide terrain information associated with a target or area of interest.
object_capability	The capability to provide object information related to identification or understanding of an object deemed a target, e.g. an air vehicle or hardened aircraft shelter.

Activities

assess_capability_evidence

Assess the capability evidence (e.g. weapon availability, weapon serviceability and information about External_Influences) and decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to establish the available weapon capability or External_Influences, to determine Destructive_Effect_Capability to the required level of specificity and certainty.

5.4.2.15.7.2 Service Dependencies

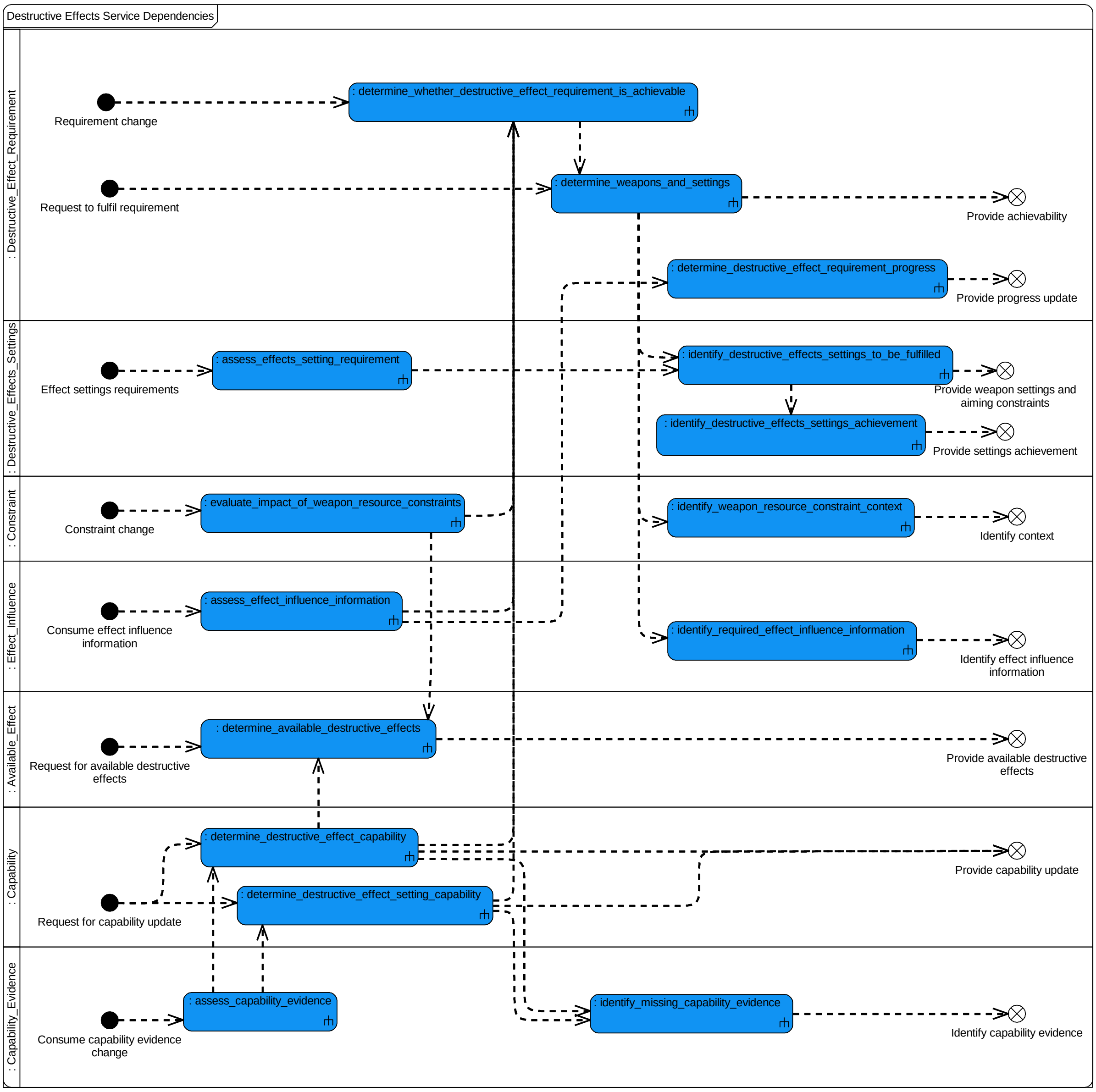

Destructive Effects Service Dependencies
: Destructive_Effect_Requirement
: Destructive_Effects_Settings
: Constraint
: Effect_Influence
: Available_Effect
: Capability
: Capability_Evidence
Requirement change
: determine_whether_destructive_effect_requirement_is_achievable
Request to fulfil requirement
: determine_weapons_and_settings
Provide achievability
: determine_destructive_effect_requirement_progress
Provide progress update
Effect settings requirements
: assess_effects_setting_requirement
: identify_destructive_effects_settings_to_be_fulfilled
Provide weapon settings and aiming constraints
: identify_destructive_effects_settings_achievement
Provide settings achievement
Constraint change
: evaluate_impact_of_weapon_resource_constraints
: identify_weapon_resource_constraint_context
Identify context
Consume effect influence information
: assess_effect_influence_information
: identify_required_effect_influence_information
Identify effect influence information
Request for available destructive effects
: determine_available_destructive_effects
Provide available destructive effects
: determine_destructive_effect_capability
Provide capability update
Request for capability update
: determine_destructive_effect_setting_capability
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 275: Destructive Effects Service Dependencies

5.4.2.16 Effectors

5.4.2.16.1 Role

The role of Effectors is to manage effector devices by controlling their state and the effects they provide.

5.4.2.16.2 Overview

Control Architecture

Effectors is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In response to a Requirement, Effectors will change the state of an Effector_Resource or command it to provide an Effect.

Examples of Use

This component will be required for control of Effector_Resources to carry out activities such as:

- Activating and deactivating heating for ice protection.
- Control of an aileron actuator.
- Bringing a jamming pod from 'off' to 'standby' state.
- Requesting resources such as power or hydraulic pressure needed to operate a device.

5.4.2.16.3 Service Summary

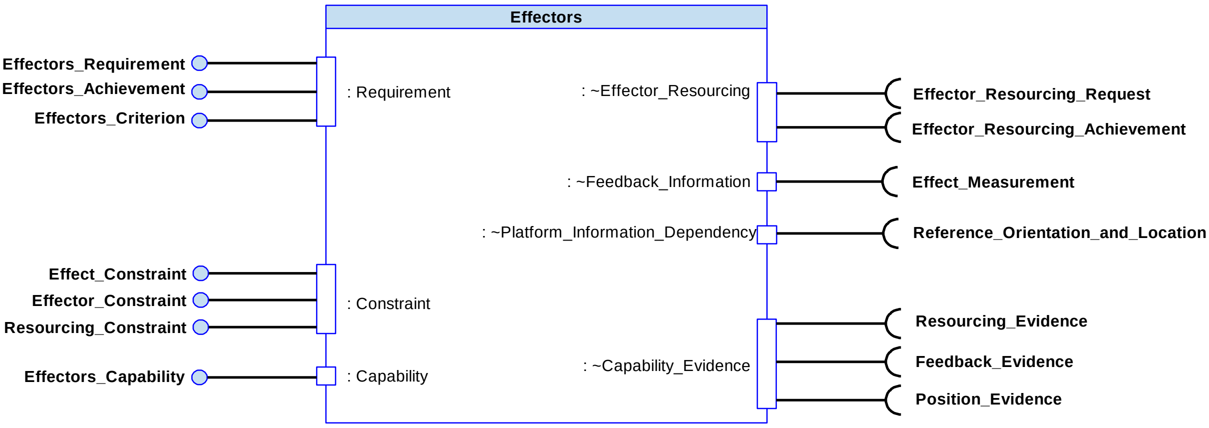


Figure 276: Effectors Service Summary

5.4.2.16.4 Responsibilities

capture_requirements_for_effector_resources

- To capture provided requirements for use of Effector_Resources (e.g. turn valve off now, select flap to position 3 in 2 seconds or increase output voltage at 3V per second for 6 seconds).

capture_measurement_criteria

- To capture Measurement_Criterion for an Effect.

capture_effector_constraints

- To capture provided constraints, e.g. EMCON.

determine_effector_solution

- To determine a solution for use of Effector_Resources that will meet given Requirements.

determine_resources_used_by_the_effector

- To determine information about effectors use of the resources.

determine_if_solution_remains_feasible

- To determine if a planned or ongoing Effector_Solution remains feasible.

control_use_of_effector

- To control the use of Effector_Resources.

identify_progress_of_effector

- To identify the progress of the Effector_Solution use against the Requirement.

assess_effector_capability

- To assess the Capability provided by Effector_Resources, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

predict_capability_progression

- To predict the progression of Effector_Resource capability over time and with use.

5.4.2.16.5 Subject Matter Semantics

The subject matter of Effectors is the Effector_Resources that can be used to provide an Effect.

Exclusions

The subject matter of Effectors does not include:

- Complex pieces of equipment with multiple functions and a highly configurable interface.

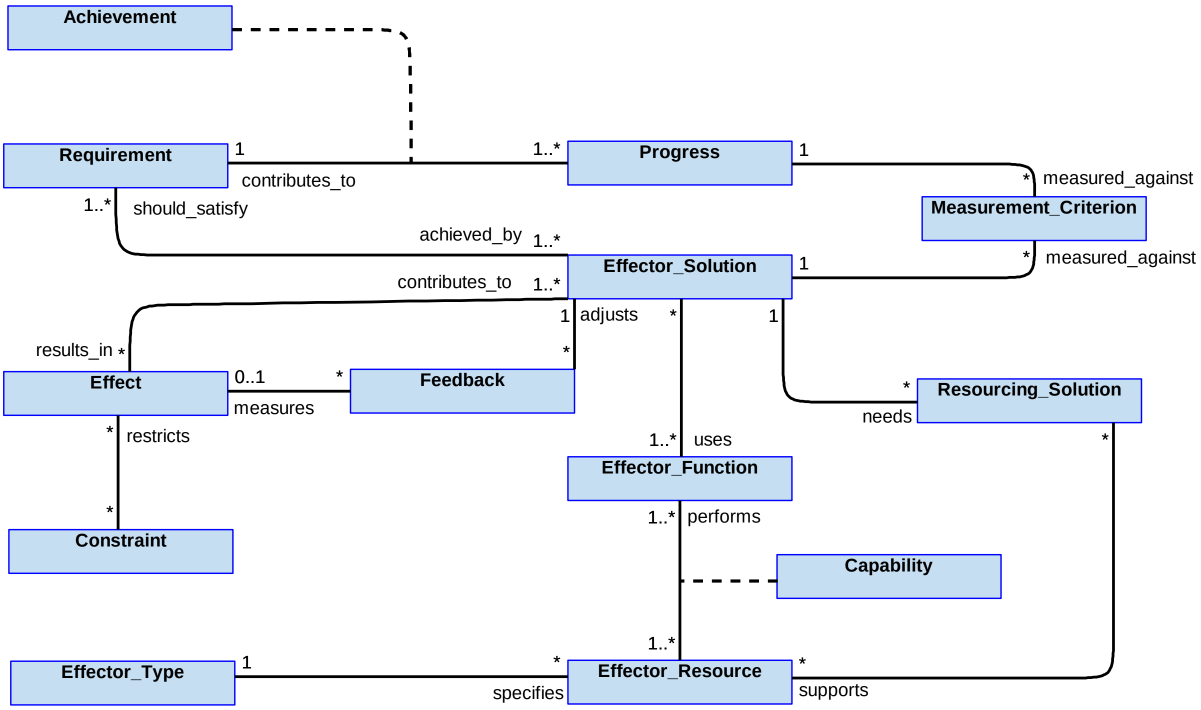


Figure 277: Effectors Semantics

5.4.2.16.5.1 Entities

Achievement

The extent to which the Progress contributes towards achieving the Requirement goal.

Capability

The ability to affect the physical environment in order to meet requirements. This takes into account the ability of Effectors to control the Effector_Resource.

Constraint

A restriction that limits the implementation of an Effector_Solution that would result in an Effect, e.g. EMCON forbidding the transmission of light or a platform process which inhibits the Effector_Solution from supporting its Effect.

Effect

An emission state or positon state of an effector, e.g. the length of extension of an actuator, the level of heat output of a heating element, or the electromagnetic wave output by a transmitter. (It does not include the consequences of the effector’s state, e.g. the movement of a structure connected to an actuator, the temperature of a space, or the propagation through space of an electromagnetic wave.)

Effector_Function

An effector operation that can be performed by an effector, e.g. heating, jamming, or changing the state from idle to on.

Effector_Resource

A real world piece of resource equipment conforming to the specified Effector_Type that can perform the Effector_Function(s) as part of the Effector_Solution(s).

Effector_Solution

A solution to satisfy the Requirement by causing a change in the physical world, through an Effect, or a change in the effector state.

Effector_Type

A type of effector that can be utilised, e.g. an aileron actuator.

Feedback

The information needed on the quantitative performance and the status of the effector to adjust the Effector_Solution to achieve the intended Effect.

Measurement_Criterion

A criterion used to determine progress and/or success.

Progress

Measure of progress as part of an activity, e.g. rudder actuator has completed 60% of its required movement.

Requirement

A requirement to manipulate an effector to change its state or produce a change to the physical environment.

Resourcing_Solution

A solution to sequence the securing of the resources needed to support the Effector_Resource in providing or achieving the Effector_Function, e.g. power and cooling.

5.4.2.16.6 Design Rationale

5.4.2.16.6.1 Assumptions

- The Effector_Resources managed by the Effectors component may be simple pieces of effecting equipment such as valve actuators.
- Effector_Resources may represent a simple interface to effecting functions of a complex piece of equipment. Such a function may be highly sophisticated (such as a pod that provides a complex jamming function), even if it presents a simple interface, for example, it can only be turned on and off.
- Effector_Resources do not represent the whole of complex pieces of equipment with multiple functions and a highly configurable interface. Such equipment is addressed using multiple components in accordance with the Interaction with Equipment PYRAMID concept.

- An Effector_Resource produces an effect on an aspect of the physical environment (e.g. an actuator that moves a rudder or a heating element that melts or prevents ice).
- Other components (e.g. Sensors) will confirm that the action of an Effector_Resource had the intended Effect on the physical environment. The Effectors component can only understand whether the Effector_Resource was correctly commanded, and whether it gave an appropriate response. Where a single piece of equipment includes both sensor and the effector elements, the effecting and sensing (and related action control) functions are considered as separate.

5.4.2.16.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Effectors:

- Data Driving - There are numerous types of Effector_Resource; this component could be configured to support any of them using data driving.
- Resource Management - The Resourcing_Solution and Effector_Resource will need to secure platform resources to deliver an Effect, e.g. power, time and spectrum when delivering a radio frequency Effect.

Extensions

- It is possible that extensions will be developed to provide an interface to specific types of equipment.

Exploitation Considerations

- The interface of the Effectors component with its Effector_Resources may be via a service interface or may be direct (i.e. not via a service interface). The interface may vary according to the type of Effector_Resource, and may change if one Effector_Resource is replaced with another. This variation is expected to be handled by data driving or extensions. (Note that resources replaced on a like-for-like basis (with the same form, fit and function) may have different failure modes and steady states. The monitoring and management of the new resource may not be considered within Effectors, but the PRA considers this in components such as Anomaly Detection and Health Assessment).
- A new Effector_Resource could be introduced during mission fit. For example, different payload bay modules may be switched according to mission requirements, and these may incorporate different Effector_Resources.
- An Effector_Resource and Effector_Solution may need to be closely coupled with related sensing resources to enable precise monitoring and control of the Effect, the Feedback_Information service provides a method for meeting this need.

5.4.2.16.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Failure of this component when controlling cooling air, de-icing heaters, etc. could cause equipment that is critical to the controlled flight of the air vehicle (e.g. flight control system computing hardware or control surfaces) to be outside the qualified operating environment. As the equipment can no longer be relied upon to operate, this could cause uncontrolled flight of the air vehicle and subsequently an uncontrolled crash. This would result in loss of the air vehicle and potentially fatalities.
- Failure of this component in certain states (e.g. weight on wheels) when controlling ionising equipment could result in the irradiation of ground crew and bystanders. This would result in long term harmful consequences for these personnel.

Where instances of this component contribute to hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.16.6.4 Security Considerations

The indicative security classification is O but will vary according to the effector.

This component forms part of the interface with effectors and as such is dependent on the Effector_Type being managed and the purpose they are put to; there are expected to be multiple instances of this component, for effectors ranging from simple heating elements or valve actuators to complex tactical effecting equipment including those carrying out electronic warfare tasks. The confidentiality, integrity and availability requirements will need to reflect this.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to effector use during the mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** of the effector state against the commanded operations. Unexpected activity may indicate that the Exploiting Platform has been compromised.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.16.7 Services

5.4.2.16.7.1 Service Definitions

5.4.2.16.7.1.1 Requirement

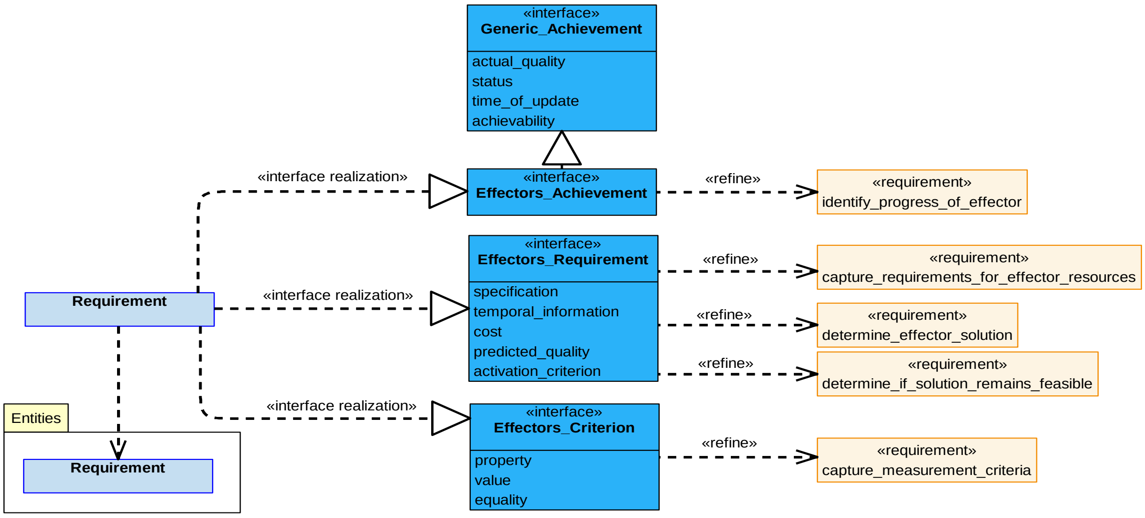
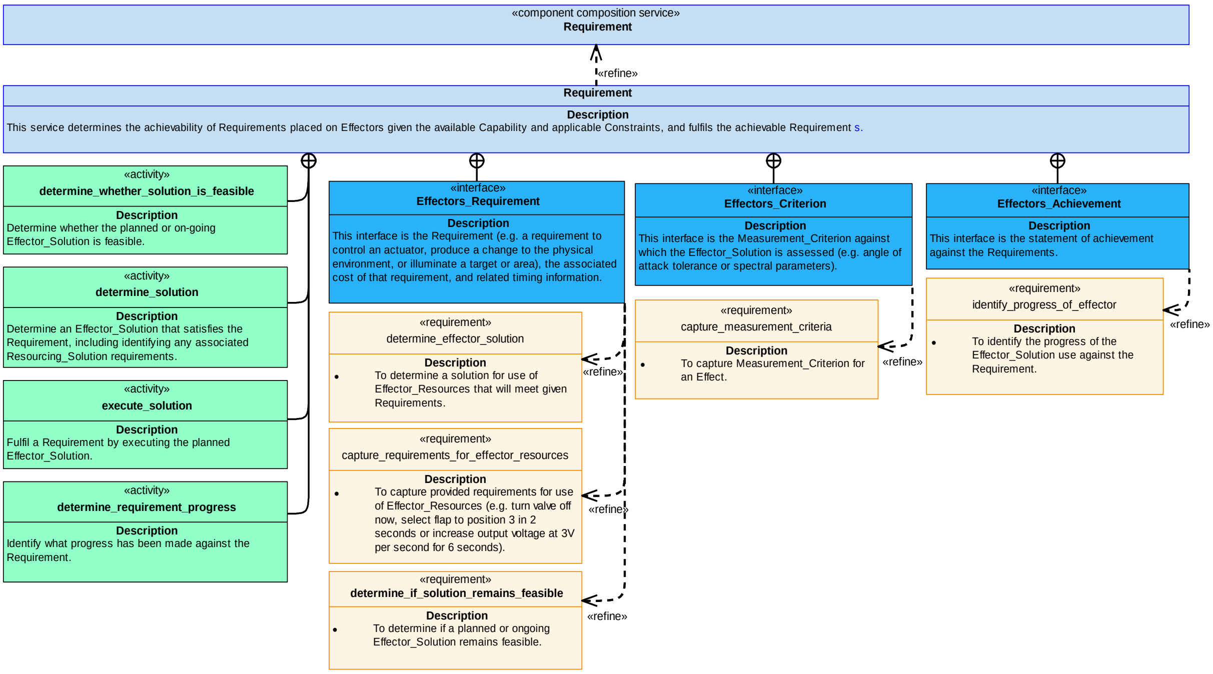


Figure 278: Requirement Service Definition



Description
This service determines the achievability of Requirements placed on Effectors given the available Capability and applicable Constraints, and fulfils the achievable Requirement s.

Description
Determine whether the planned or on-going Effector_Solution is feasible.

Description
Determine an Effector_Solution that satisfies the Requirement, including identifying any associated Resourcing_Solution requirements.

Description
Fulfil a Requirement by executing the planned Effector_Solution.

Description
Identify what progress has been made against the Requirement.

Description
This interface is the Requirement (e.g. a requirement to control an actuator, produce a change to the physical environment, or illuminate a target or area), the associated cost of that requirement, and related timing information.

Description
- To determine a solution for use of Effector_Resources that will meet given Requirements.

Description
- To capture provided requirements for use of Effector_Resources (e.g. turn valve off now, select flap to position 3 in 2 seconds or increase output voltage at 3V per second for 6 seconds).

Description
- To determine if a planned or ongoing Effector_Solution remains feasible.

Description
This interface is the Measurement_Criterion against which the Effector_Solution is assessed (e.g. angle of attack tolerance or spectral parameters).

Description
- To capture Measurement_Criterion for an Effect.

Description
This interface is the statement of achievement against the Requirements.

Description
- To identify the progress of the Effector_Solution use against the Requirement.

Figure 279: Requirement Service Policy

Requirement

This service determines the achievability of Requirements placed on Effectors given the available Capability and applicable Constraints, and fulfils the achievable Requirements.

Interfaces

Effectors_Criterion

This interface is the Measurement_Criterion against which the Effector_Solution is assessed (e.g. angle of attack tolerance or spectral parameters).

Attributes

property	The property to be measured, e.g. a specific measurement such as frequency or an expression of the importance attaching to the Requirement.
value	The amount related to the property to be measured, e.g. 50 GHz.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Effectors_Achievement

This interface is the statement of achievement against the Requirements.

Effectors_Requirement

This interface is the Requirement (e.g. a requirement to control an actuator, produce a change to the physical environment, or illuminate a target or area), the associated cost of that requirement, and related timing information.

Attributes

specification	A requirement to control an Effector_Resource or produce an Effect by a given Effector_Resource (e.g. the effect function, the order to transmit at given spectral parametrics, or adjust control surfaces).
temporal_information	A requirement to cover timing of use of effectors, such as start and end times.
cost	The cost of executing the Effector_Solution, for example: effector resources and power used, plus time taken.
predicted_quality	Acceptable quality thresholds and gradients (i.e. minimum vs ideal level) to be obtained by the Effector_Solution, specified appropriately for the type of Requirement.
activation_criterion	How and when the effectors requirement fulfilment should be triggered, i.e. once selected, the Effector_Solution will be enacted immediately, or at a particular time or location depending on the decision.

Activities

determine_solution

Determine an Effector_Solution that satisfies the Requirement, including identifying any associated Resourcing_Solution requirements.

determine_requirement_progress

Identify what progress has been made against the Requirement.

execute_solution

Fulfil a Requirement by executing the planned Effector_Solution.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Effector_Solution is feasible.

5.4.2.16.7.1.2 Effector_Resourcing

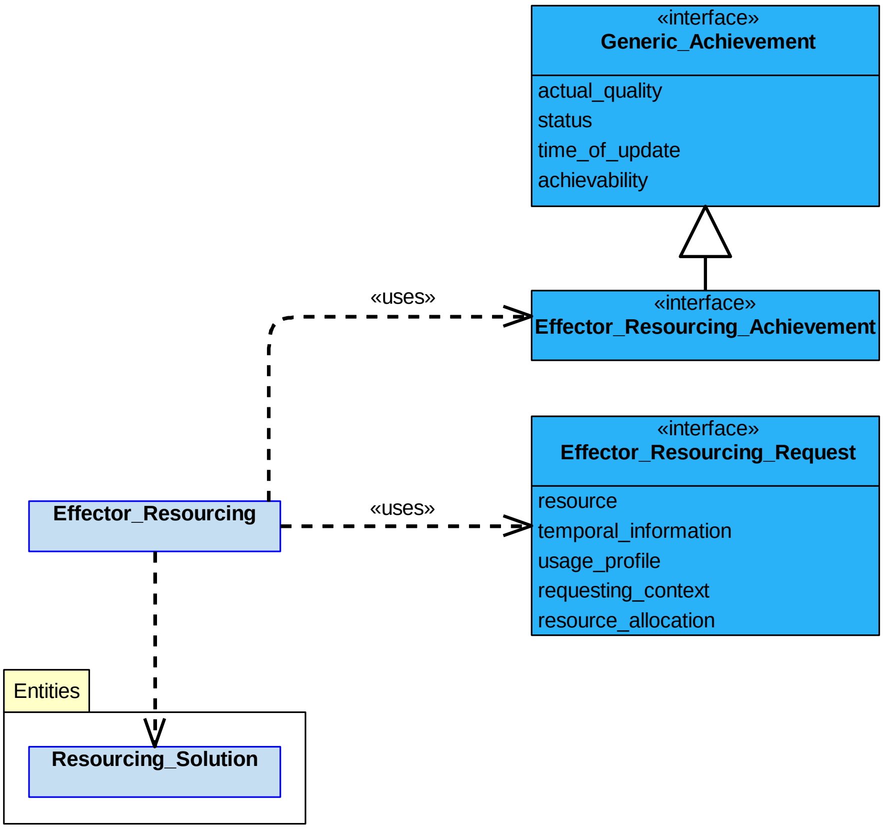


Figure 280: Effector_Resourcing Service Definition

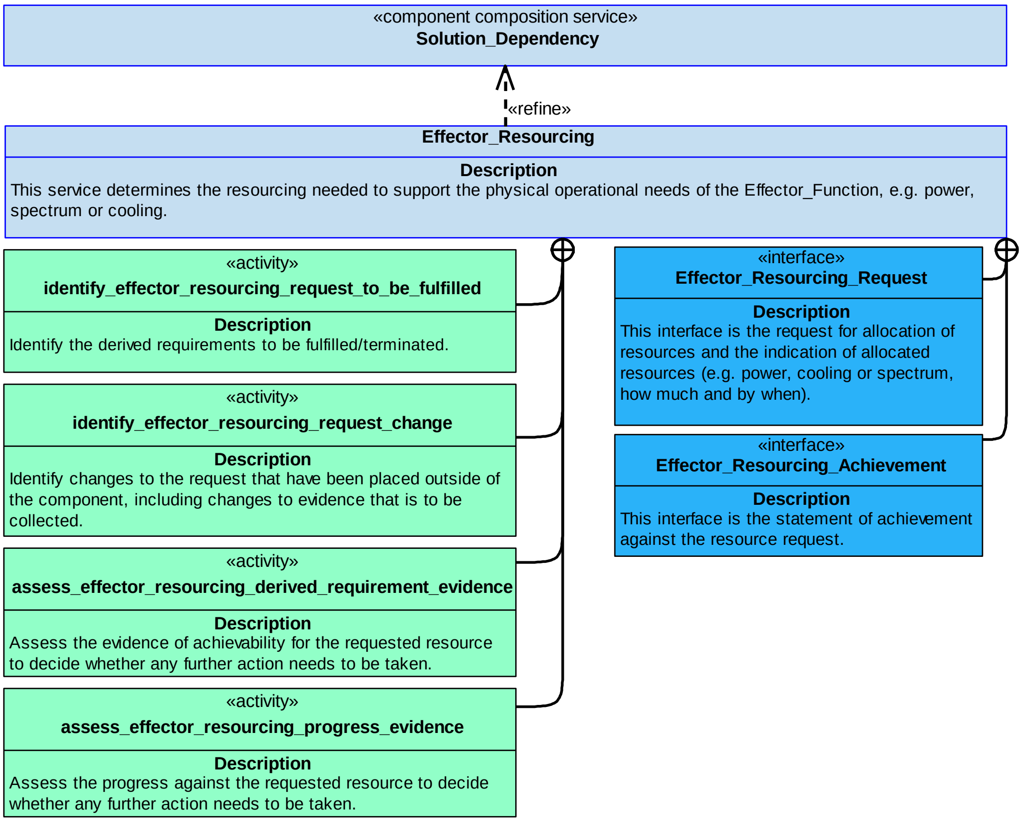

«component composition service»
Solution_Dependency
«refine»
Effector_Resourcing
Description
This service determines the resourcing needed to support the physical operational needs of the Effector_Function, e.g. power, spectrum or cooling.
«activity»
identify_effector_resourcing_request_to_be_fulfilled
Description
Identify the derived requirements to be fulfilled/terminated.
«activity»
identify_effector_resourcing_request_change
Description
Identify changes to the request that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
assess_effector_resourcing_derived_requirement_evidence
Description
Assess the evidence of achievability for the requested resource to decide whether any further action needs to be taken.
«activity»
assess_effector_resourcing_progress_evidence
Description
Assess the progress against the requested resource to decide whether any further action needs to be taken.
«interface»
Effector_Resourcing_Request
Description
This interface is the request for allocation of resources and the indication of allocated resources (e.g. power, cooling or spectrum, how much and by when).
«interface»
Effector_Resourcing_Achievement
Description
This interface is the statement of achievement against the resource request.

Figure 281: Effector_Resourcing Service Policy

Effector_Resourcing

This service determines the resourcing needed to support the physical operational needs of the Effector_Function, e.g. power, spectrum or cooling.

Interfaces

Effector_Resourcing_Request

This interface is the request for allocation of resources and the indication of allocated resources (e.g. power, cooling or spectrum, how much and by when).

Attributes

resource	The resource being requested, e.g. power or cooling.
temporal_information	Information covering timing for the requested resource, such as start and end times. This might include segments of a requested time window that must not be interrupted.
usage_profile	The quantity of resource requested for use, e.g. a one-off amount or a variable amount or 10 kW for a specified time period.
requesting_context	The information that identifies the source or reason for the request.
resource_allocation	The actual allocated resource quantity required to meet the usage_profile.

Effector_Resourcing_Achievement

This interface is the statement of achievement against the resource request.

Activities

identify_effector_resourcing_request_to_be_fulfilled

Identify the derived requirements to be fulfilled/terminated.

identify_effector_resourcing_request_change

Identify changes to the request that have been placed outside of the component, including changes to evidence that is to be collected.

assess_effector_resourcing_derived_requirement_evidence

Assess the evidence of achievability for the requested resource to decide whether any further action needs to be taken.

assess_effector_resourcing_progress_evidence

Assess the progress against the requested resource to decide whether any further action needs to be taken.

5.4.2.16.7.1.3 Feedback_Information

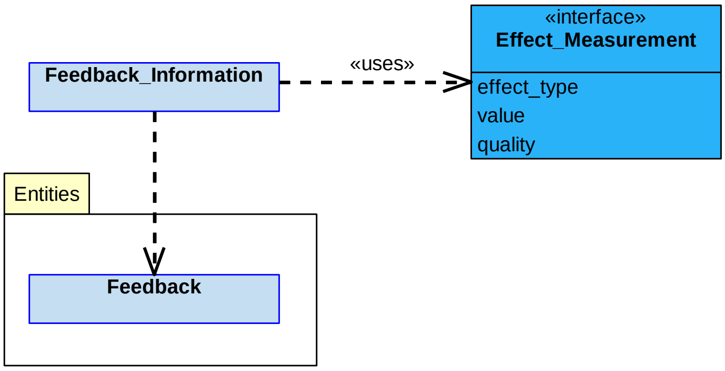


Figure 282: Feedback_Information Service Definition

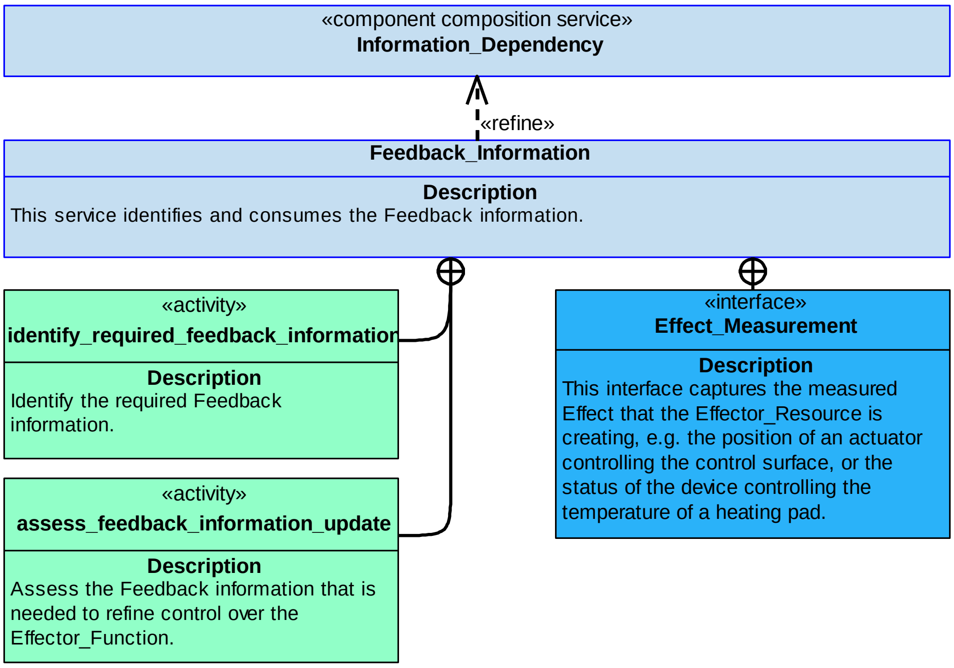

«component composition service»
Information_Dependency
«refine»
Feedback_Information
Description
This service identifies and consumes the Feedback information.
«activity»
identify_required_feedback_information
Description
Identify the required Feedback information.
«activity»
assess_feedback_information_update
Description
Assess the Feedback information that is needed to refine control over the Effector_Function.
«interface»
Effect_Measurement
Description
This interface captures the measured Effect that the Effector_Resource is creating, e.g. the position of an actuator controlling the control surface, or the status of the device controlling the temperature of a heating pad.

Figure 283: Feedback_Information Service Policy

Feedback_Information

This service identifies and consumes the Feedback information.

Interface

Effect_Measurement

This interface captures the measured Effect that the Effector_Resource is creating, e.g. the position of an actuator controlling the control surface, or the status of the device controlling the temperature of a heating pad.

Attributes

effect_type	The effect that is being measured, e.g. actuator extension or output temperature.
value	The measured value of the effect, e.g. the length in millimetres or a temperature in degrees Celsius.
quality	The quality (e.g. accuracy and certainty) in the provided response.

Activities

assess_feedback_information_update

Assess the Feedback information that is needed to refine control over the Effector_Function.

identify_required_feedback_information

Identify the required Feedback information.

5.4.2.16.7.1.4 Platform_Information_Dependency

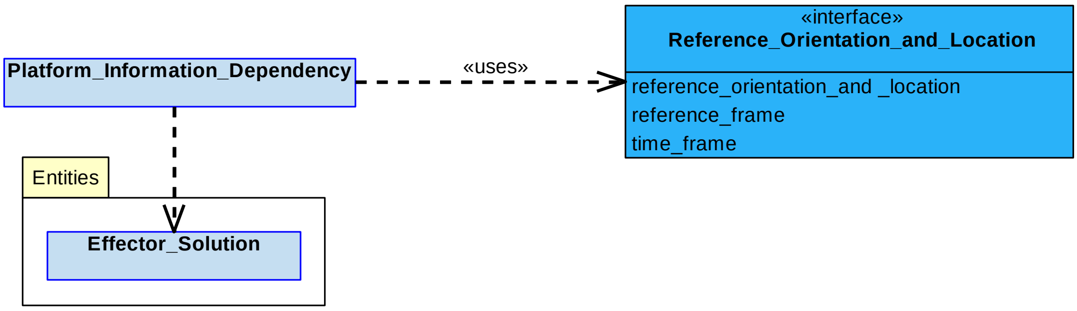

«interface»
Reference_Orientation_and_Location
reference_orientation_and _location
reference_frame
time_frame
Platform_Information_Dependency
«uses»
Entities
Effector_Solution

Figure 284: Platform_Information_Dependency Service Definition

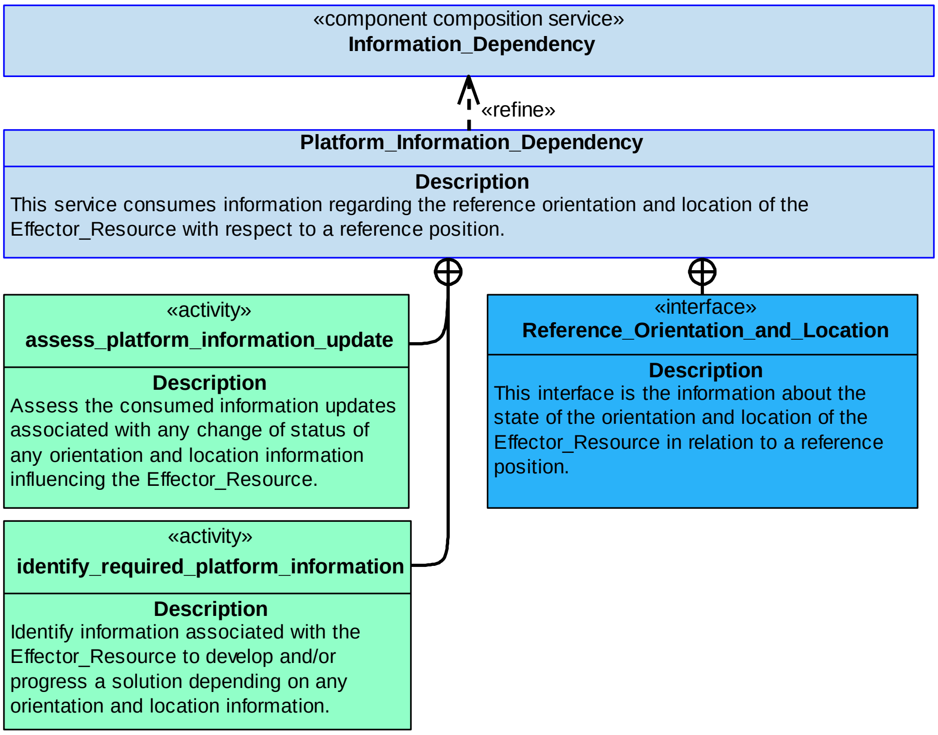

«component composition service»
Information_Dependency
«refine»
Platform_Information_Dependency
Description
This service consumes information regarding the reference orientation and location of the Effector_Resource with respect to a reference position.
«activity»
assess_platform_information_update
Description
Assess the consumed information updates associated with any change of status of any orientation and location information influencing the Effector_Resource.
«interface»
Reference_Orientation_and_Location
Description
This interface is the information about the state of the orientation and location of the Effector_Resource in relation to a reference position.
«activity»
identify_required_platform_information
Description
Identify information associated with the Effector_Resource to develop and/or progress a solution depending on any orientation and location information.

Figure 285: Platform_Information_Dependency Service Policy

Platform_Information_Dependency

This service consumes information regarding the reference orientation and location of the Effector_Resource with respect to a reference position.

Interface

Reference_Orientation_and_Location

This interface is the information about the state of the orientation and location of the Effector_Resource in relation to a reference position.

Attributes

reference_orientation_and _location	This is the offset between the reference position and the Effector_Resource position.

reference_frame	A physical reference point for the estimated data consumed.
time_frame	A temporal reference point for the estimated data consumed.

Activities

assess_platform_information_update

Assess the consumed information updates associated with any change of status of any orientation and location information influencing the Effector_Resource.

identify_required_platform_information

Identify information associated with the Effector_Resource to develop and/or progress a solution depending on any orientation and location information.

5.4.2.16.7.1.5 Constraint

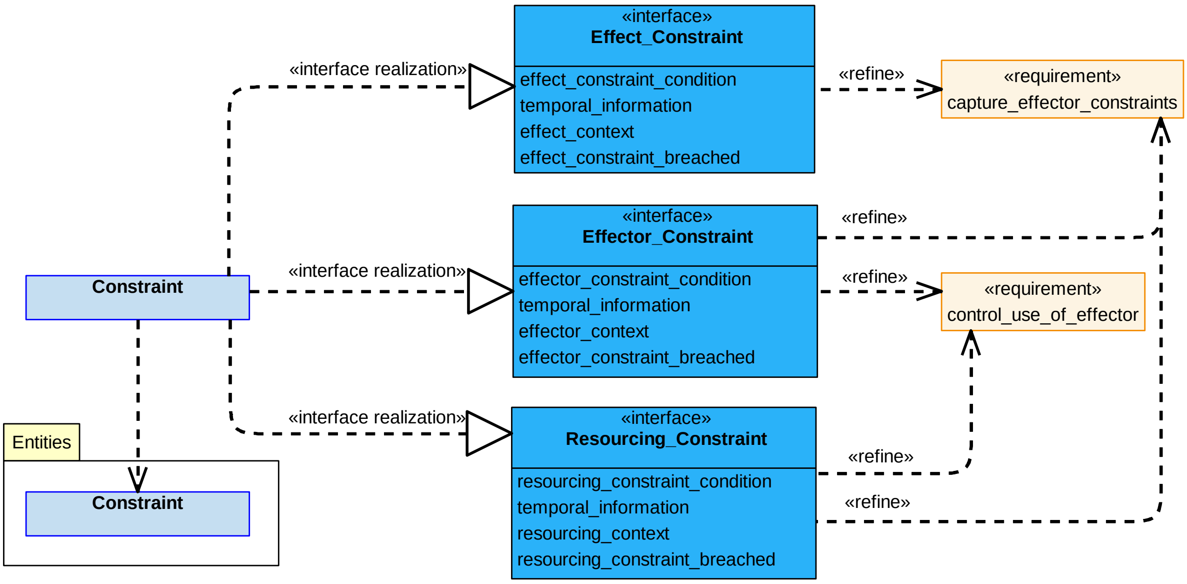


Figure 286: Constraint Service Definition

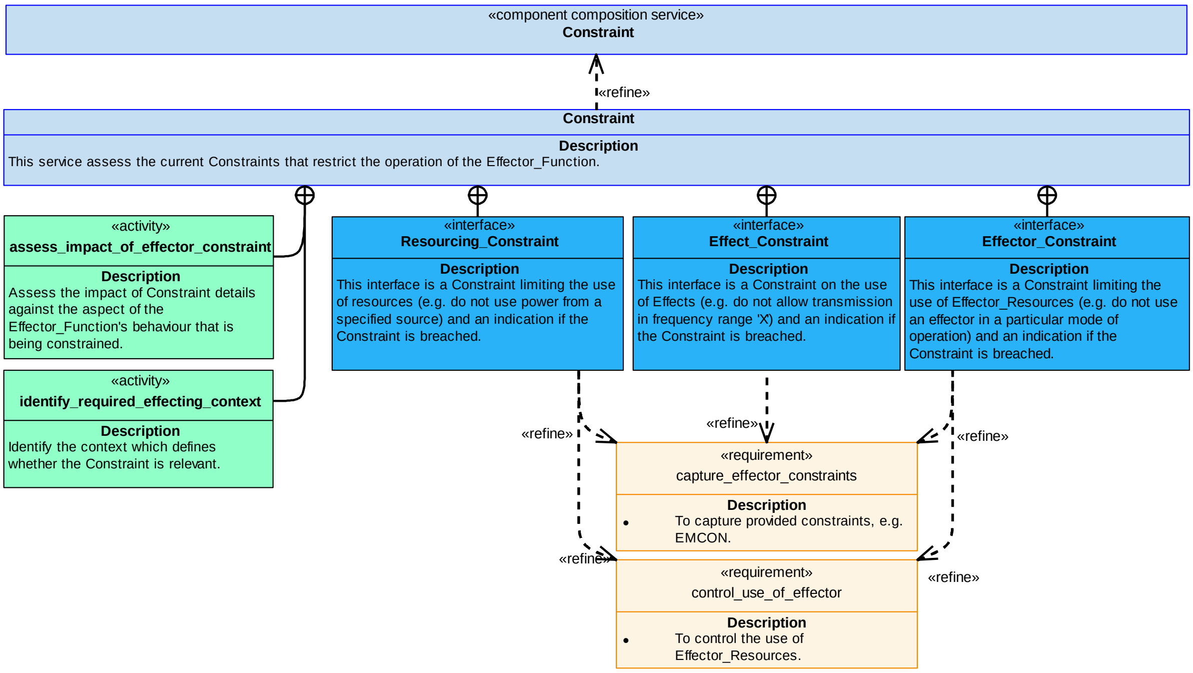

«component composition service»
Constraint
«refine»
Constraint
Description
This service assess the current Constraints that restrict the operation of the Effector_Function.
«activity»
assess_impact_of_effector_constraint
Description
Assess the impact of Constraint details against the aspect of the Effector_Function's behaviour that is being constrained.
«activity»
identify_required_effecting_context
Description
Identify the context which defines whether the Constraint is relevant.
«interface»
Resourcing_Constraint
Description
This interface is a Constraint limiting the use of resources (e.g. do not use power from a specified source) and an indication if the Constraint is breached.
«interface»
Effect_Constraint
Description
This interface is a Constraint on the use of Effects (e.g. do not allow transmission in frequency range 'X) and an indication if the Constraint is breached.
«interface»
Effector_Constraint
Description
This interface is a Constraint limiting the use of Effector_Resources (e.g. do not use an effector in a particular mode of operation) and an indication if the Constraint is breached.
«refine»
«refine»
«refine»
«refine»
«refine»
«requirement»
capture_effector_constraints
Description
To capture provided constraints, e.g. EMCON.
«requirement»
control_use_of_effector
Description
To control the use of Effector_Resources.

Figure 287: Constraint Service Policy

Constraint

This service assess the current Constraints that restrict the operation of the Effector_Function.

Interfaces

Effector_Constraint

This interface is a Constraint limiting the use of Effector_Resources (e.g. do not use an effector in a particular mode of operation) and an indication if the Constraint is breached.

Attributes

effector_constraint_condition	The specification of condition that restricts the Effector_Resource, e.g. do not use an effector in a particular mode of operation.
temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. the start and stop time of the Constraint.
effector_context	The context in which the Constraint is applicable.
effector_constraint_breached	A notification of a breach of the Constraint.

Effect_Constraint

This interface is a Constraint on the use of Effects (e.g. do not allow transmission in frequency range 'X') and an indication if the Constraint is breached.

Attributes

effect_constraint_condition	The specification of condition that restricts the Effector_Solution from delivering the Effect.

temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. the start and stop time of the Constraint.
effect_context	The context in which the Constraint is applicable.
effect_constraint_breached	A notification of a breach of the Constraint.

Resourcing_Constraint

This interface is a Constraint limiting the use of resources (e.g. do not use power from a specified source) and an indication if the Constraint is breached.

Attributes

resourcing_constraint_condition	The specification of condition that restricts the Effector_Resource use of external resources, e.g. do not use power from a specified source.
temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. the start and stop time of the Constraint.
resourcing_context	The context in which the Constraint is applicable.
resourcing_constraint_breached	A notification of a breach of the Constraint.

Activities

assess_impact_of_effector_constraint

Assess the impact of Constraint details against the aspect of the Effector_Function's behaviour that is being constrained.

identify_required_effecting_context

Identify the context which defines whether the Constraint is relevant.

5.4.2.16.7.1.6 Capability

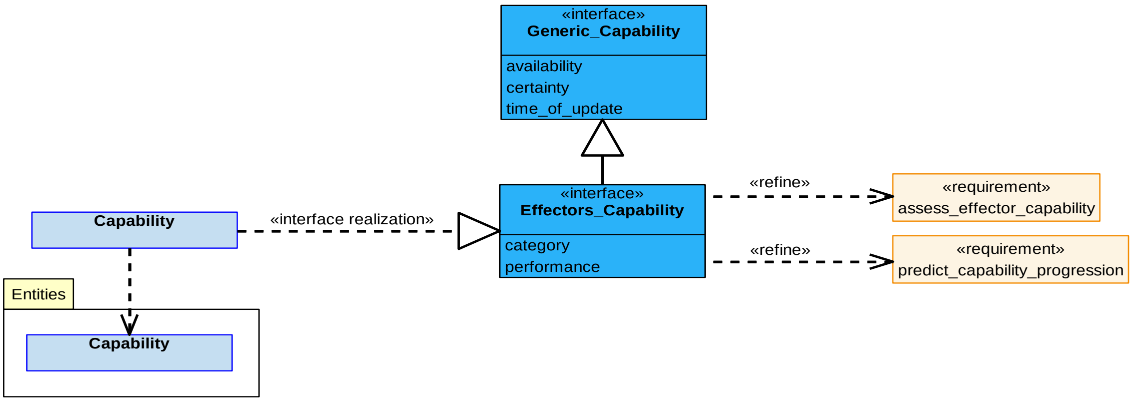


Figure 288: Capability Service Definition

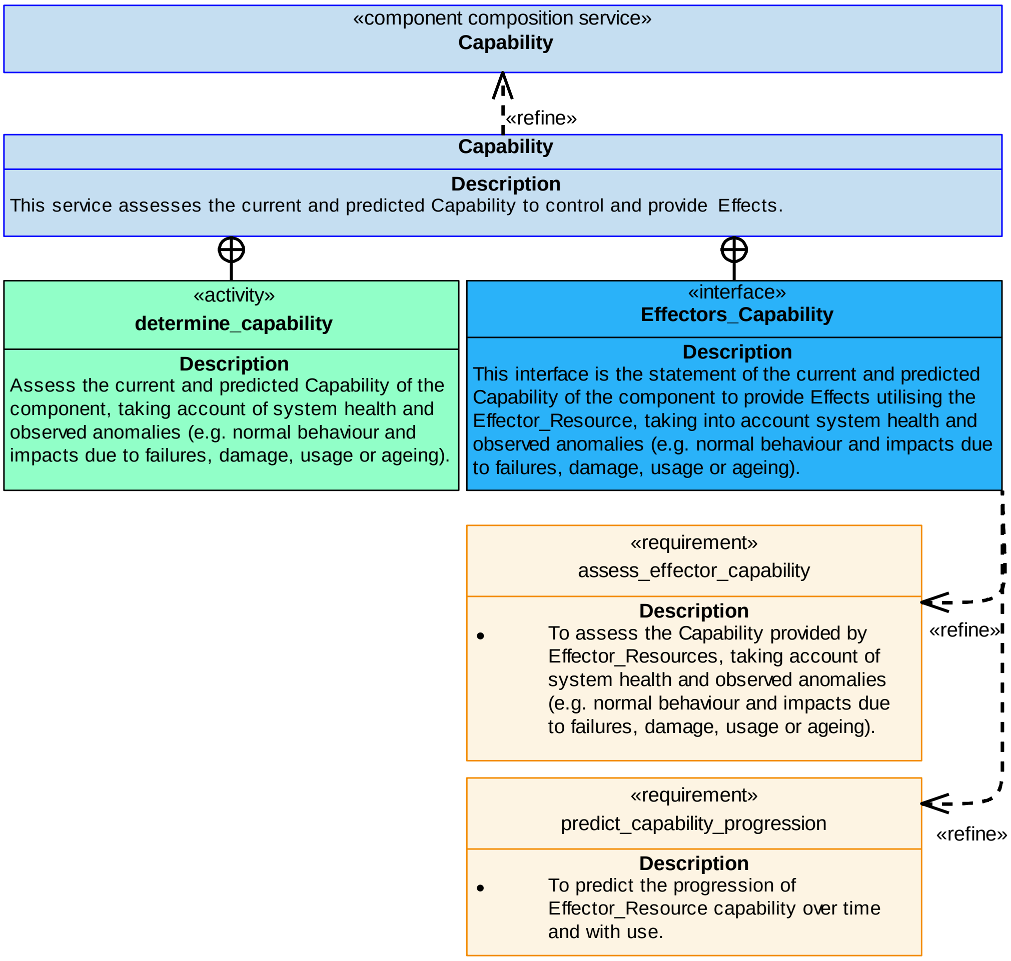

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to control and provide Effects.
«activity»
determine_capability
Description
Assess the current and predicted Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Effectors_Capability
Description
This interface is the statement of the current and predicted Capability of the component to provide Effects utilising the Effector_Resource, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«requirement»
assess_effector_capability
Description
To assess the Capability provided by Effector_Resources, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of Effector_Resource capability over time and with use.
«refine»

Figure 289: Capability Service Policy

Capability

This service assesses the current and predicted Capability to control and provide Effects.

Interface

Effectors_Capability

This interface is the statement of the current and predicted Capability of the component to provide Effects utilising the Effector_Resource, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Attributes

category	The type or category of Capability, e.g. jamming.
performance	The level of performance or effectiveness that can be achieved for this Capability.

Activity

determine_capability

Assess the current and predicted Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.16.7.1.7 Capability_Evidence

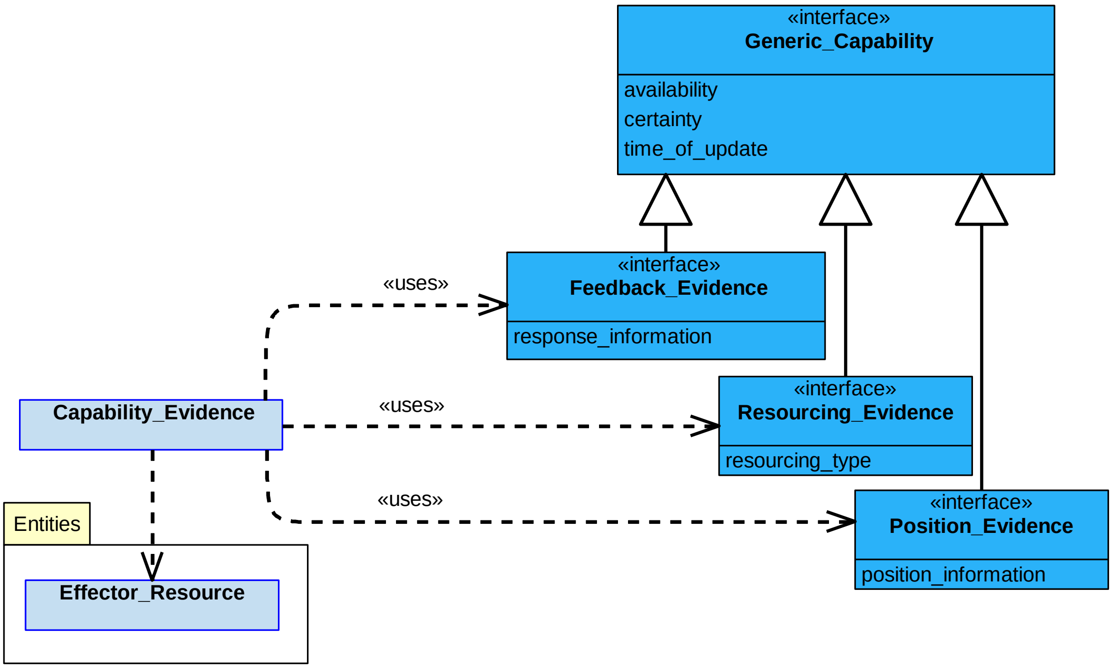
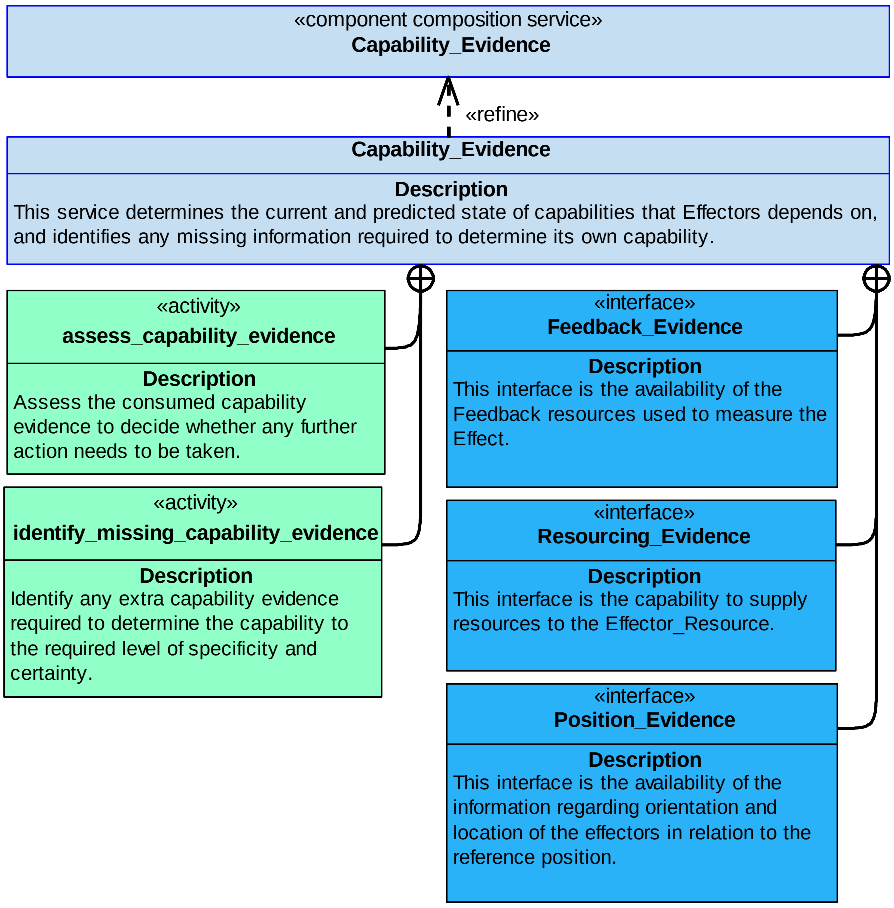


Figure 290: Capability_Evidence Service Definition



This service determines the current and predicted state of capabilities that Effectors depends on, and identifies any missing information required to determine its own capability.

Assess the consumed capability evidence to decide whether any further action needs to be taken.

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

This interface is the availability of the Feedback resources used to measure the Effect.

This interface is the capability to supply resources to the Effector_Resource.

This interface is the availability of the information regarding orientation and location of the effectors in relation to the reference position.

Figure 291: Capability_Evidence Service Policy

Capability_Evidence

This service determines the current and predicted state of capabilities that Effectors depends on, and identifies any missing information required to determine its own capability.

<u>Interfaces</u>

Feedback_Evidence

This interface is the availability of the Feedback resources used to measure the Effect.

<u>Attribute</u>

response_information	The type of information relating to the availability of Feedback.

Resourcing_Evidence

This interface is the capability to supply resources to the Effector_Resource.

<u>Attribute</u>

resourcing_type	The definition of the resource.

Position_Evidence

This interface is the availability of the information regarding orientation and location of the effectors in relation to the reference position.

<u>Attribute</u>

position_information	The availability of information relating to orientation and location.

<u>Activities</u>

assess_capability_evidence

Assess the consumed capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.16.7.2 Service Dependencies

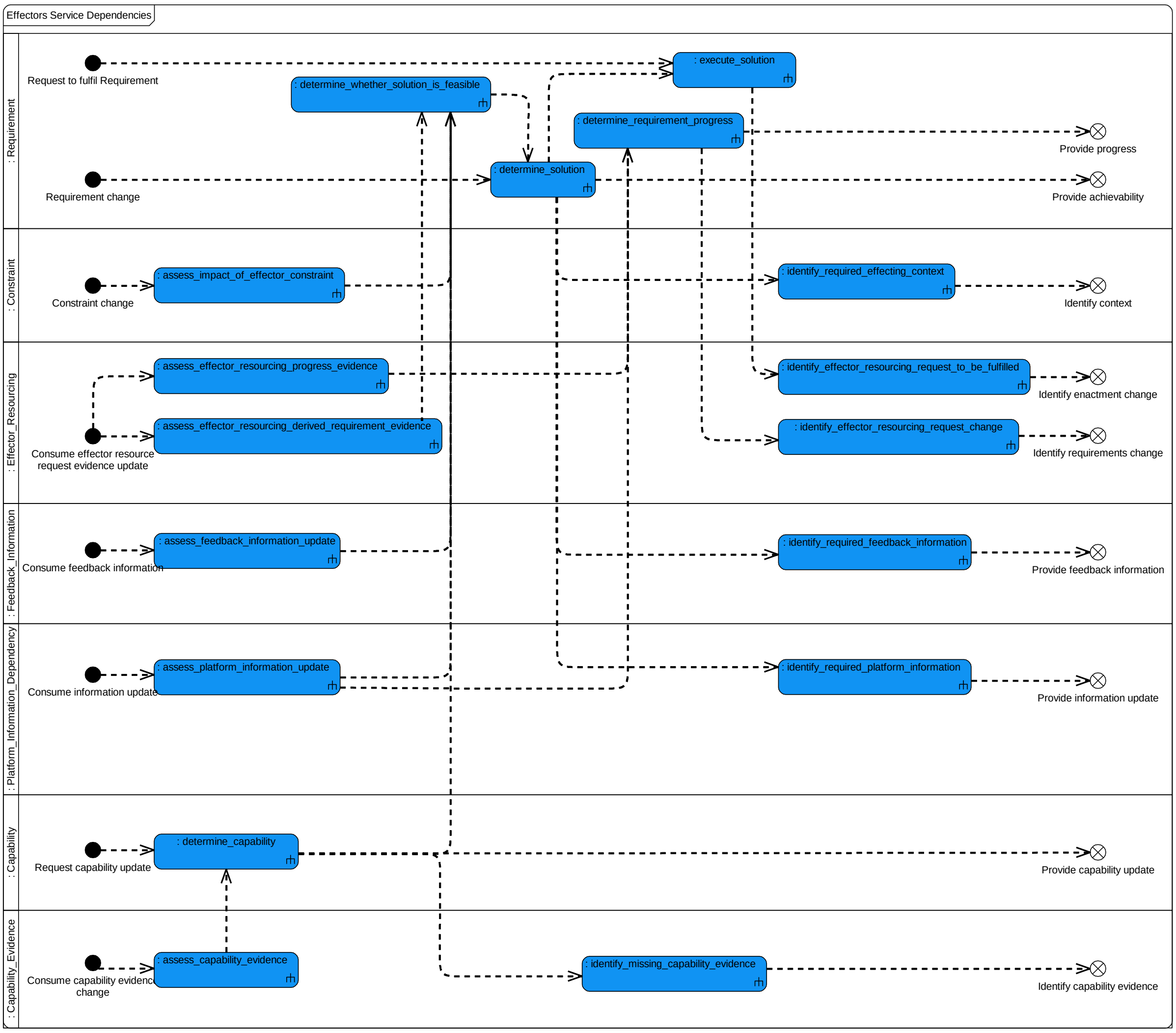

Effectors Service Dependencies
: Requirement
Request to fulfil Requirement
: determine_whether_solution_is_feasible
: execute_solution
: determine_requirement_progress
Provide progress
: determine_solution
Requirement change
Provide achievability
: Constraint
Constraint change
: assess_impact_of_effector_constraint
: identify_required_effecting_context
Identify context
: Effector_Resourcing
: assess_effector_resourcing_progress_evidence
: assess_effector_resourcing_derived_requirement_evidence
Consume effector resource request evidence update
: identify_effector_resourcing_request_to_be_fulfilled
Identify enactment change
: identify_effector_resourcing_request_change
Identify requirements change
: Feedback_Information
Consume feedback information
: assess_feedback_information_update
: identify_required_feedback_information
Provide feedback information
: Platform_Information_Dependency
Consume information update
: assess_platform_information_update
: identify_required_platform_information
Provide information update
: Capability
Request capability update
: determine_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 292: Effectors Service Dependencies

5.4.2.17 Environment Infrastructure

5.4.2.17.1 Role

The role of Environment Infrastructure is to provide information about the infrastructure that exists within an operating environment to support the operation of the Exploiting Platform.

5.4.2.17.2 Overview

Control Architecture

Environment Infrastructure is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Upon receiving a request for infrastructure information, this component will identify the relevant Infrastructure_Features and provide the information. The information may be the relationship between Infrastructure_Features (Feature_Relationship), a feature's relationship with a Reference_Item (Reference_Relationship), or further information derived from the properties of an Infrastructure_Feature.

Examples of Use

Environment Infrastructure will be required when a deployment needs to:

- Identify landing locations that satisfy received requests for information, such as regions with certain properties e.g. minimum dimensions, gradient within a range, need for authorisation, or proximity to an aircraft hangar.
- Determine whether a Reference_Item (such as a projected route) conflicts with a Feature_Type (e.g. ATS zone, or a no-fly zone).
- Identify applicable beacons that may be used for navigation.
- Determine a profile for take-off or landing.

5.4.2.17.3 Service Summary

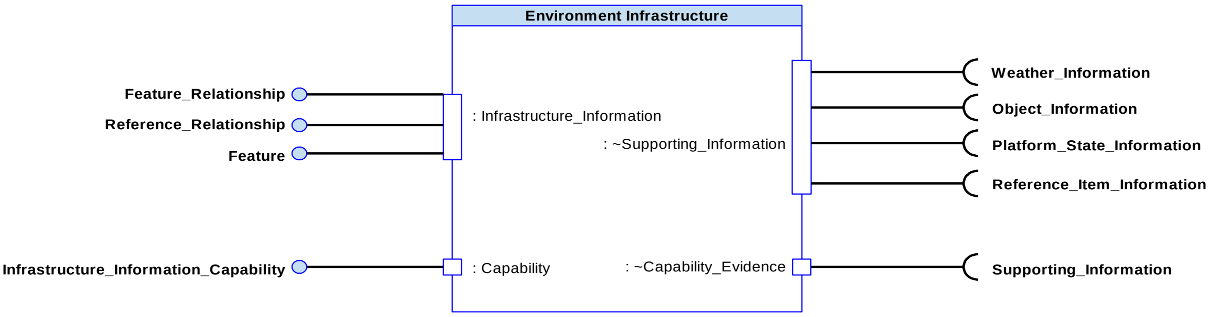

Environment Infrastructure
Feature_Relationship
Reference_Relationship
Feature
Infrastructure_Information_Capability
: Infrastructure_Information
: ~Supporting_Information
: Capability
: ~Capability_Evidence
Weather_Information
Object_Information
Platform_State_Information
Reference_Item_Information
Supporting_Information

Figure 293: Environment Infrastructure Service Summary

5.4.2.17.4 Responsibilities

capture_infrastructure_information_request

- To capture the requests for information on Infrastructure_Features, the relationship between them, or the relationship between a Reference_Item and Infrastructure_Features (e.g. a request to identify recovery locations, or suitable beacons).

determine_CTT_forced_landing_locations

- To determine available Controlled-Trajectory Termination (CTT) or forced landing locations.

determine_terminal_operation_areas

- To determine available Terminal_Operation_Areas, including those which are not formally designated by authorities.

determine_infrastructure_feature_properties

- To determine the properties of Infrastructure_Features.

determine_navigation_aids

- To determine available aids to navigation.

determine_minimum_safe_altitude

- To determine Minimum Safe Altitude (MSA) levels.

determine_TOA_profiles

- To determine the available profiles to support taxi, launch and recovery within a Terminal_Operation_Area.

determine_reference_relationship

- To determine the relationship between a Reference_Item (e.g. ownship position) and Infrastructure_Features.

determine_infrastructure_conflict

- To determine when a Reference_Item (e.g. ownship's projected route) conflicts with an Infrastructure_Feature (e.g. path enters no-fly zone or breaches MSA).

determine_feature_relationship

- To determine information on how Infrastructure_Features relate to each other.

assess_capability

- To assess the Infrastructure_Capability to provide infrastructure information taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing such as loss of an infrastructure data resource).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Infrastructure_Capability assessment (e.g. identifying that information from an infrastructure data source may not have been updated).

5.4.2.17.5 Subject Matter Semantics

The subject matter of Environment Infrastructure is the manmade features in the operating environment that are relevant to the operation of an Exploiting Platform.

Exclusions

The subject matter of Environment Infrastructure does not include:

- The size and shape of physical manmade features.
- Authorisation for entry to regions such as no-fly zones and ATS zones.

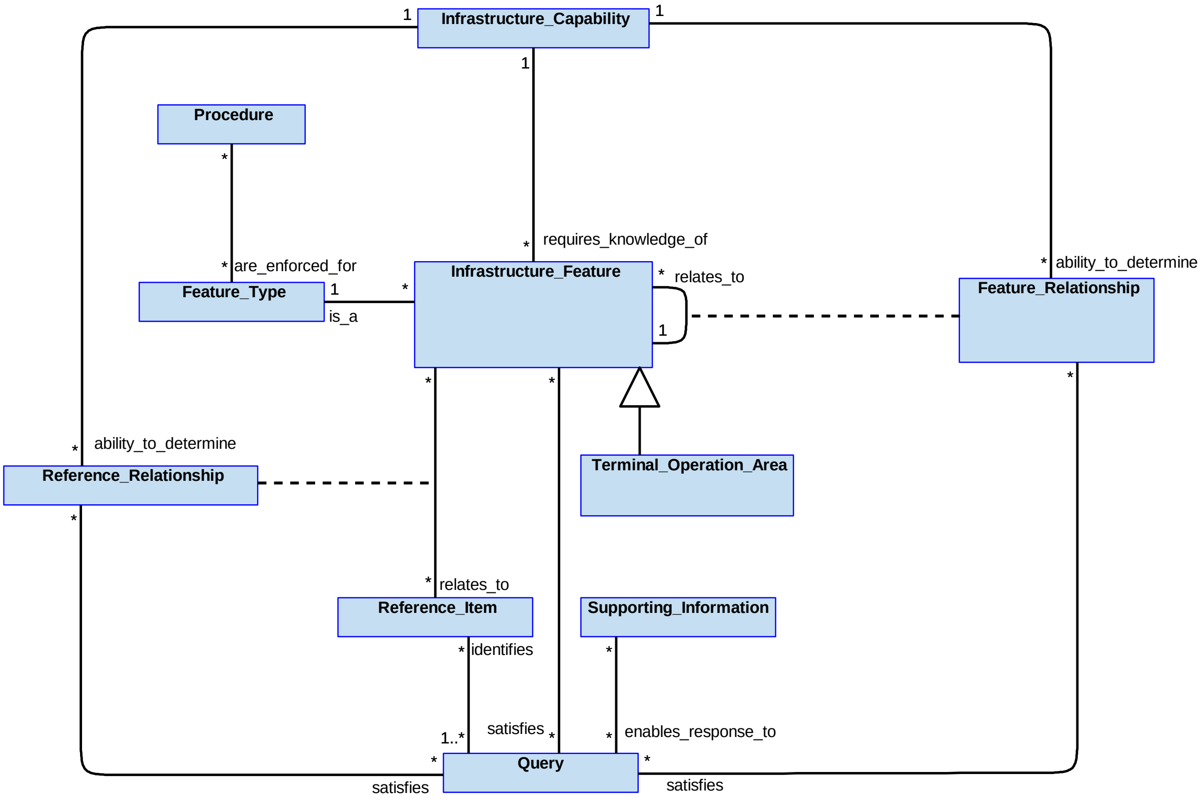


Figure 294: Environment Infrastructure Semantics

5.4.2.17.5.1 Entities

Feature_Relationship

The relationship between Infrastructure_Features, such as the distance between a Terminal_Operation_Area and no-fly zone.

Feature_Type

The type of Infrastructure_Feature, e.g. no fly-zones, airways, navigational aids, or TOAs.

Infrastructure_Capability

The ability to provide information on Infrastructure_Features, Feature_Relationships and Reference_Relationships, and the ability to determine such relationships.

Infrastructure_Feature

A feature in the operating environment, e.g. a specific runway or a beacon.

Procedure

The established set of rules or actions that should be followed by traffic when using a Feature_Type, e.g. not entering a specific no-fly zones.

Query

A request to determine a set of properties relating to an Infrastructure_Feature, a Feature_Relationship, or a Reference_Relationship.

Reference_Item

A specific item in, or a reference to, the physical world whose relative properties are the subject of a request, e.g. ownship position, a sensor owned by the Exploiting Platform, or a projected route. Where an item has a spatial extent the position of this extent will be defined.

Reference_Relationship

The relationship between an Infrastructure_Feature and a Reference_Item, e.g. the conflict between ownship's projected route and a no-fly zone.

Supporting_Information

Information used to determine the response to a Query, e.g. information about the operating environment, or relating to a Reference_Item.

Terminal_Operation_Area

A defined area on land or water (including any buildings, installations, ships and equipment) intended to be used either wholly or in part for the arrival, departure and surface movement of aircraft (including helicopters).

5.4.2.17.6 Design Rationale

5.4.2.17.6.1 Assumptions

- Environment Infrastructure will represent Infrastructure_Features that must not be entered such as no-fly zones or volumes below MSA levels.
- Environment Infrastructure provides Procedures and constraints based on properties of Infrastructure_Features (e.g. only vertical-landing allowed at a terminal or authorisation required for an airspace region). The security classifications of these Procedures will depend on whether they are civil or military.
- The above Procedures, constraints, and properties may be safety critical, thus appropriate protection to avoid spoofing (e.g. of ATS) and to defend against unauthorised manipulation, etc. may be required.
- Knowledge of a vehicle's location will not be held within this component. It should instead be the request for this component's service that specifies the location of interest.

5.4.2.17.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Environment Infrastructure:

- Data Driving - Environment Infrastructure understands the structure of the operating environment and Procedures to be utilised within that environment. The structure of, and procedures for, the environment will be subject to occasional change during the lifetime of an Exploiting Programme and could be data-driven.

Exploitation Considerations

- Environment Infrastructure may define the location of an Infrastructure_Feature in absolute terms, or relative to the location of a different component's entity (such location information will be held with the entity it belongs to, within the component that owns said entity). For example, Environment Infrastructure may define a no-fly zone that coincides with a physical boundary defined in Geography (e.g. a country border), or it may define the no-fly zone using an offset from a tactical entity held in Tactical Objects.

5.4.2.17.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- The most critical use for this component is considered to be provision of runway data (including elevation, lengths and slope) used to determine take-off and landing performance data (e.g. maximum take-off mass, maximum landing mass and V1). If this data is incorrect then an air vehicle could take-off or land above a safe mass, potentially resulting in an uncontrolled crash, resulting in loss of air vehicle and fatalities.

5.4.2.17.6.4 Security Considerations

The indicative component security classification is O.

This component provides information about the infrastructure that exists within the operating environment, e.g. Air Traffic Services (ATS) controlled areas and Terminal_Operation_Areas (TOA). For civil authority-controlled areas, this information is O, however this could be up to SNEO for military-controlled areas. It may be appropriate to have different instances (and implementations) within different security domains, these instances could be required to coordinate; separation would be enforced outside of these instances. The component is not expected to require absolute knowledge of vehicle location but in reasoning about a location of interest, approximate own position may be divulged.

This component is considered a subject of interest for an adversary and a likely target for a cyber attack and will need appropriate protection. Loss of integrity or availability could affect safe operation of the vehicle in relation to operations in these areas.

The component may be expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** of external data sources (e.g. the local ATC) that provide the properties of the Infrastructure_Features etc.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **Supporting Secure Remote Operation** of uncrewed aircraft around a TOA.

The component is not expected to directly implement security enforcing functions but relies on the integrity of external data sources.

5.4.2.17.7 Services

5.4.2.17.7.1 Service Definitions

5.4.2.17.7.1.1 Infrastructure_Information

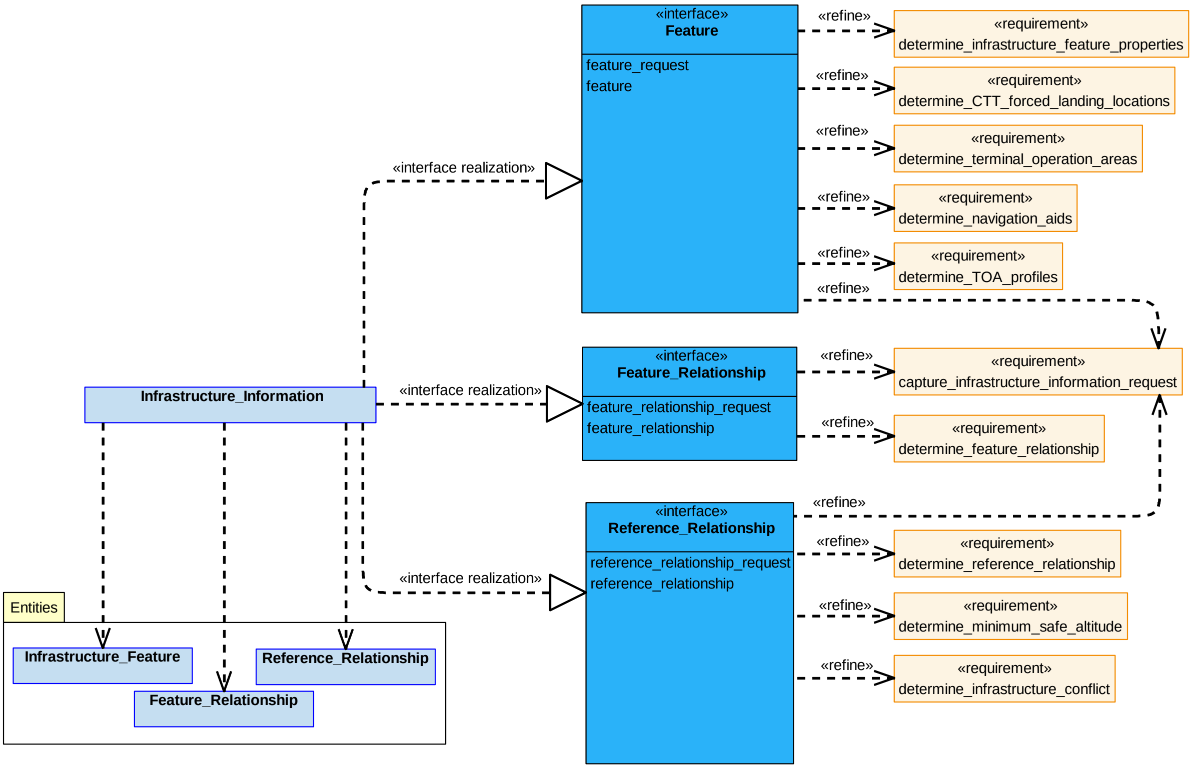


Figure 295: Infrastructure_Information Service Definition

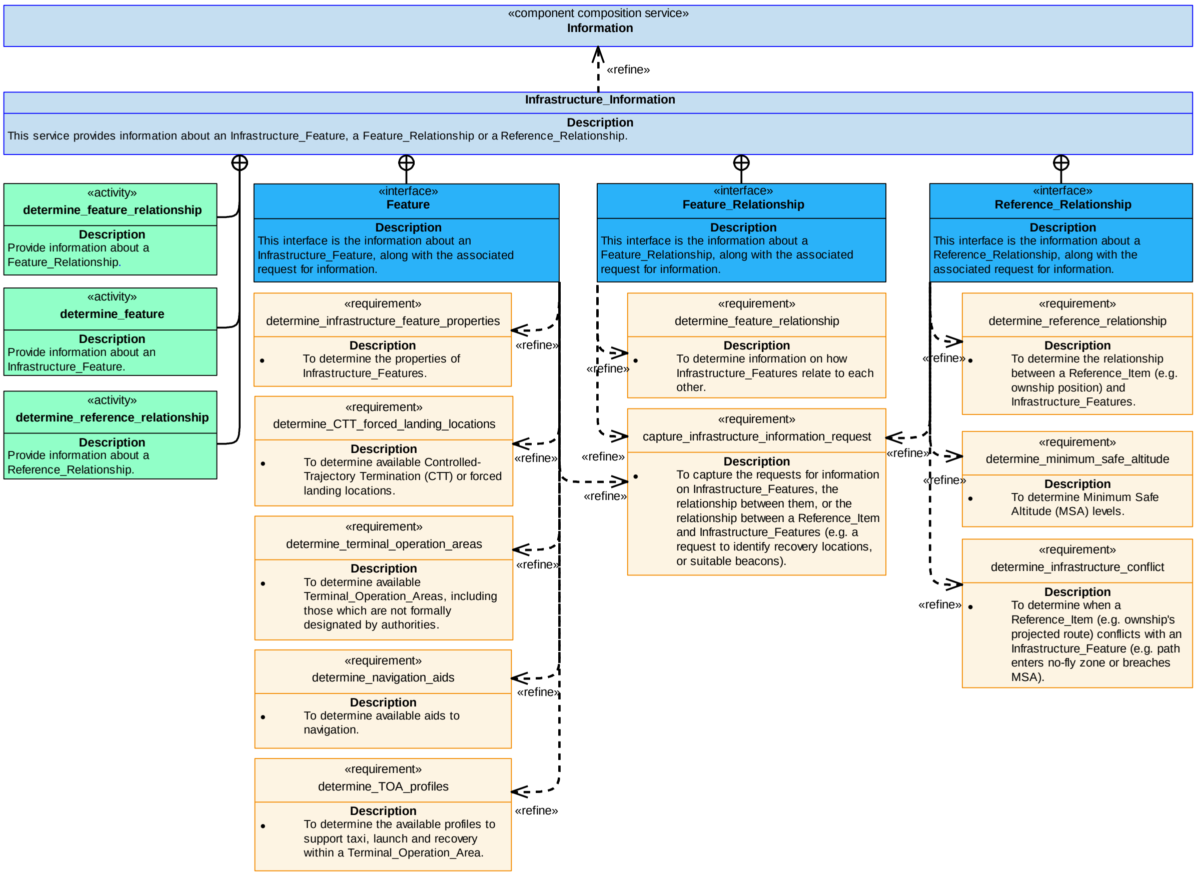

«component composition service»
Information
«refine»
Infrastructure_Information
Description
This service provides information about an Infrastructure_Feature, a Feature_Relationship or a Reference_Relationship.
«activity»
determine_feature_relationship
Description
Provide information about a Feature_Relationship.
«activity»
determine_feature
Description
Provide information about an Infrastructure_Feature.
«activity»
determine_reference_relationship
Description
Provide information about a Reference_Relationship.
«interface»
Feature
Description
This interface is the information about an Infrastructure_Feature, along with the associated request for information.
«interface»
Feature_Relationship
Description
This interface is the information about a Feature_Relationship, along with the associated request for information.
«interface»
Reference_Relationship
Description
This interface is the information about a Reference_Relationship, along with the associated request for information.
«requirement»
determine_infrastructure_feature_properties
Description
To determine the properties of Infrastructure_Features.
«requirement»
determine_CTT_forced_landing_locations
Description
To determine available Controlled-Trajectory Termination (CTT) or forced landing locations.
«requirement»
determine_terminal_operation_areas
Description
To determine available Terminal_Operation_Areas, including those which are not formally designated by authorities.
«requirement»
determine_navigation_aids
Description
To determine available aids to navigation.
«requirement»
determine_TOA_profiles
Description
To determine the available profiles to support taxi, launch and recovery within a Terminal_Operation_Area.
«requirement»
determine_feature_relationship
Description
To determine information on how Infrastructure_Features relate to each other.
«requirement»
capture_infrastructure_information_request
Description
To capture the requests for information on Infrastructure_Features, the relationship between them, or the relationship between a Reference_Item and Infrastructure_Features (e.g. a request to identify recovery locations, or suitable beacons).
«requirement»
determine_reference_relationship
Description
To determine the relationship between a Reference_Item (e.g. ownship position) and Infrastructure_Features.
«requirement»
determine_minimum_safe_altitude
Description
To determine Minimum Safe Altitude (MSA) levels.
«requirement»
determine_infrastructure_conflict
Description
To determine when a Reference_Item (e.g. ownship's projected route) conflicts with an Infrastructure_Feature (e.g. path enters no-fly zone or breaches MSA).
«refine»

Figure 296: Infrastructure_Information Service Policy

Infrastructure_Information

This service provides information about an Infrastructure_Feature, a Feature_Relationship or a Reference_Relationship.

Interfaces

Feature_Relationship

This interface is the information about a Feature_Relationship, along with the associated request for information.

Attributes

feature_relationship_request	The definition of the request for information about a Feature_Relationship.
feature_relationship	The details of the relationship between Infrastructure_Features, e.g. the distance between two beacons.

Reference_Relationship

This interface is the information about a Reference_Relationship, along with the associated request for information.

Attributes

reference_relationship_request	The definition of the request for information about a Reference_Relationship.
reference_relationship	The details of the relationship between Infrastructure_Features and Reference_Items, e.g. the distance between an air vehicle and a navigation beacon.

Feature

This interface is the information about an Infrastructure_Feature, along with the associated request for information.

Attributes

feature_request	The definition of the request for information about an Infrastructure_Feature.
feature	The details of the Infrastructure_Feature, e.g. location or Feature_Type, or a take-off or landing profile for a Terminal_Operation_Area.

Activities

determine_feature

Provide information about an Infrastructure_Feature.

determine_feature_relationship

Provide information about a Feature_Relationship.

determine_reference_relationship

Provide information about a Reference_Relationship.

5.4.2.17.7.1.2 Supporting_Information

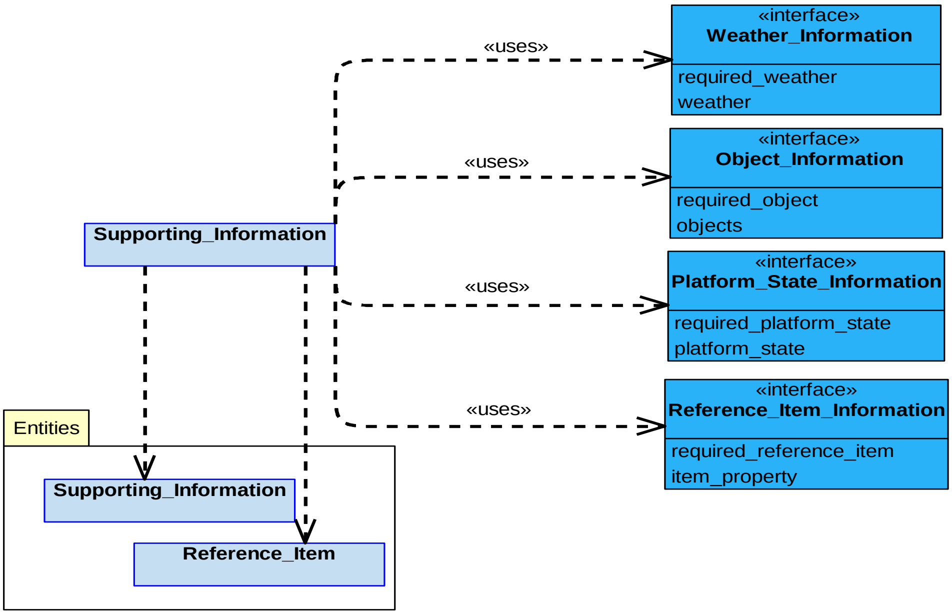


Figure 297: Supporting_Information Service Definition

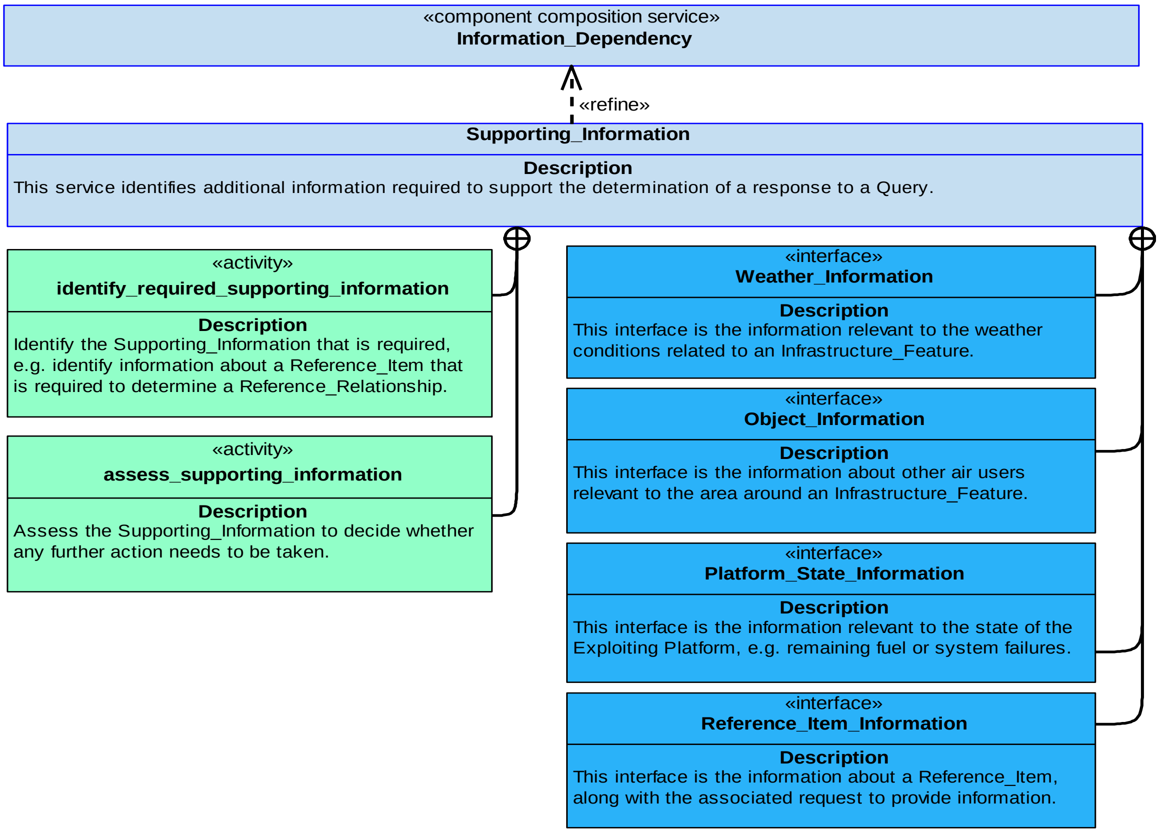

«component composition service»
Information_Dependency
«refine»
Supporting_Information
Description
This service identifies additional information required to support the determination of a response to a Query.
«activity»
identify_required_supporting_information
Description
Identify the Supporting_Information that is required, e.g. identify information about a Reference_Item that is required to determine a Reference_Relationship.
«activity»
assess_supporting_information
Description
Assess the Supporting_Information to decide whether any further action needs to be taken.
«interface»
Weather_Information
Description
This interface is the information relevant to the weather conditions related to an Infrastructure_Feature.
«interface»
Object_Information
Description
This interface is the information about other air users relevant to the area around an Infrastructure_Feature.
«interface»
Platform_State_Information
Description
This interface is the information relevant to the state of the Exploiting Platform, e.g. remaining fuel or system failures.
«interface»
Reference_Item_Information
Description
This interface is the information about a Reference_Item, along with the associated request to provide information.

Figure 298: Supporting_Information Service Policy

Supporting_Information

This service identifies additional information required to support the determination of a response to a Query.

<u>Interfaces</u>

Weather_Information

This interface is the information relevant to the weather conditions related to an Infrastructure_Feature.

<u>Attributes</u>

required_weather	The identification of the information need associated with the weather conditions related to an Infrastructure_Feature, e.g. the need for information regarding the weather at an identified airfield.
weather	The weather conditions that apply to the Infrastructure_Feature.

Object_Information

This interface is the information about other air users relevant to the area around an Infrastructure_Feature.

Attributes

required_object	The identification of the information need associated with other air users relevant to the area around an Infrastructure_Feature, e.g. the need for information regarding other air users in an area around an identified airfield.
objects	The objects (e.g. other air users) operating in the area around an Infrastructure_Feature.

Platform_State_Information

This interface is the information relevant to the state of the Exploiting Platform, e.g. remaining fuel or system failures.

Attributes

required_platform_state	The identification of the information need associated with the state of the Exploiting Platform, e.g. the need for information regarding the remaining fuel on the air vehicle.
platform_state	The state of the Exploiting Platform, e.g. remaining fuel or system failures.

Reference_Item_Information

This interface is the information about a Reference_Item, along with the associated request to provide information.

Attributes

required_reference_item	The identification of the information need associated with a Reference_Item, e.g. the need for information regarding a navigation beacon location.
item_property	The properties about a Reference_Item, e.g. size or location.

Activities

assess_supporting_information

Assess the Supporting_Information to decide whether any further action needs to be taken.

identify_required_supporting_information

Identify the Supporting_Information that is required, e.g. identify information about a Reference_Item that is required to determine a Reference_Relationship.

5.4.2.17.7.1.3 Capability

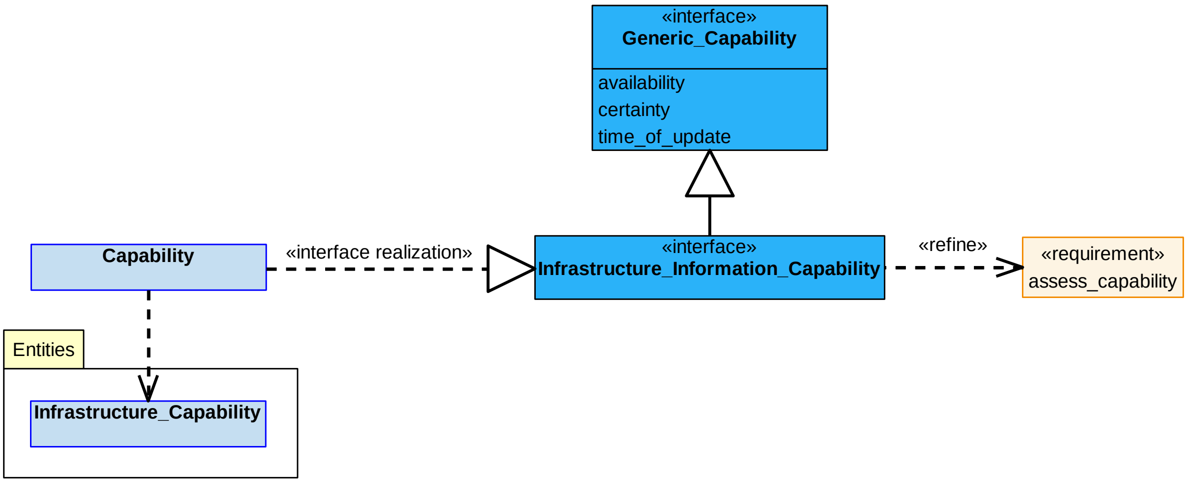


Figure 299: Capability Service Definition

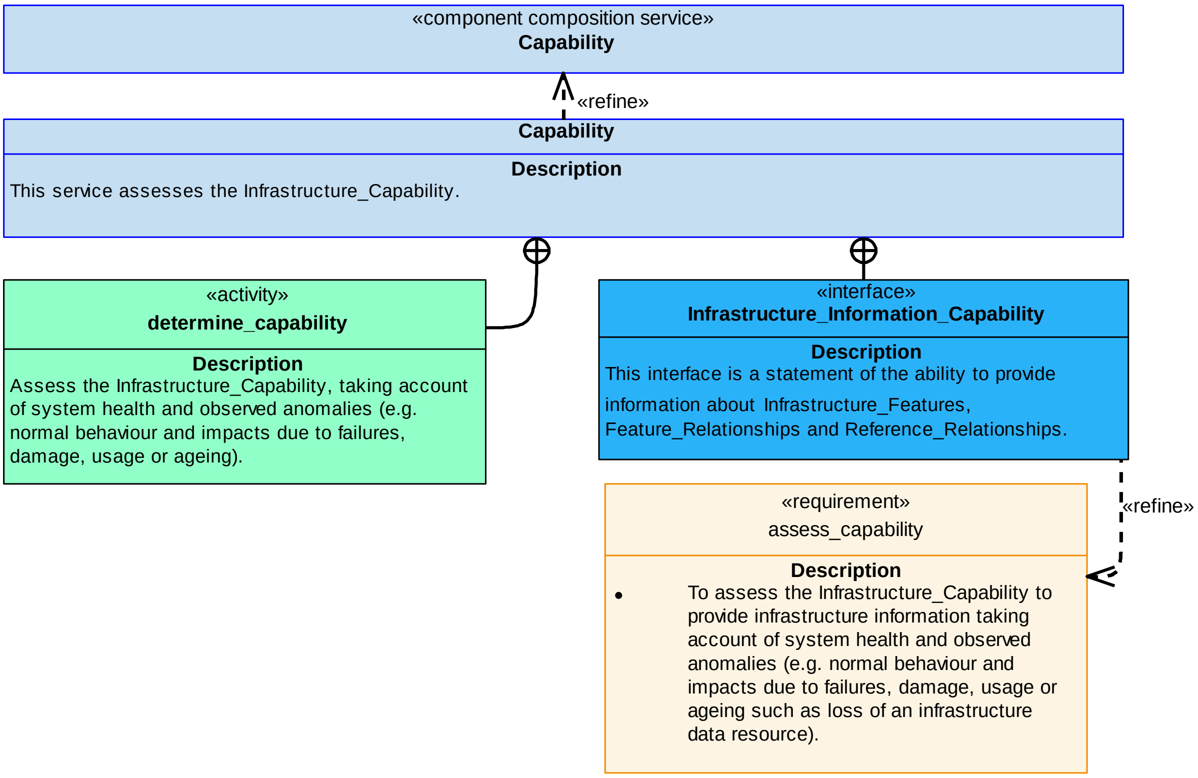

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the Infrastructure_Capability.
«activity»
determine_capability
Description
Assess the Infrastructure_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Infrastructure_Information_Capability
Description
This interface is a statement of the ability to provide information about Infrastructure_Features, Feature_Relationships and Reference_Relationships.
«refine»
«requirement»
assess_capability
Description
To assess the Infrastructure_Capability to provide infrastructure information taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing such as loss of an infrastructure data resource).

Figure 300: Capability Service Policy

Capability

This service assesses the Infrastructure_Capability.

Interface

Infrastructure_Information_Capability

This interface is a statement of the ability to provide information about Infrastructure_Features, Feature_Relationships and Reference_Relationships.

Activity

determine_capability

Assess the Infrastructure_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.17.7.1.4 Capability_Evidence

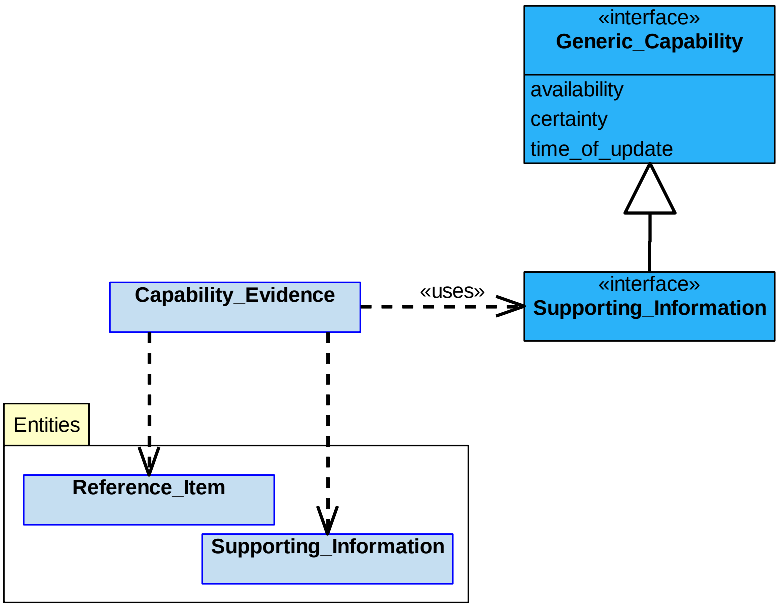


Figure 301: Capability_Evidence Service Definition

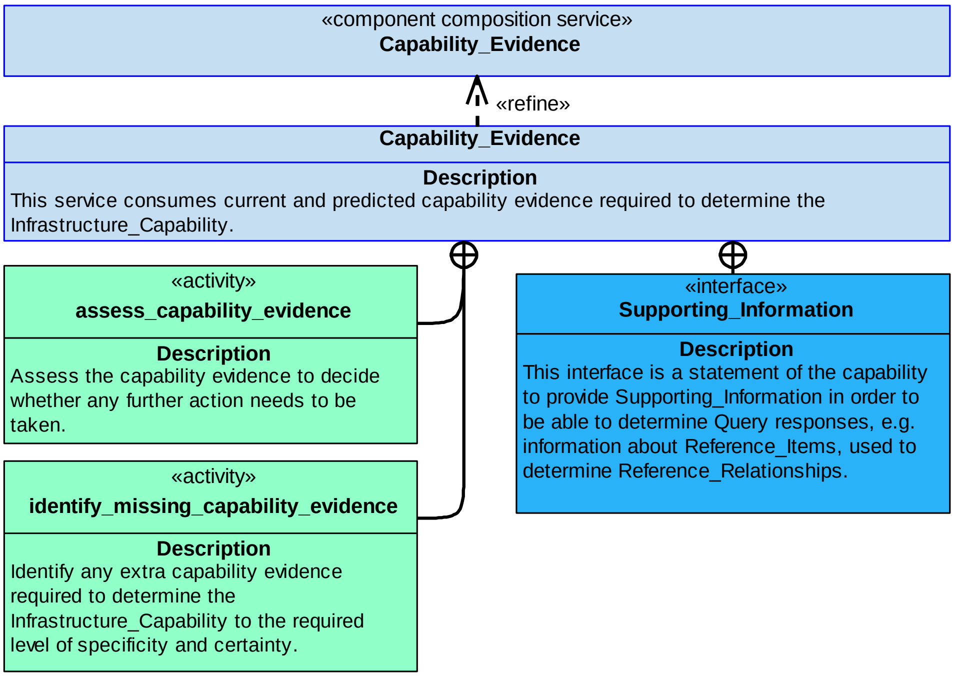

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability evidence required to determine the Infrastructure_Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Infrastructure_Capability to the required level of specificity and certainty.
«interface»
Supporting_Information
Description
This interface is a statement of the capability to provide Supporting_Information in order to be able to determine Query responses, e.g. information about Reference_Items, used to determine Reference_Relationships.

Figure 302: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability evidence required to determine the Infrastructure_Capability.

Interface

Supporting_Information

This interface is a statement of the capability to provide Supporting_Information in order to be able to determine Query responses, e.g. information about Reference_Items, used to determine Reference_Relationships.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Infrastructure_Capability to the required level of specificity and certainty.

5.4.2.17.7.2 Service Dependencies

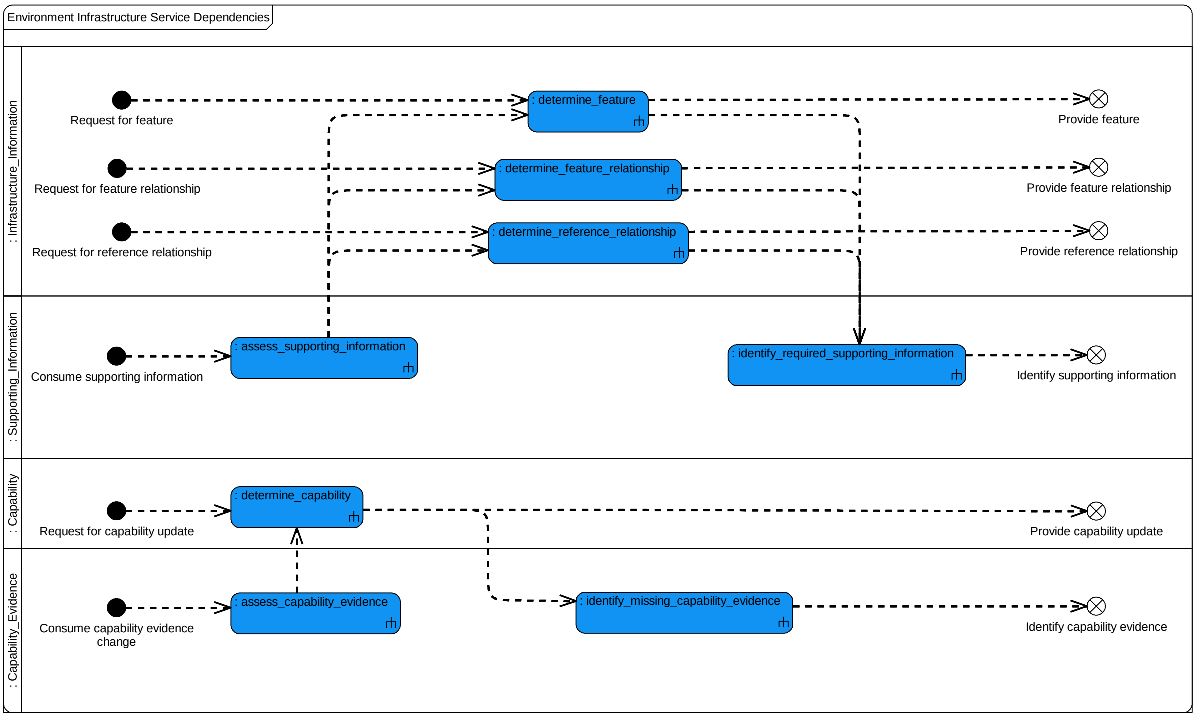

Environment Infrastructure Service Dependencies
: Infrastructure_Information
: Supporting_Information
: Capability
: Capability_Evidence
Request for feature
: determine_feature
Provide feature
Request for feature relationship
: determine_feature_relationship
Provide feature relationship
Request for reference relationship
: determine_reference_relationship
Provide reference relationship
Consume supporting information
: assess_supporting_information
: identify_required_supporting_information
Identify supporting information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 303: Environment Infrastructure Service Dependencies

5.4.2.18 Environment Integration

5.4.2.18.1 Role

The role of Environment Integration is to manage the integration of the Exploiting Platform (e.g. an air vehicle) with the physical operating environment.

5.4.2.18.2 Overview

Control Architecture

Environment Integration is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Environment Integration receives a Requirement to perform an action related to integration with the operating environment. Environment Integration determines an Integration_Solution to achieve this, taking into account Integration_Capability, Constraints, Integration_Settings, Protocols, and identifying the Pre-conditions that need to be satisfied. Environment Integration enacts the Integration_Solution using the System_Capability, and monitors the Integration_Solution to determine the effectiveness until the Outcome has been met.

Examples of Use

Environment Integration will be required to:

- Coordinate activities associated with air mobility activities (e.g. air-to-air refuelling).
- Coordinate transitioning between regions in the operating environment, including the setting of any identification codes received from ATS.
- Understand flight levels.
- Request authorisations and changes to communication end points related to environment integration.
- Plan and coordinate the conditions needed for take-off and landing.

5.4.2.18.3 Service Summary

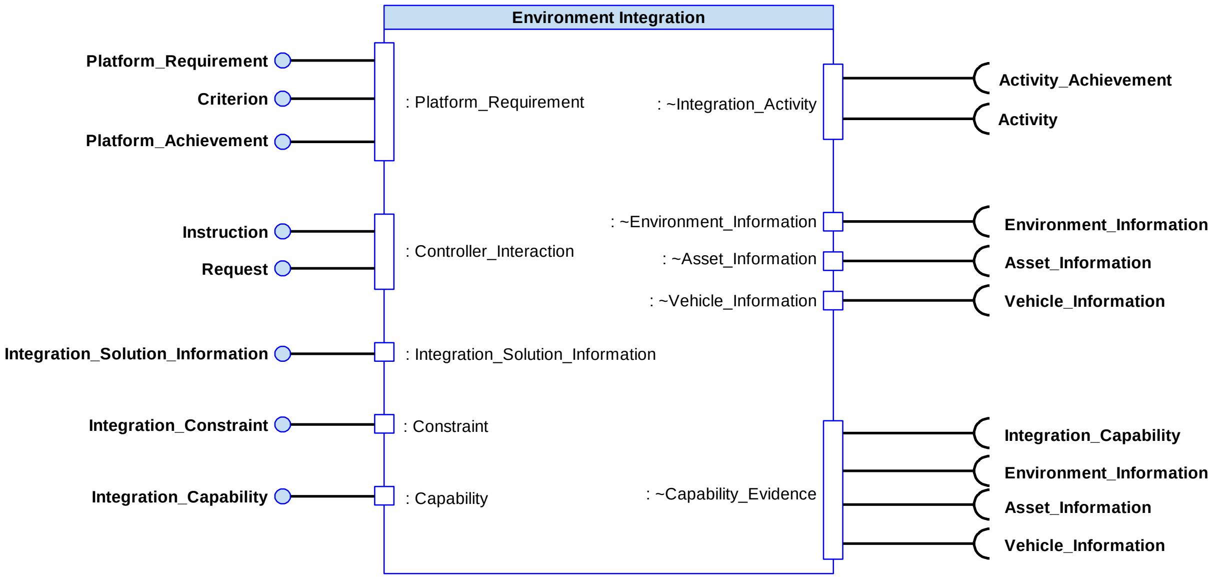


Figure 304: Environment Integration Service Summary

5.4.2.18.4 Responsibilities

capture_requirements_for_environment_integration_actions

- To capture provided Requirements (e.g. coordinate ATC transition) for environment integration actions.

capture_measurement_criteria_for_environment_integration_actions

- To capture provided Measurement_Criterion/criteria that an Integration_Solution and its Outcomes will be measured against.

capture_environment_integration_constraints

- To capture provided Constraints for environment integration actions.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Integration_Capability and Constraints.

determine_environment_integration_solution

- To determine an Integration_Solution that meets the given Requirements within provided Constraints using the available System_Capability.

determine_predicted_quality_of_environment_integration_solution

- To determine the predicted quality of the Integration_Solution against given Measurement_Criterion/criteria.

determine_integration_settings

- To determine the Integration_Settings for the operating environment.

determine_applicable_environment_integration_rules

- To determine the currently applicable Protocols for integrating into the operating environment.

identify_environment_integration_pre_conditions

- To identify Pre-conditions required to support the Integration_Solution or an Action_Step of the Integration_Solution.

coordinate_environment_integration_solution

- To coordinate the execution of an Integration_Solution.

coordinate_interactions_with_controlling_service

- To generate and interpret automated interactions with the Controlling_Service.

identify_progress_of_environment_integration_solution

- To identify the progress of an Integration_Solution against the Requirements.

determine_actual_quality_of_environment_integration_deliverables

- To determine the actual quality of the Integration_Solution against given Measurement_Criterion/criteria.

provide_integration_solution_information

- To provide information relating to an Integration_Solution.

assess_environment_integration_capability

- To assess the Integration_Capability to perform environment integration actions (e.g. respond to ATC instruction) taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

predict_capability_progression

- To predict the progression of Environment Integration's Integration_Capability over time and with use.

5.4.2.18.5 Subject Matter Semantics

The subject matter of Environment Integration is the solutions that enable an Exploiting Platform (e.g. an air vehicle) to integrate with the external environment, including both civil and military airspaces. This includes, but is not limited to, transiting and transitioning, terminal operations, determining behavioural rules based on the environment and performing automated and non-human interactions with controlling services.

Exclusions

The subject matter of Environment Integration does not include:

- The impact of the environment on route planning.

- The overall structure of the environment; although relevant aspects needed for integrating an Exploiting Platform into the environment are obtained and understood by the component, such as the location of marker beacons for instrumented landings.
- Details for communications, such as a communications plan or specific communications technology details, that are beyond the scope of the specific high level protocols that are necessary for integration with the environment.

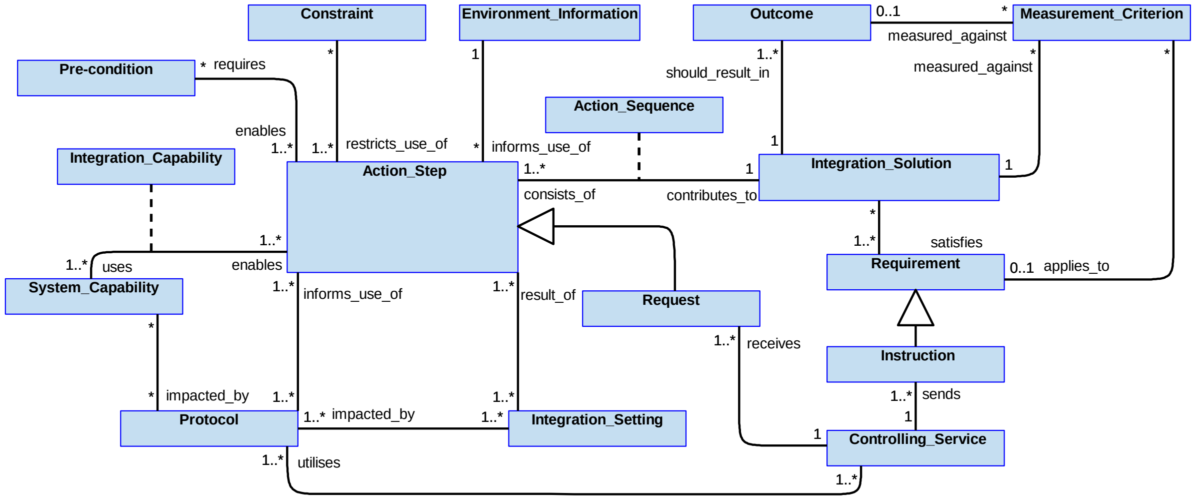


Figure 305: Environment Integration Semantics

5.4.2.18.5.1 Entities

Action_Sequence

The order in which Action_Steps must be performed to achieve an Integration_Solution.

Action_Step

An action that, when performed, achieves (or helps to achieve) integration with the operating environment, e.g. requesting ATC approvals, entering an airspace corridor and maintaining a safe MSD from terrain or obstacles in the operating environment.

Constraint

An externally imposed restriction, e.g. aviation rules or the amount of fuel left on the Exploiting Platform.

Controlling_Service

The controlling service, e.g. ATS or the dominant air vehicle involved in air-to-air refuelling, that the Exploiting Platform is interacting with in its current operating environment.

Environment_Information

Data describing the environment with which the platform will integrate, e.g. obstacles, vehicles, or weather formations.

Instruction

A command that instructs the Exploiting Platform to achieve a particular result in relation to integration with the operating environment.

Integration_Capability

The capability to execute environment integration actions via the utilisation of a System_Capability.

Integration_Setting

An integration setting for the particular operating environment, e.g. desired MSD, an altimeter setting, or an identification code.

Integration_Solution

The solution to integrate with the environment, e.g. the solution to transition between two regions in the operating environment or to enact Terminal Operation Area activities (e.g. taxi, launch, or recovery).

Measurement_Criterion

A criterion that the quality of an Integration_Solution and its Outcomes will be measured against.

Outcome

An outcome that will integrate the Exploiting Platform with the operating environment, e.g. a completed transition between regions.

Pre-condition

A condition that must be true, e.g. a transition point has been reached.

Protocol

A set of rules and procedures that are applicable to the operating environment.

Request

A request sent from the Exploiting Platform for a particular reason, e.g. to request authorisation from ATS to deviate from a cleared route or to enact a communications change upon reaching the boundary between two Controlling_Service control areas.

Requirement

A requirement to achieve a result relating to integration with the operating environment, e.g. a terminal operation such as launch.

System_Capability

A capability of the Exploiting Platform that is available for use, e.g. the communications system or vehicle routing.

5.4.2.18.6 Design Rationale

5.4.2.18.6.1 Assumptions

None.

5.4.2.18.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Environment Integration:

- Data Driving - This PYRAMID concept is applicable because the rules for integrating into an operating environment vary depending on the jurisdiction being operated in, so in order to cope with this variation an approach such as data driving should be considered. This allows the component to be reusable between multiple Exploiting Programmes.

Extensions

- Environment Integration will need to manage integration with different types of environment (e.g. civil airspace or a hostile area of operation). This variable nature of the procedures required for different types of environment could be handled by extension components.

Exploitation Considerations

- There could be a single or multiple variants of the Environment Integration component to manage integration with different types of environment (e.g. civil airspace or a hostile area of operation).
- Taxiing and launch from a Terminal Operation Area may involve requesting and gaining ATS approval to taxi to runway and enact the solution.

5.4.2.18.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- The conclusion of DAL A is driven by Integration_Settings being erroneous. For example an incorrect altimeter setting (pressure altitude reference such as QNH) would result in the incorrect pressure altitude being flown. In the worst case this may result in inadvertent flight into terrain.
- Whilst other height sources (GNSS and radar altitude) provide additional barriers to prevent impact with terrain, they are not considered robust enough to reduce the indicative DAL:
  - GNSS is not high integrity and may be jammed.
  - Radar altitude is ineffective for preventing flight into terrain in some circumstances, such as near cliffs or steep terrain.
- This analysis is conservative and driven by uncrewed air vehicles - for a manned air vehicle the additional situation awareness of the crew may be sufficient to reduce the DAL.

5.4.2.18.6.4 Security Considerations

The indicative security classification is O, however higher classification instances are expected.

This component is responsible for integration with the operating environment, the details of which will range from O (e.g. civil airspace or other public systems) to SNEO (for military operations areas). Where there are multiple security domains and multiple instances of the component, these may need to communicate with each other; separation will not be provided by this component and boundary protection will be required.

The component is expected to at least partially satisfy security related functions by:

- **Identifying Data Sources**, e.g. an air traffic controller or aircraft. Spoofing of other users in the environment is a particular concern as this could impact the behaviour of the component in the integrations performed.
- **Logging of Security Data** for authentication and authorisation successes and failures for later forensic examination.
- **Maintaining Audit Records** to support non-repudiation of instructions received in the course of operations.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.18.7 Services

5.4.2.18.7.1 Service Definitions

5.4.2.18.7.1.1 Platform_Requirement

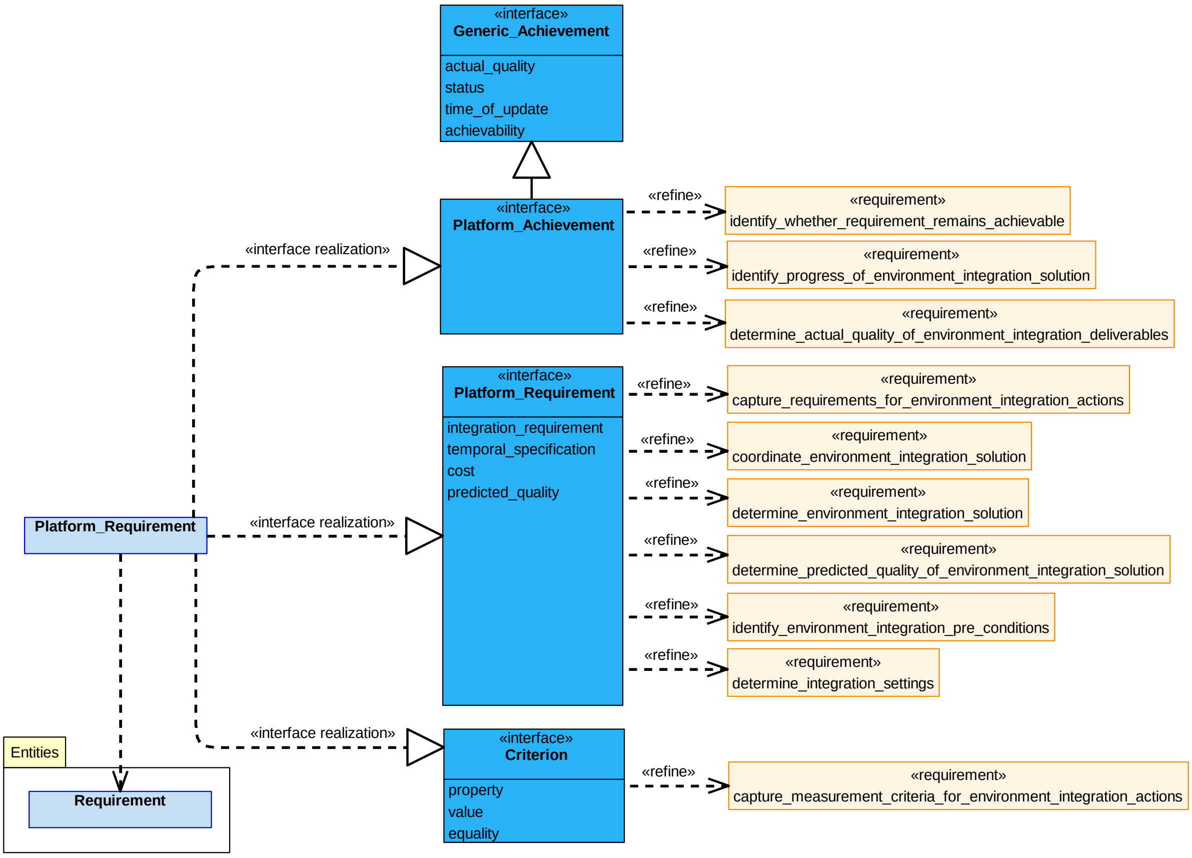


Figure 306: Platform_Requirement Service Definition

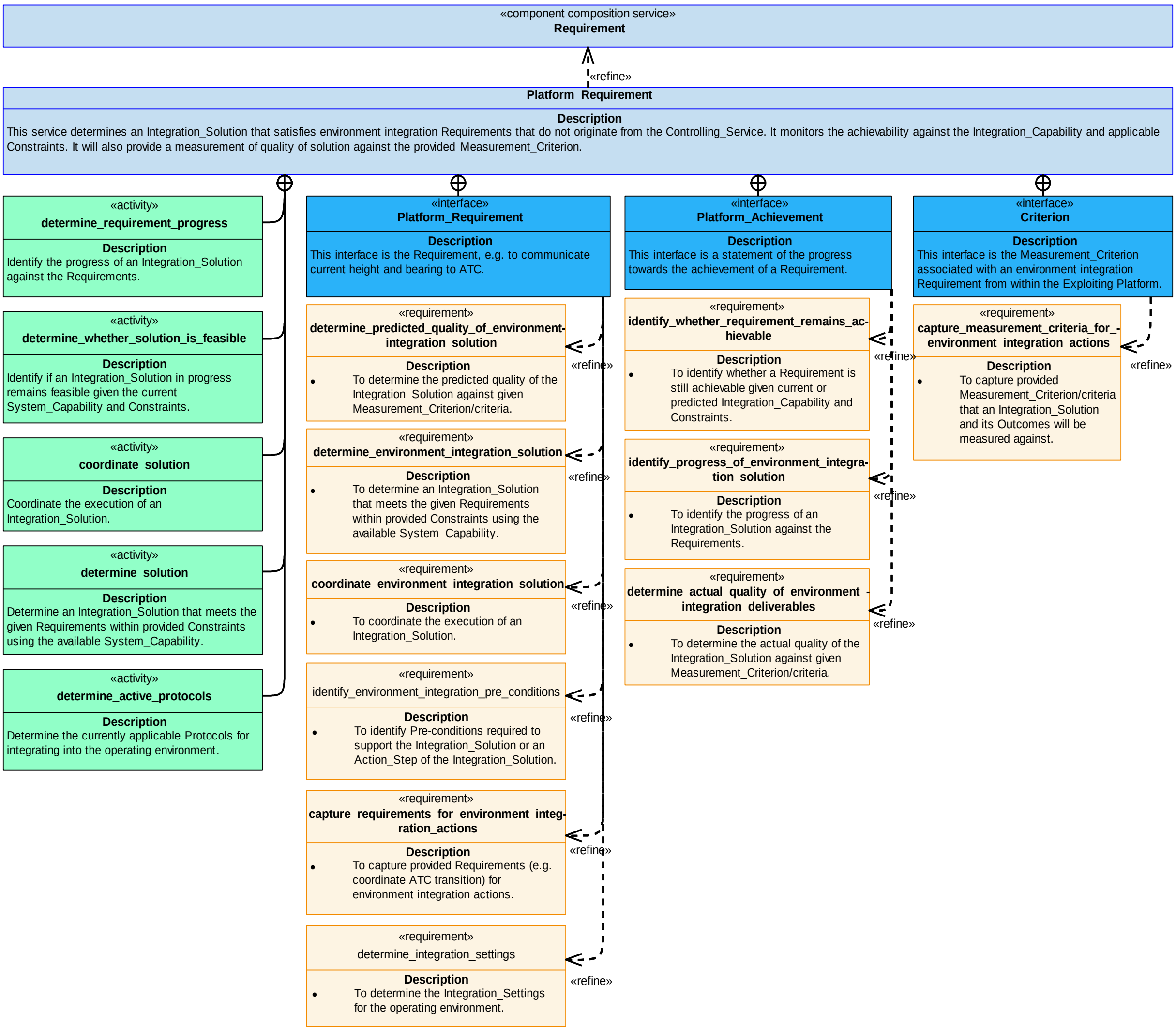

«component composition service»
Requirement
«refine»
Platform_Requirement
Description
This service determines an Integration_Solution that satisfies environment integration Requirements that do not originate from the Controlling_Service. It monitors the achievability against the Integration_Capability and applicable Constraints. It will also provide a measurement of quality of solution against the provided Measurement_Criterion.
«activity»
determine_requirement_progress
Description
Identify the progress of an Integration_Solution against the Requirements.
«activity»
determine_whether_solution_is_feasible
Description
Identify if an Integration_Solution in progress remains feasible given the current System_Capability and Constraints.
«activity»
coordinate_solution
Description
Coordinate the execution of an Integration_Solution.
«activity»
determine_solution
Description
Determine an Integration_Solution that meets the given Requirements within provided Constraints using the available System_Capability.
«activity»
determine_active_protocols
Description
Determine the currently applicable Protocols for integrating into the operating environment.
«interface»
Platform_Requirement
Description
This interface is the Requirement, e.g. to communicate current height and bearing to ATC.
«requirement»
determine_predicted_quality_of_environment_integration_solution
Description
To determine the predicted quality of the Integration_Solution against given Measurement_Criterion/criteria.
«refine»
«requirement»
determine_environment_integration_solution
Description
To determine an Integration_Solution that meets the given Requirements within provided Constraints using the available System_Capability.
«refine»
«requirement»
coordinate_environment_integration_solution
Description
To coordinate the execution of an Integration_Solution.
«refine»
«requirement»
identify_environment_integration_pre_conditions
Description
To identify Pre-conditions required to support the Integration_Solution or an Action_Step of the Integration_Solution.
«refine»
«requirement»
capture_requirements_for_environment_integration_actions
Description
To capture provided Requirements (e.g. coordinate ATC transition) for environment integration actions.
«refine»
«requirement»
determine_integration_settings
Description
To determine the Integration_Settings for the operating environment.
«refine»
«interface»
Platform_Achievement
Description
This interface is a statement of the progress towards the achievement of a Requirement.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Requirement is still achievable given current or predicted Integration_Capability and Constraints.
«refine»
«requirement»
identify_progress_of_environment_integration_solution
Description
To identify the progress of an Integration_Solution against the Requirements.
«refine»
«requirement»
determine_actual_quality_of_environment_integration_deliverables
Description
To determine the actual quality of the Integration_Solution against given Measurement_Criterion/criteria.
«refine»
«interface»
Criterion
Description
This interface is the Measurement_Criterion associated with an environment integration Requirement from within the Exploiting Platform.
«requirement»
capture_measurement_criteria_for_environment_integration_actions
Description
To capture provided Measurement_Criterion/criteria that an Integration_Solution and its Outcomes will be measured against.
«refine»

Figure 307: Platform_Requirement Service Policy

Platform_Requirement

This service determines an Integration_Solution that satisfies environment integration Requirements that do not originate from the Controlling_Service. It monitors the achievability against the Integration_Capability and applicable Constraints. It will also provide a measurement of quality of solution against the provided Measurement_Criterion.

Interfaces

Criterion

This interface is the Measurement_Criterion associated with an environment integration Requirement from within the Exploiting Platform.

Attributes

property	The property to be measured, e.g. current height.
value	The measured value of the property, e.g. 10,000 feet.

equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Platform_Requirement

This interface is the Requirement, e.g. to communicate current height and bearing to ATC.

Attributes

integration_requirement	A requirement specifying how the platform needs to integrate into the environment, e.g. transition to a different flight state such as launch, or to traverse from one waypoint to another.
temporal_specification	Timing information such as start time and duration.
cost	The cost of executing the solution, for example: resources used or time taken.
predicted_quality	How well the proposed integration solution is predicted to satisfy the requirement.

Platform_Achievement

This interface is a statement of the progress towards the achievement of a Requirement.

Activities

coordinate_solution

Coordinate the execution of an Integration_Solution.

determine_whether_solution_is_feasible

Identify if an Integration_Solution in progress remains feasible given the current System_Capability and Constraints.

determine_solution

Determine an Integration_Solution that meets the given Requirements within provided Constraints using the available System_Capability.

determine_requirement_progress

Identify the progress of an Integration_Solution against the Requirements.

determine_active_protocols

Determine the currently applicable Protocols for integrating into the operating environment.

5.4.2.18.7.1.2 Controller_Interaction

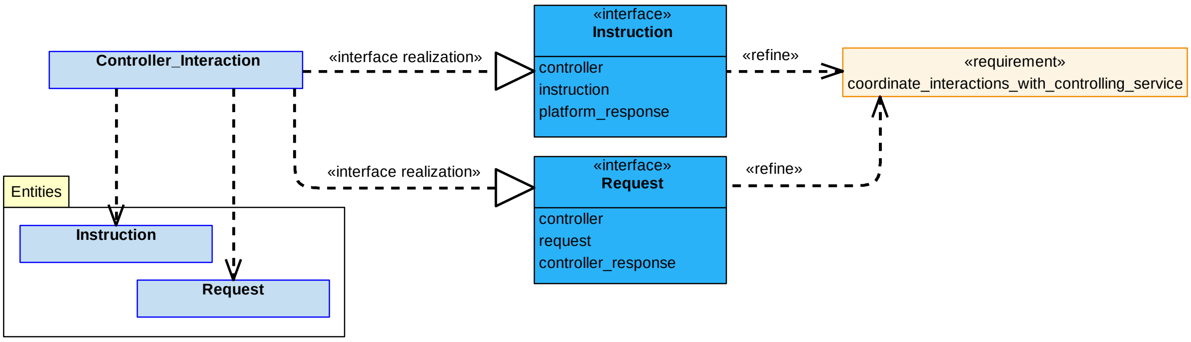


Figure 308: Controller_Interaction Service Definition

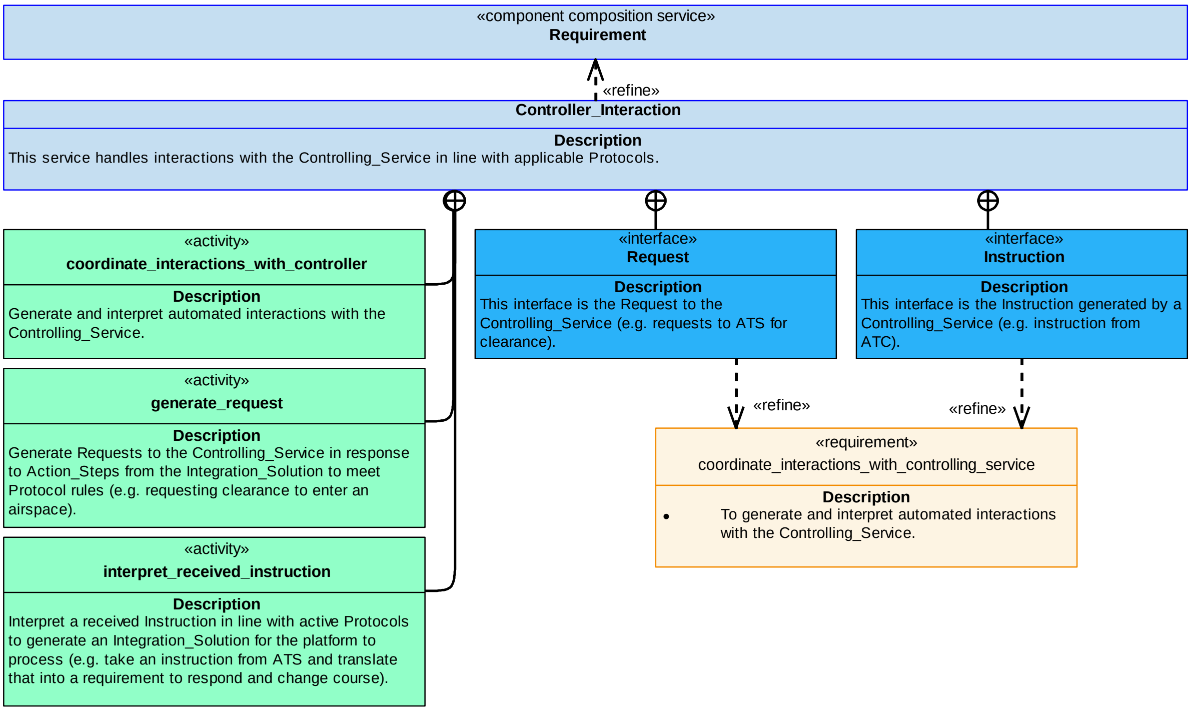

«component composition service»
Requirement
«refine»
Controller_Interaction
Description
This service handles interactions with the Controlling_Service in line with applicable Protocols.
«activity»
coordinate_interactions_with_controller
Description
Generate and interpret automated interactions with the Controlling_Service.
«activity»
generate_request
Description
Generate Requests to the Controlling_Service in response to Action_Steps from the Integration_Solution to meet Protocol rules (e.g. requesting clearance to enter an airspace).
«activity»
interpret_received_instruction
Description
Interpret a received Instruction in line with active Protocols to generate an Integration_Solution for the platform to process (e.g. take an instruction from ATS and translate that into a requirement to respond and change course).
«interface»
Request
Description
This interface is the Request to the Controlling_Service (e.g. requests to ATS for clearance).
«interface»
Instruction
Description
This interface is the Instruction generated by a Controlling_Service (e.g. instruction from ATC).
«refine»
«refine»
«requirement»
coordinate_interactions_with_controlling_service
Description
To generate and interpret automated interactions with the Controlling_Service.

Figure 309: Controller_Interaction Service Policy

Controller_Interaction

This service handles interactions with the Controlling_Service in line with applicable Protocols.

Interfaces

Instruction

This interface is the Instruction generated by a Controlling_Service (e.g. instruction from ATC).

Attributes

controller	Controlling_Service that has provided the instruction (e.g. ATS).
instruction	An Instruction for the platform to follow (e.g. achieve a certain flight level or heading).

platform_response	Response from the platform to the Controlling_Service (e.g. acknowledge, read back instruction, or unable).

Request

This interface is the Request to the Controlling_Service (e.g. requests to ATS for clearance).

Attributes

controller	Controlling_Service to which a request is being directed (e.g. ATS).
request	The Request being made to the Controlling_Service (e.g. permission to transit an area, clearance for take-off, or request for data).
controller_response	The response from the Controlling_Service (e.g. clearance granted or request declined).

Activities

interpret_received_instruction

Interpret a received Instruction in line with active Protocols to generate an Integration_Solution for the platform to process (e.g. take an instruction from ATS and translate that into a requirement to respond and change course).

coordinate_interactions_with_controller

Generate and interpret automated interactions with the Controlling_Service.

generate_request

Generate Requests to the Controlling_Service in response to Action_Steps from the Integration_Solution to meet Protocol rules (e.g. requesting clearance to enter an airspace).

5.4.2.18.7.1.3 Integration_Activity

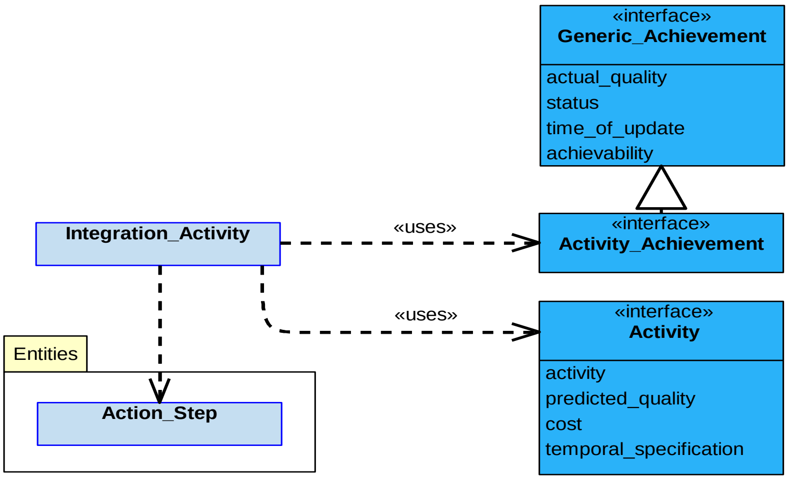


Figure 310: Integration_Activity Service Definition

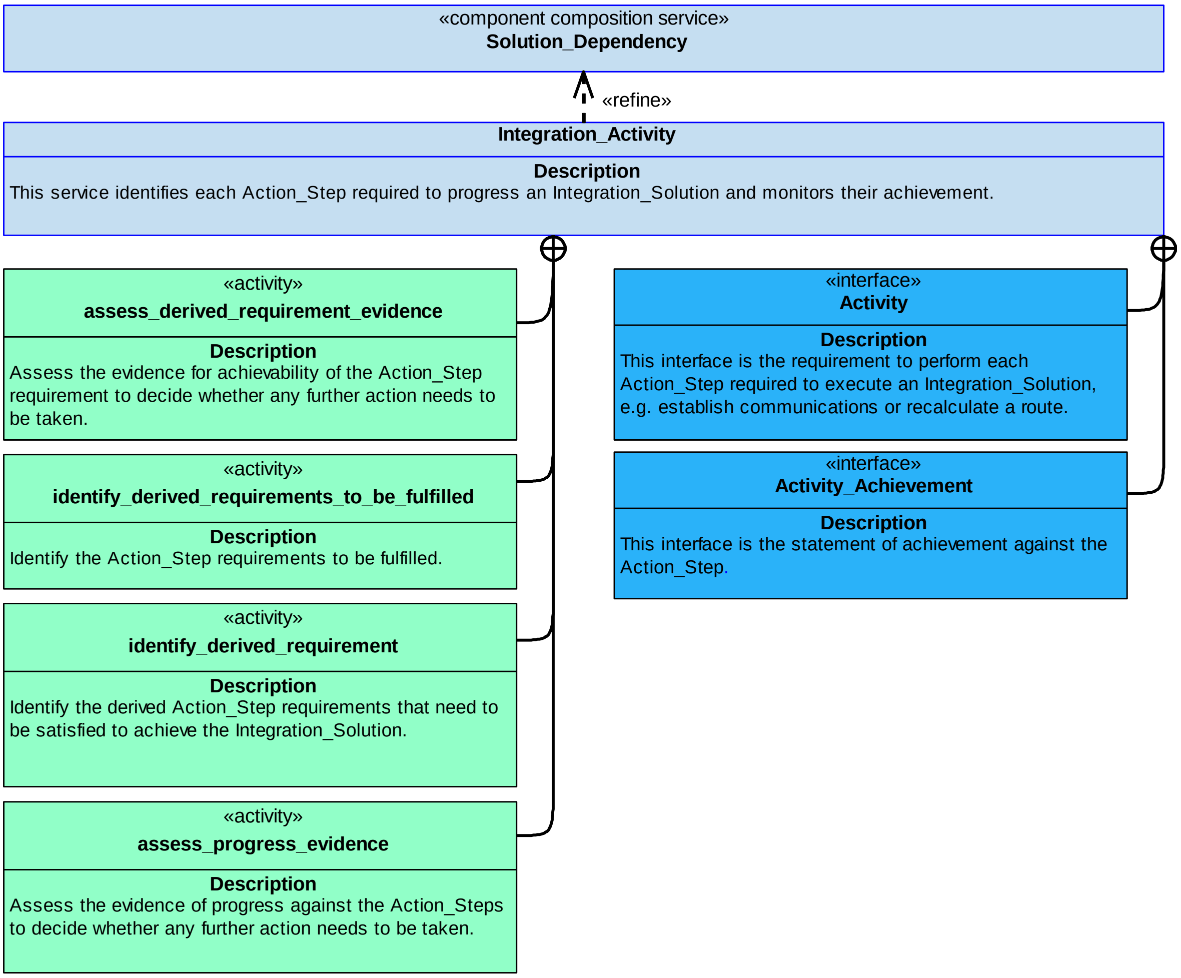

«component composition service»
Solution_Dependency
«refine»
Integration_Activity
Description
This service identifies each Action_Step required to progress an Integration_Solution and monitors their achievement.
«activity»
assess_derived_requirement_evidence
Description
Assess the evidence for achievability of the Action_Step requirement to decide whether any further action needs to be taken.
«activity»
identify_derived_requirements_to_be_fulfilled
Description
Identify the Action_Step requirements to be fulfilled.
«activity»
identify_derived_requirement
Description
Identify the derived Action_Step requirements that need to be satisfied to achieve the Integration_Solution.
«activity»
assess_progress_evidence
Description
Assess the evidence of progress against the Action_Steps to decide whether any further action needs to be taken.
«interface»
Activity
Description
This interface is the requirement to perform each Action_Step required to execute an Integration_Solution, e.g. establish communications or recalculate a route.
«interface»
Activity_Achievement
Description
This interface is the statement of achievement against the Action_Step.

Figure 311: Integration_Activity Service Policy

Integration_Activity

This service identifies each Action_Step required to progress an Integration_Solution and monitors their achievement.

Interfaces

Activity

This interface is the requirement to perform each Action_Step required to execute an Integration_Solution, e.g. establish communications or recalculate a route.

Attributes

activity	An activity to be performed to achieve the integration objective, e.g. determine route to waypoint, change an Integration_Setting, maintain a safe MSD from a terrain feature, or request authorisation from an operator.
predicted_quality	How well the planned solution is predicted to satisfy the requirement.
cost	The cost of executing the solution, for example: resources used or time taken.
temporal_specification	Timing information such as start time and duration.

Activity_Achievement

This interface is the statement of achievement against the Action_Step.

Activities

assess_progress_evidence

Assess the evidence of progress against the Action_Steps to decide whether any further action needs to be taken.

identify_derived_requirement

Identify the derived Action_Step requirements that need to be satisfied to achieve the Integration_Solution.

identify_derived_requirements_to_be_fulfilled

Identify the Action_Step requirements to be fulfilled.

assess_derived_requirement_evidence

Assess the evidence for achievability of the Action_Step requirement to decide whether any further action needs to be taken.

5.4.2.18.7.1.4 Integration_Solution_Information

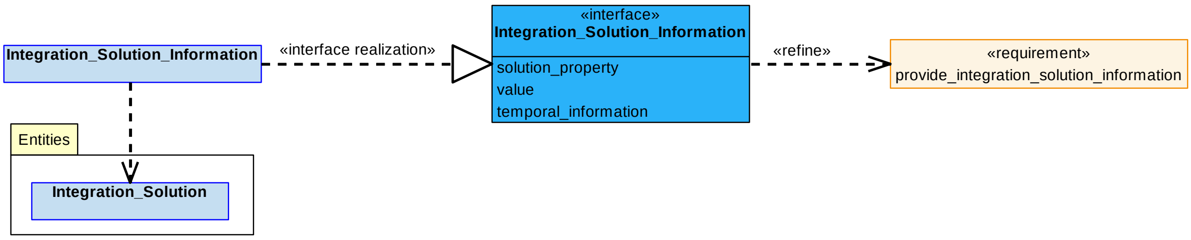


Figure 312: Integration_Solution_Information Service Definition

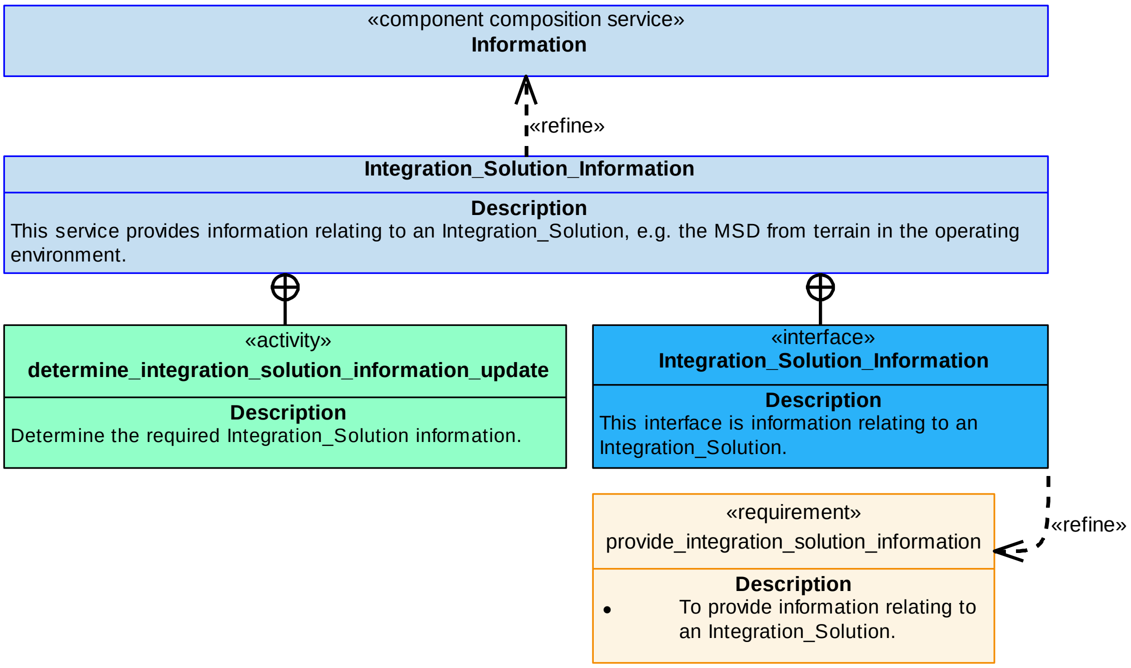

«component composition service»
Information
«refine»
Integration_Solution_Information
Description
This service provides information relating to an Integration_Solution, e.g. the MSD from terrain in the operating environment.
«activity»
determine_integration_solution_information_update
Description
Determine the required Integration_Solution information.
«interface»
Integration_Solution_Information
Description
This interface is information relating to an Integration_Solution.
«requirement»
provide_integration_solution_information
Description
To provide information relating to an Integration_Solution.
«refine»

Figure 313: Integration_Solution_Information Service Policy

Integration_Solution_Information

This service provides information relating to an Integration_Solution, e.g. the MSD from terrain in the operating environment.

Interface

Integration_Solution_Information

This interface is information relating to an Integration_Solution.

Attributes

solution_property	The information property to be provided.
value	The value of the solution_property.
temporal_information	Information covering timing, such as start and end times.

Activity

determine_integration_solution_information_update

Determine the required Integration_Solution information.

5.4.2.18.7.1.5 Asset_Information

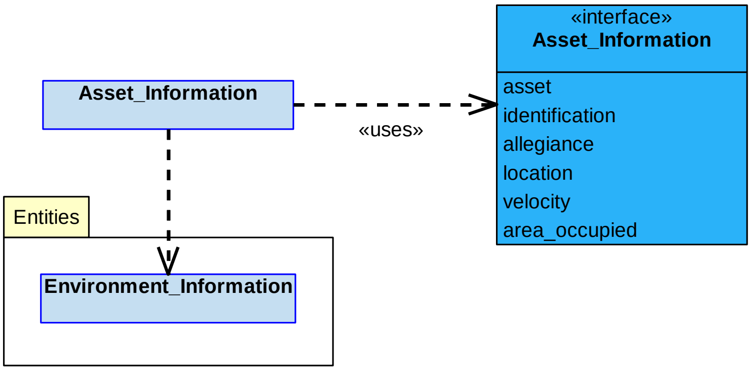


Figure 314: Asset_Information Service Definition

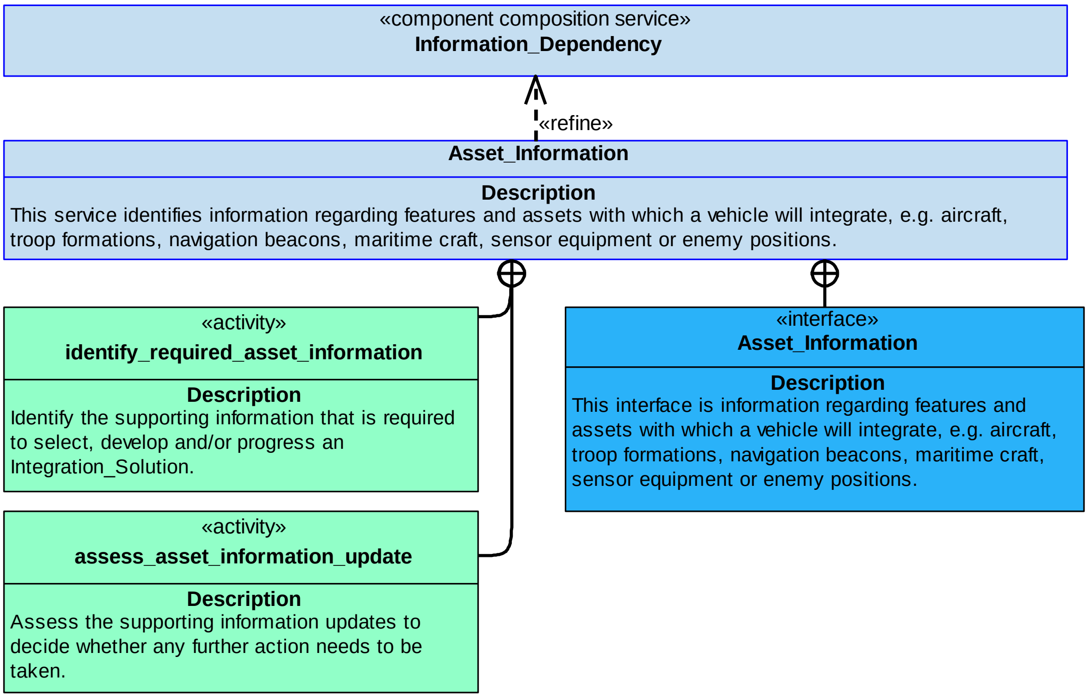

«component composition service»
Information_Dependency
«refine»
Asset_Information
Description
This service identifies information regarding features and assets with which a vehicle will integrate, e.g. aircraft, troop formations, navigation beacons, maritime craft, sensor equipment or enemy positions.
«activity»
identify_required_asset_information
Description
Identify the supporting information that is required to select, develop and/or progress an Integration_Solution.
«activity»
assess_asset_information_update
Description
Assess the supporting information updates to decide whether any further action needs to be taken.
«interface»
Asset_Information
Description
This interface is information regarding features and assets with which a vehicle will integrate, e.g. aircraft, troop formations, navigation beacons, maritime craft, sensor equipment or enemy positions.

Figure 315: Asset_Information Service Policy

Asset_Information

This service identifies information regarding features and assets with which a vehicle will integrate, e.g. aircraft, troop formations, navigation beacons, maritime craft, sensor equipment or enemy positions.

Interface

Asset_Information

This interface is information regarding features and assets with which a vehicle will integrate, e.g. aircraft, troop formations, navigation beacons, maritime craft, sensor equipment or enemy positions.

Attributes

asset	Asset type, e.g. fuel tanker, enemy jet, aircraft carrier, navigation beacon or formation of soldiers.
identification	Data to identify an asset relevant to the environment integration, e.g. tail identifier.
allegiance	Allegiance of an asset, e.g. Friend, Foe, Neutral, or Unknown.
location	Data to identify an asset position, e.g. the position of a vehicle or troop formation.
velocity	Data to identify an asset speed and direction, e.g. if the asset is stationary or in motion.
area_occupied	The area occupied by an asset, e.g. the area occupied by a ground force.

Activities

assess_asset_information_update

Assess the supporting information updates to decide whether any further action needs to be taken.

identify_required_asset_information

Identify the supporting information that is required to select, develop and/or progress an Integration_Solution.

5.4.2.18.7.1.6 Environment_Information

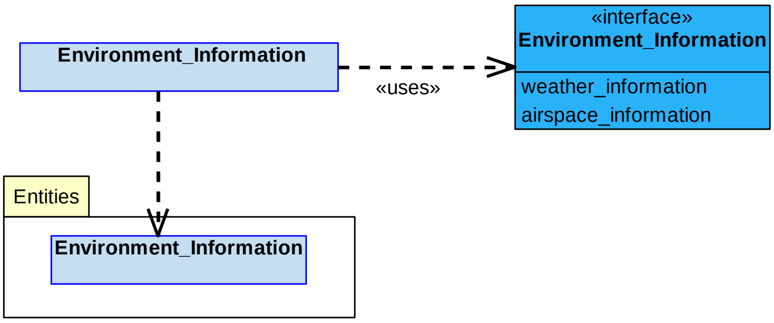


Figure 316: Environment_Information Service Definition

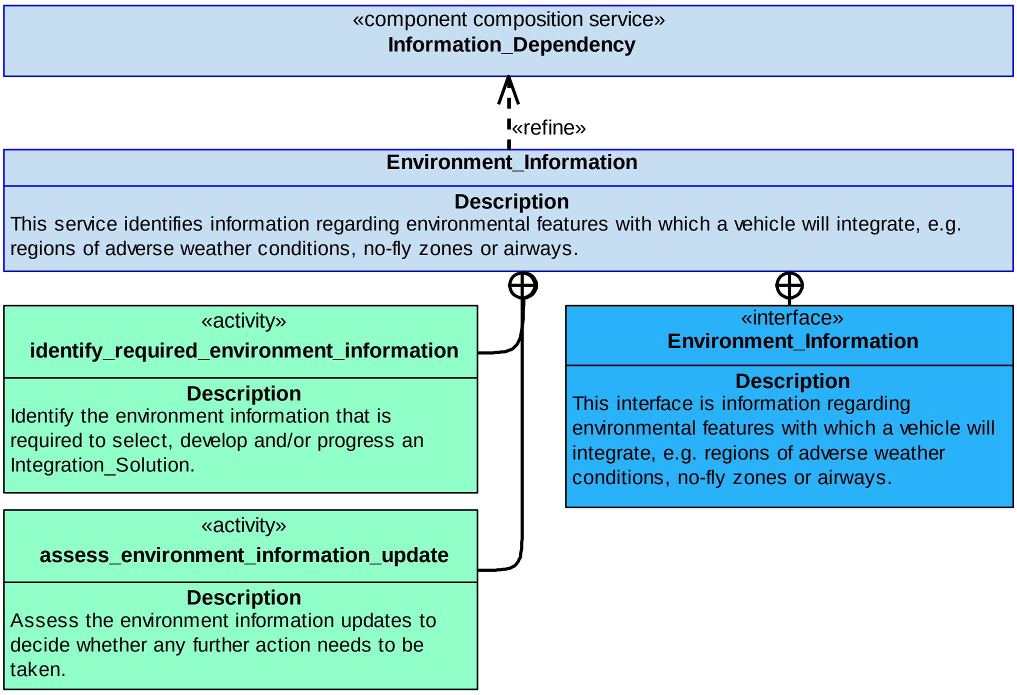

«component composition service»
Information_Dependency
«refine»
Environment_Information
Description
This service identifies information regarding environmental features with which a vehicle will integrate, e.g. regions of adverse weather conditions, no-fly zones or airways.
«activity»
identify_required_environment_information
Description
Identify the environment information that is required to select, develop and/or progress an Integration_Solution.
«activity»
assess_environment_information_update
Description
Assess the environment information updates to decide whether any further action needs to be taken.
«interface»
Environment_Information
Description
This interface is information regarding environmental features with which a vehicle will integrate, e.g. regions of adverse weather conditions, no-fly zones or airways.

Figure 317: Environment_Information Service Policy

Environment_Information

This service identifies information regarding environmental features with which a vehicle will integrate, e.g. regions of adverse weather conditions, no-fly zones or airways.

<u>Interface</u>

Environment_Information

This interface is information regarding environmental features with which a vehicle will integrate, e.g. regions of adverse weather conditions, no-fly zones or airways.

<u>Attributes</u>

weather_information	Weather information that the vehicle must account for in its solution, e.g. adverse weather conditions.
airspace_information	Relevant airspace information the vehicle will have to integrate with, e.g. regions that need to be avoided.

<u>Activities</u>

assess_environment_information_update

Assess the environment information updates to decide whether any further action needs to be taken.

identify_required_environment_information

Identify the environment information that is required to select, develop and/or progress an Integration_Solution.

5.4.2.18.7.1.7 Vehicle_Information

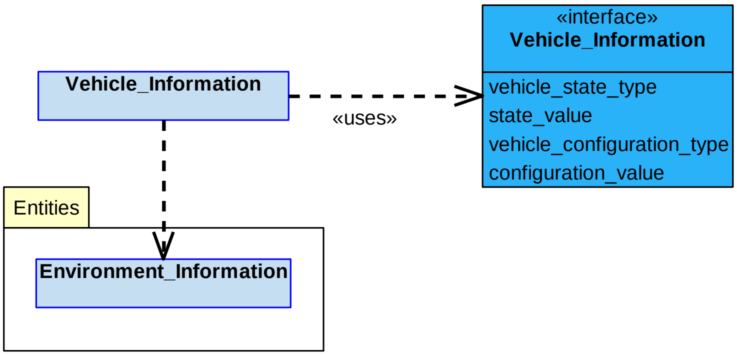


Figure 318: Vehicle_Information Service Definition

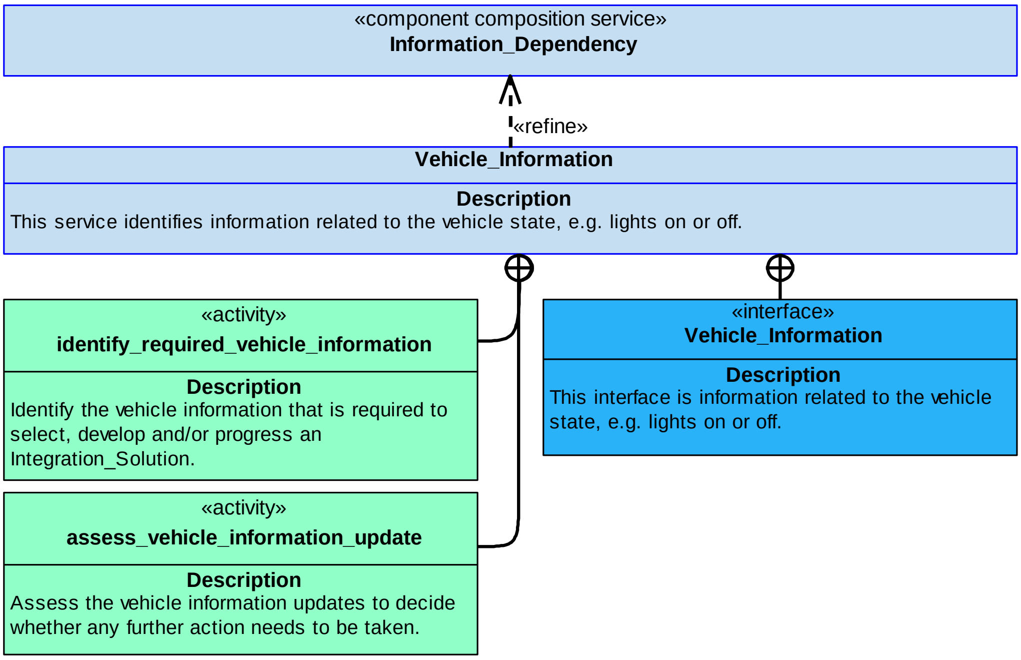

«component composition service»
Information_Dependency
«refine»
Vehicle_Information
Description
This service identifies information related to the vehicle state, e.g. lights on or off.
«activity»
identify_required_vehicle_information
Description
Identify the vehicle information that is required to select, develop and/or progress an Integration_Solution.
«interface»
Vehicle_Information
Description
This interface is information related to the vehicle state, e.g. lights on or off.
«activity»
assess_vehicle_information_update
Description
Assess the vehicle information updates to decide whether any further action needs to be taken.

Figure 319: Vehicle_Information Service Policy

Vehicle_Information

This service identifies information related to the vehicle state, e.g. lights on or off.

Interface

Vehicle_Information

This interface is information related to the vehicle state, e.g. lights on or off.

Attributes

vehicle_state_type	The type of information relating to the vehicle state, such as altitude, airspeed, pitch or roll.
state_value	The value of the vehicle_state_type.

vehicle_configuration_type	The type of information relating to the vehicle configuration, such as door or landing wheel configuration status.
configuration_value	The value of the vehicle_configuration_type.

Activities

identify_required_vehicle_information

Identify the vehicle information that is required to select, develop and/or progress an Integration_Solution.

assess_vehicle_information_update

Assess the vehicle information updates to decide whether any further action needs to be taken.

5.4.2.18.7.1.8 Constraint

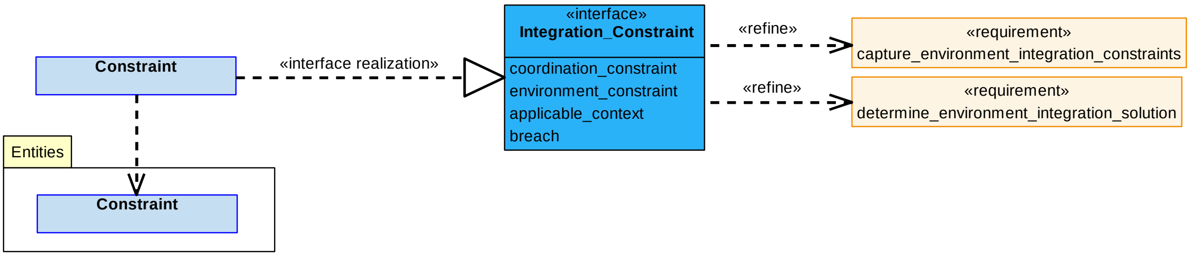


Figure 320: Constraint Service Definition

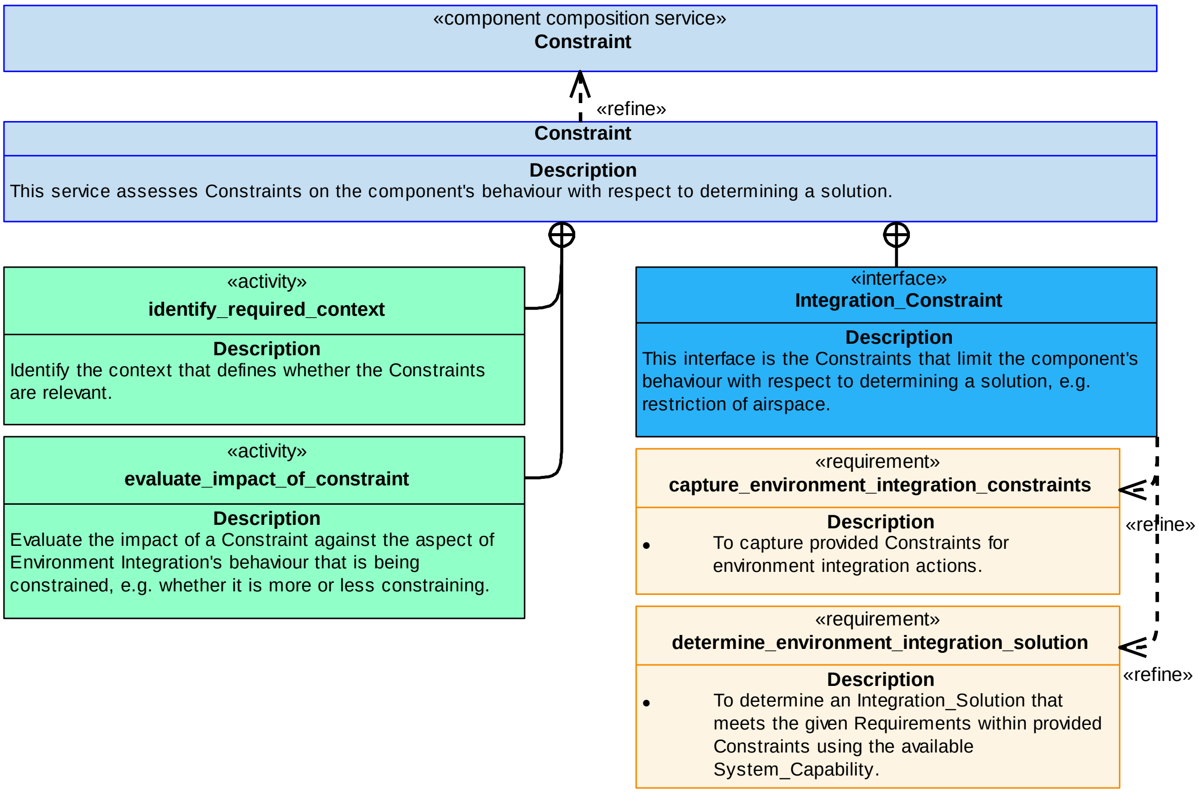

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints on the component's behaviour with respect to determining a solution.
«activity»
identify_required_context
Description
Identify the context that defines whether the Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of a Constraint against the aspect of Environment Integration's behaviour that is being constrained, e.g. whether it is more or less constraining.
«interface»
Integration_Constraint
Description
This interface is the Constraints that limit the component's behaviour with respect to determining a solution, e.g. restriction of airspace.
«requirement»
capture_environment_integration_constraints
Description
• To capture provided Constraints for environment integration actions.
«refine»
«requirement»
determine_environment_integration_solution
Description
• To determine an Integration_Solution that meets the given Requirements within provided Constraints using the available System_Capability.
«refine»

Figure 321: Constraint Service Policy

Constraint

This service assesses Constraints on the component's behaviour with respect to determining a solution.

Interface

Integration_Constraint

This interface is the Constraints that limit the component's behaviour with respect to determining a solution, e.g. restriction of airspace.

Attributes

coordination_constraint	A Constraint that will restrict the ability of the component to coordinate with other platforms, e.g. if it is necessary to maintain radio silence, the system will not be able to communicate with ATC and other vehicles.
environment_constraint	An externally imposed Constraint that the operating platform must comply with, e.g. avoid specified airspace.
applicable_context	The context in which the Constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of a Constraint against the aspect of Environment Integration's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context that defines whether the Constraints are relevant.

5.4.2.18.7.1.9 Capability

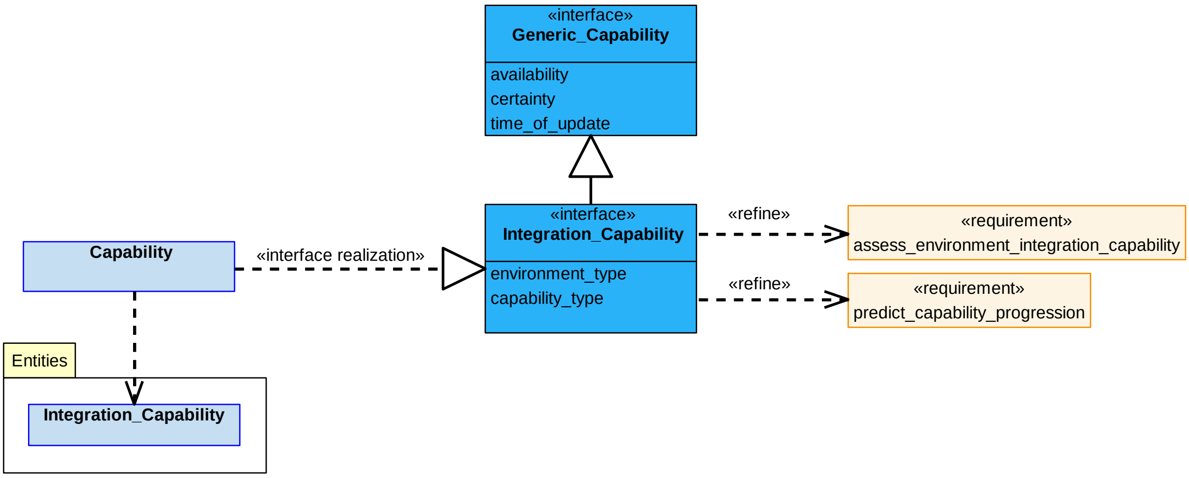


Figure 322: Capability Service Definition

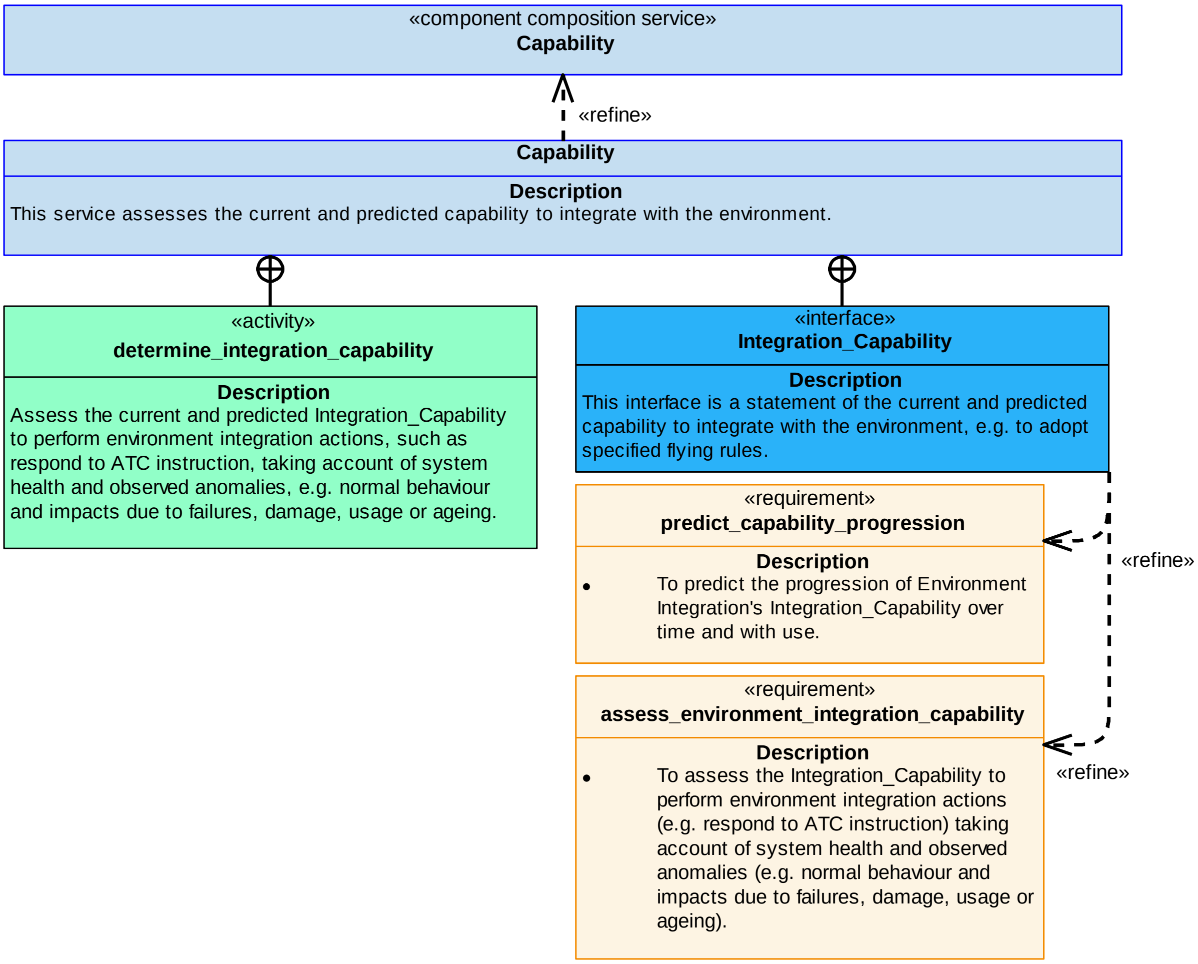

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability to integrate with the environment.
«activity»
determine_integration_capability
Description
Assess the current and predicted Integration_Capability to perform environment integration actions, such as respond to ATC instruction, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.
«interface»
Integration_Capability
Description
This interface is a statement of the current and predicted capability to integrate with the environment, e.g. to adopt specified flying rules.
«requirement»
predict_capability_progression
Description
To predict the progression of Environment Integration's Integration_Capability over time and with use.
«refine»
«requirement»
assess_environment_integration_capability
Description
To assess the Integration_Capability to perform environment integration actions (e.g. respond to ATC instruction) taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 323: Capability Service Policy

Capability

This service assesses the current and predicted capability to integrate with the environment.

Interface

Integration_Capability

This interface is a statement of the current and predicted capability to integrate with the environment, e.g. to adopt specified flying rules.

Attributes

environment_type	The type of environment that the component is capable of integrating with, e.g. civil or military airspace.
capability_type	The type of integration capability, e.g. traverse airspace.

Activity

determine_integration_capability

Assess the current and predicted Integration_Capability to perform environment integration actions, such as respond to ATC instruction, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.

5.4.2.18.7.1.10 Capability_Evidence

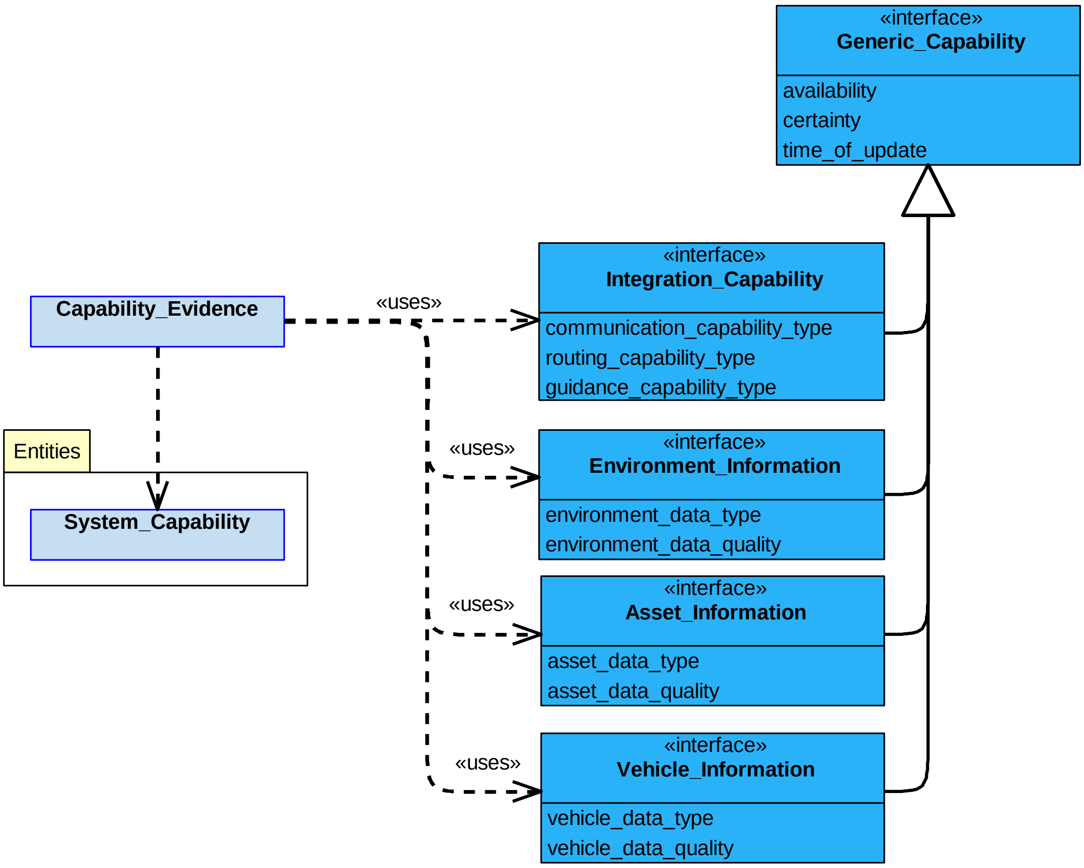


Figure 324: Capability_Evidence Service Definition

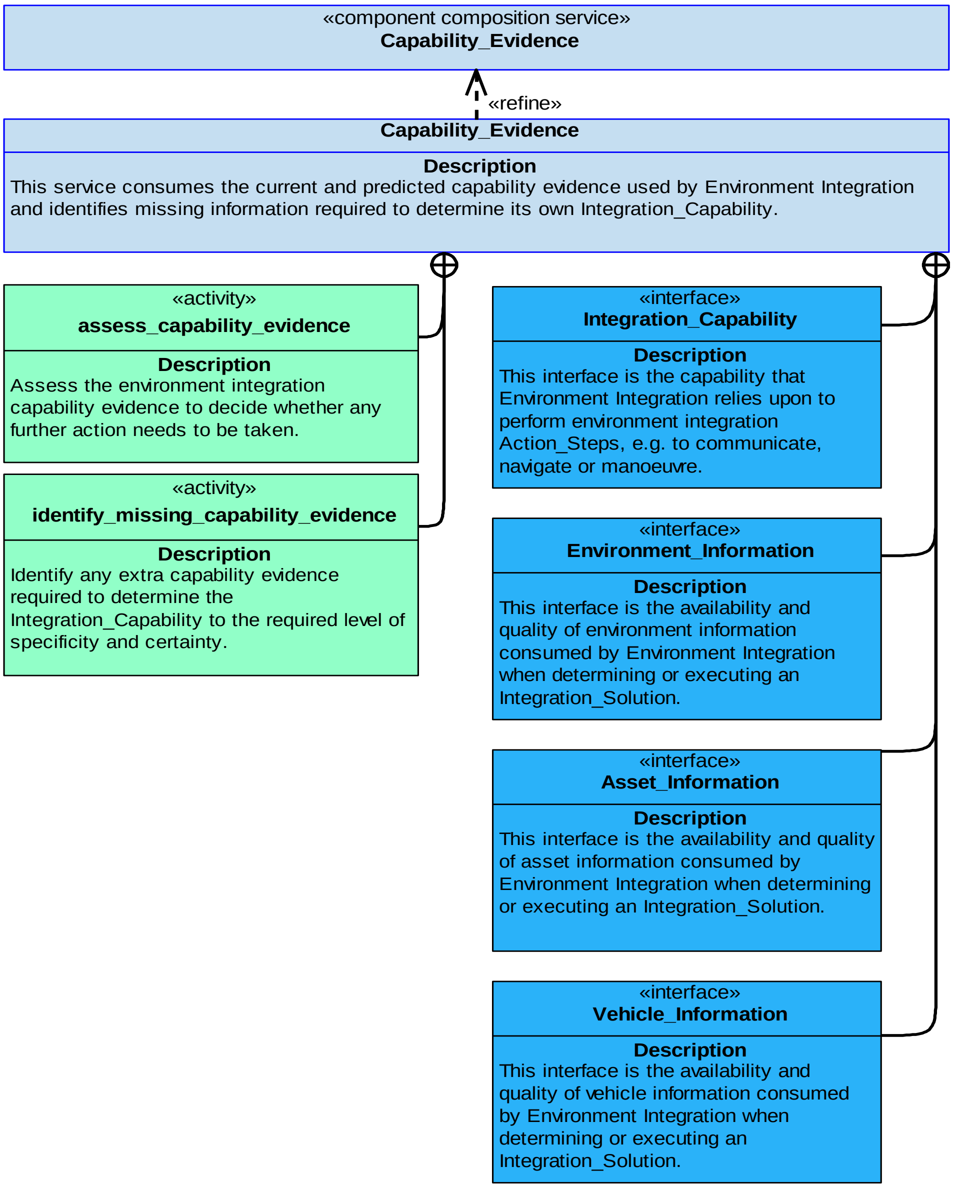

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted capability evidence used by Environment Integration and identifies missing information required to determine its own Integration_Capability.
«activity»
assess_capability_evidence
Description
Assess the environment integration capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Integration_Capability to the required level of specificity and certainty.
«interface»
Integration_Capability
Description
This interface is the capability that Environment Integration relies upon to perform environment integration Action_Steps, e.g. to communicate, navigate or manoeuvre.
«interface»
Environment_Information
Description
This interface is the availability and quality of environment information consumed by Environment Integration when determining or executing an Integration_Solution.
«interface»
Asset_Information
Description
This interface is the availability and quality of asset information consumed by Environment Integration when determining or executing an Integration_Solution.
«interface»
Vehicle_Information
Description
This interface is the availability and quality of vehicle information consumed by Environment Integration when determining or executing an Integration_Solution.

Figure 325: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capability evidence used by Environment Integration and identifies missing information required to determine its own Integration_Capability.

Interfaces

Integration_Capability

This interface is the capability that Environment Integration relies upon to perform environment integration Action_Steps, e.g. to communicate, navigate or manoeuvre.

Attributes

communication_capability_type	The communications capability to which this capability evidence applies (e.g. a communications link with the Controlling_Service).

routing_capability_type	The routing capability to which this capability evidence applies (e.g. the ability to determine the route to a specified location).
guidance_capability_type	The guidance capability to which this capability evidence applies (e.g. the ability to control the vehicle speed).

Environment_Information

This interface is the availability and quality of environment information consumed by Environment Integration when determining or executing an Integration_Solution.

Attributes

environment_data_type	The type of data being reported on, e.g. airport information.
environment_data_quality	Indication of the quality of data that can be provided.

Asset_Information

This interface is the availability and quality of asset information consumed by Environment Integration when determining or executing an Integration_Solution.

Attributes

asset_data_type	The type of data being reported on, e.g. aircraft positions.
asset_data_quality	Indication of the quality of data that can be provided.

Vehicle_Information

This interface is the availability and quality of vehicle information consumed by Environment Integration when determining or executing an Integration_Solution.

Attributes

vehicle_data_type	The type of data being reported on, e.g. internal vehicle state.
vehicle_data_quality	Indication of the quality of data that can be provided.

Activities

assess_capability_evidence

Assess the environment integration capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Integration_Capability to the required level of specificity and certainty.

5.4.2.18.7.2 Service Dependencies

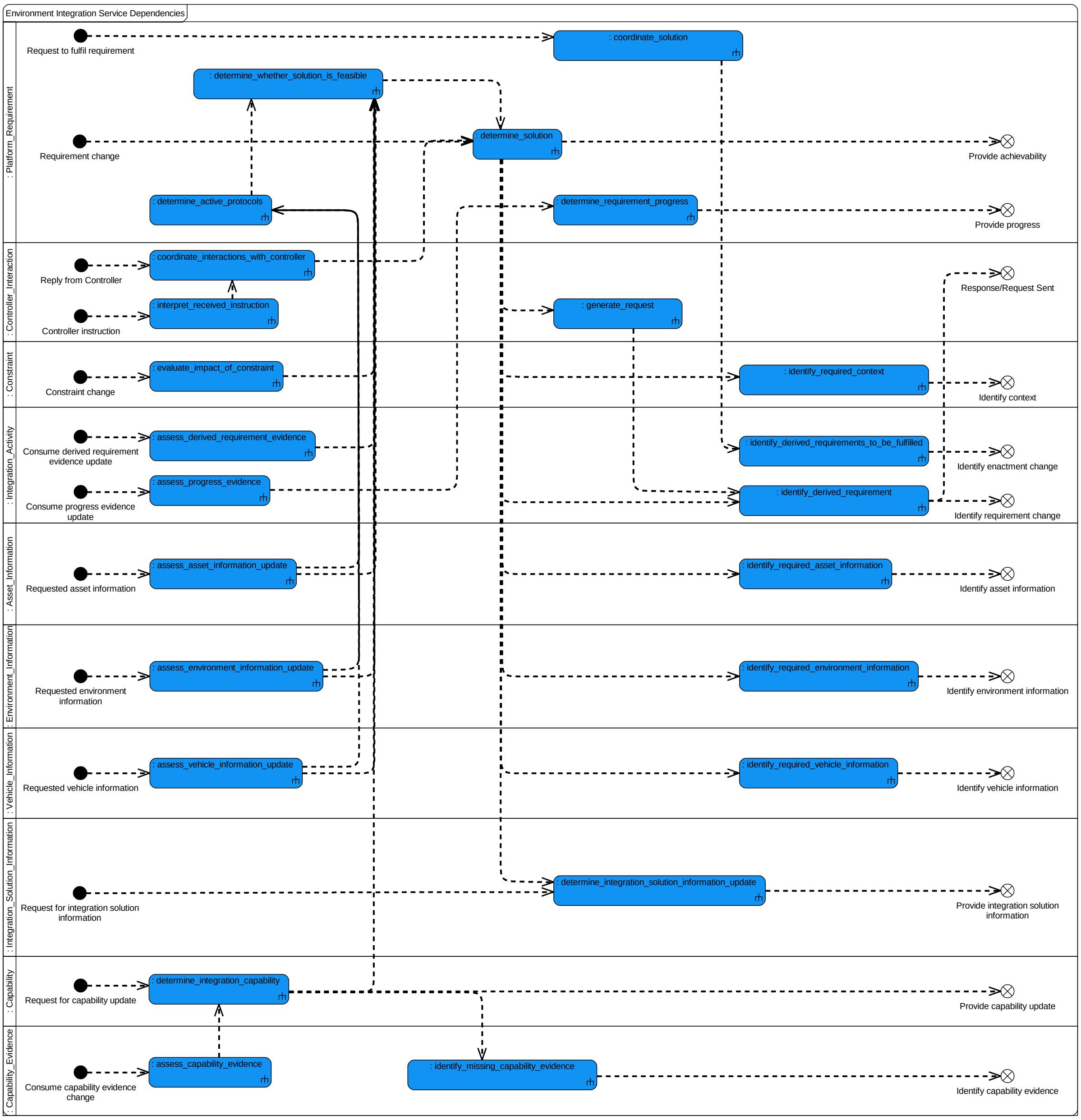

Environment Integration Service Dependencies
: Platform_Requirement
: Controller_Interaction
: Constraint
: Integration_Activity
: Asset_Information
: Environment_Information
: Vehicle_Information
: Integration_Solution_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
: coordinate_solution
: determine_whether_solution_is_feasible
Requirement change
determine_solution
Provide achievability
: determine_active_protocols
: determine_requirement_progress
Provide progress
: coordinate_interactions_with_controller
Reply from Controller
Response/Request Sent
: interpret_received_instruction
: generate_request
Controller instruction
: evaluate_impact_of_constraint
Constraint change
: identify_required_context
Identify context
: assess_derived_requirement_evidence
Consume derived requirement evidence update
: identify_derived_requirements_to_be_fulfilled
Identify enactment change
: assess_progress_evidence
Consume progress evidence update
: identify_derived_requirement
Identify requirement change
: assess_asset_information_update
Requested asset information
: identify_required_asset_information
Identify asset information
: assess_environment_information_update
Requested environment information
: identify_required_environment_information
Identify environment information
: assess_vehicle_information_update
Requested vehicle information
: identify_required_vehicle_information
Identify vehicle information
Request for integration solution information
: determine_integration_solution_information_update
Provide integration solution information
: determine_integration_capability
Request for capability update
Provide capability update
: assess_capability_evidence
Consume capability evidence change
: identify_missing_capability_evidence
Identify capability evidence

Figure 326: Environment Integration Service Dependencies

5.4.2.19 Environmental Conditioning

5.4.2.19.1 Role

The role of Environmental Conditioning is to control the environmental properties of environmental zones.

5.4.2.19.2 Overview

Control Architecture

Environmental Conditioning is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Environmental Conditioning captures Zone_Requirements to control an Environmental_Property (e.g. temperature, pressure or humidity) or properties of one or more Environmental_Zones. Based on Measurements of the controlled Environmental_Property (or properties) and/or Measurements of the Conditioning_Mechanism used to control those properties (e.g. pressure of engine bleed air), Environmental Conditioning controls Conditioning_Mechanisms (e.g. by the control of an effecting medium) in order to satisfy the Zone_Requirements for each Environmental_Zone.

Examples of Use

Environmental Conditioning will be used for:

- UAV control system station air conditioning to maintain air temperature and humidity.
- Cooling of electronic equipment in a vehicle bay.
- Anti-icing of aircraft aerofoils.

5.4.2.19.3 Service Summary

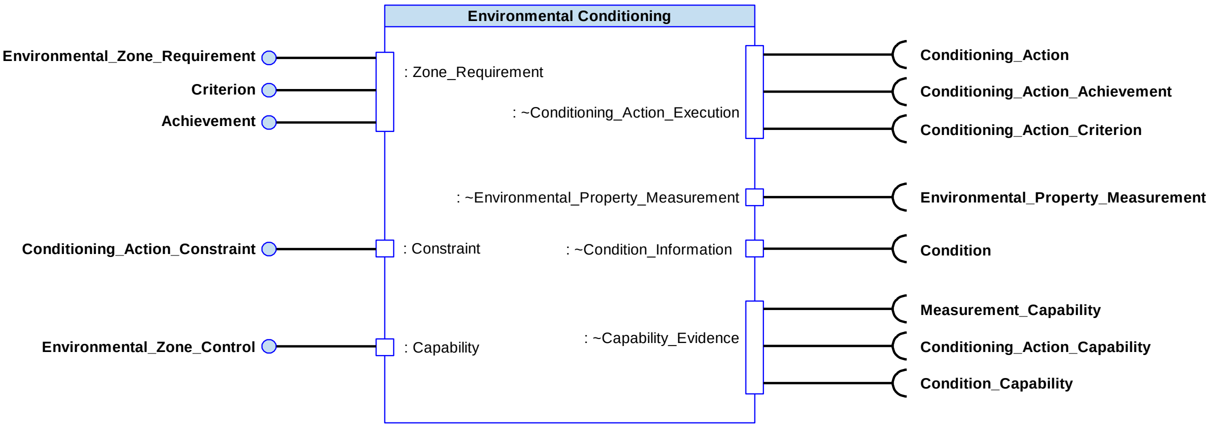

Environmental Conditioning
Environmental_Zone_Requirement
Criterion
Achievement
Conditioning_Action_Constraint
Environmental_Zone_Control
: Zone_Requirement
: ~Conditioning_Action_Execution
: ~Environmental_Property_Measurement
: Constraint
: ~Condition_Information
: Capability
: ~Capability_Evidence
Conditioning_Action
Conditioning_Action_Achievement
Conditioning_Action_Criterion
Environmental_Property_Measurement
Condition
Measurement_Capability
Conditioning_Action_Capability
Condition_Capability

Figure 327: Environmental Conditioning Service Summary

5.4.2.19.4 Responsibilities

capture_zone_requirements

- To capture given Zone_Requirements for an Environmental_Zone (e.g. a required temperature range).

capture_measurement_criteria

- To capture provided Measurement_Criterion for each Zone_Requirement (e.g. response time).

capture_constraints

- To capture provided Constraints for Conditioning_Mechanisms (e.g. heating of an Environmental_Zone is not permitted whilst a door is open).

determine_solution

- To determine a Conditioning_Procedure that meets Zone_Requirements and Constraints using a Conditioning_Mechanism.

identify_solution_in_progress_remains_feasible

- To identify whether a Conditioning_Procedure in progress remains feasible given current resources.

coordinate_solution

- To coordinate the execution of a Conditioning_Procedure via the use of Conditioning_Mechanisms.

identify_progress_of_solution

- To identify the progress of a Conditioning_Procedure against the Zone_Requirements.

determine_quality_of_solution

- To determine the quality of a proposed Conditioning_Procedure against given Measurement_Criterion or criteria.

determine_quality_of_deliverables

- To determine the quality of the Environmental_Property or properties controlled by executing a Conditioning_Procedure, measured against given Zone_Requirements and Measurement_Criterion or criteria.

assess_environmental_conditioning_capability

- To assess capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the capability assessment.

predict_capability_progression

- To predict the progression of capability over time and with use.

5.4.2.19.5 Subject Matter Semantics

The subject matter of Environmental Conditioning is the Environmental_Property (or properties) of Environmental_Zones and resources that can be used to control them.

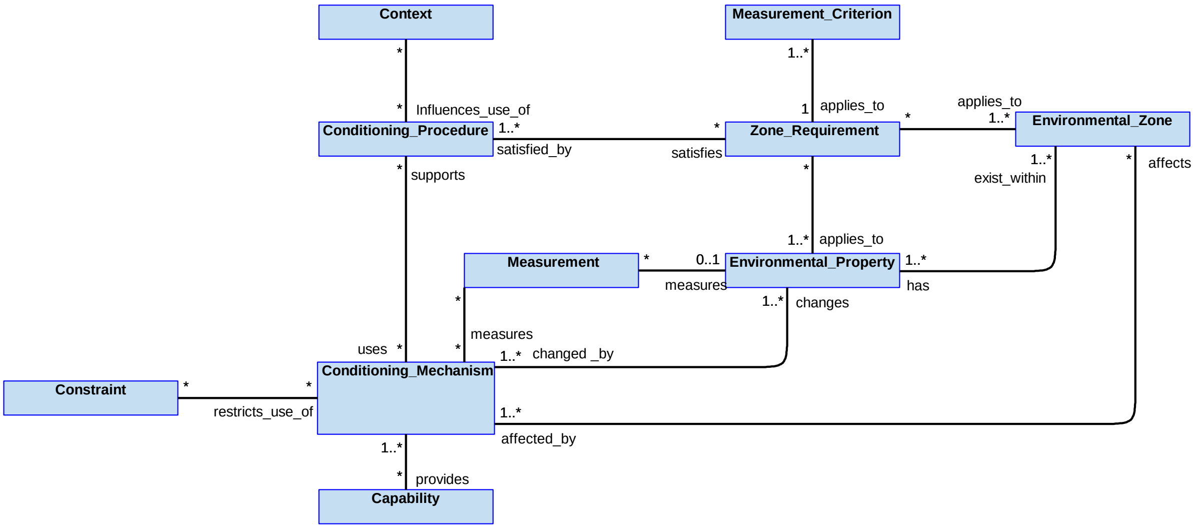


Figure 328: Environmental Conditioning Semantics

5.4.2.19.5.1 Entities

Capability

The ability to effect a change in environmental condition.

Conditioning_Mechanism

The medium, process, or means of directly or indirectly effecting an environment's properties, e.g. engine bleed air being used to provide heating or cooling, direct heating, pressure release, or use of a chemical coating. Note the conditioning mechanism may be outside the control of the component (or system), such as an external source of heat impacting an Environmental_Property being managed.

Conditioning_Procedure

An action, or set of actions that can be performed to cause a desired change in an environmental condition or property. This includes the use of direct or indirect environmental effects as provided by available Conditioning_Mechanisms, e.g. the controlled application of both heating and cooling to reduce humidity (while maintaining a constant temperature and pressure) or the use of chemical processes to affect the hydrophobic properties of a surface.

Constraint

A restriction on the application of the available capability, e.g. restricting the use of a Conditioning_Mechanism due to safety interlocks.

Context

Information relating to current operating conditions or vehicle configuration, which affects how environmental conditioning can be achieved.

Environmental_Property

A property of the physical environment that can be effected (e.g. density, acidity, drag coefficient, temperature or pressure).

Environmental_Zone

An area, volume or region whose Environmental_Property (or properties) can be affected, e.g. a cargo bay, UAV control system cabin or aerofoil surface.

Measurement

A measured value of a property of an environment or conditioning mechanism, including data on the measurement quality and capability to provide measurement information.

Measurement_Criterion

A criterion used to evaluate the effectiveness and achievement of a requirement, e.g. timeliness or quality.

Zone_Requirement

A requirement to control environmental aspects of an Environmental_Zone. For example maintaining the Environmental_Zone at a desired temperature, pressure and humidity.

5.4.2.19.6 Design Rationale

5.4.2.19.6.1 Assumptions

- Environmental Conditioning can be used to control any type of Environmental_Property, not just temperature.

5.4.2.19.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Environmental Conditioning:

- Data Driving - the component can be data-driven to cater for different types of Environmental_Property and Environmental_Zone.
- Resource Management - since Conditioning_Mechanisms are likely to require a significant proportion of available system managed resources, such as electrical power, and the Conditioning_Mechanism used may have other functions (e.g. engine bleed air).

Extensions

- Extension components could be used to provide different environmental conditioning solutions for an Environmental_Zone, though in many cases the environmental conditioning solutions are likely to be bespoke to an Exploiting Platform.

Exploitation Considerations

- An exploitation may include multiple instances of Environmental Conditioning with differing safety integrity levels (e.g. Environmental Conditioning as part of aircrew life support as opposed to Environmental Conditioning of weapons bays or to maintain sensor performance).
- Whilst Environmental Conditioning is defined as an action component, it is unlikely that it will receive requirements directly from Tasks - other action components will provide requirements for environmental conditioning based on the physical resources they require to perform their actions.

5.4.2.19.6.3 Safety Considerations

The indicative IDAL is DAL A*.*

The rationale behind this is:

- For environmental conditioning actions relating to the control of an air vehicle, failure of this component could cause equipment that is critical to the controlled flight of the air vehicle (e.g. flight control system computing hardware) to be outside the qualified operating environment. As the equipment can no longer be relied upon to operate, this could cause uncontrolled flight of the air vehicle and subsequently an uncontrolled crash. This would result in loss of the air vehicle and potentially fatalities.

5.4.2.19.6.4 Security Considerations

The indicative security classification is O-S.

This component is responsible for the control of heating and cooling, etc. of parts of the Exploiting Platform that require specific conditioning. In general, this would drive an indicative security classification of O-S. Where the conditioning requirements may indicate use of specific equipment, there may need to be greater confidentiality assigned. If the integrity of demands for, and availability of conditioning is compromised, the combat effectiveness of the Exploiting Platform may be reduced, e.g. through loss of computing resources due to overheating. The component is considered a legitimate target for cyber attack and due to the risk to integrity and availability, appropriate protection is required. This is one of a series of components that will assist in identifying if form and fit integrity has been interfered with.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Information** relating to possible tamper events.
- **Maintaining Audit Records** to support accountability of conditioning applied.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

- Performing **System Status and Monitoring** of the demanded conditioning against that supplied, detecting unexpected conditions including hot or cold spots, etc.
- Providing **Warnings and Notifications** of overheating, etc.

The component is involved in satisfying security enforcing functions relating to:

- **Detecting Security Breaches** through identifying conditioning states that may indicate the physical security has been compromised (e.g. an unauthorised item is attached to the Exploiting Platform that is generating heat or an authorised item is generating excessive heat as a result of being tampered with).

5.4.2.19.7 Services

5.4.2.19.7.1 Service Definitions

5.4.2.19.7.1.1 Zone_Requirement

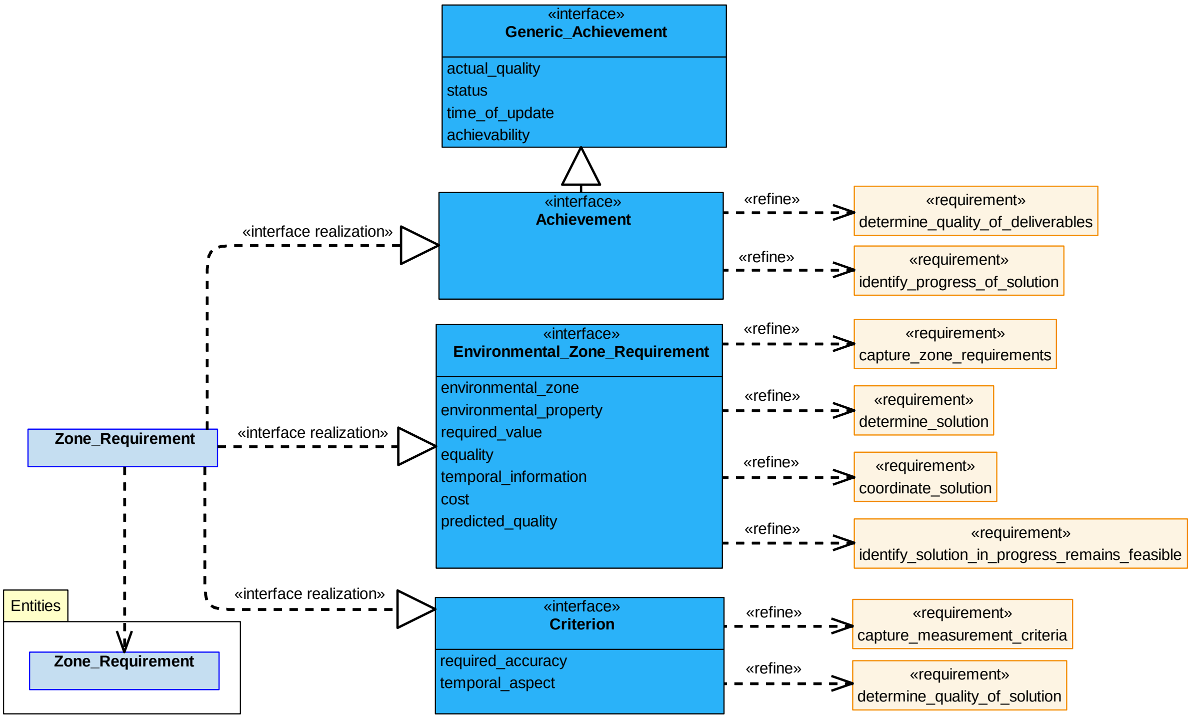
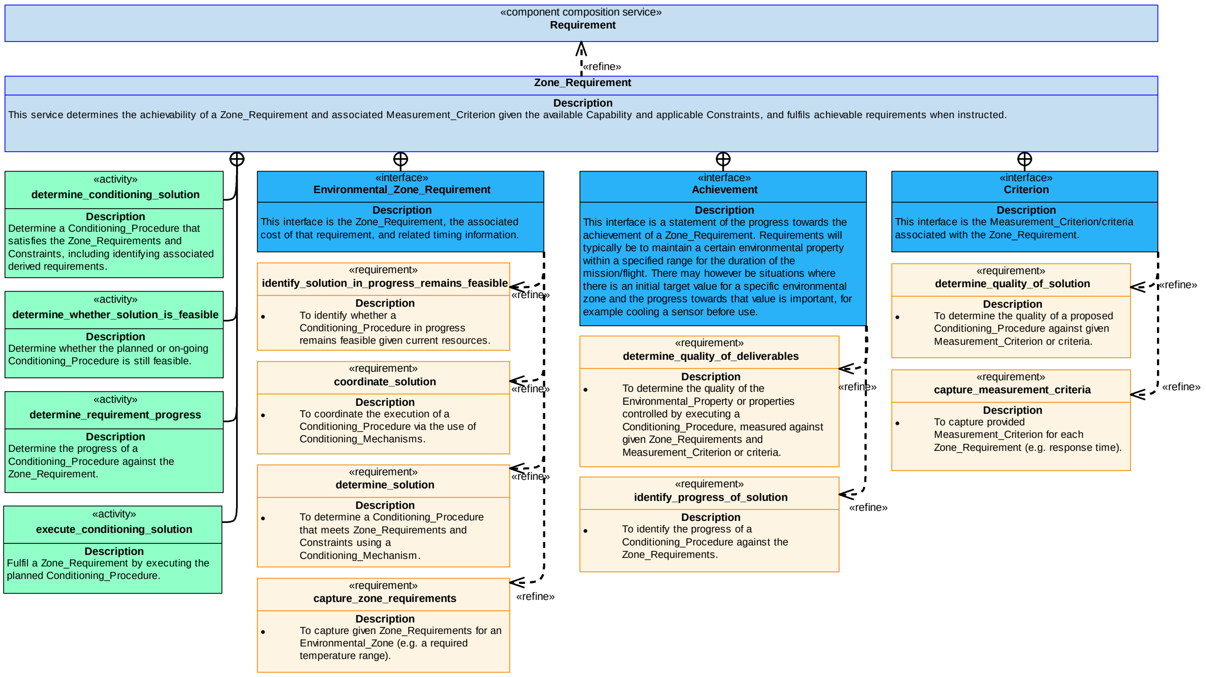


Figure 329: Zone_Requirement Service Definition



This service determines the achievability of a Zone_Requirement and associated Measurement_Criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Determine a Conditioning_Procedure that satisfies the Zone_Requirements and Constraints, including identifying associated derived requirements.

Determine whether the planned or on-going Conditioning_Procedure is still feasible.

Determine the progress of a Conditioning_Procedure against the Zone_Requirement.

Fulfil a Zone_Requirement by executing the planned Conditioning_Procedure.

This interface is the Zone_Requirement, the associated cost of that requirement, and related timing information.

- To identify whether a Conditioning_Procedure in progress remains feasible given current resources.

- To coordinate the execution of a Conditioning_Procedure via the use of Conditioning_Mechanisms.

- To determine a Conditioning_Procedure that meets Zone_Requirements and Constraints using a Conditioning_Mechanism.

- To capture given Zone_Requirements for an Environmental_Zone (e.g. a required temperature range).

This interface is a statement of the progress towards the achievement of a Zone_Requirement. Requirements will typically be to maintain a certain environmental property within a specified range for the duration of the mission/flight. There may however be situations where there is an initial target value for a specific environmental zone and the progress towards that value is important, for example cooling a sensor before use.

- To determine the quality of the Environmental_Property or properties controlled by executing a Conditioning_Procedure, measured against given Zone_Requirements and Measurement_Criterion or criteria.

- To identify the progress of a Conditioning_Procedure against the Zone_Requirements.

This interface is the Measurement_Criterion/criteria associated with the Zone_Requirement.

- To determine the quality of a proposed Conditioning_Procedure against given Measurement_Criterion or criteria.

- To capture provided Measurement_Criterion for each Zone_Requirement (e.g. response time).

Figure 330: Zone_Requirement Service Policy

Zone_Requirement

This service determines the achievability of a Zone_Requirement and associated Measurement_Criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Environmental_Zone_Requirement

This interface is the Zone_Requirement, the associated cost of that requirement, and related timing information.

Attributes

environmental_zone	The Environmental_Zone (or zones) that the requirement relates to.
environmental_property	The Environmental_Property to be controlled.
required_value	The required value of the Environmental_Property.
equality	The relationship between the required_value and any limit on the measurement, e.g. less than, or equal to.
temporal_information	Information covering timing, for example the time required to condition an environmental zone prior to a specific event.
cost	The cost of executing a Conditioning_Procedure, e.g. resources used or time taken.
predicted_quality	How well the Conditioning_Procedure is predicted to satisfy the requirement.

Achievement

This interface is a statement of the progress towards the achievement of a Zone_Requirement. Requirements will typically be to maintain a certain environmental property within a specified range for the duration of the mission/flight. There may however be situations where there is an initial target value for a specific environmental zone and the progress towards that value is important, for example cooling a sensor before use.

Criterion

This interface is the Measurement_Criterion/criteria associated with the Zone_Requirement.

Attributes

required_accuracy	A measure of how accurately the required value should be achieved.
temporal_aspect	A measure of the temporal aspects of the Conditioning_Procedure, e.g. the amount of time that a value strays outside of a desired range.

Activities

determine_requirement_progress

Determine the progress of a Conditioning_Procedure against the Zone_Requirement.

determine_conditioning_solution

Determine a Conditioning_Procedure that satisfies the Zone_Requirements and Constraints, including identifying associated derived requirements.

execute_conditioning_solution

Fulfil a Zone_Requirement by executing the planned Conditioning_Procedure.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Conditioning_Procedure is still feasible.

5.4.2.19.7.1.2 Conditioning_Action_Execution

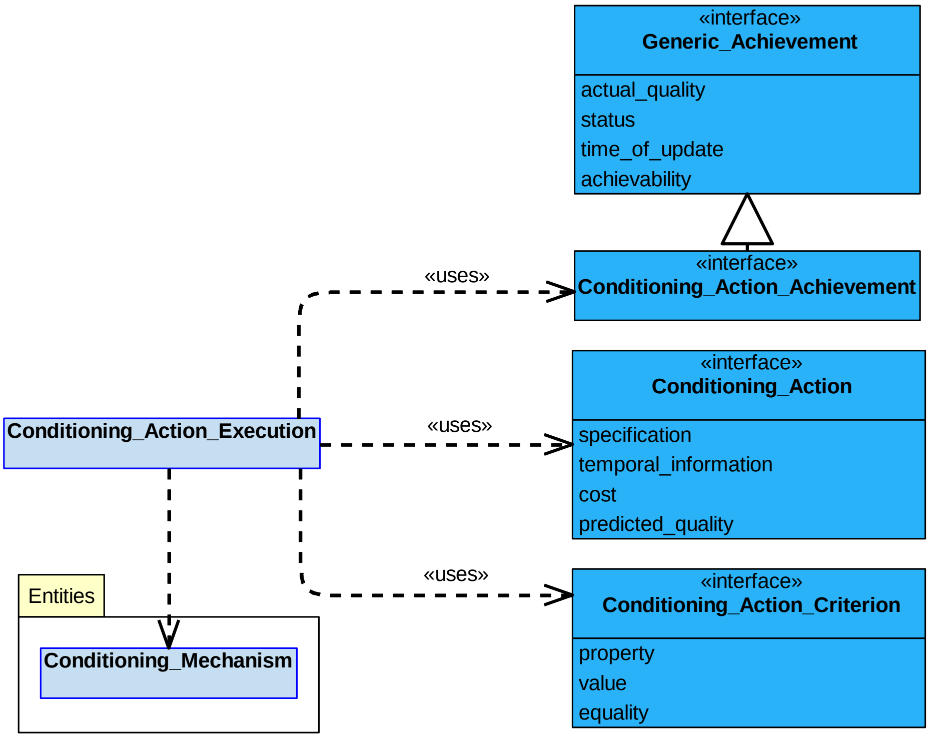


Figure 331: Conditioning_Action_Execution Service Definition

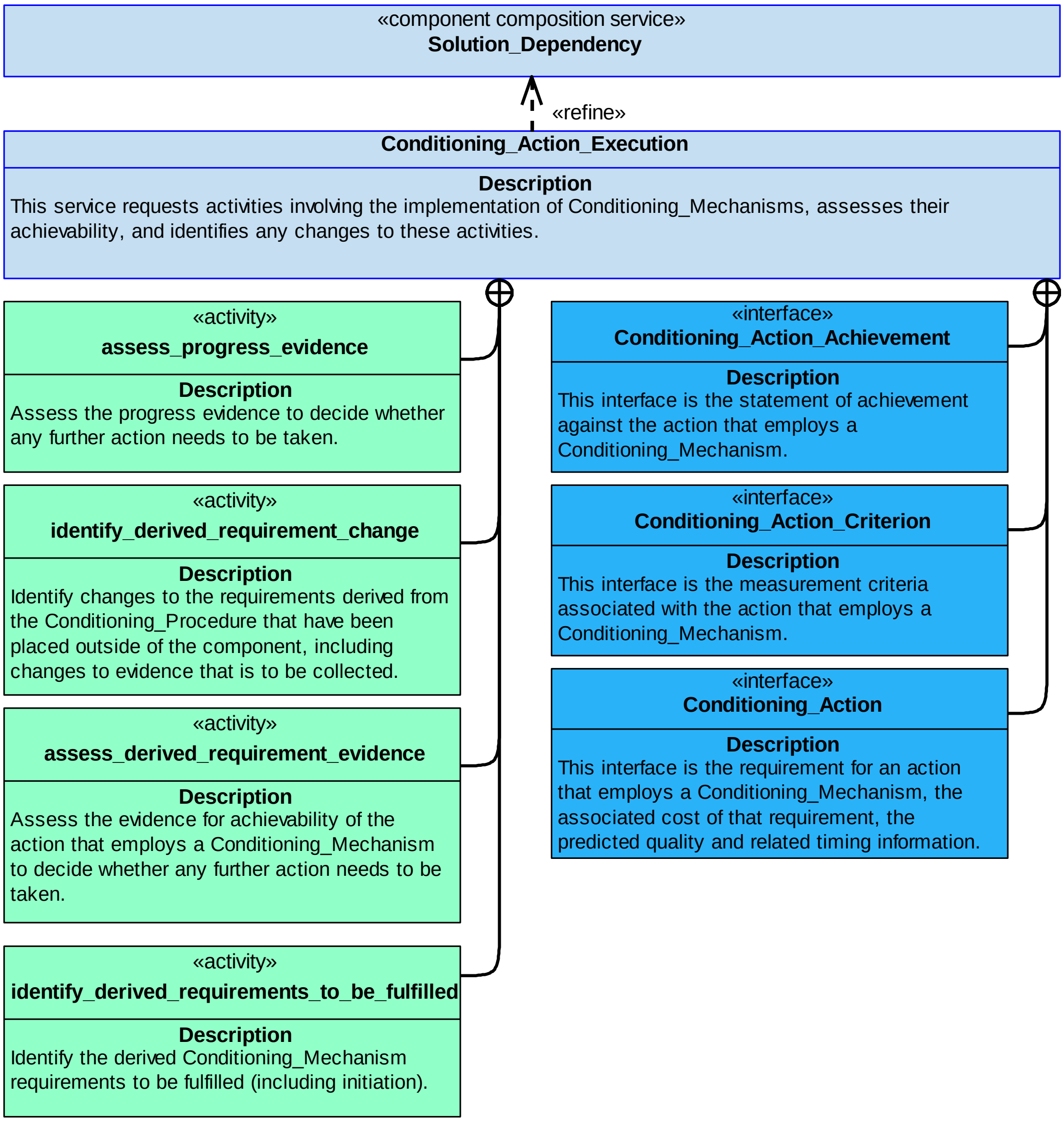

«component composition service»
Solution_Dependency
«refine»
Conditioning_Action_Execution
Description
This service requests activities involving the implementation of Conditioning_Mechanisms, assesses their achievability, and identifies any changes to these activities.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
identify_derived_requirement_change
Description
Identify changes to the requirements derived from the Conditioning_Procedure that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
assess_derived_requirement_evidence
Description
Assess the evidence for achievability of the action that employs a Conditioning_Mechanism to decide whether any further action needs to be taken.
«activity»
identify_derived_requirements_to_be_fulfilled
Description
Identify the derived Conditioning_Mechanism requirements to be fulfilled (including initiation).
«interface»
Conditioning_Action_Achievement
Description
This interface is the statement of achievement against the action that employs a Conditioning_Mechanism.
«interface»
Conditioning_Action_Criterion
Description
This interface is the measurement criteria associated with the action that employs a Conditioning_Mechanism.
«interface»
Conditioning_Action
Description
This interface is the requirement for an action that employs a Conditioning_Mechanism, the associated cost of that requirement, the predicted quality and related timing information.

Figure 332: Conditioning_Action_Execution Service Policy

Conditioning_Action_Execution

This service requests activities involving the implementation of Conditioning_Mechanisms, assesses their achievability, and identifies any changes to these activities.

Interfaces

Conditioning_Action

This interface is the requirement for an action that employs a Conditioning_Mechanism, the associated cost of that requirement, the predicted quality and related timing information.

Attributes

specification	The definition of the action.
temporal_information	Information covering timing, for example the length of time to apply a change in environmental condition.

cost	The cost of executing the action, for example: resources used or time taken.
predicted_quality	How well the planned the action is predicted to satisfy the requirement.

Conditioning_Action_Criterion

This interface is the measurement criteria associated with the action that employs a Conditioning_Mechanism.

Attributes

property	The property to be measured, e.g. temperature of anti-icing heater mats or volume of forced cooled air provided.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Conditioning_Action_Achievement

This interface is the statement of achievement against the action that employs a Conditioning_Mechanism.

Activities

assess_derived_requirement_evidence

Assess the evidence for achievability of the action that employs a Conditioning_Mechanism to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_derived_requirement_change

Identify changes to the requirements derived from the Conditioning_Procedure that have been placed outside of the component, including changes to evidence that is to be collected.

identify_derived_requirements_to_be_fulfilled

Identify the derived Conditioning_Mechanism requirements to be fulfilled (including initiation).

5.4.2.19.7.1.3 Environmental_Property_Measurement

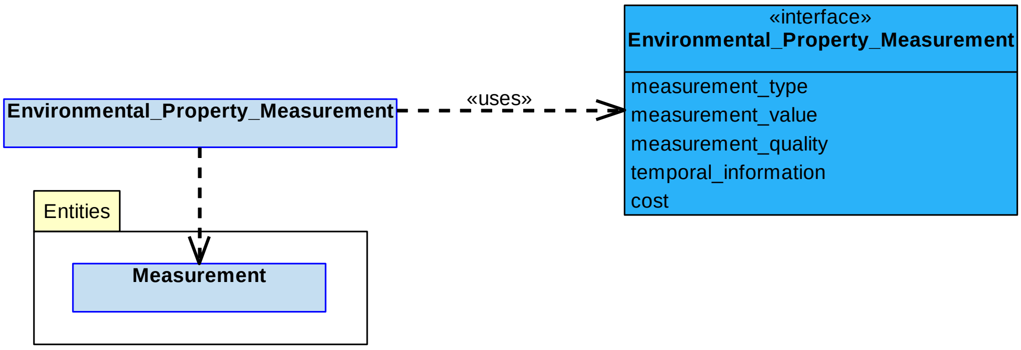


Figure 333: Environmental_Property_Measurement Service Definition

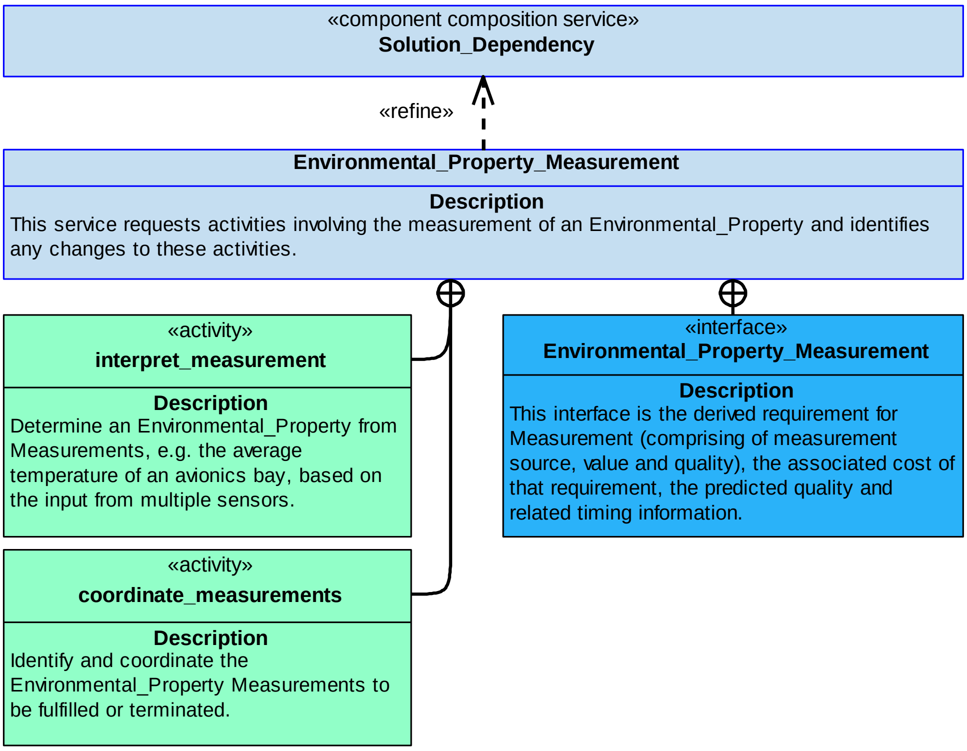

«component composition service»
Solution_Dependency
«refine»
Environmental_Property_Measurement
Description
This service requests activities involving the measurement of an Environmental_Property and identifies any changes to these activities.
«activity»
interpret_measurement
Description
Determine an Environmental_Property from Measurements, e.g. the average temperature of an avionics bay, based on the input from multiple sensors.
«interface»
Environmental_Property_Measurement
Description
This interface is the derived requirement for Measurement (comprising of measurement source, value and quality), the associated cost of that requirement, the predicted quality and related timing information.
«activity»
coordinate_measurements
Description
Identify and coordinate the Environmental_Property Measurements to be fulfilled or terminated.

Figure 334: Environmental_Property_Measurement Service Policy

Environmental_Property_Measurement

This service requests activities involving the measurement of an Environmental_Property and identifies any changes to these activities.

<u>Interface</u>

Environmental_Property_Measurement

This interface is the derived requirement for Measurement (comprising of measurement source, value and quality), the associated cost of that requirement, the predicted quality and related timing information.

<u>Attributes</u>

measurement_type	The Environmental_Property being measured.
measurement_value	The Environmental_Property Measurement value.
measurement_quality	The quality of the returned measurement value, e.g. accuracy and precision.
temporal_information	Information covering timing, e.g. the frequency that the measurements are taken over a certain period.
cost	The cost of executing the Measurement solution, e.g. resources used or time taken.

Activities

interpret_measurement

Determine an Environmental_Property from Measurements, e.g. the average temperature of an avionics bay, based on the input from multiple sensors.

coordinate_measurements

Identify and coordinate the Environmental_Property Measurements to be fulfilled or terminated.

5.4.2.19.7.1.4 Condition_Information

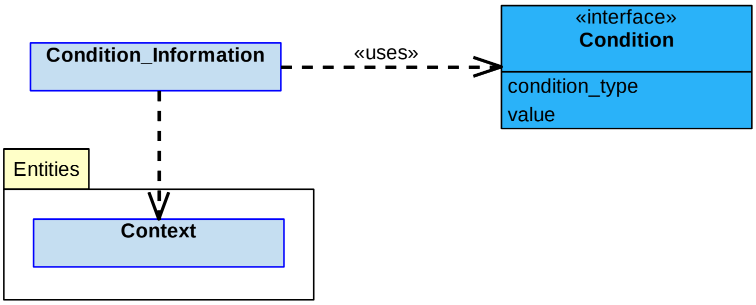


Figure 335: Condition_Information Service Definition

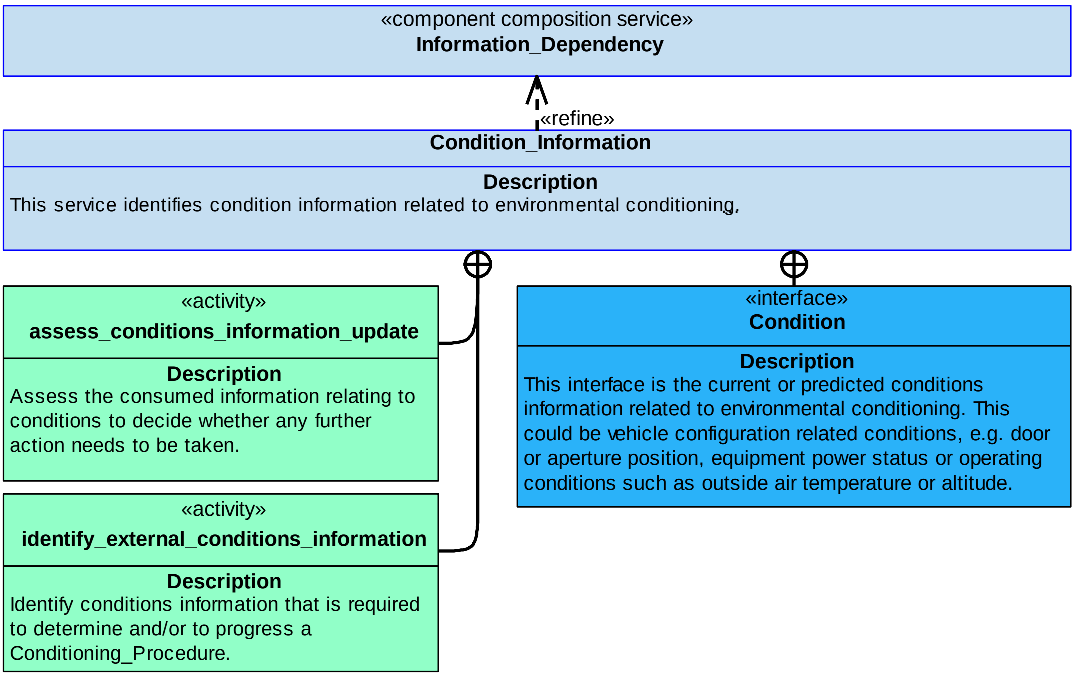

«component composition service»
Information_Dependency
«refine»
Condition_Information
Description
This service identifies condition information related to environmental conditioning.
«activity»
assess_conditions_information_update
Description
Assess the consumed information relating to conditions to decide whether any further action needs to be taken.
«activity»
identify_external_conditions_information
Description
Identify conditions information that is required to determine and/or to progress a Conditioning_Procedure.
«interface»
Condition
Description
This interface is the current or predicted conditions information related to environmental conditioning. This could be vehicle configuration related conditions, e.g. door or aperture position, equipment power status or operating conditions such as outside air temperature or altitude.

Figure 336: Condition_Information Service Policy

Condition_Information

This service identifies condition information related to environmental conditioning.

Interface

Condition

This interface is the current or predicted conditions information related to environmental conditioning. This could be vehicle configuration related conditions, e.g. door or aperture position, equipment power status or operating conditions such as outside air temperature or altitude.

Attributes

condition_type	The type of condition.
value	A value or state of the related condition.

Activities

assess_conditions_information_update

Assess the consumed information relating to conditions to decide whether any further action needs to be taken.

identify_external_conditions_information

Identify conditions information that is required to determine and/or to progress a Conditioning_Procedure.

5.4.2.19.7.1.5 Constraint

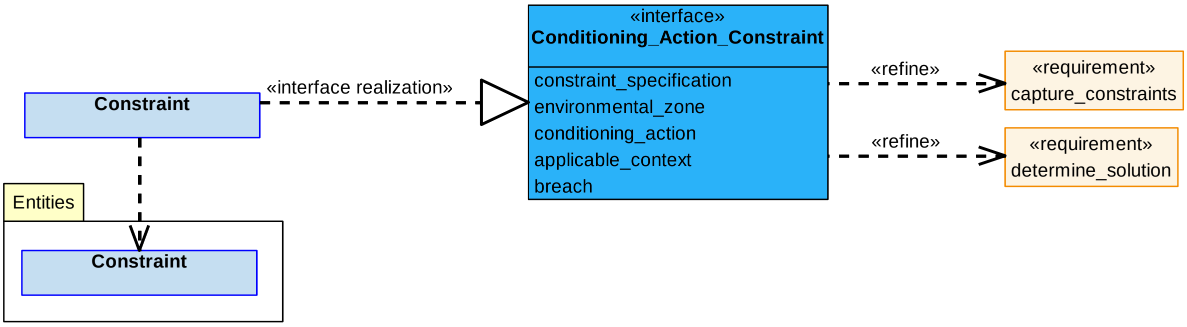


Figure 337: Constraint Service Definition

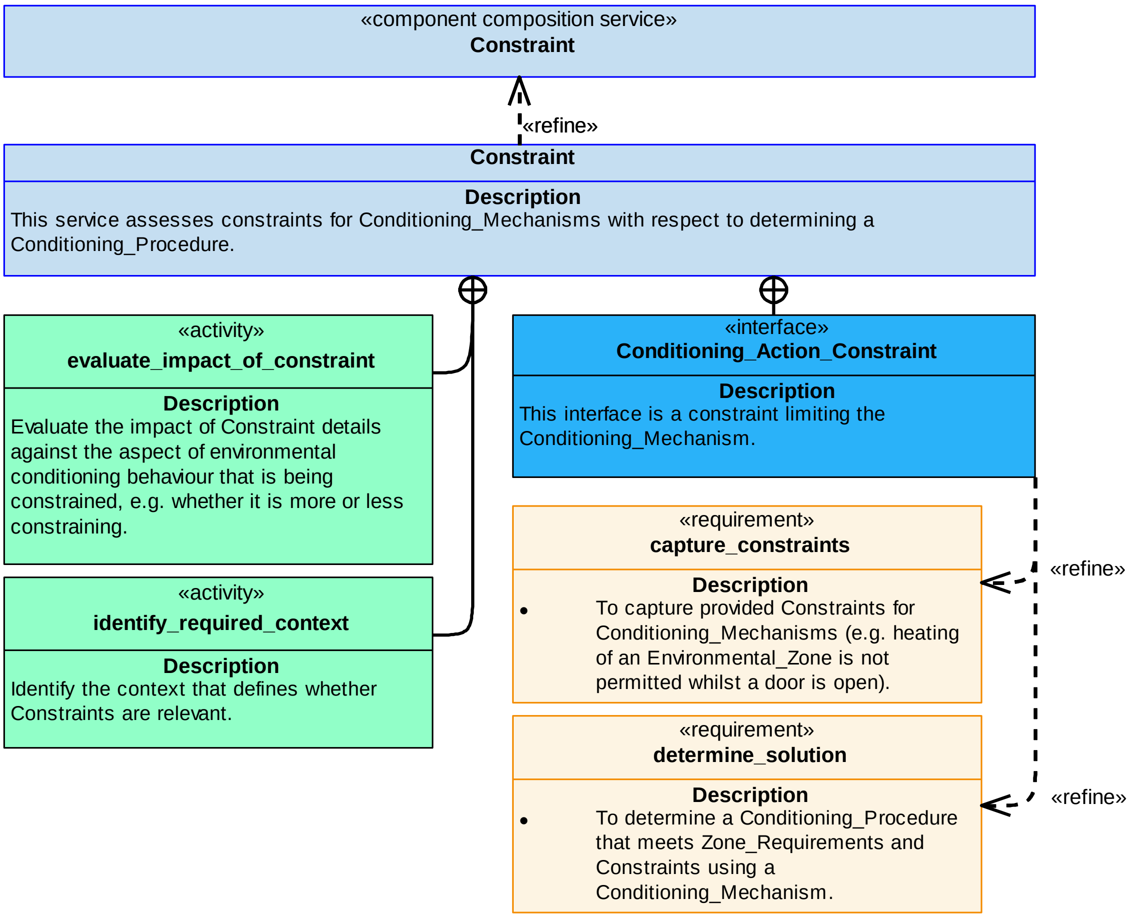

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses constraints for Conditioning_Mechanisms with respect to determining a Conditioning_Procedure.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of environmental conditioning behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context that defines whether Constraints are relevant.
«interface»
Conditioning_Action_Constraint
Description
This interface is a constraint limiting the Conditioning_Mechanism.
«requirement»
capture_constraints
Description
To capture provided Constraints for Conditioning_Mechanisms (e.g. heating of an Environmental_Zone is not permitted whilst a door is open).
«refine»
«requirement»
determine_solution
Description
To determine a Conditioning_Procedure that meets Zone_Requirements and Constraints using a Conditioning_Mechanism.
«refine»

Figure 338: Constraint Service Policy

Constraint

This service assesses constraints for Conditioning_Mechanisms with respect to determining a Conditioning_Procedure.

Interface

Conditioning_Action_Constraint

This interface is a constraint limiting the Conditioning_Mechanism.

Attributes

constraint_specification	A Constraint that will restrict the Conditioning_Mechanism.
environmental_zone	The Environmental_Zone (or zones) that the Constraint relates to.
conditioning_action	The Conditioning_Mechanism that the Constraint relates to.
applicable_context	The Context in which the Constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of environmental conditioning behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context that defines whether Constraints are relevant.

5.4.2.19.7.1.6 Capability

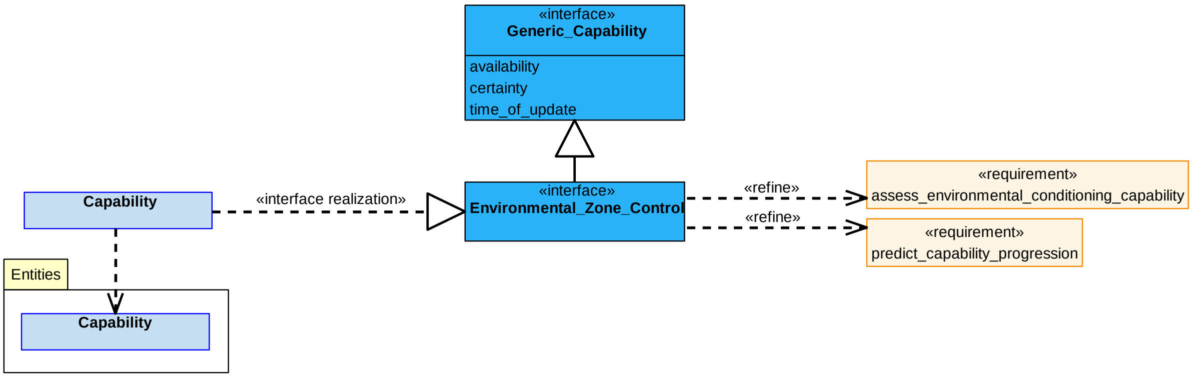


Figure 339: Capability Service Definition

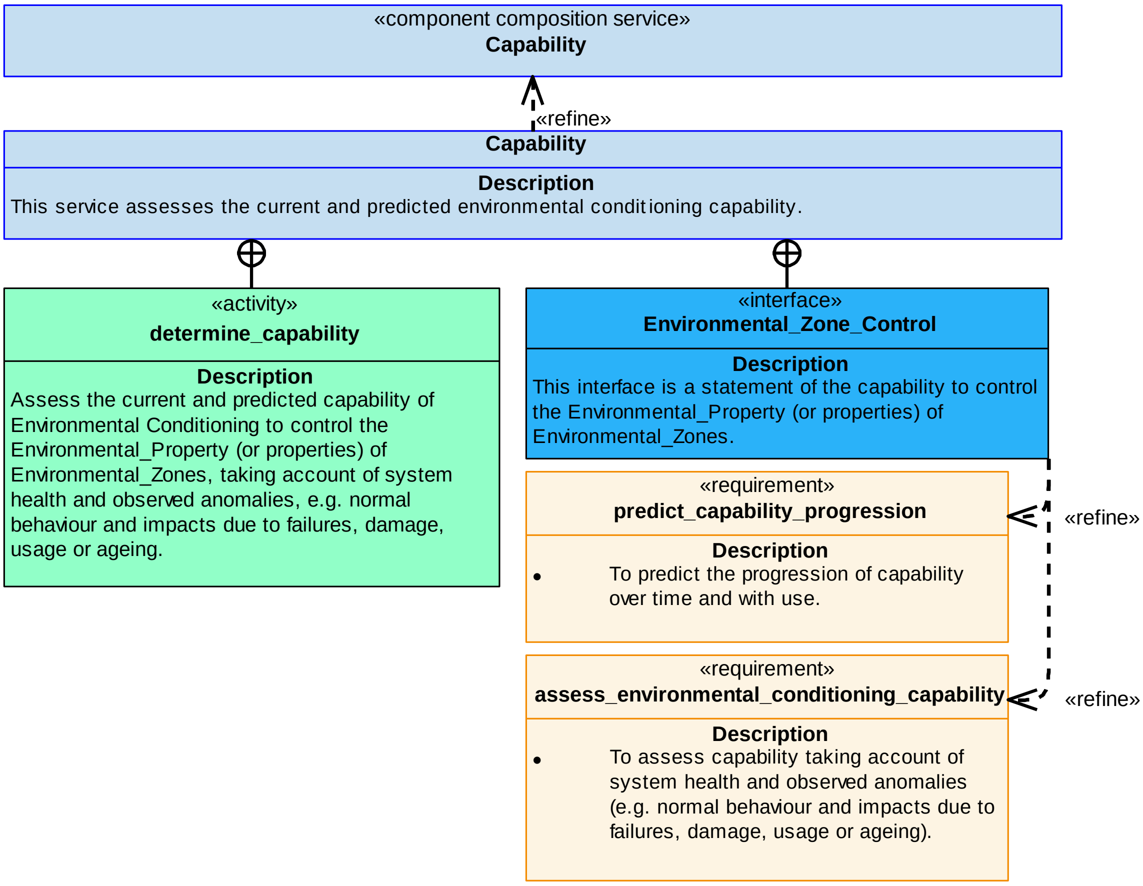

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted environmental conditioning capability.
«activity»
determine_capability
Description
Assess the current and predicted capability of Environmental Conditioning to control the Environmental_Property (or properties) of Environmental_Zones, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.
«interface»
Environmental_Zone_Control
Description
This interface is a statement of the capability to control the Environmental_Property (or properties) of Environmental_Zones.
«requirement»
predict_capability_progression
Description
• To predict the progression of capability over time and with use.
«refine»
«requirement»
assess_environmental_conditioning_capability
Description
• To assess capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 340: Capability Service Policy

Capability

This service assesses the current and predicted environmental conditioning capability.

Interface

Environmental_Zone_Control

This interface is a statement of the capability to control the Environmental_Property (or properties) of Environmental_Zones.

Activity

determine_capability

Assess the current and predicted capability of Environmental Conditioning to control the Environmental_Property (or properties) of Environmental_Zones, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.

5.4.2.19.7.1.7 Capability_Evidence

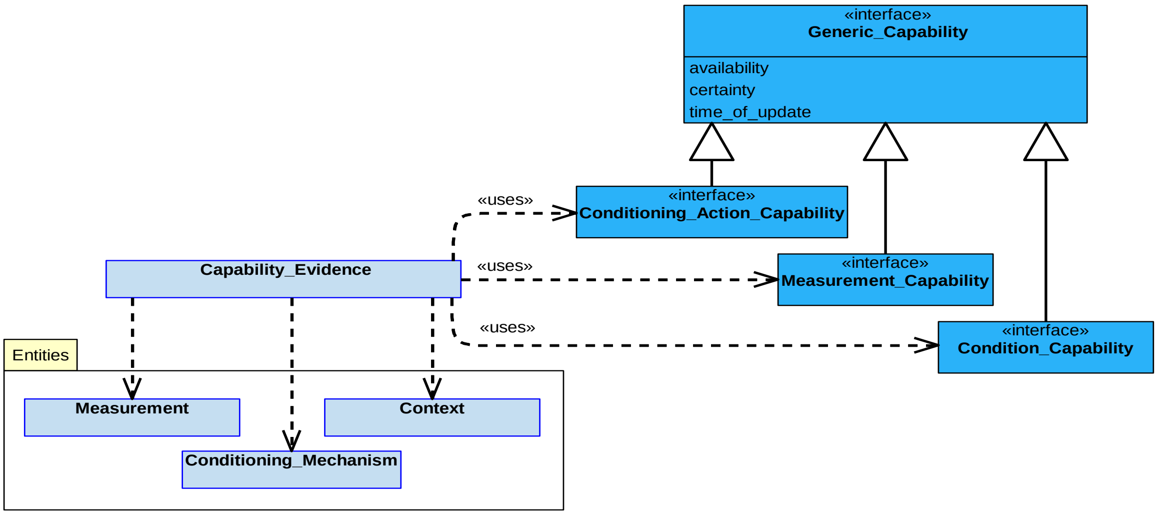
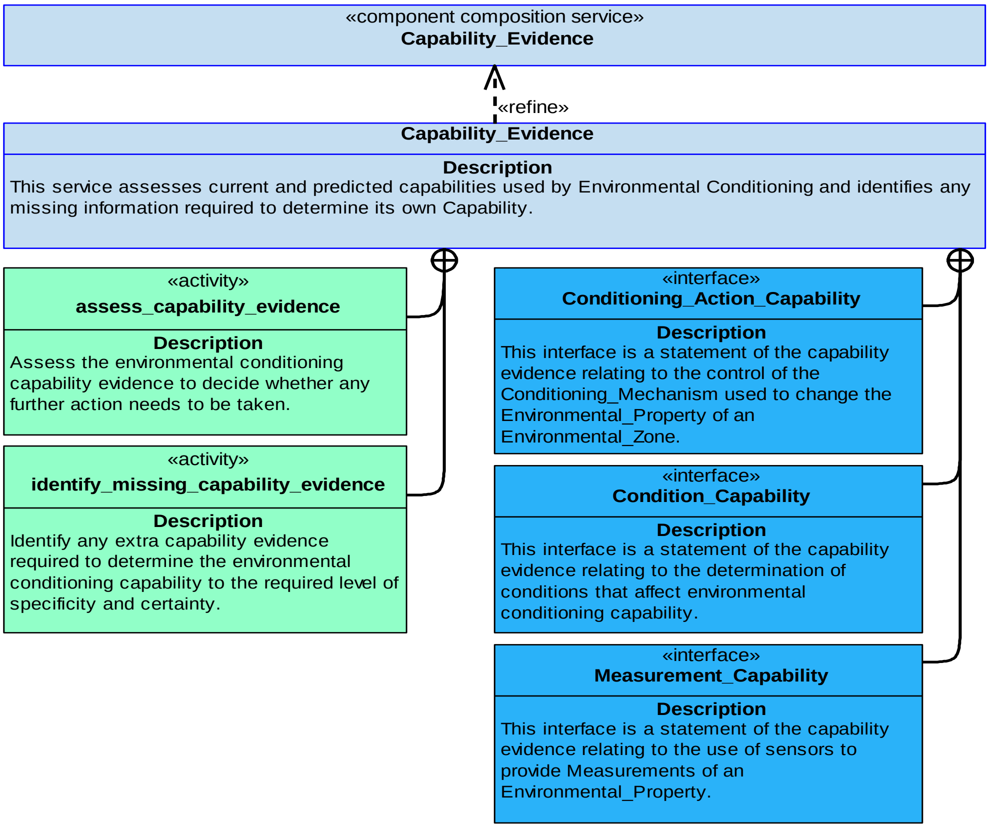


Figure 341: Capability_Evidence Service Definition



Description

This service assesses current and predicted capabilities used by Environmental Conditioning and identifies any missing information required to determine its own Capability.

«activity»
assess_capability_evidence

Description

Assess the environmental conditioning capability evidence to decide whether any further action needs to be taken.

«activity»
identify_missing_capability_evidence

Description

Identify any extra capability evidence required to determine the environmental conditioning capability to the required level of specificity and certainty.

«interface»
Conditioning_Action_Capability

Description

This interface is a statement of the capability evidence relating to the control of the Conditioning_Mechanism used to change the Environmental_Property of an Environmental_Zone.

«interface»
Condition_Capability

Description

This interface is a statement of the capability evidence relating to the determination of conditions that affect environmental conditioning capability.

«interface»
Measurement_Capability

Description

This interface is a statement of the capability evidence relating to the use of sensors to provide Measurements of an Environmental_Property.

Figure 342: Capability_Evidence Service Policy

Capability_Evidence

This service assesses current and predicted capabilities used by Environmental Conditioning and identifies any missing information required to determine its own Capability.

Interfaces

Conditioning_Action_Capability

This interface is a statement of the capability evidence relating to the control of the Conditioning_Mechanism used to change the Environmental_Property of an Environmental_Zone.

Measurement_Capability

This interface is a statement of the capability evidence relating to the use of sensors to provide Measurements of an Environmental_Property.

Condition_Capability

This interface is a statement of the capability evidence relating to the determination of conditions that affect environmental conditioning capability.

Activities

assess_capability_evidence

Assess the environmental conditioning capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the environmental conditioning capability to the required level of specificity and certainty.

5.4.2.19.7.2 Service Dependencies

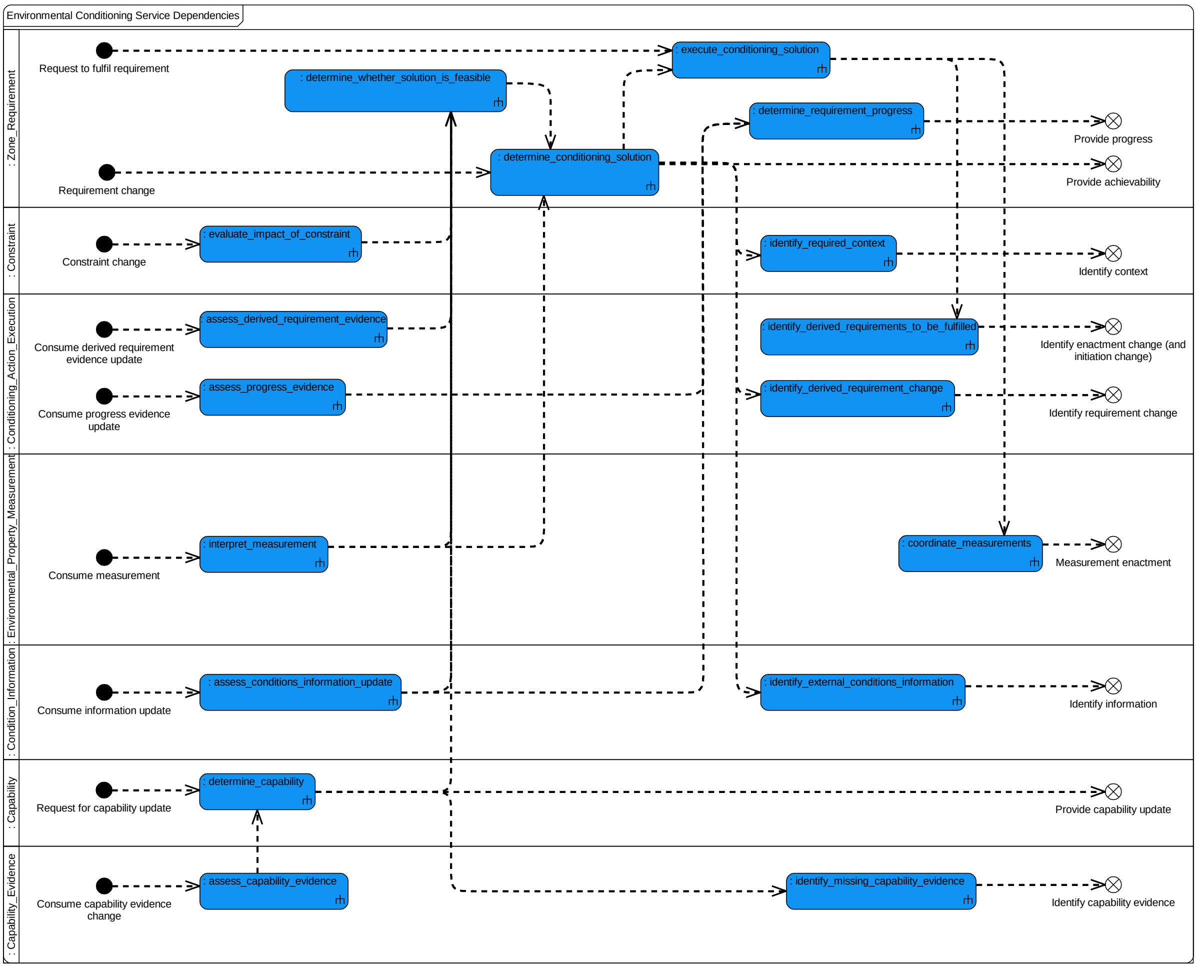

Environmental Conditioning Service Dependencies
: Zone_Requirement
: Constraint
: Conditioning_Action_Execution
: Environmental_Property_Measurement
: Condition_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_conditioning_solution
: determine_whether_solution_is_feasible
: determine_requirement_progress
Provide progress
Provide achievability
Requirement change
: determine_conditioning_solution
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume derived requirement evidence update
: assess_derived_requirement_evidence
: identify_derived_requirements_to_be_fulfilled
Identify enactment change (and initiation change)
Consume progress evidence update
: assess_progress_evidence
: identify_derived_requirement_change
Identify requirement change
Consume measurement
: interpret_measurement
: coordinate_measurements
Measurement enactment
Consume information update
: assess_conditions_information_update
: identify_external_conditions_information
Identify information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 343: Environmental Conditioning Service Dependencies

5.4.2.20 Flights

5.4.2.20.1 Role

The role of Flights is to manage the composition of flights that Exploiting Platforms participate in.

5.4.2.20.2 Overview

Control Architecture

Flights is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Upon receiving a Requirement, this component will use a Resource to manage Flight composition (e.g. it may use a communication resource to identify or change the Role of an aircraft).

Examples of Use

- Where a Flight does not have fixed membership, e.g. if there is a requirement to merge one Flight with another.
- Where there may be an advantage in transferring the flight lead Role, e.g. if there is a high risk of losing the flight lead, or if different Members are more suitable leads in different situations.

5.4.2.20.3 Service Summary

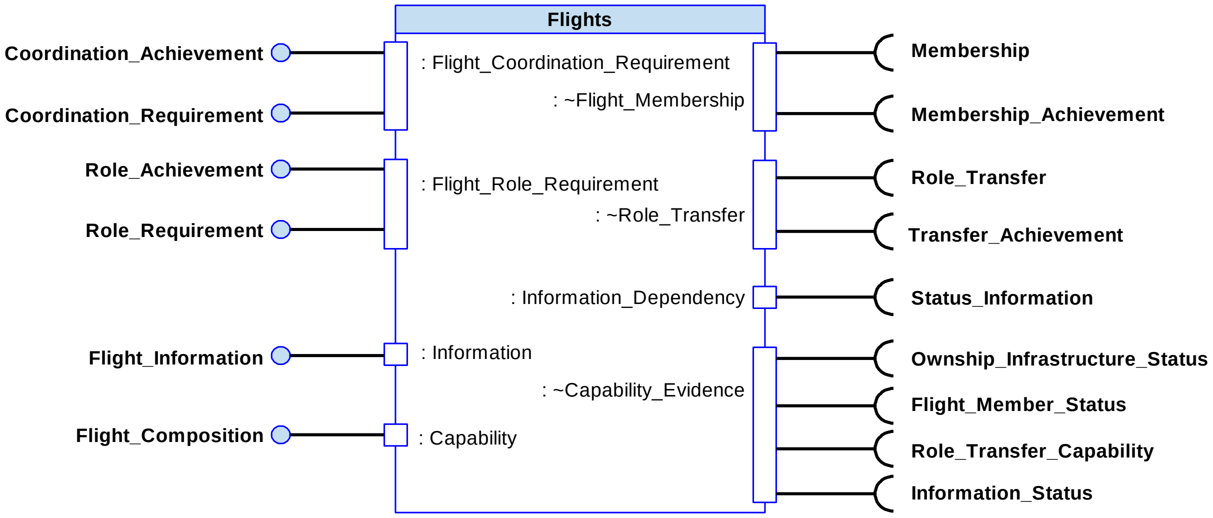


Figure 344: Flights Service Summary

5.4.2.20.4 Responsibilities

capture_requirements

- To capture provided Requirements related to Flight membership (including requests to join or leave a Flight and the rules and assessments applicable to joining).

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Capability.

coordinate_flight_member_departure

- To coordinate departure of a Member from a Flight.

coordinate_flight_member_arrival

- To coordinate arrival of a Member into a Flight.

coordinate_role_handover

- To coordinate a pre-planned Role handover to an eligible Member.

coordinate_role_takeover

- To coordinate an unplanned Role takeover due to loss of capability suffered by a Member.

maintain_flight_control_structure

- To manage the Flight in accordance with the Control_Structure (e.g. maintain the hierarchy).

identify_achievement

- To identify what has been achieved against the Requirement.

identify_flight_members

- To identify the Members that make up a Flight.

identify_role_absence

- To identify when a Role is no longer being fulfilled (e.g. the flight lead is not capable of staying in command due to being destroyed or damaged).

assess_capability

- To assess the Capability to manage Flight composition taking account of system health and observed anomalies.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Capability assessment (e.g. identifying that the status of a Member is not being updated).

predict_capability_progression

- To predict the progression of Flights' Capability over time and with use (e.g. predicting that capability will reduce because communication transmission capacity is deteriorating).

5.4.2.20.5 Subject Matter Semantics

The subject matter of Flights is the status and management of the composition of a Flight.

Exclusions

The subject matter of Flights does not include:

- The spatial arrangement of vehicles.

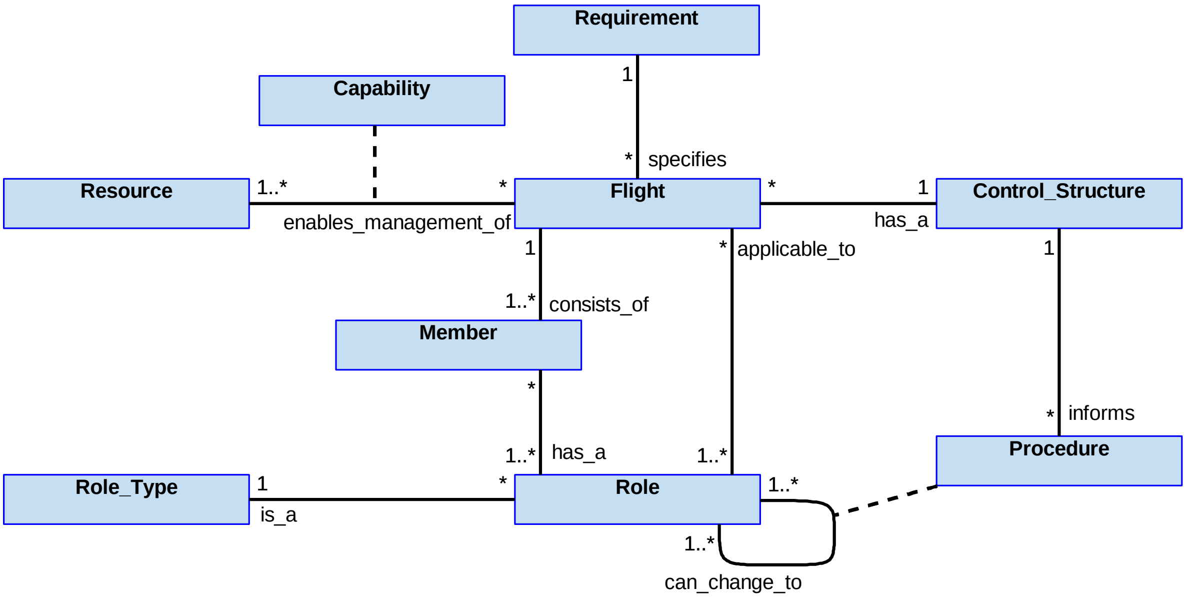


Figure 345: Flights Semantics

5.4.2.20.5.1 Entities

Capability

The capability of this component to manage Flight composition.

Control_Structure

The structure of governance (e.g. hierarchy or non-hierarchical swarm) and associated rules that a Flight's Members adhere to.

Flight

A collection of one or more aircraft that have the potential to or have been specifically tasked with achieving a pre-defined outcome with roles and tasks undertaken by the flight constituent parts to achieve the overall mission.

Member

Any aircraft that forms part of a current Flight. Each member acts in support of the overall Flight aims and in support of the other flight members.

Procedure

The established process for a role change (e.g. flight lead handover).

Requirement

A requirement to alter Flight composition (e.g. a request to join a Flight from a prospective Member).

Resource

A resource that this component uses to manage Flight composition (e.g. a communication resource).

Role

The current role of an asset in relation to a particular Flight.

Role_Type

A category of Role (e.g. flight lead, deputy, identified potential or former Member).

5.4.2.20.6 Design Rationale

5.4.2.20.6.1 Assumptions

- A Member will not be part of more than one Flight at a time.

5.4.2.20.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Flights:

- Data Driving - Procedures for managing a Flight are likely to vary between operators, use of data driving should be considered to accommodate this.
- Multi-Vehicle Coordination - This component enables coordinated interaction between vehicles, therefore the Multi-Vehicle Coordination PYRAMID concept is applicable.

5.4.2.20.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component would mean that coordination of the mission objectives across the available assets was not fulfilled, which is not a safety concern. It is expected that other components in the Control Architecture would ensure that any actions were performed safely (e.g. Interlocks) and perform mitigating actions to accommodate failures or a change in circumstances (e.g. Tasks). However, failure of the component would cause an increase in workload for crew, which is considered a "Significant Reduction in Safety Margins" (severity major). Therefore, the indicative IDAL is DAL C.

5.4.2.20.6.4 Security Considerations

The indicative security classification is SNEO.

This component manages the composition of Flights in support of mission objectives, and where it carries the role of flight lead, it may modify the roles of flight members, therefore the component is assumed to be SNEO. The confidentiality, integrity and availability of flight interactions will need to be protected.

The security of communications channels used to coordinate between flight members is not a function of this component.

The component may be expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** as trusted members (or potential members) of the flight.
- **Logging of Security Data** relating to member role assignments for later forensic examination.
- **Maintaining Audit Records** to support non-repudiation of instructions and role changes, etc. in the course of operations of the flight.
- **System Status and Monitoring** through coordinating and monitoring the available assets. Unexpected flight activity may indicate that one or more flight members have been compromised by a cyber adversary.

This component is considered unlikely to implement security enforcing functions.

5.4.2.20.7 Services

5.4.2.20.7.1 Service Definitions

5.4.2.20.7.1.1 Flight_Coordination_Requirement

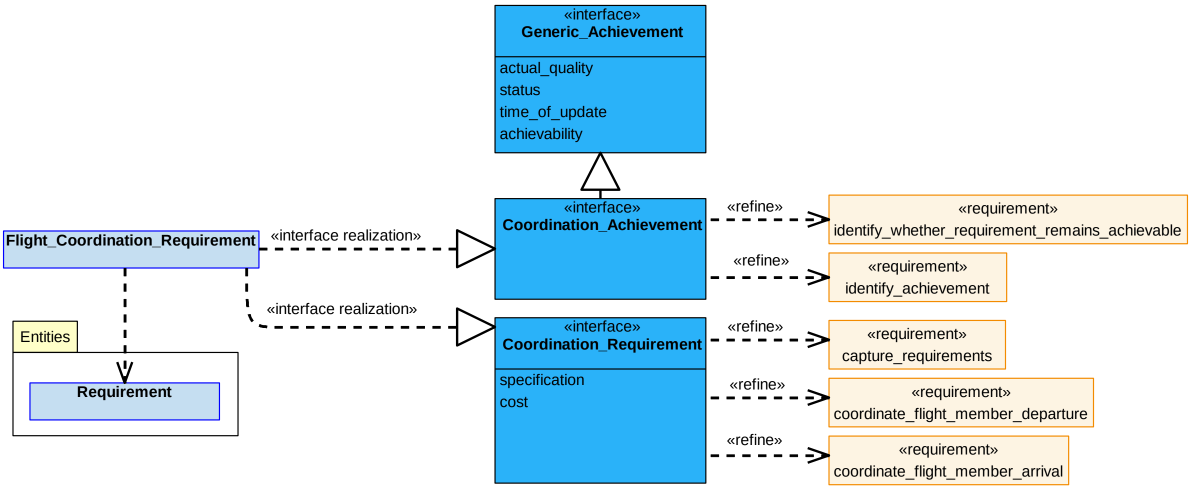


Figure 346: Flight_Coordination_Requirement Service Definition

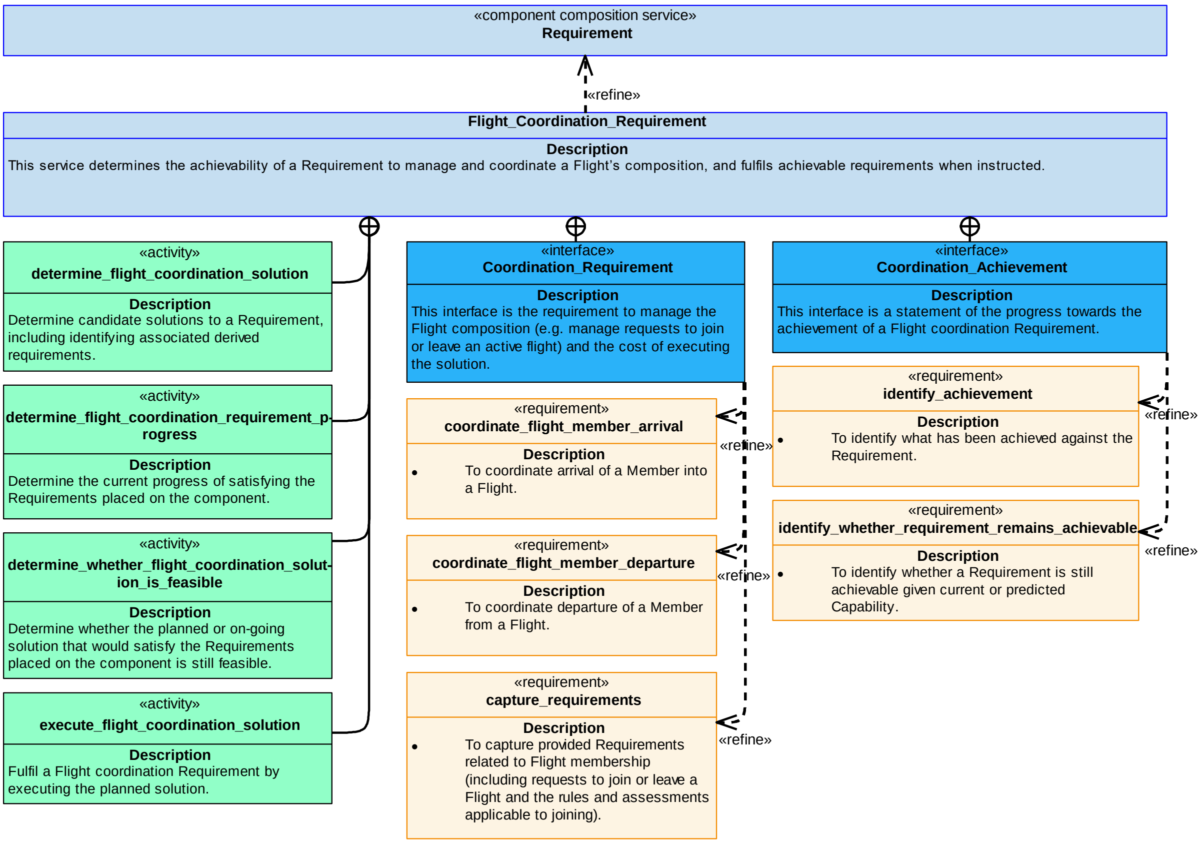

«component composition service»
Requirement
«refine»
Flight_Coordination_Requirement
Description
This service determines the achievability of a Requirement to manage and coordinate a Flight's composition, and fulfils achievable requirements when instructed.
«activity»
determine_flight_coordination_solution
Description
Determine candidate solutions to a Requirement, including identifying associated derived requirements.
«activity»
determine_flight_coordination_requirement_progress
Description
Determine the current progress of satisfying the Requirements placed on the component.
«activity»
determine_whether_flight_coordination_solution_is_feasible
Description
Determine whether the planned or on-going solution that would satisfy the Requirements placed on the component is still feasible.
«activity»
execute_flight_coordination_solution
Description
Fulfil a Flight coordination Requirement by executing the planned solution.
«interface»
Coordination_Requirement
Description
This interface is the requirement to manage the Flight composition (e.g. manage requests to join or leave an active flight) and the cost of executing the solution.
«requirement»
coordinate_flight_member_arrival
Description
To coordinate arrival of a Member into a Flight.
«refine»
«requirement»
coordinate_flight_member_departure
Description
To coordinate departure of a Member from a Flight.
«refine»
«requirement»
capture_requirements
Description
To capture provided Requirements related to Flight membership (including requests to join or leave a Flight and the rules and assessments applicable to joining).
«refine»
«interface»
Coordination_Achievement
Description
This interface is a statement of the progress towards the achievement of a Flight coordination Requirement.
«requirement»
identify_achievement
Description
To identify what has been achieved against the Requirement.
«refine»
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Requirement is still achievable given current or predicted Capability.
«refine»

Figure 347: Flight_Coordination_Requirement Service Policy

Flight_Coordination_Requirement

This service determines the achievability of a Requirement to manage and coordinate a Flight's composition, and fulfils achievable requirements when instructed.

Interfaces

Coordination_Requirement

This interface is the requirement to manage the Flight composition (e.g. manage requests to join or leave an active flight) and the cost of executing the solution.

Attributes

specification	The definition of the requirement placed on the component.
cost	The cost of executing the solution, for example: resources used or time taken.

Coordination_Achievement

This interface is a statement of the progress towards the achievement of a Flight coordination Requirement.

Activities

determine_flight_coordination_solution

Determine candidate solutions to a Requirement, including identifying associated derived requirements.

determine_flight_coordination_requirement_progress

Determine the current progress of satisfying the Requirements placed on the component.

determine_whether_flight_coordination_solution_is_feasible

Determine whether the planned or on-going solution that would satisfy the Requirements placed on the component is still feasible.

execute_flight_coordination_solution

Fulfil a Flight coordination Requirement by executing the planned solution.

5.4.2.20.7.1.2 Flight_Role_Requirement

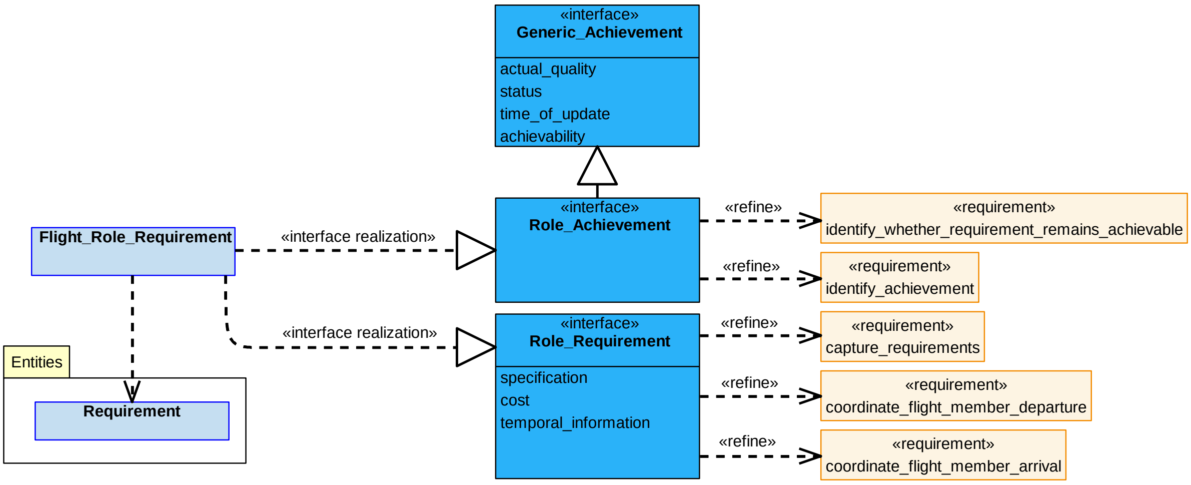
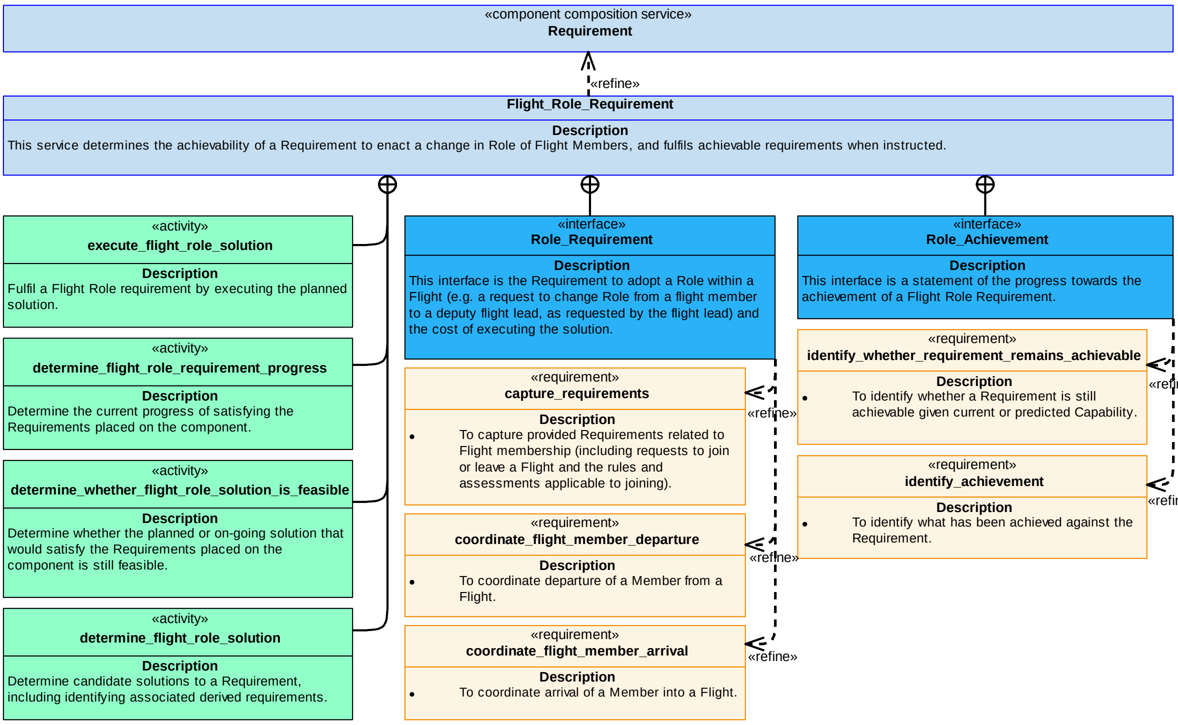


Figure 348: Flight_Role_Requirement Service Definition



This service determines the achievability of a Requirement to enact a change in Role of Flight Members, and fulfils achievable requirements when instructed.

Fulfil a Flight Role requirement by executing the planned solution.

Determine the current progress of satisfying the Requirements placed on the component.

Determine whether the planned or on-going solution that would satisfy the Requirements placed on the component is still feasible.

Determine candidate solutions to a Requirement, including identifying associated derived requirements.

This interface is the Requirement to adopt a Role within a Flight (e.g. a request to change Role from a flight member to a deputy flight lead, as requested by the flight lead) and the cost of executing the solution.

This interface is a statement of the progress towards the achievement of a Flight Role Requirement.

- To capture provided Requirements related to Flight membership (including requests to join or leave a Flight and the rules and assessments applicable to joining).

- To coordinate departure of a Member from a Flight.

- To coordinate arrival of a Member into a Flight.

- To identify whether a Requirement is still achievable given current or predicted Capability.

- To identify what has been achieved against the Requirement.

Figure 349: Flight_Role_Requirement Service Policy

Flight_Role_Requirement

This service determines the achievability of a Requirement to enact a change in Role of Flight Members, and fulfils achievable requirements when instructed.

Interfaces

Role_Requirement

This interface is the Requirement to adopt a Role within a Flight (e.g. a request to change Role from a flight member to a deputy flight lead, as requested by the flight lead) and the cost of executing the solution.

Attributes

specification	The definition of the requirement placed on the component, e.g. a flight lead requesting that a non-flight member change its Role to a flight member.
cost	The cost of executing the solution, for example: resources used or time taken.
temporal_information	Information covering timing, such as start and end times.

Role_Achievement

This interface is a statement of the progress towards the achievement of a Flight Role Requirement.

Activities

determine_flight_role_solution

Determine candidate solutions to a Requirement, including identifying associated derived requirements.

determine_flight_role_requirement_progress

Determine the current progress of satisfying the Requirements placed on the component.

determine_whether_flight_role_solution_is_feasible

Determine whether the planned or on-going solution that would satisfy the Requirements placed on the component is still feasible.

execute_flight_role_solution

Fulfil a Flight Role requirement by executing the planned solution.

5.4.2.20.7.1.3 Role_Transfer

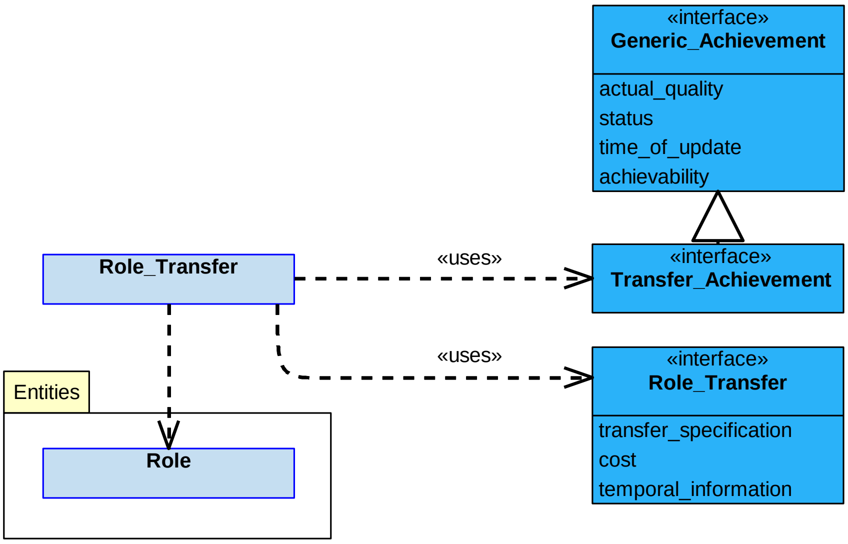


Figure 350: Role_Transfer Service Definition

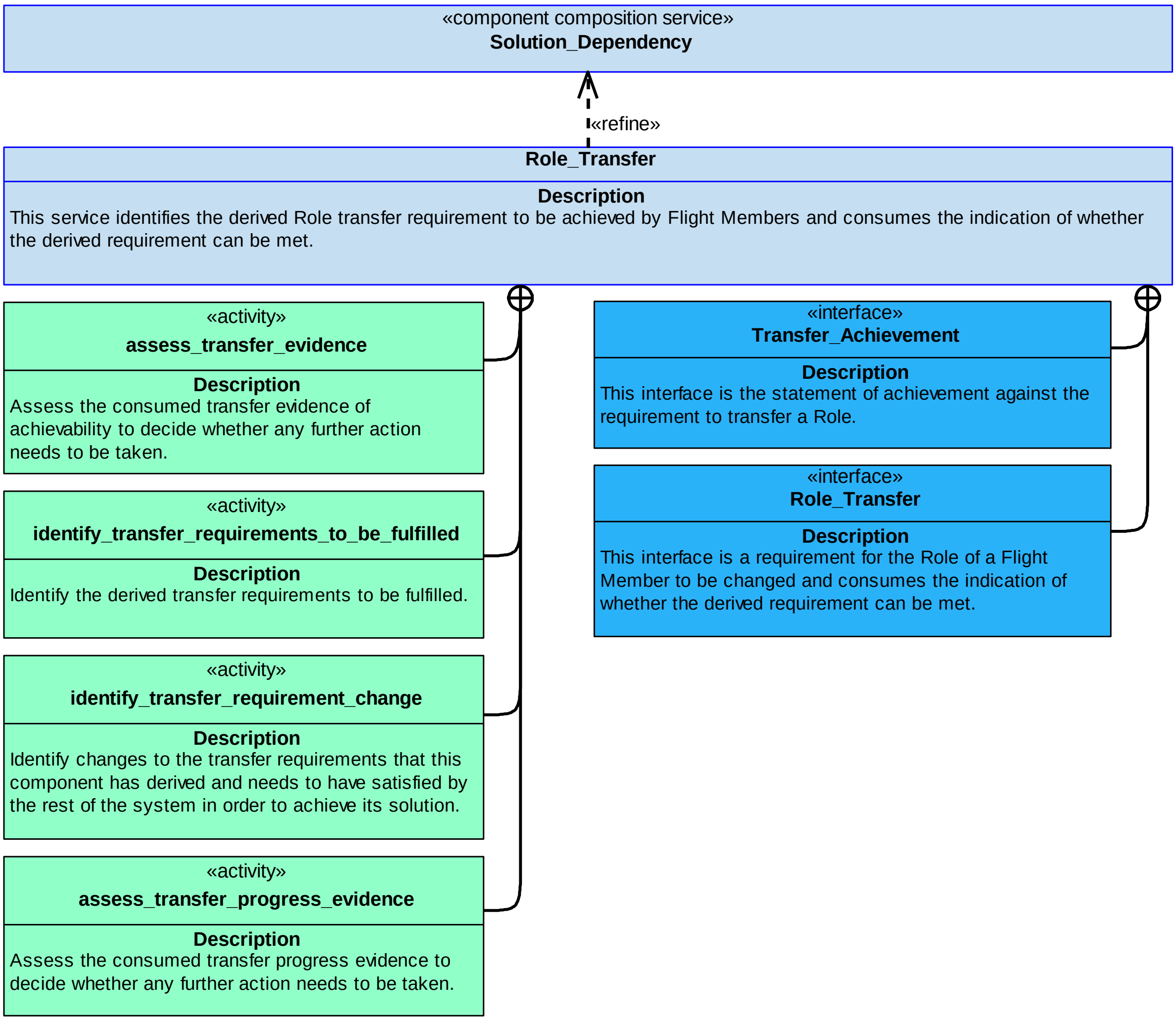

«component composition service»
Solution_Dependency
«refine»
Role_Transfer
Description
This service identifies the derived Role transfer requirement to be achieved by Flight Members and consumes the indication of whether the derived requirement can be met.
«activity»
assess_transfer_evidence
Description
Assess the consumed transfer evidence of achievability to decide whether any further action needs to be taken.
«activity»
identify_transfer_requirements_to_be_fulfilled
Description
Identify the derived transfer requirements to be fulfilled.
«activity»
identify_transfer_requirement_change
Description
Identify changes to the transfer requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution.
«activity»
assess_transfer_progress_evidence
Description
Assess the consumed transfer progress evidence to decide whether any further action needs to be taken.
«interface»
Transfer_Achievement
Description
This interface is the statement of achievement against the requirement to transfer a Role.
«interface»
Role_Transfer
Description
This interface is a requirement for the Role of a Flight Member to be changed and consumes the indication of whether the derived requirement can be met.

Figure 351: Role_Transfer Service Policy

Role_Transfer

This service identifies the derived Role transfer requirement to be achieved by Flight Members and consumes the indication of whether the derived requirement can be met.

Interfaces

Role_Transfer

This interface is a requirement for the Role of a Flight Member to be changed and consumes the indication of whether the derived requirement can be met.

Attributes

transfer_specification	The definition of the Role transfer requirement.
cost	The cost of executing the solution, for example: resources used or time taken.
temporal_information	Information covering timing, such as start and end times.

Transfer_Achievement

This interface is the statement of achievement against the requirement to transfer a Role.

Activities

assess_transfer_evidence

Assess the consumed transfer evidence of achievability to decide whether any further action needs to be taken.

assess_transfer_progress_evidence

Assess the consumed transfer progress evidence to decide whether any further action needs to be taken.

identify_transfer_requirement_change

Identify changes to the transfer requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution.

identify_transfer_requirements_to_be_fulfilled

Identify the derived transfer requirements to be fulfilled.

5.4.2.20.7.1.4 Flight_Membership

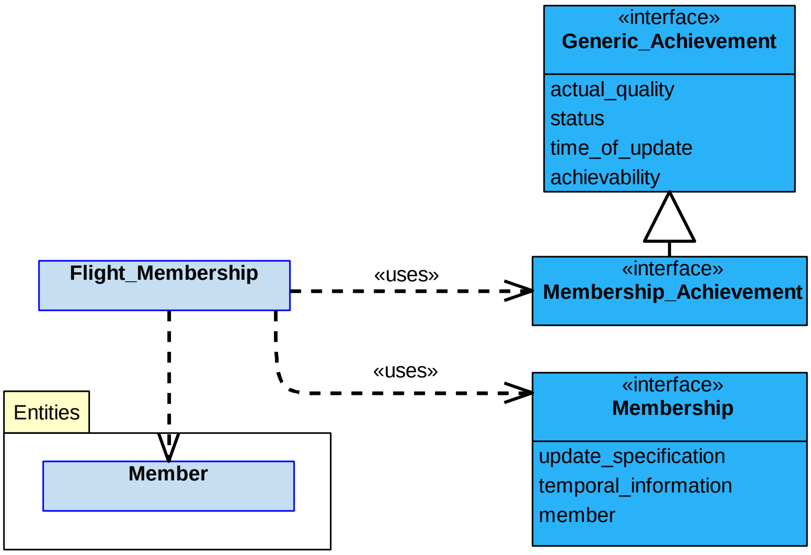


Figure 352: Flight_Membership Service Definition

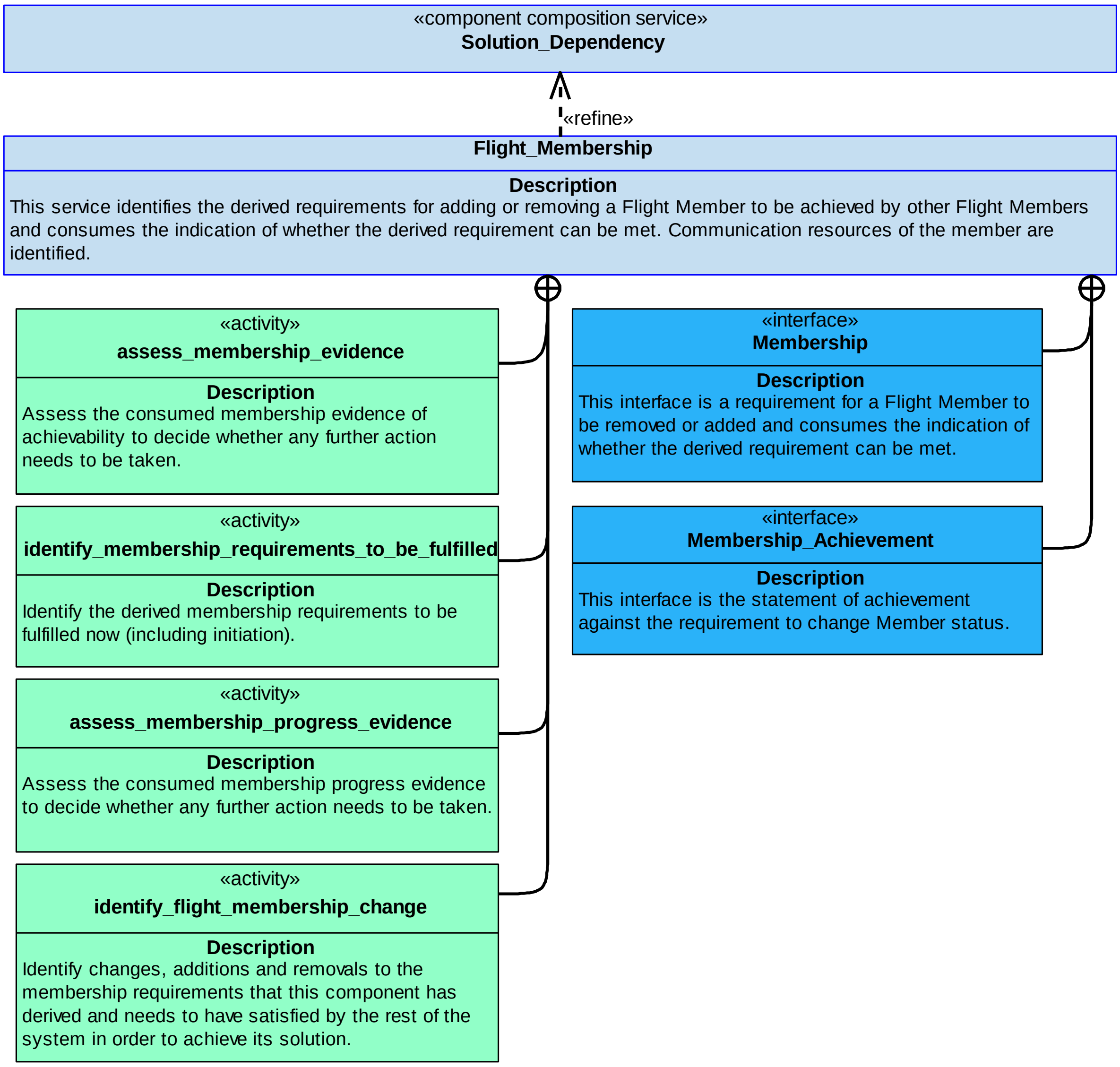

«component composition service»
Solution_Dependency
«refine»
Flight_Membership
Description
This service identifies the derived requirements for adding or removing a Flight Member to be achieved by other Flight Members and consumes the indication of whether the derived requirement can be met. Communication resources of the member are identified.
«activity»
assess_membership_evidence
Description
Assess the consumed membership evidence of achievability to decide whether any further action needs to be taken.
«activity»
identify_membership_requirements_to_be_fulfilled
Description
Identify the derived membership requirements to be fulfilled now (including initiation).
«activity»
assess_membership_progress_evidence
Description
Assess the consumed membership progress evidence to decide whether any further action needs to be taken.
«activity»
identify_flight_membership_change
Description
Identify changes, additions and removals to the membership requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution.
«interface»
Membership
Description
This interface is a requirement for a Flight Member to be removed or added and consumes the indication of whether the derived requirement can be met.
«interface»
Membership_Achievement
Description
This interface is the statement of achievement against the requirement to change Member status.

Figure 353: Flight_Membership Service Policy

Flight_Membership

This service identifies the derived requirements for adding or removing a Flight Member to be achieved by other Flight Members and consumes the indication of whether the derived requirement can be met. Communication resources of the member are identified.

Interfaces

Membership

This interface is a requirement for a Flight Member to be removed or added and consumes the indication of whether the derived requirement can be met.

Attributes

update_specification	The definition of the requirement to update the Flight Members.
temporal_information	Information covering timing, such as start and end times.

member	The Flight Member.

Membership_Achievement

This interface is the statement of achievement against the requirement to change Member status.

Activities

assess_membership_evidence

Assess the consumed membership evidence of achievability to decide whether any further action needs to be taken.

assess_membership_progress_evidence

Assess the consumed membership progress evidence to decide whether any further action needs to be taken.

identify_flight_membership_change

Identify changes, additions and removals to the membership requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution.

identify_membership_requirements_to_be_fulfilled

Identify the derived membership requirements to be fulfilled now (including initiation).

5.4.2.20.7.1.5 Information

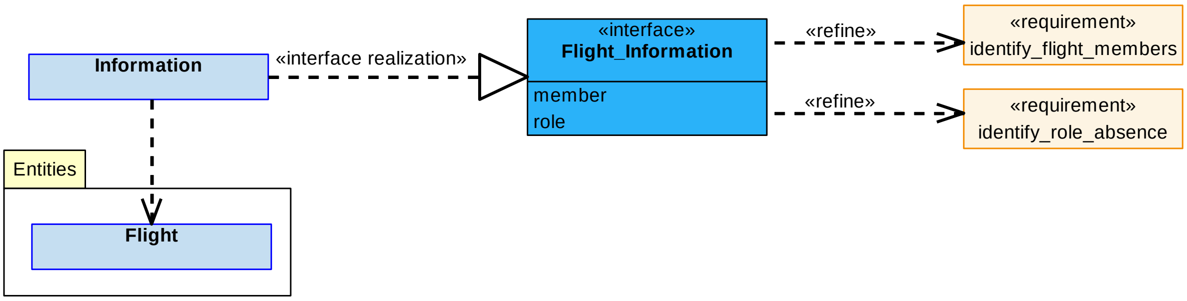


Figure 354: Information Service Definition

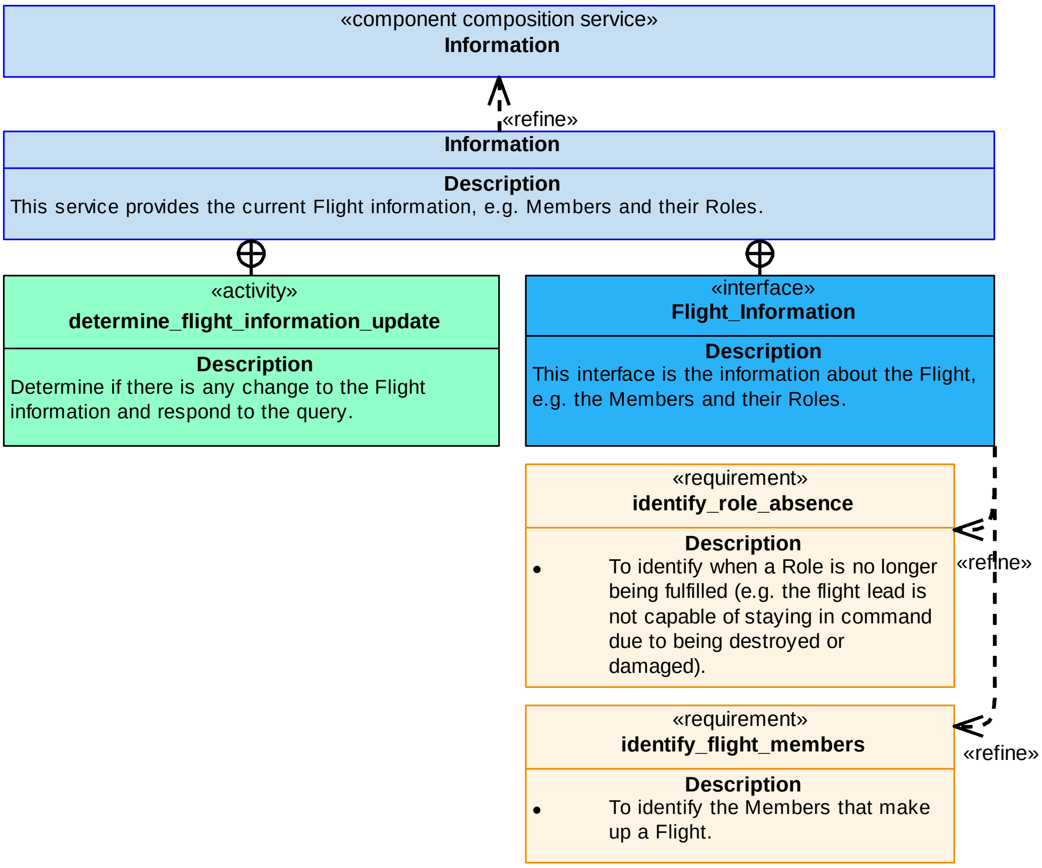

«component composition service»
Information
«refine»
Information
Description
This service provides the current Flight information, e.g. Members and their Roles.
«activity»
determine_flight_information_update
Description
Determine if there is any change to the Flight information and respond to the query.
«interface»
Flight_Information
Description
This interface is the information about the Flight, e.g. the Members and their Roles.
«requirement»
identify_role_absence
Description
To identify when a Role is no longer being fulfilled (e.g. the flight lead is not capable of staying in command due to being destroyed or damaged).
«refine»
«requirement»
identify_flight_members
Description
To identify the Members that make up a Flight.
«refine»

Figure 355: Information Service Policy

Information

This service provides the current Flight information, e.g. Members and their Roles.

Interface

Flight_Information

This interface is the information about the Flight, e.g. the Members and their Roles.

Attributes

member	The Flight Member.
role	The current role of a flight Member in relation to a particular Flight.

Activity

determine_flight_information_update

Determine if there is any change to the Flight information and respond to the query.

5.4.2.20.7.1.6 Information_Dependency

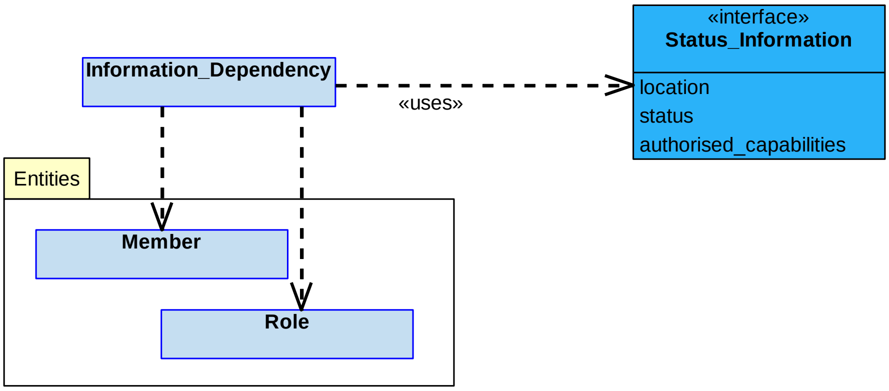


Figure 356: Information_Dependency Service Definition

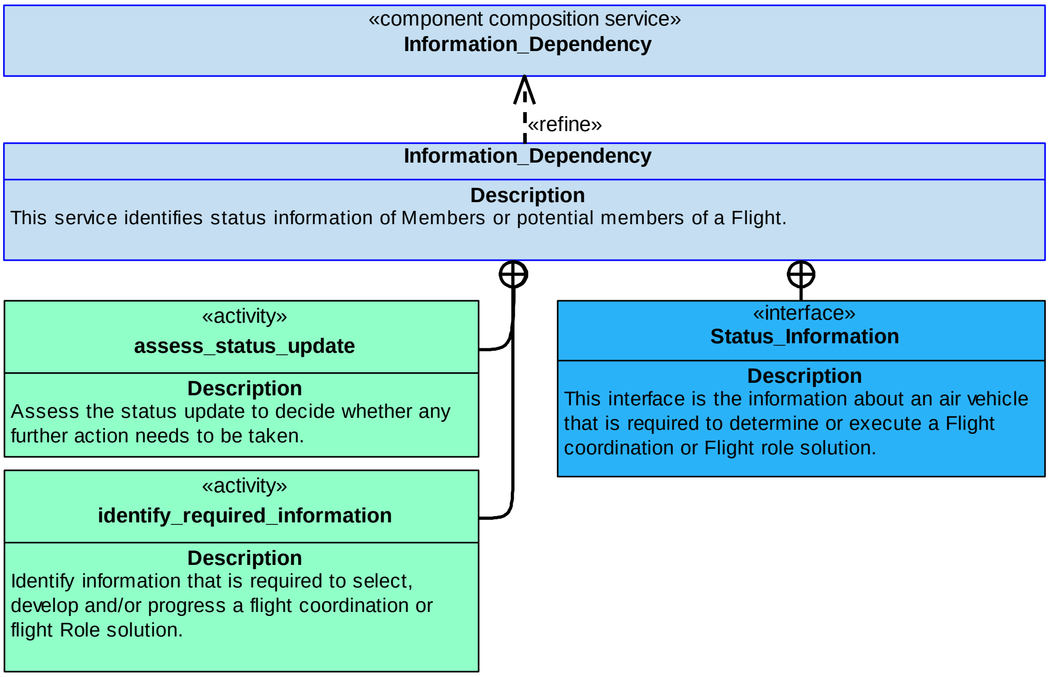

«component composition service»
Information_Dependency
«refine»
Information_Dependency
Description
This service identifies status information of Members or potential members of a Flight.
«activity»
assess_status_update
Description
Assess the status update to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify information that is required to select, develop and/or progress a flight coordination or flight Role solution.
«interface»
Status_Information
Description
This interface is the information about an air vehicle that is required to determine or execute a Flight coordination or Flight role solution.

Figure 357: Information_Dependency Service Policy

Information_Dependency

This service identifies status information of Members or potential members of a Flight.

Interface

Status_Information

This interface is the information about an air vehicle that is required to determine or execute a Flight coordination or Flight role solution.

Attributes

location	The spatial location a Member or potential member of a Flight.

status	The status of a Member or potential member of a Flight, e.g. combat status.
authorised_capabilities	The allowable capability of an air vehicle in terms of its contribution to a Flight, e.g. combat or operator ability.

Activities

assess_status_update

Assess the status update to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to select, develop and/or progress a flight coordination or flight Role solution.

5.4.2.20.7.1.7 Capability

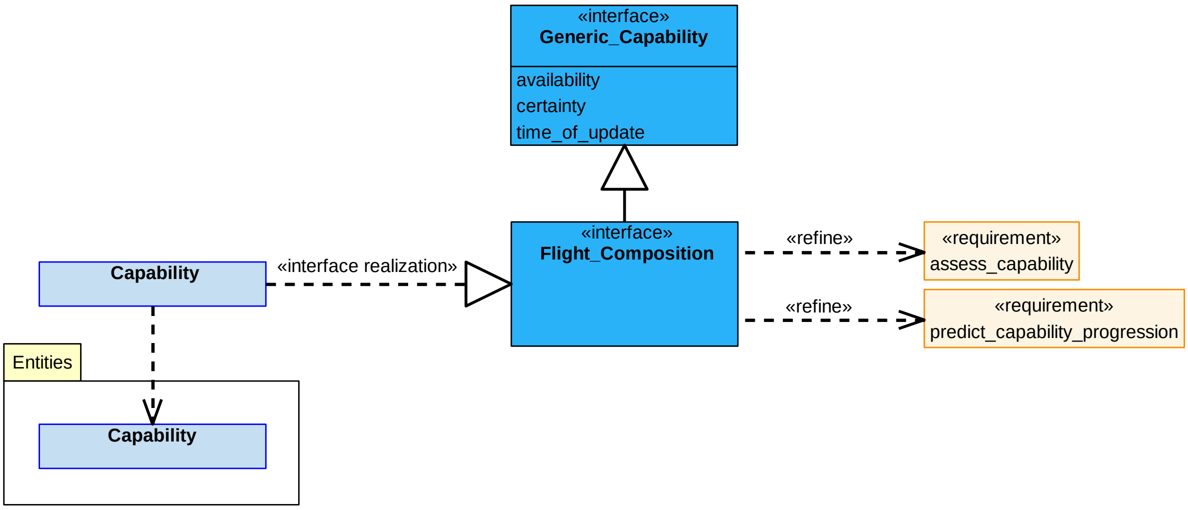


Figure 358: Capability Service Definition

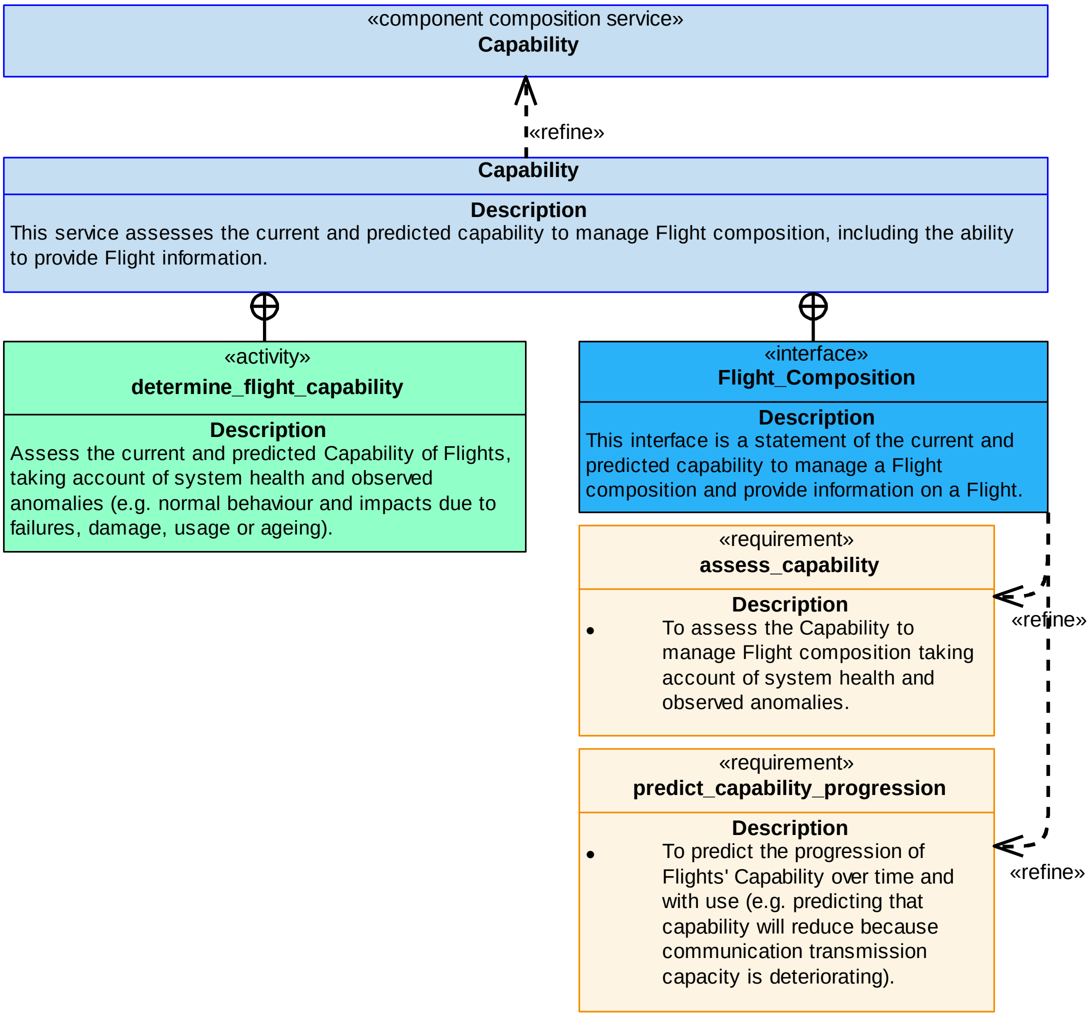

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability to manage Flight composition, including the ability to provide Flight information.
«activity»
determine_flight_capability
Description
Assess the current and predicted Capability of Flights, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Flight_Composition
Description
This interface is a statement of the current and predicted capability to manage a Flight composition and provide information on a Flight.
«requirement»
assess_capability
Description
• To assess the Capability to manage Flight composition taking account of system health and observed anomalies.
«refine»
«requirement»
predict_capability_progression
Description
• To predict the progression of Flights' Capability over time and with use (e.g. predicting that capability will reduce because communication transmission capacity is deteriorating).
«refine»

Figure 359: Capability Service Policy

Capability

This service assesses the current and predicted capability to manage Flight composition, including the ability to provide Flight information.

Interface

Flight_Composition

This interface is a statement of the current and predicted capability to manage a Flight composition and provide information on a Flight.

Activity

determine_flight_capability

Assess the current and predicted Capability of Flights, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.20.7.1.8 Capability_Evidence

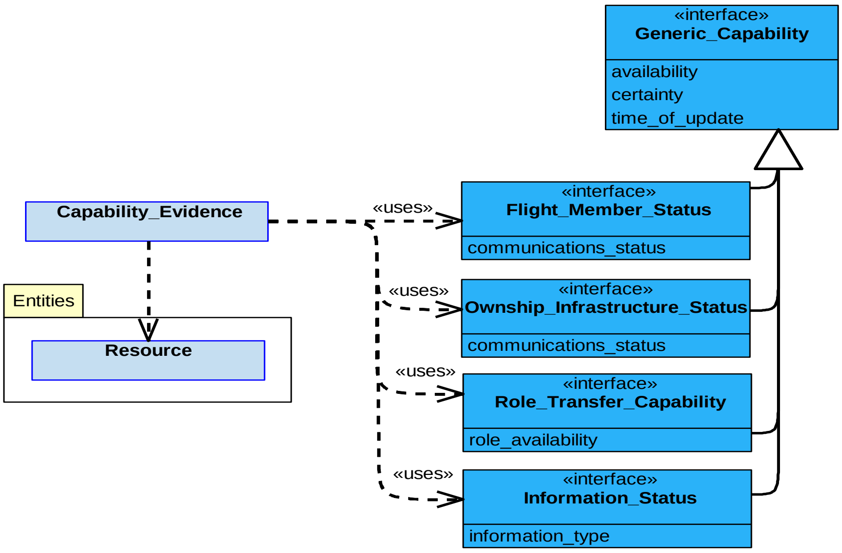
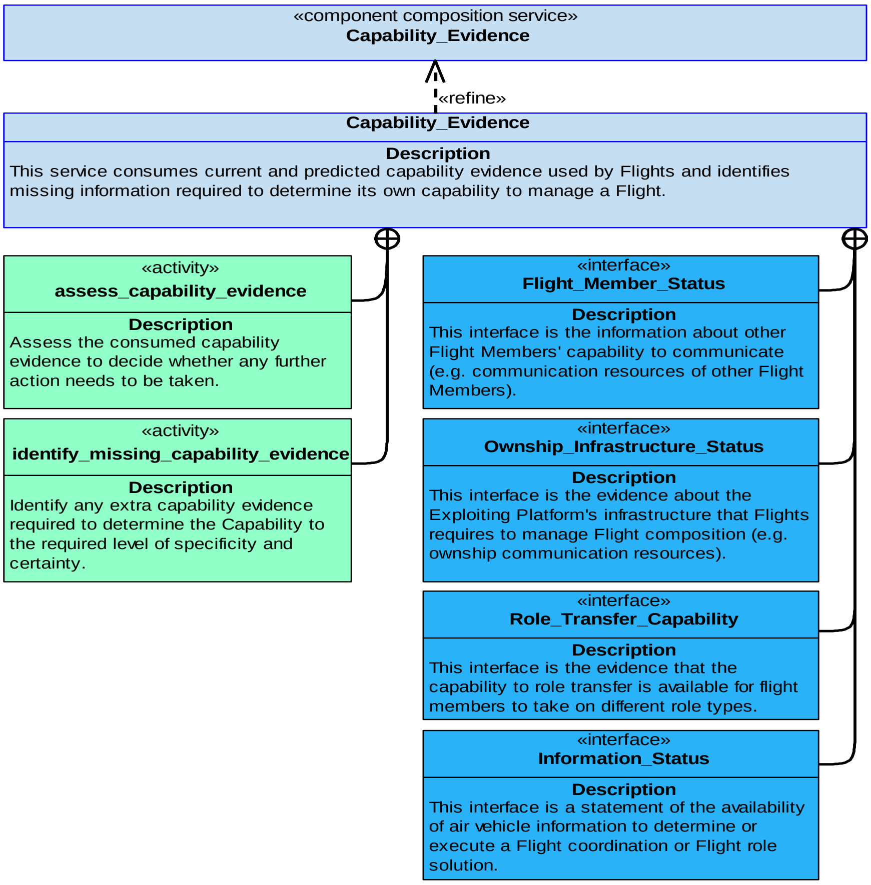


Figure 360: Capability_Evidence Service Definition



This service consumes current and predicted capability evidence used by Flights and identifies missing information required to determine its own capability to manage a Flight.

Assess the consumed capability evidence to decide whether any further action needs to be taken.

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

This interface is the information about other Flight Members' capability to communicate (e.g. communication resources of other Flight Members).

This interface is the evidence about the Exploiting Platform's infrastructure that Flights requires to manage Flight composition (e.g. ownship communication resources).

This interface is the evidence that the capability to role transfer is available for flight members to take on different role types.

This interface is a statement of the availability of air vehicle information to determine or execute a Flight coordination or Flight role solution.

Figure 361: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability evidence used by Flights and identifies missing information required to determine its own capability to manage a Flight.

Interfaces

Ownship_Infrastructure_Status

This interface is the evidence about the Exploiting Platform's infrastructure that Flights requires to manage Flight composition (e.g. ownship communication resources).

Attribute

communications_status	Status of communications resource (e.g. radio).

Flight_Member_Status

This interface is the information about other Flight Members' capability to communicate (e.g. communication resources of other Flight Members).

Attribute

communications_status	Status of communications resource (e.g. radio).

Role_Transfer_Capability

This interface is the evidence that the capability to role transfer is available for flight members to take on different role types.

Attribute

role_availability	Roles that flight members are capable of performing.

Information_Status

This interface is a statement of the availability of air vehicle information to determine or execute a Flight coordination or Flight role solution.

Attribute

information_type	The type of information available.

Activities

assess_capability_evidence

Assess the consumed capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.20.7.2 Service Dependencies

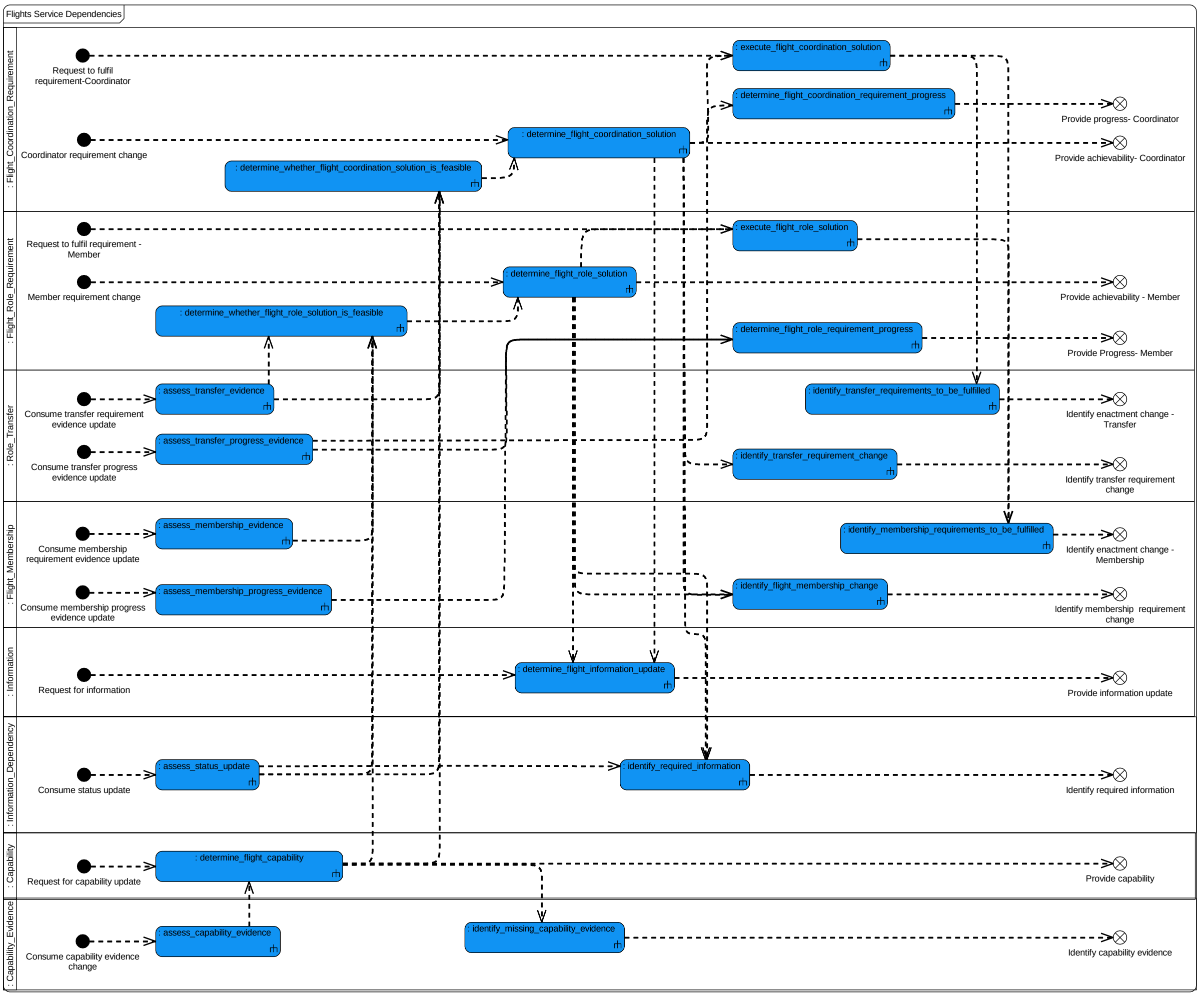

Flights Service Dependencies
: Flight_Coordination_Requirement
Request to fulfil requirement-Coordinator
Coordinator requirement change
: determine_whether_flight_coordination_solution_is_feasible
: determine_flight_coordination_solution
: execute_flight_coordination_solution
: determine_flight_coordination_requirement_progress
Provide progress- Coordinator
Provide achievability- Coordinator
: Flight_Role_Requirement
Request to fulfil requirement - Member
Member requirement change
: determine_whether_flight_role_solution_is_feasible
: determine_flight_role_solution
: execute_flight_role_solution
: determine_flight_role_requirement_progress
Provide achievability - Member
Provide Progress- Member
: Role_Transfer
Consume transfer requirement evidence update
Consume transfer progress evidence update
: assess_transfer_evidence
: assess_transfer_progress_evidence
: identify_transfer_requirements_to_be_fulfilled
: identify_transfer_requirement_change
Identify enactment change - Transfer
Identify transfer requirement change
: Flight_Membership
Consume membership requirement evidence update
Consume membership progress evidence update
: assess_membership_evidence
: assess_membership_progress_evidence
: identify_membership_requirements_to_be_fulfilled
: identify_flight_membership_change
Identify enactment change - Membership
Identify membership requirement change
: Information
Request for information
: determine_flight_information_update
Provide information update
: Information_Dependency
Consume status update
: assess_status_update
: identify_required_information
Identify required information
: Capability
Request for capability update
: determine_flight_capability
Provide capability
: Capability_Evidence
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 362: Flights Service Dependencies

5.4.2.21 Fluids

5.4.2.21.1 Role

The role of Fluids is to manage the storage and transfer of fluids.

5.4.2.21.2 Overview

Control Architecture

Fluids is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Following the reception of a Fluid_Management_Requirement, the Fluids component determines the optimum solution for how to distribute contents between Containers via Distribution_Paths, and coordinates the subsequent Fluid_Management_Solution by means of Distribution_Mechanisms (e.g. valves and pumps).

Examples of Use

Fluids will be used where:

- Fuel requires distributing in order to achieve propulsion requirements.
- Fluid requires distributing around the system for Exploiting Platform balance purposes.
- Fluid requires dumping to achieve a reduction in Exploiting Platform weight.
- Fluids require agitation or mixing.

5.4.2.21.3 Service Summary

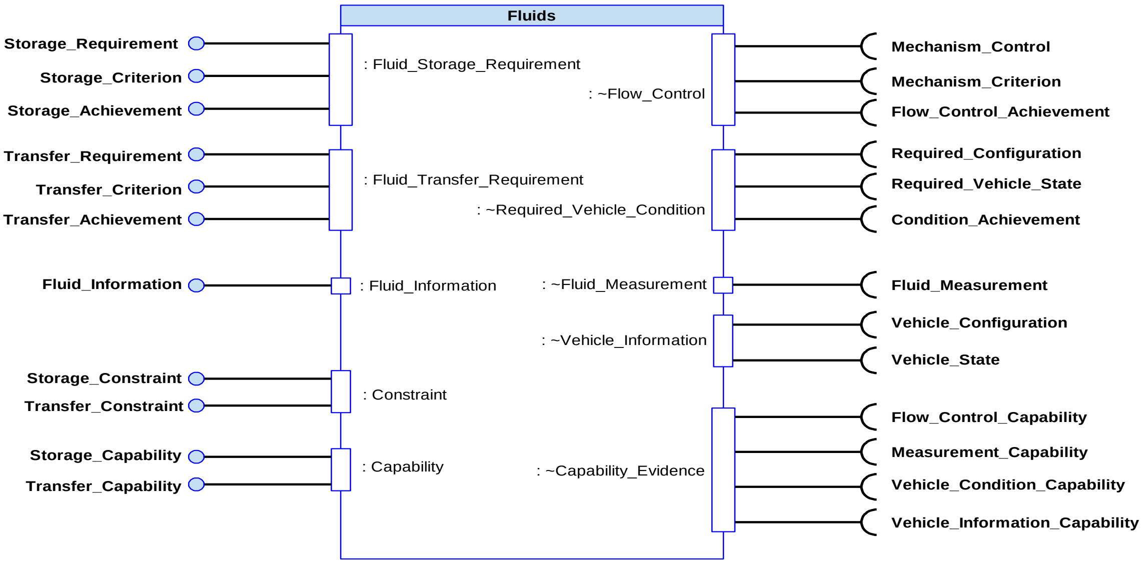


Figure 363: Fluids Service Summary

5.4.2.21.4 Responsibilities

capture_transfer_requirements

- To capture Transfer_Requirements to transfer fluid.

capture_storage_requirements

- To capture Storage_Requirements to store fluid.

capture_fluid_management_criteria

- To capture provided measurement criteria for fluid management.

capture_constraints

- To capture Constraints related to fluid storage and transfer.

update_achievability

- To identify whether a Fluid_Management_Requirement is still achievable given current or predicted Capability and Constraints.

determine_transfer_solution

- To determine a fluid transfer solution.

determine_storage_solution

- To determine a fluid storage solution.

determine_predicted_fluid_management_quality

- To determine the predicted quality of a proposed Fluid_Management_Solution against given measurement criteria.

identify_distribution_pre-conditions

- To identify Vehicle_States required to support fluid distribution.

coordinate_fluid_storage

- To coordinate a solution for the storage of fluid.

coordinate_fluid_transfer

- To coordinate a solution for the transfer of fluid.

identify_fluid_distribution_progress

- To identify the progress of a fluid distribution solution against a Fluid_Management_Requirement.

determine_actual_fluid_management_quality

- To determine the actual quality of a proposed Fluid_Management_Solution against given measurement criteria.

monitor_fluid_properties

- To monitor the properties of fluid (e.g. quantity or temperature).

assess_fluid_management_capability

- To assess the Capability to store and transfer fluid, taking account of system health and anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of Capability over time and with use.

5.4.2.21.5 Subject Matter Semantics

The subject matter of Fluids is the management of fluids, encompassing the Containers that can be used for fluid storage purposes, and the Distribution_Paths and mechanisms that can be used to distribute fluids.

Exclusions

The subject matter of Fluids does not include:

- The physical operation of valves and pumps.
- Taking direct measurements of the attributes of Container contents (e.g. fuel level).
- Determining what can be achieved with the contents of the Container (e.g. the range of an air vehicle based on the amount of fuel remaining), only the existence and distribution is in scope.

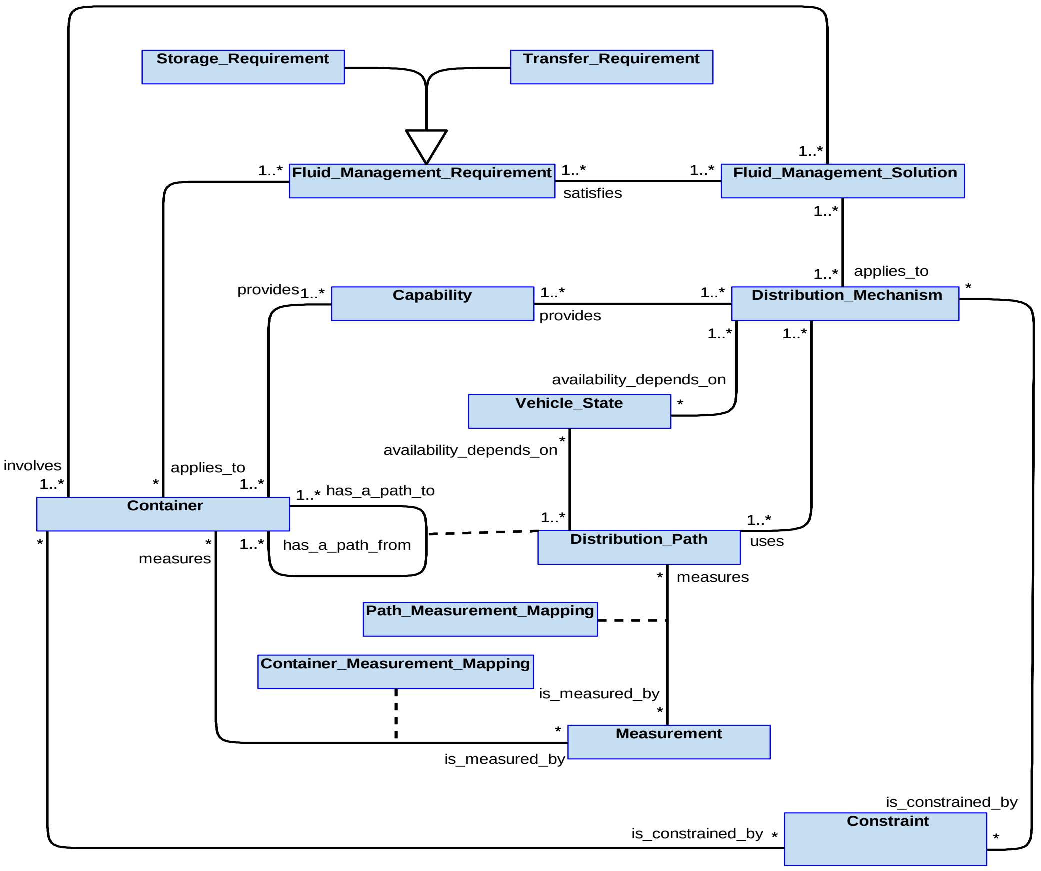


Figure 364: Fluids Semantics

5.4.2.21.5.1 Entities

Capability

The range of ways to move and store fluid. This will depend on the fluid system's physical infrastructure, material properties and health.

Constraint

A constraint on fluid storage and the mechanisms by which fluids are transferred (e.g. limitations of valves or pumps).

Container

A space with an interface that fluid can travel across. This includes storage receptacles, and also the spaces that they connect to even where they are not the responsibility of this component, such as an engine, another vehicle involved in refuelling, or empty space around the Exploiting Platform.

Container_Measurement_Mapping

The mapping of how Measurements relate to the properties of a Container's fluid contents.

Distribution_Mechanism

A mechanism by which fluids are distributed between Containers via Distribution_Paths (e.g. valves or pumps), including the capabilities of the transfer device.

Distribution_Path

A physical connection between Containers that fluids can move along.

Fluid_Management_Solution

A sequence of steps that together provide a solution for the storage or transfer of fluid.

Fluid_Management_Requirement

A requirement to manage storage and transfer of fluids.

Measurement

A measurement from which information on a fluid can be derived, such as a flow rate or a fluid level measurement.

Path_Measurement_Mapping

The mapping of how Measurements relate to the properties of fluid in Distribution_Paths.

Storage_Requirement

A requirement to store fluid in a certain way, including any required distribution or monitoring of the fluid while in storage (e.g. a requirement to ensure a feed tank is kept 80% full).

Transfer_Requirement

A requirement to transfer fluid (e.g. to transfer fuel from a tank Container to the fluid interface of an engine Container or to vent water to the atmosphere).

Vehicle_State

A state or configuration of the vehicle that affects whether certain mechanisms or paths can be used in certain ways for fluid management (e.g. interlock state or regime conditions).

5.4.2.21.6 Design Rationale

5.4.2.21.6.1 Assumptions

None.

5.4.2.21.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Fluids:

- Data Driving - As the quantity and type of Containers are expected to vary between Exploiting Platforms, the use of data driving in this component should be considered.

Extensions

- The use of extension components for Fluids may be appropriate to cater for varying content types associated with a particular Exploiting Platform.

Exploitation Considerations

- An exploitation may wish to include multiple instances of Fluids to cater for the management of different types of contents (e.g. fuel, coolant or oxygen).
- As an alternative to taking flow Measurements, the level in a Container can be used to infer the flow between Containers. Similarly, as an alternative to taking level Measurements, fluid level in a Container can be determined from flow Measurements.

5.4.2.21.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component could cause an out of balance condition if Container contents were transferred incorrectly, resulting in uncontrolled flight. This could lead to loss of structural integrity of the Exploiting Platform and/or an uncontrolled crash. The result is likely to be loss of the Exploiting Platform and fatalities.
- Additionally, this component could cause a loss of thrust if the amount of fuel available in fuel Containers is incorrectly reported. This could also be catastrophic. However, in many cases the Exploiting Platform would still be controllable, and so fatalities may be avoided (e.g. air vehicle crashes in location clear of people and/or crew eject).

5.4.2.21.6.4 Security Considerations

The indicative security classification is O-S.

This component is responsible for managing Container content and its transfer, which without knowledge of other performance data is considered O-S. Where level of use of a fluid (such as fuel) may indicate a level of performance, this may lead to greater controls on confidentiality. If the integrity of demands for, and availability of fluids is compromised, the combat effectiveness of the Exploiting Platform may be reduced, e.g. through loss fuel to engines. The component is considered a legitimate target for cyber attack and due to the risk to integrity and availability, appropriate protection is required. This is one of a series of components that will assist in identifying if form and fit integrity has been interfered with.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Information** relating to possible tamper events.
- **Maintaining Audit Records** to support accountability of fluid usage.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** of Container contents and its transfer, and notifying of available fuel or other fluids.
- Providing **Warnings and Notifications** of fluid leakage, etc.

The component is involved in satisfying security enforcing functions relating to:

- **Detecting Security Breaches** through identifying fluid states that may indicate the physical security has been compromised (e.g. an unauthorised item is attached to the Exploiting Platform that is diverting fluids for its own use).

5.4.2.21.7 Services

5.4.2.21.7.1 Service Definitions

5.4.2.21.7.1.1 Fluid_Storage_Requirement

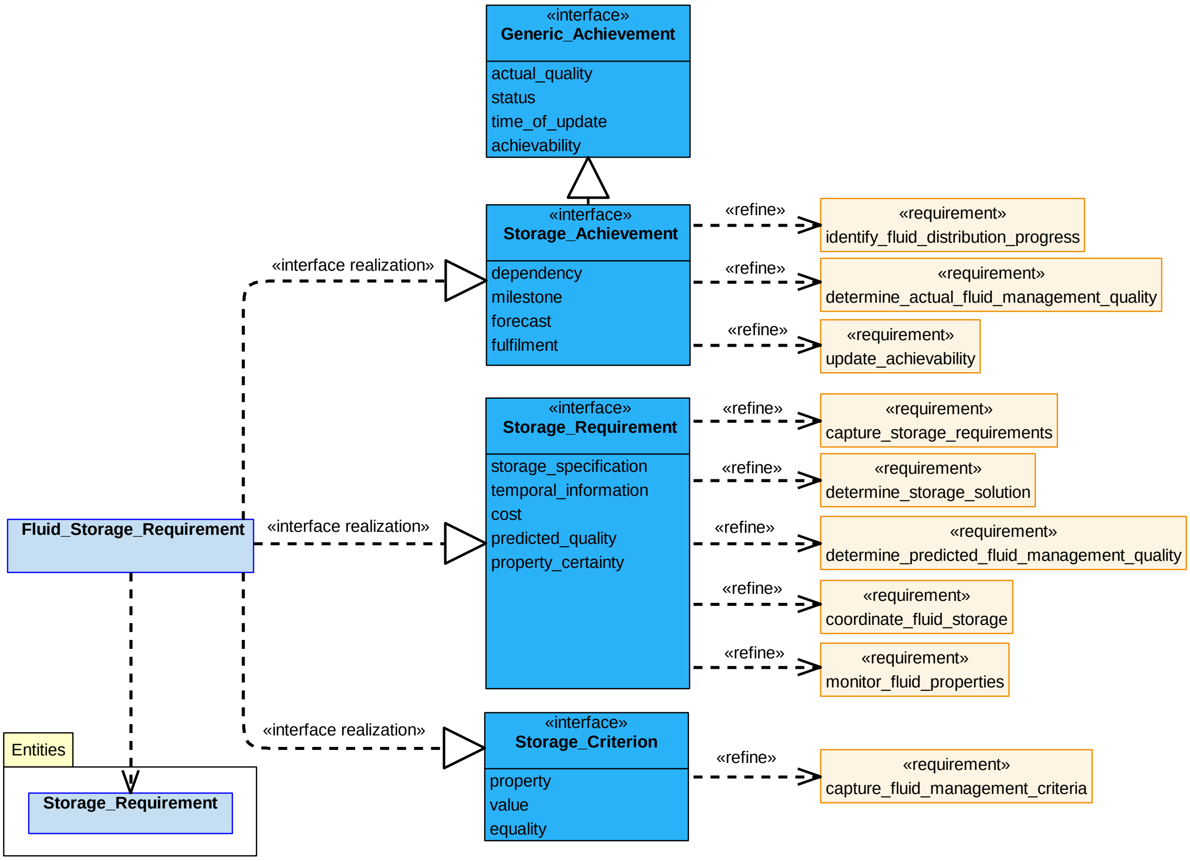


Figure 365: Fluid_Storage_Requirement Service Definition

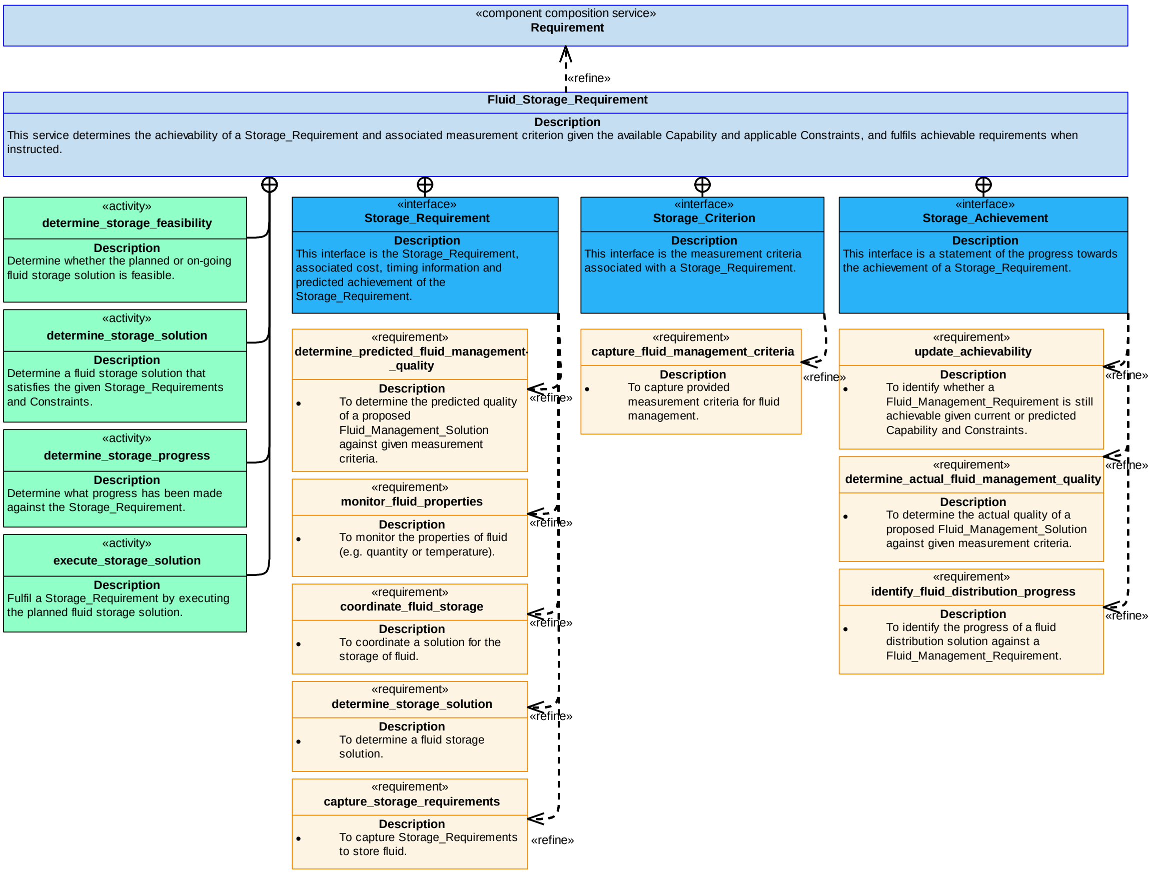

«component composition service»
Requirement
«refine»
Fluid_Storage_Requirement
Description
This service determines the achievability of a Storage_Requirement and associated measurement criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_storage_feasibility
Description
Determine whether the planned or on-going fluid storage solution is feasible.
«activity»
determine_storage_solution
Description
Determine a fluid storage solution that satisfies the given Storage_Requirements and Constraints.
«activity»
determine_storage_progress
Description
Determine what progress has been made against the Storage_Requirement.
«activity»
execute_storage_solution
Description
Fulfil a Storage_Requirement by executing the planned fluid storage solution.
«interface»
Storage_Requirement
Description
This interface is the Storage_Requirement, associated cost, timing information and predicted achievement of the Storage_Requirement.
«interface»
Storage_Criterion
Description
This interface is the measurement criteria associated with a Storage_Requirement.
«interface»
Storage_Achievement
Description
This interface is a statement of the progress towards the achievement of a Storage_Requirement.
«requirement»
determine_predicted_fluid_management_quality
Description
To determine the predicted quality of a proposed Fluid_Management_Solution against given measurement criteria.
«refine»
«requirement»
monitor_fluid_properties
Description
To monitor the properties of fluid (e.g. quantity or temperature).
«refine»
«requirement»
coordinate_fluid_storage
Description
To coordinate a solution for the storage of fluid.
«refine»
«requirement»
determine_storage_solution
Description
To determine a fluid storage solution.
«refine»
«requirement»
capture_storage_requirements
Description
To capture Storage_Requirements to store fluid.
«refine»
«requirement»
capture_fluid_management_criteria
Description
To capture provided measurement criteria for fluid management.
«refine»
«requirement»
update_achievability
Description
To identify whether a Fluid_Management_Requirement is still achievable given current or predicted Capability and Constraints.
«refine»
«requirement»
determine_actual_fluid_management_quality
Description
To determine the actual quality of a proposed Fluid_Management_Solution against given measurement criteria.
«refine»
«requirement»
identify_fluid_distribution_progress
Description
To identify the progress of a fluid distribution solution against a Fluid_Management_Requirement.
«refine»

Figure 366: Fluid_Storage_Requirement Service Policy

Fluid_Storage_Requirement

This service determines the achievability of a Storage_Requirement and associated measurement criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

<u>Interfaces</u>

Storage_Requirement

This interface is the Storage_Requirement, associated cost, timing information and predicted achievement of the Storage_Requirement.

<u>Attributes</u>

storage_specification	The definition of the Storage_Requirement. For example: a requirement to store fluid for weight balance purposes, or so that a specific property can be monitored.
temporal_information	Information covering timing of a solution to the Storage_Requirement, such as start and end times.
cost	The cost of executing a storage solution, such as the resources used or time taken.
predicted_quality	How well the planned solution is predicted to satisfy the Storage_Requirement, e.g. the predicted storage temperature compared with the specified temperature.

property_certainty	The certainty of the calculated properties.

Storage_Achievement

This interface is a statement of the progress towards the achievement of a Storage_Requirement.

Attributes

dependency	The reliance of a Storage_Requirement on the progress of fluid control requirements and fluid measurements.
milestone	A significant point in progress towards the Storage_Requirement, e.g. a Container has been filled, or a volume of fluid has been successfully stored.
forecast	A forecast of the remaining transfer work required (e.g. time to complete or volume still to distribute within storage).
fulfilment	The comparison of achievement against a storage forecast or requirement parameter (e.g. percentage complete in terms of time or mass).

Storage_Criterion

This interface is the measurement criteria associated with a Storage_Requirement.

Attributes

property	A property that is a measure of the quality or cost of a storage solution, such as the amount of fluid stored in the specified storage conditions, time taken to complete the storage solution, or fluid masses for balance considerations. Any measurable properties specified in the requirement should be criterion properties.
value	The measured value of the property, e.g. 100 litres, 1 bar, or 60 seconds.
equality	The relationship between the value and any limit on the property, e.g. less than, or equal to.

Activities

determine_storage_feasibility

Determine whether the planned or on-going fluid storage solution is feasible.

determine_storage_solution

Determine a fluid storage solution that satisfies the given Storage_Requirements and Constraints.

execute_storage_solution

Fulfil a Storage_Requirement by executing the planned fluid storage solution.

determine_storage_progress

Determine what progress has been made against the Storage_Requirement.

5.4.2.21.7.1.2 Fluid_Transfer_Requirement

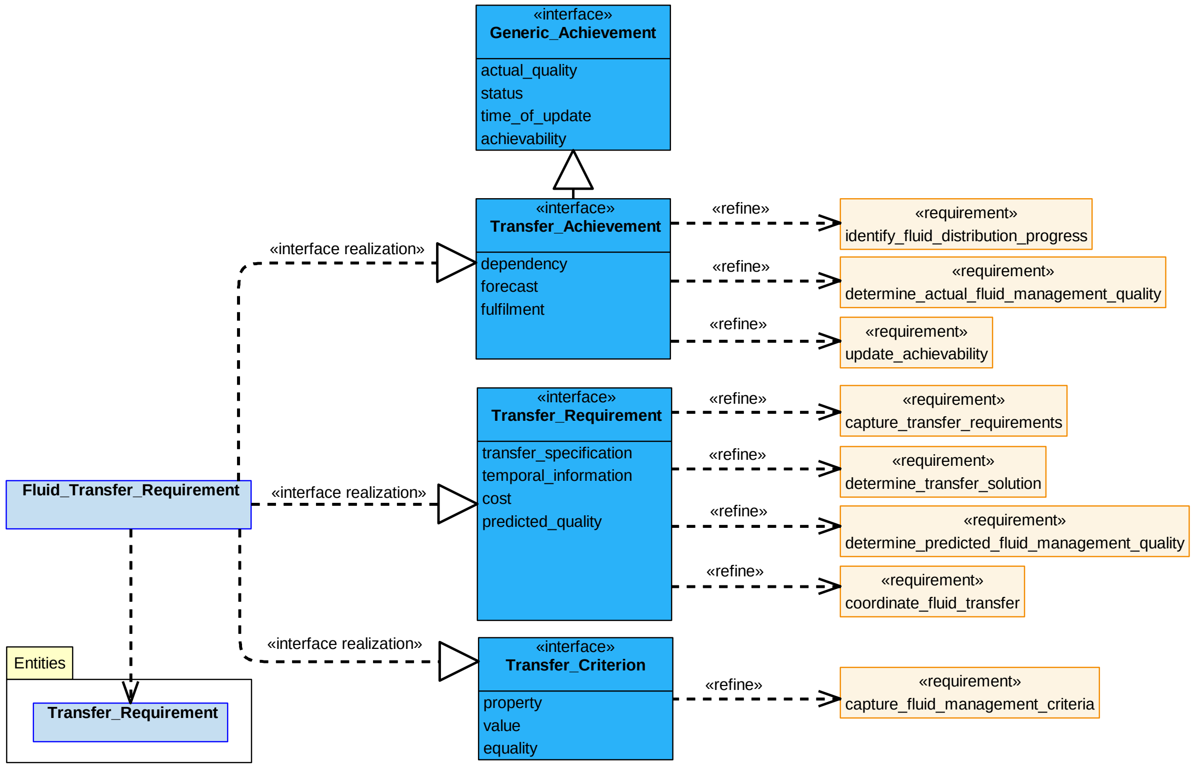


Figure 367: Fluid_Transfer_Requirement Service Definition

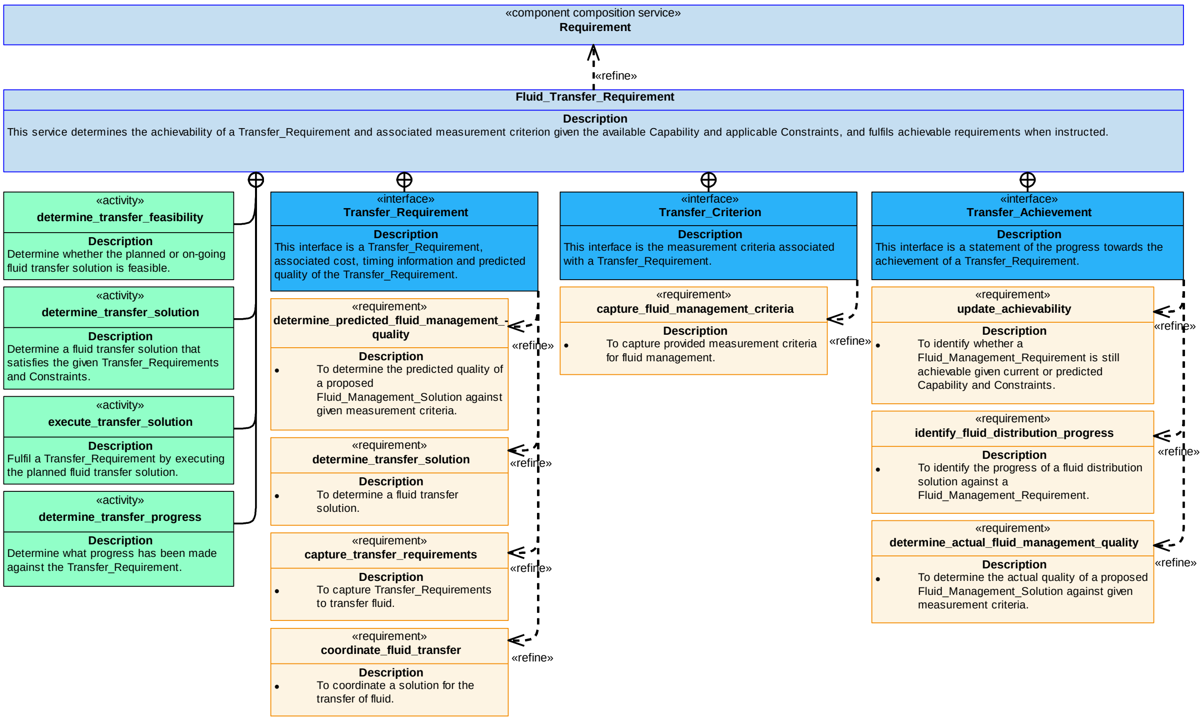

«component composition service»
Requirement
«refine»
Fluid_Transfer_Requirement
Description
This service determines the achievability of a Transfer_Requirement and associated measurement criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_transfer_feasibility
Description
Determine whether the planned or on-going fluid transfer solution is feasible.
«activity»
determine_transfer_solution
Description
Determine a fluid transfer solution that satisfies the given Transfer_Requirements and Constraints.
«activity»
execute_transfer_solution
Description
Fulfil a Transfer_Requirement by executing the planned fluid transfer solution.
«activity»
determine_transfer_progress
Description
Determine what progress has been made against the Transfer_Requirement.
«interface»
Transfer_Requirement
Description
This interface is a Transfer_Requirement, associated cost, timing information and predicted quality of the Transfer_Requirement.
«requirement»
determine_predicted_fluid_management_-quality
Description
To determine the predicted quality of a proposed Fluid_Management_Solution against given measurement criteria.
«refine»
«requirement»
determine_transfer_solution
Description
To determine a fluid transfer solution.
«refine»
«requirement»
capture_transfer_requirements
Description
To capture Transfer_Requirements to transfer fluid.
«refine»
«requirement»
coordinate_fluid_transfer
Description
To coordinate a solution for the transfer of fluid.
«refine»
«interface»
Transfer_Criterion
Description
This interface is the measurement criteria associated with a Transfer_Requirement.
«requirement»
capture_fluid_management_criteria
Description
To capture provided measurement criteria for fluid management.
«refine»
«interface»
Transfer_Achievement
Description
This interface is a statement of the progress towards the achievement of a Transfer_Requirement.
«requirement»
update_achievability
Description
To identify whether a Fluid_Management_Requirement is still achievable given current or predicted Capability and Constraints.
«refine»
«requirement»
identify_fluid_distribution_progress
Description
To identify the progress of a fluid distribution solution against a Fluid_Management_Requirement.
«refine»
«requirement»
determine_actual_fluid_management_quality
Description
To determine the actual quality of a proposed Fluid_Management_Solution against given measurement criteria.
«refine»

Figure 368: Fluid_Transfer_Requirement Service Policy

Fluid_Transfer_Requirement

This service determines the achievability of a Transfer_Requirement and associated measurement criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Transfer_Requirement

This interface is a Transfer_Requirement, associated cost, timing information and predicted quality of the Transfer_Requirement.

Attributes

transfer_specification	The definition of the Transfer_Requirement. For example: a requirement to transfer an amount of water for emission to the atmosphere; to supply fuel to an engine at a set flow rate; or to transfer fluid for heat or weight balance purposes.
temporal_information	Information covering timing of a solution to the Transfer_Requirement, such as start and end times.
cost	The cost of executing a transfer solution, such as the resources used or time taken.
predicted_quality	How well the planned solution is predicted to satisfy the Transfer_Requirement, e.g. the predicted amount of transferable fluid compared with the specified amount.

Transfer_Achievement

This interface is a statement of the progress towards the achievement of a Transfer_Requirement.

Attributes

dependency	The reliance of a transfer requirement on the progress of fluid control requirements and fluid measurements.
forecast	A forecast of the remaining transfer work required (e.g. time to complete or volume still to transfer).
fulfilment	The comparison of achievement against a transfer forecast or requirement parameter (e.g. percentage complete in terms of time or volume).

Transfer_Criterion

This interface is the measurement criteria associated with a Transfer_Requirement.

Attributes

property	A property that is a measure of the quality or cost of a transfer solution, such as the amount of fluid successfully transferred, or the time taken to complete the transfer solution. Any measurable properties specified in the requirement should be criterion properties.
value	The measured value of the property, e.g. 1 litre per second, 50 litres, or 60 seconds.
equality	The relationship between the value and any limit on the property, e.g. less than, or equal to.

Activities

determine_transfer_feasibility

Determine whether the planned or on-going fluid transfer solution is feasible.

determine_transfer_solution

Determine a fluid transfer solution that satisfies the given Transfer_Requirements and Constraints.

execute_transfer_solution

Fulfil a Transfer_Requirement by executing the planned fluid transfer solution.

determine_transfer_progress

Determine what progress has been made against the Transfer_Requirement.

5.4.2.21.7.1.3 Flow_Control

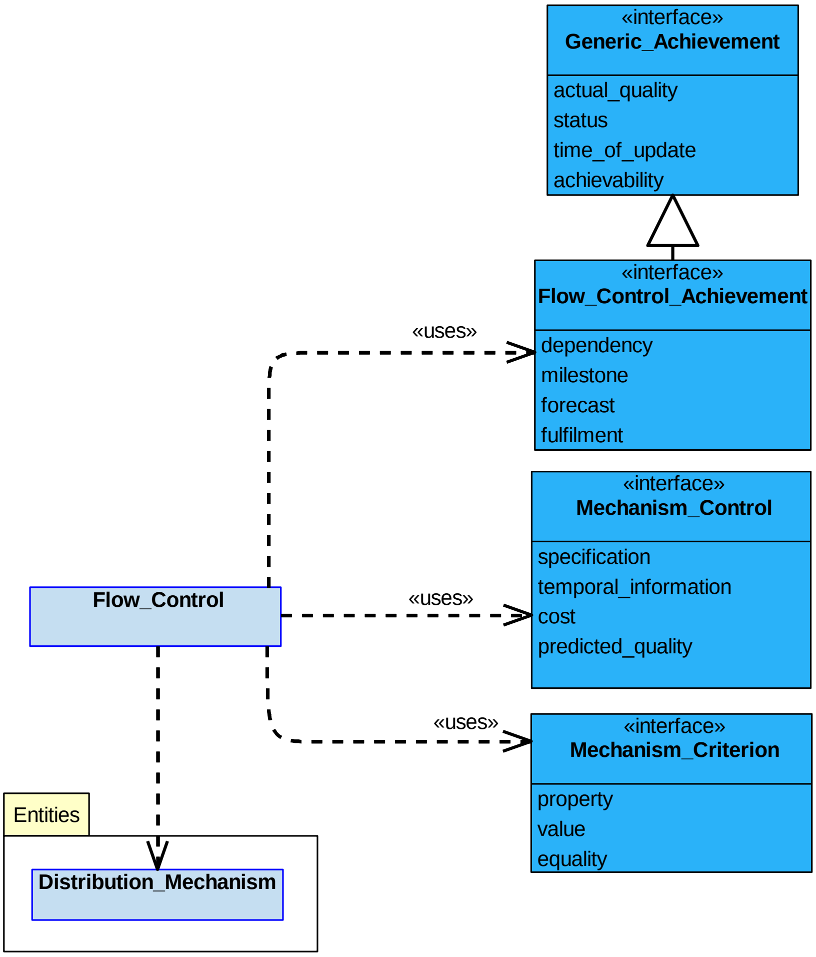


Figure 369: Flow_Control Service Definition

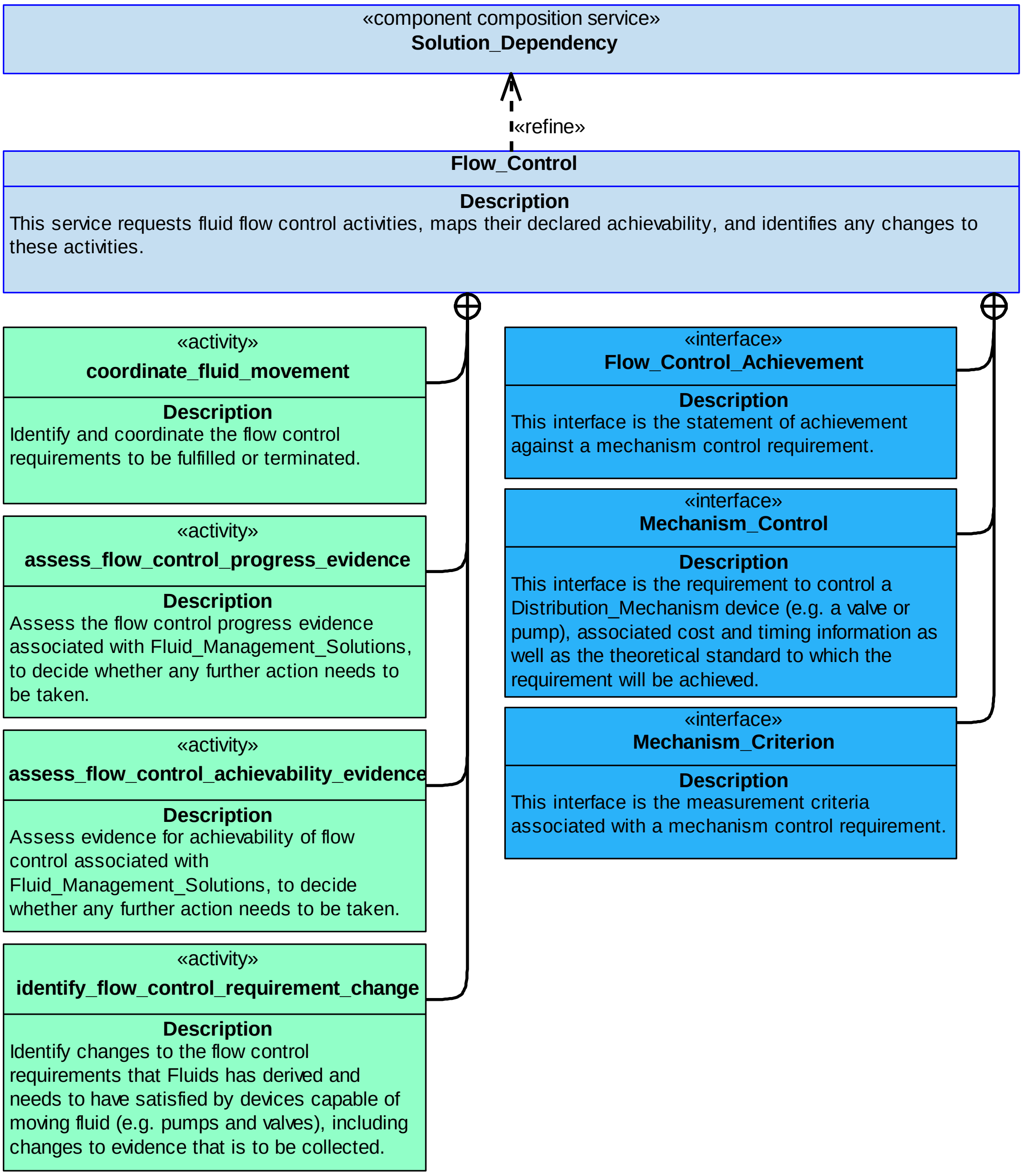

«component composition service»
Solution_Dependency
«refine»
Flow_Control
Description
This service requests fluid flow control activities, maps their declared achievability, and identifies any changes to these activities.
«activity»
coordinate_fluid_movement
Description
Identify and coordinate the flow control requirements to be fulfilled or terminated.
«activity»
assess_flow_control_progress_evidence
Description
Assess the flow control progress evidence associated with Fluid_Management_Solutions, to decide whether any further action needs to be taken.
«activity»
assess_flow_control_achievability_evidence
Description
Assess evidence for achievability of flow control associated with Fluid_Management_Solutions, to decide whether any further action needs to be taken.
«activity»
identify_flow_control_requirement_change
Description
Identify changes to the flow control requirements that Fluids has derived and needs to have satisfied by devices capable of moving fluid (e.g. pumps and valves), including changes to evidence that is to be collected.
«interface»
Flow_Control_Achievement
Description
This interface is the statement of achievement against a mechanism control requirement.
«interface»
Mechanism_Control
Description
This interface is the requirement to control a Distribution_Mechanism device (e.g. a valve or pump), associated cost and timing information as well as the theoretical standard to which the requirement will be achieved.
«interface»
Mechanism_Criterion
Description
This interface is the measurement criteria associated with a mechanism control requirement.

Figure 370: Flow_Control Service Policy

Flow_Control

This service requests fluid flow control activities, maps their declared achievability, and identifies any changes to these activities.

Interfaces

Mechanism_Control

This interface is the requirement to control a Distribution_Mechanism device (e.g. a valve or pump), associated cost and timing information as well as the theoretical standard to which the requirement will be achieved.

Attributes

specification	The definition of the mechanism control requirement, e.g. fully open valve or turn off pump.
temporal_information	Information covering mechanism control timings, such as start and end times.
cost	The cost of executing a mechanism control solution, such as the resources used or time taken.
predicted_quality	How well the planned solution is predicted to satisfy the mechanism control requirement, e.g. the predicted effect of a pump on flow rate compared with the specified effect.

Flow_Control_Achievement

This interface is the statement of achievement against a mechanism control requirement.

Attributes

dependency	The reliance of a flow control requirements on the progress of mechanism control and fluid measurements.
milestone	A significant point in progress towards the Storage_Requirement, e.g. a valve has reached its movement limit, or a pump has reached full output.
forecast	A forecast of the remaining flow work required (e.g. time to complete or flow rate to reach).
fulfilment	The comparison of achievement against a flow control forecast or requirement parameter (e.g. percentage complete in terms of time or flow rate).

Mechanism_Criterion

This interface is the measurement criteria associated with a mechanism control requirement.

Attributes

property	A property that is a measure of the quality or cost of a flow control solution, such as a mechanism's effect on flow rate or pressure.
value	The measured value of the property, e.g. 1 litre per second flow increase or 50 metres pump pressure head.
equality	The relationship between the value and any limit on the property, e.g. less than, or equal to.

Activities

coordinate_fluid_movement

Identify and coordinate the flow control requirements to be fulfilled or terminated.

assess_flow_control_achievability_evidence

Assess evidence for achievability of flow control associated with Fluid_Management_Solutions, to decide whether any further action needs to be taken.

assess_flow_control_progress_evidence

Assess the flow control progress evidence associated with Fluid_Management_Solutions, to decide whether any further action needs to be taken.

identify_flow_control_requirement_change

Identify changes to the flow control requirements that Fluids has derived and needs to have satisfied by devices capable of moving fluid (e.g. pumps and valves), including changes to evidence that is to be collected.

5.4.2.21.7.1.4 Required_Vehicle_Condition

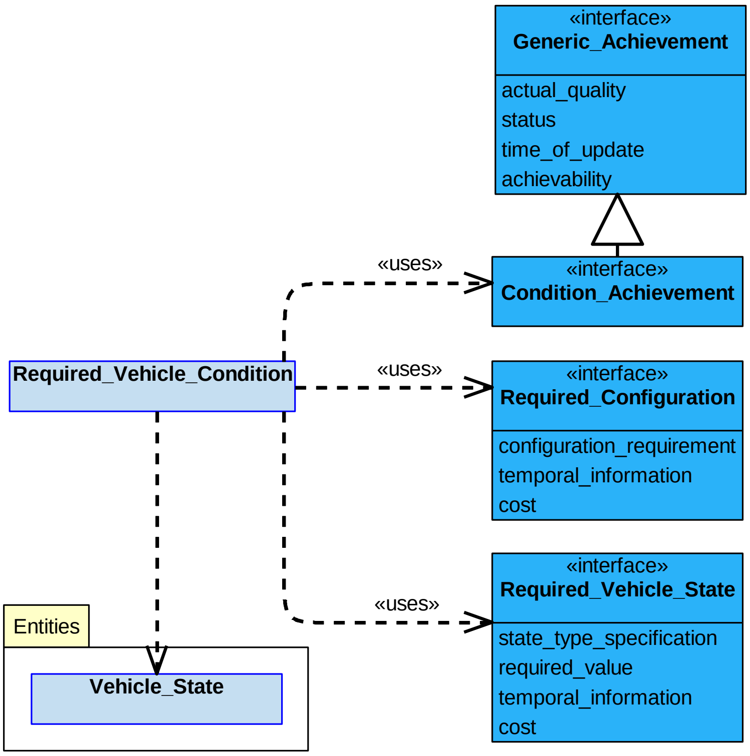
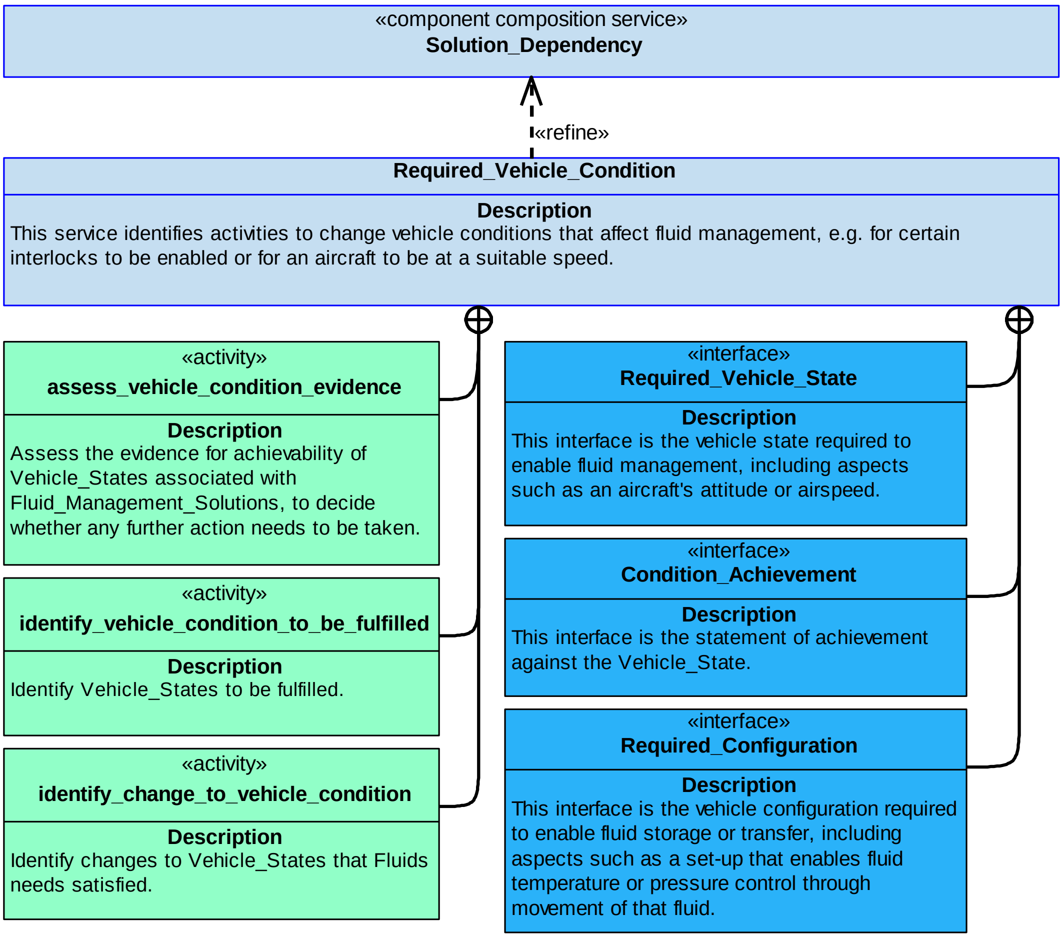


Figure 371: Required_Vehicle_Condition Service Definition



«component composition service»
Solution_Dependency
«refine»
Required_Vehicle_Condition
Description

This service identifies activities to change vehicle conditions that affect fluid management, e.g. for certain interlocks to be enabled or for an aircraft to be at a suitable speed.

«activity»
assess_vehicle_condition_evidence
Description

Assess the evidence for achievability of Vehicle_States associated with Fluid_Management_Solutions, to decide whether any further action needs to be taken.

«activity»
identify_vehicle_condition_to_be_fulfilled
Description

Identify Vehicle_States to be fulfilled.

Identify changes to Vehicle_States that Fluids needs satisfied.

This interface is the vehicle state required to enable fluid management, including aspects such as an aircraft's attitude or airspeed.

This interface is the statement of achievement against the Vehicle_State.

This interface is the vehicle configuration required to enable fluid storage or transfer, including aspects such as a set-up that enables fluid temperature or pressure control through movement of that fluid.

Figure 372: Required_Vehicle_Condition Service Policy

Required_Vehicle_Condition

This service identifies activities to change vehicle conditions that affect fluid management, e.g. for certain interlocks to be enabled or for an aircraft to be at a suitable speed.

Interfaces

Required_Configuration

This interface is the vehicle configuration required to enable fluid storage or transfer, including aspects such as a set-up that enables fluid temperature or pressure control through movement of that fluid.

Attributes

configuration_requirement	The specification of the vehicle configuration required to enable fluid distribution.
temporal_information	Information covering timing, such as when the required configuration should be enacted and for how long.
cost	The cost of fulfilling the derived configuration requirement to enable fluid distribution, such as the resources used or time taken.

Required_Vehicle_State

This interface is the vehicle state required to enable fluid management, including aspects such as an aircraft's attitude or airspeed.

Attributes

state_type_specification	The specification of the vehicle state required to enable fluid distribution.
required_value	The required value of the state property.
temporal_information	Information covering timing, such as when the state is required and for how long.
cost	The cost of fulfilling the derived vehicle state requirement to enable fluid distribution, such as the resources used or time taken.

Condition_Achievement

This interface is the statement of achievement against the Vehicle_State.

Activities

identify_vehicle_condition_to_be_fulfilled

Identify Vehicle_States to be fulfilled.

identify_change_to_vehicle_condition

Identify changes to Vehicle_States that Fluids needs satisfied.

assess_vehicle_condition_evidence

Assess the evidence for achievability of Vehicle_States associated with Fluid_Management_Solutions, to decide whether any further action needs to be taken.

5.4.2.21.7.1.5 Fluid_Information

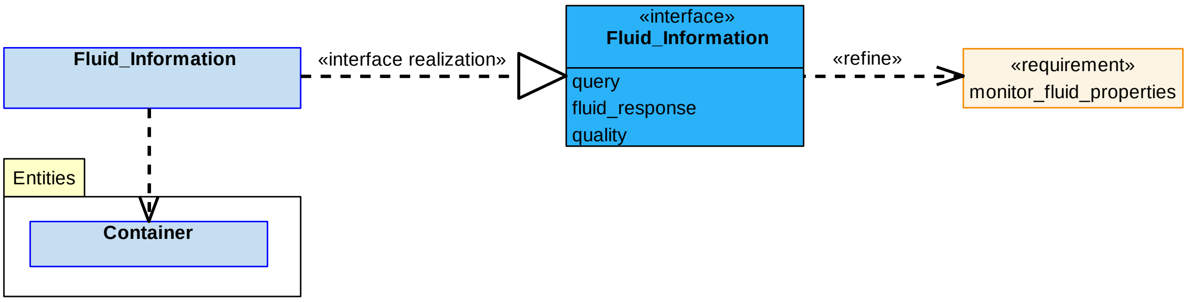


Figure 373: Fluid_Information Service Definition

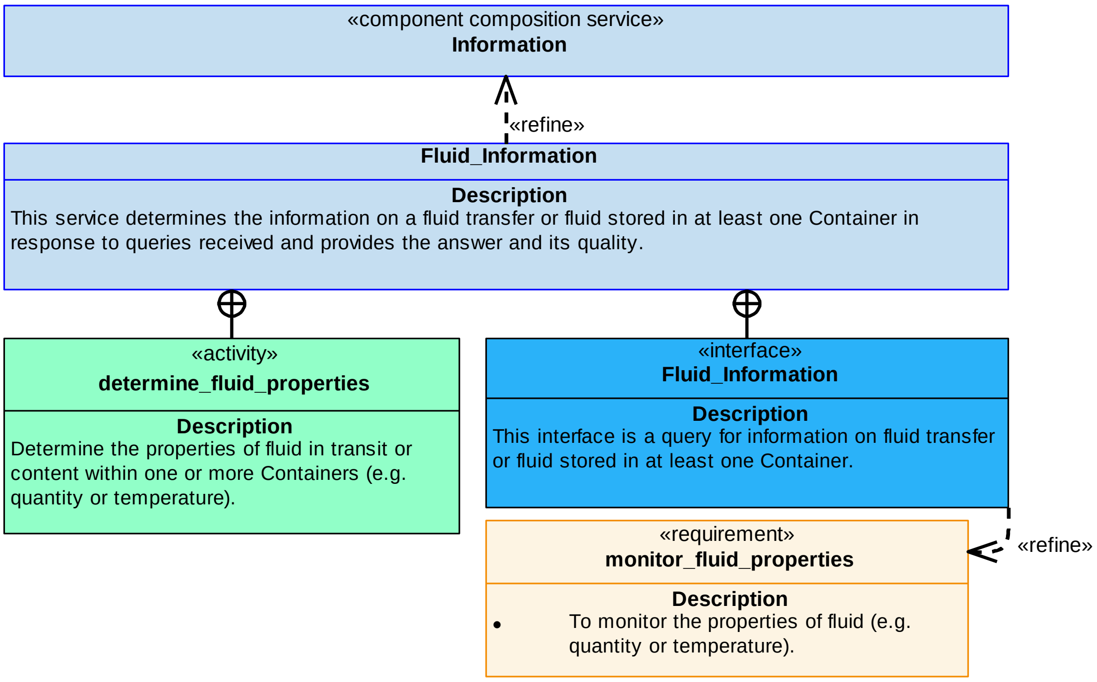

«component composition service»
Information
«refine»
Fluid_Information
Description
This service determines the information on a fluid transfer or fluid stored in at least one Container in response to queries received and provides the answer and its quality.
«activity»
determine_fluid_properties
Description
Determine the properties of fluid in transit or content within one or more Containers (e.g. quantity or temperature).
«interface»
Fluid_Information
Description
This interface is a query for information on fluid transfer or fluid stored in at least one Container.
«refine»
«requirement»
monitor_fluid_properties
Description
To monitor the properties of fluid (e.g. quantity or temperature).

Figure 374: Fluid_Information Service Policy

Fluid_Information

This service determines the information on a fluid transfer or fluid stored in at least one Container in response to queries received and provides the answer and its quality.

<u>Interface</u>

Fluid_Information

This interface is a query for information on fluid transfer or fluid stored in at least one Container.

<u>Attributes</u>

query	The request for information on a fluid transfer or stored fluid property, e.g. which Container fluid is stored in, amount of fluid, or available fluid for transfer.
fluid_response	The property of the fluid transfer or stored fluid returned in response to the query, e.g. the amount of fluid stored across a group of Containers, the specific Container that specified fluid is contained within, or amount of fluid in transit.
quality	The quality (e.g. accuracy and certainty) in the provided response.

<u>Activity</u>

determine_fluid_properties

Determine the properties of fluid in transit or content within one or more Containers (e.g. quantity or temperature).

5.4.2.21.7.1.6 Fluid_Measurement

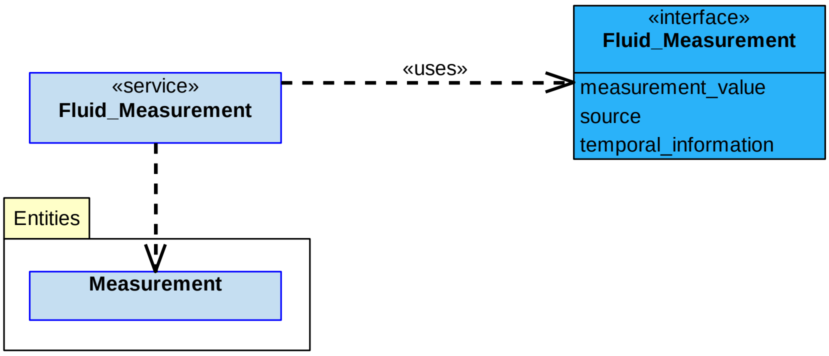


Figure 375: Fluid_Measurement Service Definition

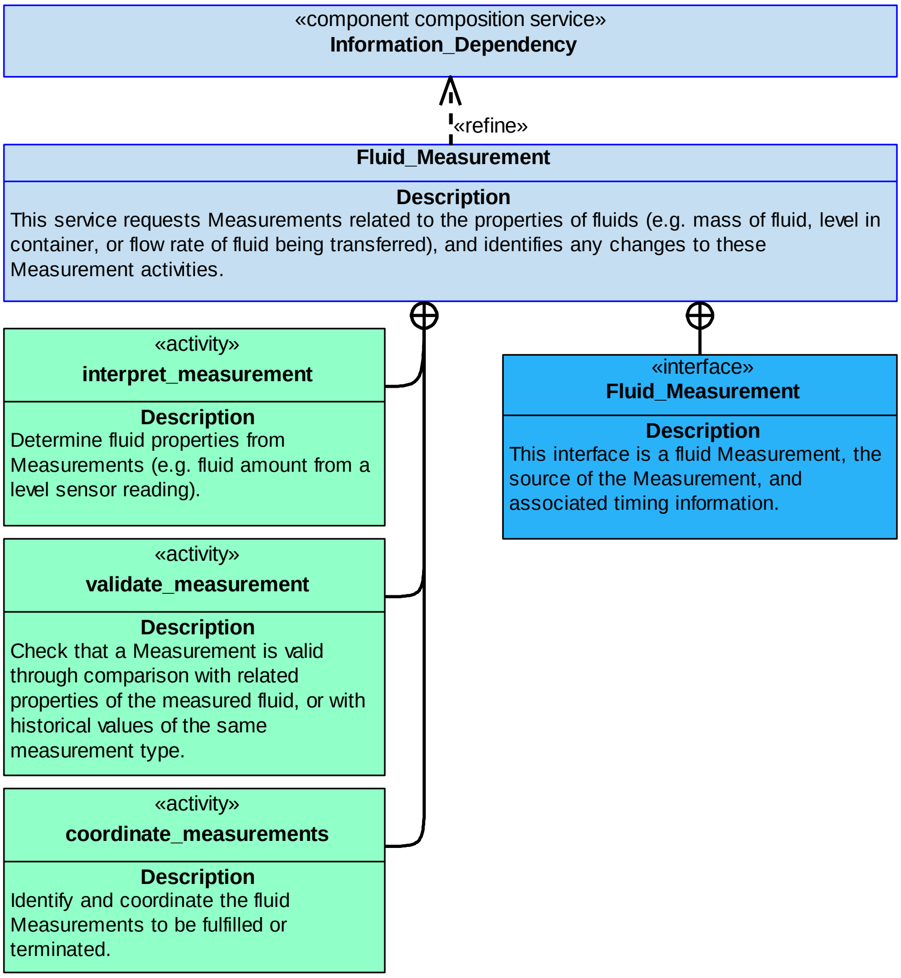

«component composition service»
Information_Dependency
«refine»
Fluid_Measurement
Description
This service requests Measurements related to the properties of fluids (e.g. mass of fluid, level in container, or flow rate of fluid being transferred), and identifies any changes to these Measurement activities.
«activity»
interpret_measurement
Description
Determine fluid properties from Measurements (e.g. fluid amount from a level sensor reading).
«activity»
validate_measurement
Description
Check that a Measurement is valid through comparison with related properties of the measured fluid, or with historical values of the same measurement type.
«activity»
coordinate_measurements
Description
Identify and coordinate the fluid Measurements to be fulfilled or terminated.
«interface»
Fluid_Measurement
Description
This interface is a fluid Measurement, the source of the Measurement, and associated timing information.

Figure 376: Fluid_Measurement Service Policy

Fluid_Measurement

This service requests Measurements related to the properties of fluids (e.g. mass of fluid, level in container, or flow rate of fluid being transferred), and identifies any changes to these Measurement activities.

Interface

Fluid_Measurement

This interface is a fluid Measurement, the source of the Measurement, and associated timing information.

Attributes

measurement_value	The fluid Measurement value.
source	The sensor capable of making the Measurement (e.g. sensor type, or whether it measures fluid in a Container or a Distribution_Path).
temporal_information	Information covering Measurement timings, such as start and end times.

Activities

interpret_measurement

Determine fluid properties from Measurements (e.g. fluid amount from a level sensor reading).

validate_measurement

Check that a Measurement is valid through comparison with related properties of the measured fluid, or with historical values of the same measurement type.

coordinate_measurements

Identify and coordinate the fluid Measurements to be fulfilled or terminated.

5.4.2.21.7.1.7 Vehicle_Information

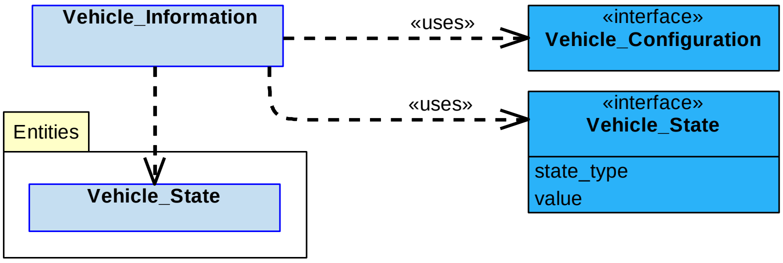


Figure 377: Vehicle_Information Service Definition

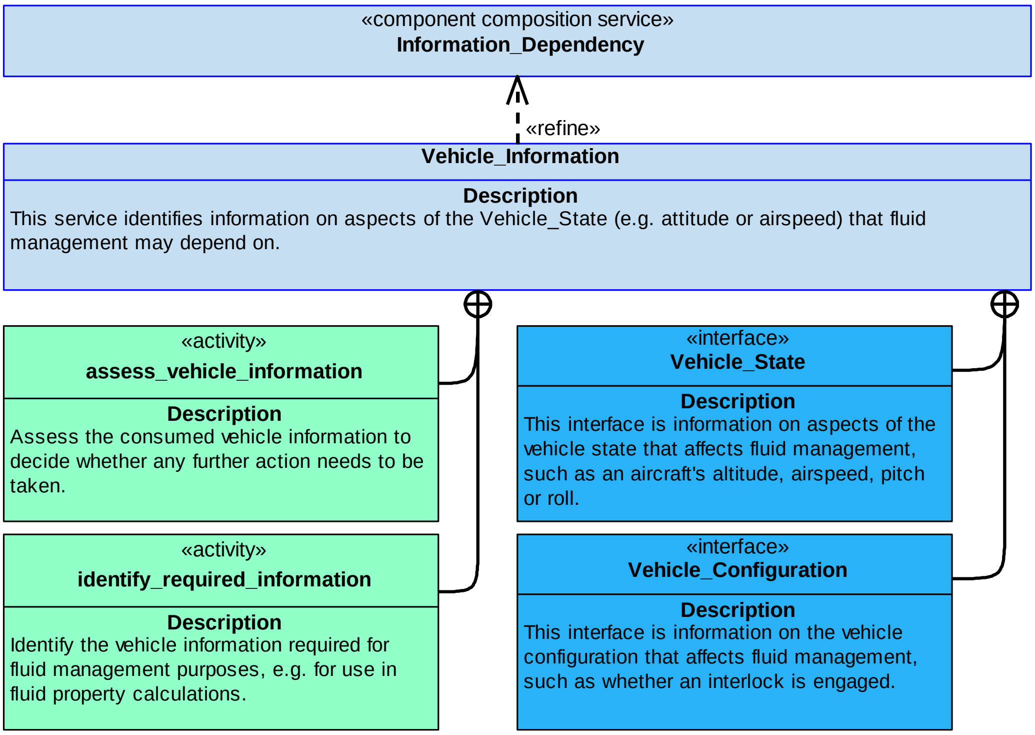

«component composition service»
Information_Dependency
«refine»
Vehicle_Information
Description
This service identifies information on aspects of the Vehicle_State (e.g. attitude or airspeed) that fluid management may depend on.
«activity»
assess_vehicle_information
Description
Assess the consumed vehicle information to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify the vehicle information required for fluid management purposes, e.g. for use in fluid property calculations.
«interface»
Vehicle_State
Description
This interface is information on aspects of the vehicle state that affects fluid management, such as an aircraft's altitude, airspeed, pitch or roll.
«interface»
Vehicle_Configuration
Description
This interface is information on the vehicle configuration that affects fluid management, such as whether an interlock is engaged.

Figure 378: Vehicle_Information Service Policy

Vehicle_Information

This service identifies information on aspects of the Vehicle_State (e.g. attitude or airspeed) that fluid management may depend on.

Interfaces

Vehicle_Configuration

This interface is information on the vehicle configuration that affects fluid management, such as whether an interlock is engaged.

Vehicle_State

This interface is information on aspects of the vehicle state that affects fluid management, such as an aircraft's altitude, airspeed, pitch or roll.

Attributes

state_type	The type of information relating to the vehicle state, such as an aircraft's altitude, airspeed, pitch or roll.
value	The value of the state property.

Activities

assess_vehicle_information

Assess the consumed vehicle information to decide whether any further action needs to be taken.

identify_required_information

Identify the vehicle information required for fluid management purposes, e.g. for use in fluid property calculations.

5.4.2.21.7.1.8 Constraint

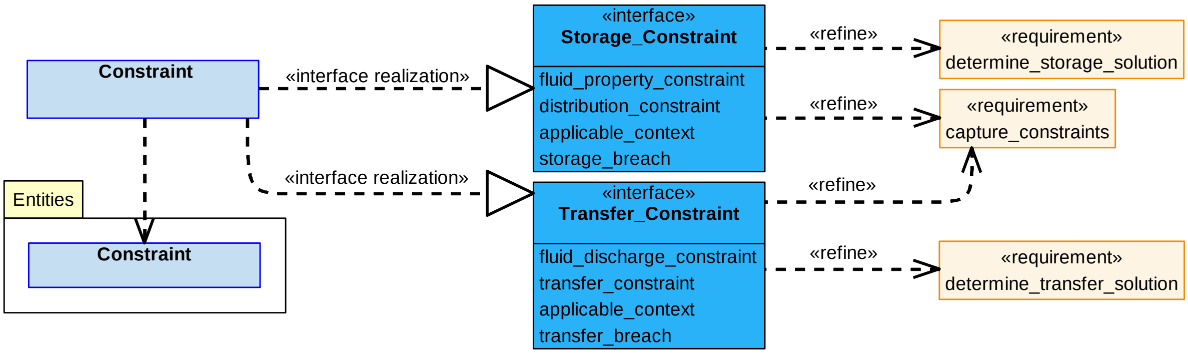
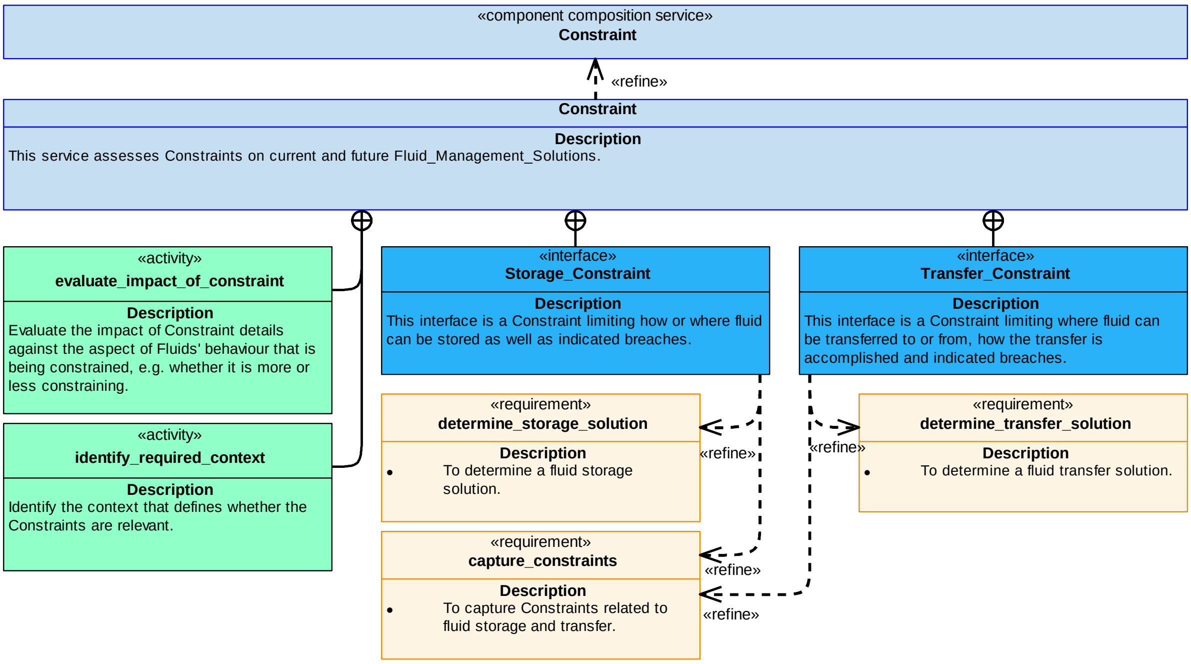


Figure 379: Constraint Service Definition



This service assesses Constraints on current and future Fluid_Management_Solutions.

Evaluate the impact of Constraint details against the aspect of Fluids' behaviour that is being constrained, e.g. whether it is more or less constraining.

Identify the context that defines whether the Constraints are relevant.

This interface is a Constraint limiting how or where fluid can be stored as well as indicated breaches.

This interface is a Constraint limiting where fluid can be transferred to or from, how the transfer is accomplished and indicated breaches.

- To determine a fluid storage solution.

- To capture Constraints related to fluid storage and transfer.

- To determine a fluid transfer solution.

Figure 380: Constraint Service Policy

Constraint

This service assesses Constraints on current and future Fluid_Management_Solutions.

Interfaces

Transfer_Constraint

This interface is a Constraint limiting where fluid can be transferred to or from, how the transfer is accomplished and indicated breaches.

Attributes

fluid_discharge_constraint	Constraints on the release of fluids from the Exploiting Platform.
transfer_constraint	Constraints on the transfer of fluids within the Exploiting Platform.
applicable_context	The context in which the transfer_constraint is applicable.
transfer_breach	A statement that an internal or external transfer_constraint has been breached.

Storage_Constraint

This interface is a Constraint limiting how or where fluid can be stored as well as indicated breaches.

Attributes

fluid_property_constraint	Limits on the properties of fluid in storage, e.g. temperature or emissivity limits.
distribution_constraint	The allowable fluid distribution across Containers and Distribution_Paths in the Exploiting Platform. For example, the allowable fluid mass distribution due to vehicle balance implications.
applicable_context	The context in which the distribution_constraint is applicable.
storage_breach	A statement that the distribution_constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of Fluids' behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context that defines whether the Constraints are relevant.

5.4.2.21.7.1.9 Capability

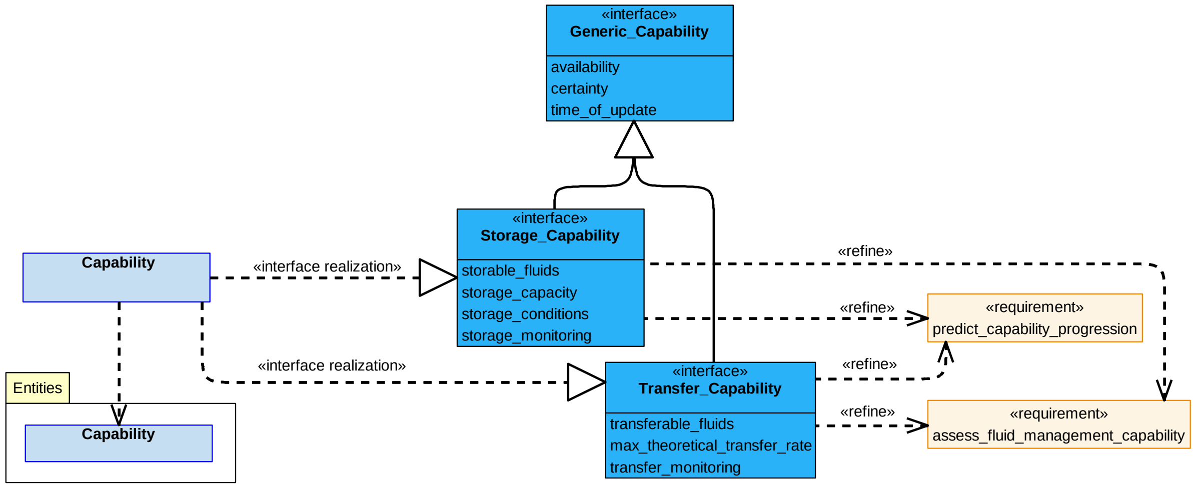
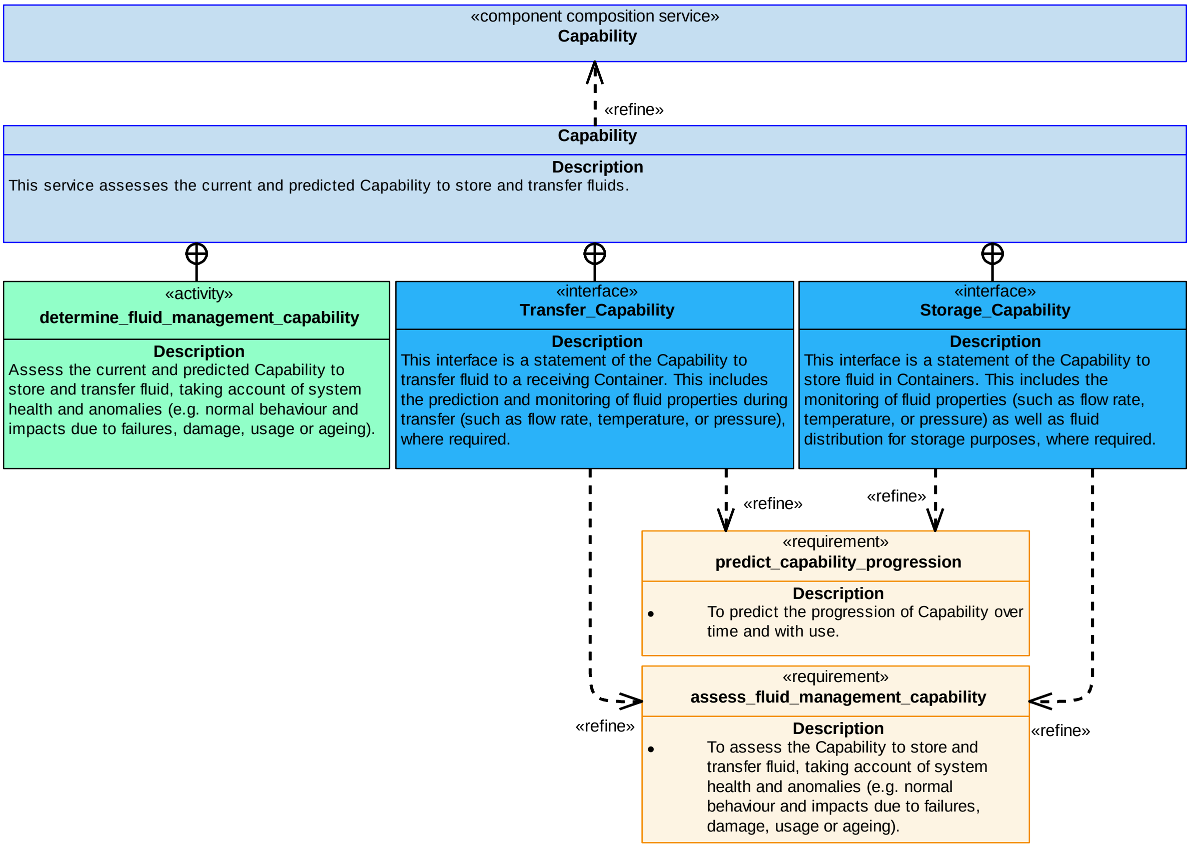


Figure 381: Capability Service Definition



This service assesses the current and predicted Capability to store and transfer fluids.

Assess the current and predicted Capability to store and transfer fluid, taking account of system health and anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

This interface is a statement of the Capability to transfer fluid to a receiving Container. This includes the prediction and monitoring of fluid properties during transfer (such as flow rate, temperature, or pressure), where required.

This interface is a statement of the Capability to store fluid in Containers. This includes the monitoring of fluid properties (such as flow rate, temperature, or pressure) as well as fluid distribution for storage purposes, where required.

- To predict the progression of Capability over time and with use.

- To assess the Capability to store and transfer fluid, taking account of system health and anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Figure 382: Capability Service Policy

Capability

This service assesses the current and predicted Capability to store and transfer fluids.

Interfaces

Transfer_Capability

This interface is a statement of the Capability to transfer fluid to a receiving Container. This includes the prediction and monitoring of fluid properties during transfer (such as flow rate, temperature, or pressure), where required.

Attributes

transferable_fluids	Information about which fluids can be transferred to a receiving Container.
max_theoretical_transfer_rate	The maximum transfer rate that Fluids can theoretically provide.
transfer_monitoring	The properties of the fluid that can be monitored during transfer.

Storage_Capability

This interface is a statement of the Capability to store fluid in Containers. This includes the monitoring of fluid properties (such as flow rate, temperature, or pressure) as well as fluid distribution for storage purposes, where required.

Attributes

storable_fluids	Information about which fluids can be stored.
storage_capacity	The amount of fluid that can be stored.
storage_conditions	The conditions in which a fluid can be stored (e.g. allowable ranges of temperature and pressure).
storage_monitoring	The properties of a stored fluid that can be monitored.

Activity

determine_fluid_management_capability

Assess the current and predicted Capability to store and transfer fluid, taking account of system health and anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.21.7.1.10 Capability_Evidence

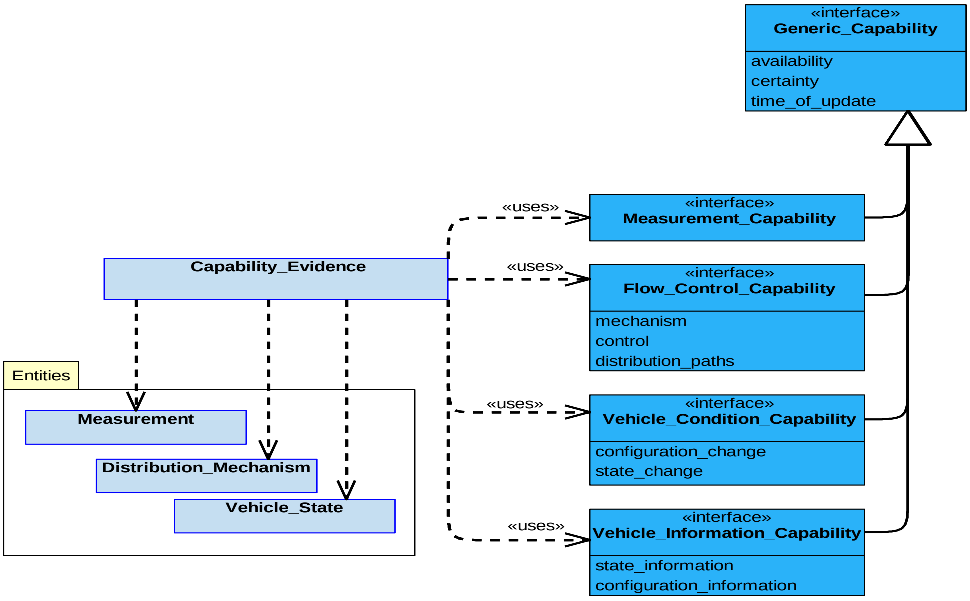


Figure 383: Capability_Evidence Service Definition

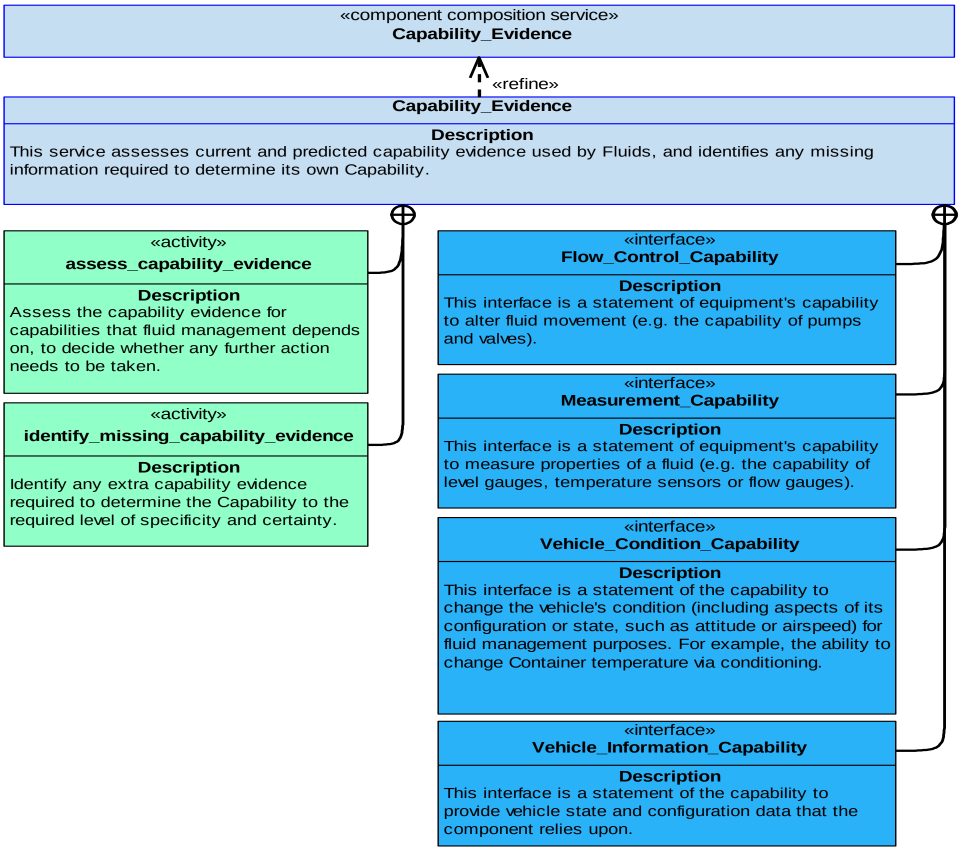

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service assesses current and predicted capability evidence used by Fluids, and identifies any missing information required to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence for capabilities that fluid management depends on, to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Flow_Control_Capability
Description
This interface is a statement of equipment's capability to alter fluid movement (e.g. the capability of pumps and valves).
«interface»
Measurement_Capability
Description
This interface is a statement of equipment's capability to measure properties of a fluid (e.g. the capability of level gauges, temperature sensors or flow gauges).
«interface»
Vehicle_Condition_Capability
Description
This interface is a statement of the capability to change the vehicle's condition (including aspects of its configuration or state, such as attitude or airspeed) for fluid management purposes. For example, the ability to change Container temperature via conditioning.
«interface»
Vehicle_Information_Capability
Description
This interface is a statement of the capability to provide vehicle state and configuration data that the component relies upon.

Figure 384: Capability Evidence Service Policy

Capability_Evidence

This service assesses current and predicted capability evidence used by Fluids, and identifies any missing information required to determine its own Capability.

Interfaces

Measurement_Capability

This interface is a statement of equipment's capability to measure properties of a fluid (e.g. the capability of level gauges, temperature sensors or flow gauges).

Flow_Control_Capability

This interface is a statement of equipment's capability to alter fluid movement (e.g. the capability of pumps and valves).

Attributes

mechanism	The type of mechanism that can be used for flow control, such as a pump or a valve.
control	The degree of fluid control that the mechanism can provide, such as whether the mechanism control is binary or variable.
distribution_paths	The Distribution_Paths that can be involved in flow control, such as the range of paths that may be connected via a valve.

Vehicle_Condition_Capability

This interface is a statement of the capability to change the vehicle's condition (including aspects of its configuration or state, such as attitude or airspeed) for fluid management purposes. For example, the ability to change Container temperature via conditioning.

Attributes

configuration_change	An indication of whether the vehicle configuration can be changed.
state_change	An indication of whether an aspect of the vehicle state can be changed.

Vehicle_Information_Capability

This interface is a statement of the capability to provide vehicle state and configuration data that the component relies upon.

Attributes

state_information	An indication of how well an aspect of the vehicle state can be established.
configuration_information	An indication of how well the vehicle configuration can be established.

Activities

assess_capability_evidence

Assess the capability evidence for capabilities that fluid management depends on, to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.21.7.2 Service Dependencies

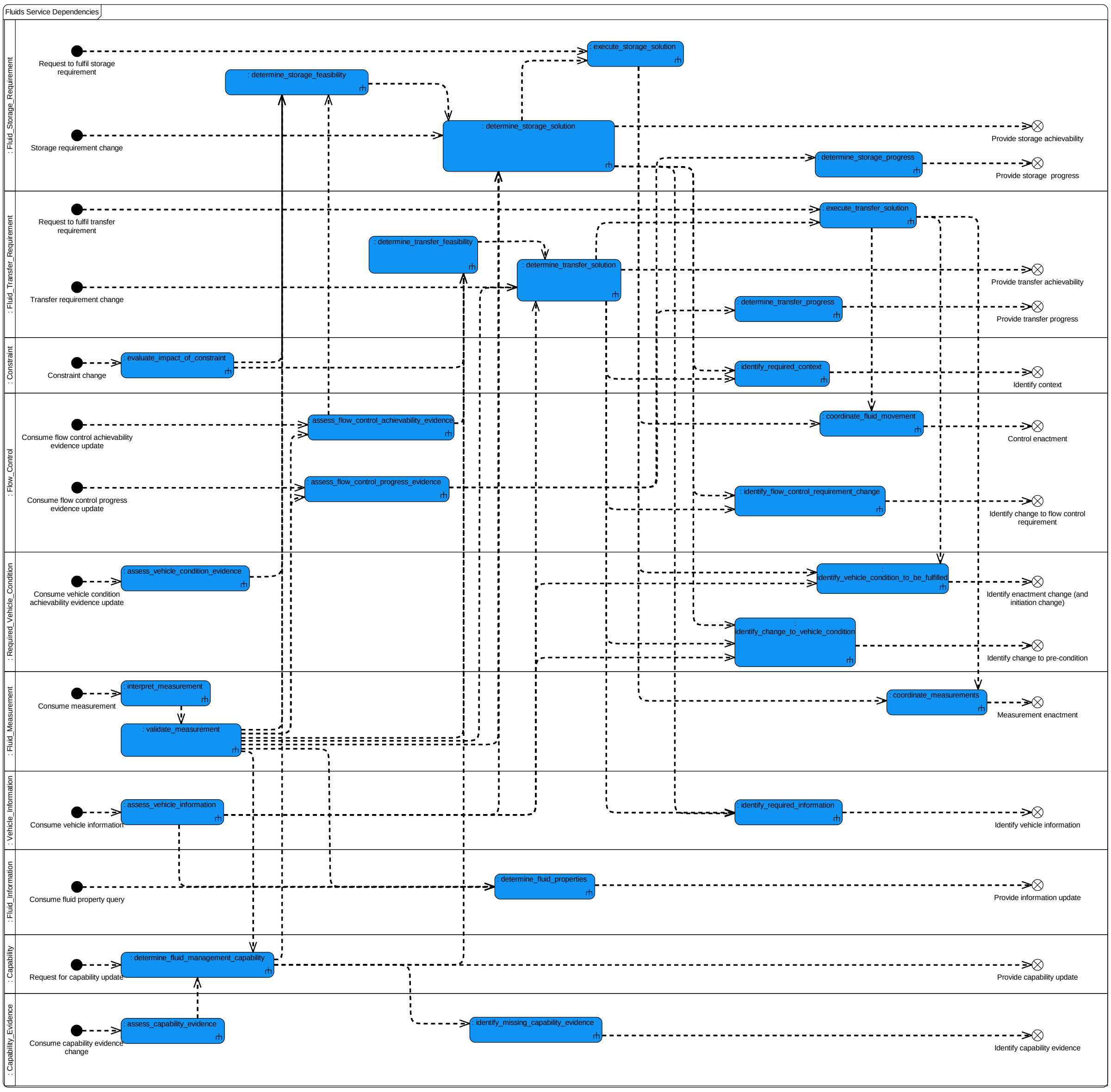

Fluids Service Dependencies
: Fluid_Storage_Requirement
Request to fulfil storage requirement
Storage requirement change
: determine_storage_feasibility
: execute_storage_solution
: determine_storage_solution
: determine_storage_progress
Provide storage achievability
Provide storage progress
: Fluid_Transfer_Requirement
Request to fulfil transfer requirement
Transfer requirement change
: determine_transfer_feasibility
: determine_transfer_solution
: execute_transfer_solution
: determine_transfer_progress
Provide transfer achievability
Provide transfer progress
: Constraint
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
: Flow_Control
Consume flow control achievability evidence update
Consume flow control progress evidence update
: assess_flow_control_achievability_evidence
: assess_flow_control_progress_evidence
: coordinate_fluid_movement
Control enactment
: identify_flow_control_requirement_change
Identify change to flow control requirement
: Required_Vehicle_Condition
Consume vehicle condition achievability evidence update
: assess_vehicle_condition_evidence
: identify_vehicle_condition_to_be_fulfilled
Identify enactment change (and initiation change)
: identify_change_to_vehicle_condition
Identify change to pre-condition
: Fluid_Measurement
Consume measurement
: interpret_measurement
: validate_measurement
: coordinate_measurements
Measurement enactment
: Vehicle_Information
Consume vehicle information
: assess_vehicle_information
: identify_required_information
Identify vehicle information
: Fluid_Information
Consume fluid property query
: determine_fluid_properties
Provide information update
: Capability
Request for capability update
: determine_fluid_management_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 385: Fluids Service Dependencies

5.4.2.22 Formations

5.4.2.22.1 Role

The role of Formations is to coordinate and execute changes in the relative positions of vehicles moving in a formation.

5.4.2.22.2 Overview

Control Architecture

Formations is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a formation Requirement is received from a tasking agent, Formations will determine a Formation_Solution, using a Formation_Pattern if appropriate. The Formation_Solution will specify how Controllable_Vehicles are coordinated to maintain a Delivered_Formation that satisfies the formation Requirement within applicable Constraints. The Delivered_Formation may include Non-controllable_Vehicles that cannot be controlled in the Formation_Solution. Once the Formation_Solution is agreed with the tasking agent it can be implemented, with each Controllable_Vehicle dynamically following the Formation_Solution.

Examples of Use

Formations is required where a group of Formation_Member vehicles are required to move in a coordinated manner. For example:

- Formations ensures air traffic zoning compliance by coordinating the relative positions of Formation_Members.
- Formations retains the relative proximity between Formation_Member vehicles during transit to ensure preparedness in case other types of operational manoeuvre become necessary (e.g. defensive, combat).
- Formations maintains the relative position between Formation_Member vehicles and a Non-controllable_Vehicle (e.g. escorting a potential adversary or formatting on an allied tanker aircraft).

5.4.2.22.3 Service Summary

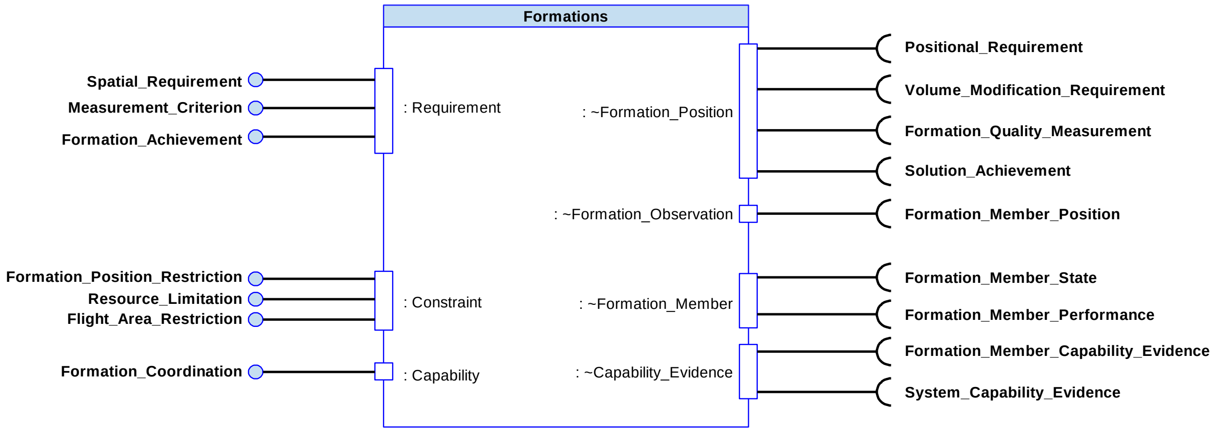


Figure 386: Formations Service Summary

5.4.2.22.4 Responsibilities

capture_formation_requirements

- To capture given formation Requirements (e.g. number of Formation_Members, required positional relationships between Formation_Members, and lead vehicle identification).

capture_measurement_criteria_for_formations

- To capture provided Measurement_Criterion/criteria for Formation_Solutions and Delivered_Formations.

capture_formation_constraints

- To capture given Constraints affecting formations (e.g. EMCON).

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Capability and Constraints.

determine_formation_solution

- To determine a Formation_Solution that meets the given Requirements, within Constraints and Formation_Member Capability.

determine_predicted_quality_of_formation_solution

- To determine the predicted quality of a proposed Formation_Solution against given Measurement_Criterion/criteria.

identify_pre-conditions

- To identify Pre-conditions in support of a Formation_Solution.

coordinate_formation_solution

- To execute an agreed Formation_Solution by coordinating the positions of Controllable_Vehicles.

identify_progress_of_formation_solution

- To identify the progress of a Formation_Solution against the given Requirements.

determine_actual_quality_of_deliverables

- To determine the quality of the Delivered_Formation provided by a solution, measured against given Requirements and Measurement_Criterion/criteria.

assess_formation_capability

- To assess the Capability to plan and execute Formation_Solutions taking account of Formation_Members' health and observed anomalies.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of Formations Capability over time and with use.

5.4.2.22.5 Subject Matter Semantics

The subject matter of Formations is the relative spatial positioning of Formation_Members.

Exclusions

The subject matter of Formations does not include:

- The management of flight composition.
- The management of the routing of vehicles.

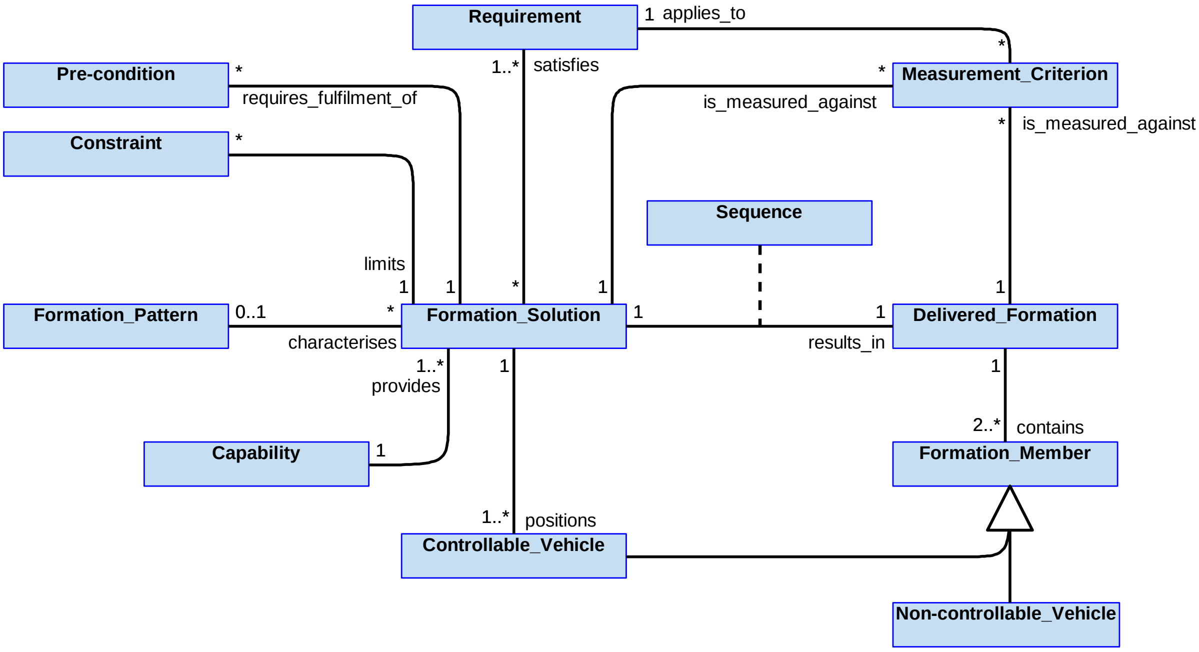


Figure 387: Formations Semantics

5.4.2.22.5.1 Entities

Capability

An ability to produce a Formation_Solution taking into account the capabilities of the individual Controllable_Vehicles.

Constraint

A restriction on when or how a Formation_Solution is applied (e.g. minimum separation distance or minimum altitude).

Controllable_Vehicle

A vehicle whose relative position can be controlled by the Formations component.

Delivered_Formation

An actual formation achieved as a result of implementing the Formation_Solution using the Formation_Members.

Formation_Member

A vehicle belonging to the group of vehicles that are part of the formation.

Formation_Pattern

A pattern of spatial arrangement that can be applied to a group of vehicles (Formation_Members). For example: a formation could be patterned on a diamond arrangement or a single line of vehicles.

Formation_Solution

Criteria and activities that must be applied to position Controllable_Vehicle vehicles, to support the joining, maintenance and leaving of the formation. For example, Controllable_Vehicle vehicles may be instructed to carry out activities such as specific positional manoeuvres (e.g. increase altitude to 25,000 feet), or they may be directed to apply criteria dynamically in response to the changing positions of other Formation_Members (e.g. maintain a particular bearing and distance from Formation_Member x).

Measurement_Criterion

A criterion which the quality of a Formation_Solution and its Delivered_Formation will be measured against (e.g. the minimum separation of any pair of Formation_Members).

Non-controllable_Vehicle

A vehicle whose relative position cannot be controlled by the Formations component.

Pre-condition

A condition that must be met before a Formation_Solution can be implemented (e.g. a Formation_Member knows the position of the other vehicles).

Requirement

A requirement to form and maintain a coordinated spatial pattern between Formation_Member vehicles.

Sequence

The temporal order of activities that make up the Formation_Solution that result in a Delivered_Formation. For example, Formation_Member A must increase altitude by 1000 feet before Formation_Member B.

5.4.2.22.6 Design Rationale

5.4.2.22.6.1 Assumptions

- Members of a formation need not be members of a flight. Formations can include non-flight-members such as fuel tankers.
- Formation_Patterns are likely to vary between missions and Exploiting Platforms.
- Formations is not concerned with understanding the potential trajectories of Formation_Members when determining Formation_Solutions.

5.4.2.22.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Formations:

- Data Driving - Entities such as Formation_Pattern, Constraint, Pre-condition and Measurement_Criterion could be configured using a data-driven approach.
- Constraint Management - Formations must generate Formation_Solutions within the bounds of given Constraints in accordance with the Constraint Management PYRAMID concept.
- Autonomy - Maintenance of Formations could involve a degree of autonomous movement of Formation_Member vehicles, which must be in accordance with the Autonomy PYRAMID concept.
- Multi-Vehicle Coordination - Governs some aspects of how Formation_Member vehicles coordinate with each other at various levels of the Control Architecture.

Extensions

- Formation_Patterns could be implemented as an extension set, in accordance with the Component Extensions PYRAMID concept. This would enable separately pre-configured, reusable patterns to be utilised in the generation of Formation_Solutions.

Exploitation Considerations

- An instance of Formations is likely to be required on all Formation_Member Controllable_Vehicles that would participate in a formation (in accordance with Multi-Vehicle Coordination). Exceptions exist, such as using a PYRAMID non-compliant vehicle as the lead vehicle in the formation.

5.4.2.22.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

Failure of this component could result in a formation being flown incorrectly. For example:

- Flying closer than intended by the crew to another air vehicle, resulting in an unacceptable risk of mid-air collision.
- Configuring the air vehicle incorrectly for flying in close proximity to another air vehicle (e.g. high power transmission not inhibited).

Therefore, it is considered that there is a reasonable likelihood of a catastrophic accident and so an indicative IDAL of DAL A is appropriate.

5.4.2.22.6.4 Security Considerations

The indicative security classification is SNEO.

This component coordinates the spatial positioning between members of the formation and will therefore require a degree of information about those vehicles (performance data, behaviour, etc.) in order to determine the Formation_Solutions that support the mission objectives using the appropriate operational tactics. Such information is likely to have a classification of SNEO. Where the component is coordinating the behaviour of uncrewed aircraft, including swarms, loss of confidentiality will lead to predictability of behaviour. The integrity and availability of flight interactions will also need to be protected to prevent unwanted behaviour.

The security of communications channels used to coordinate between formation members is not a function of this component.

The component may be expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** as trusted members of the formation.
- **Logging of Security Data** of member authentication for later forensic examination.
- **Maintaining Audit Records** to support non-repudiation of instructions for positional changes, etc. in the course of operations of the formation.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **System Status and Monitoring** through coordinating and monitoring the available assets. Unexpected formation changes may indicate that one or more formation members have been compromised by a cyber adversary.

This component is considered unlikely to implement security enforcing functions.

5.4.2.22.7 Services

5.4.2.22.7.1 Service Definitions

5.4.2.22.7.1.1 Requirement

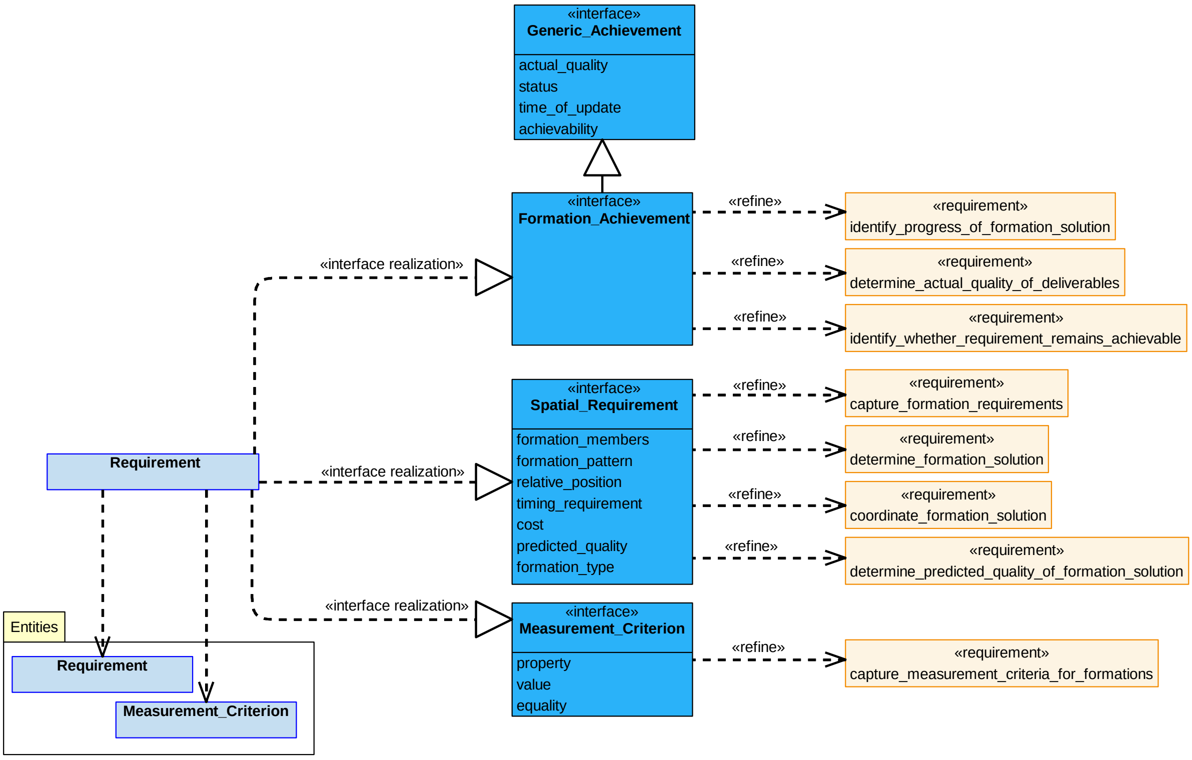


Figure 388: Requirement Service Definition

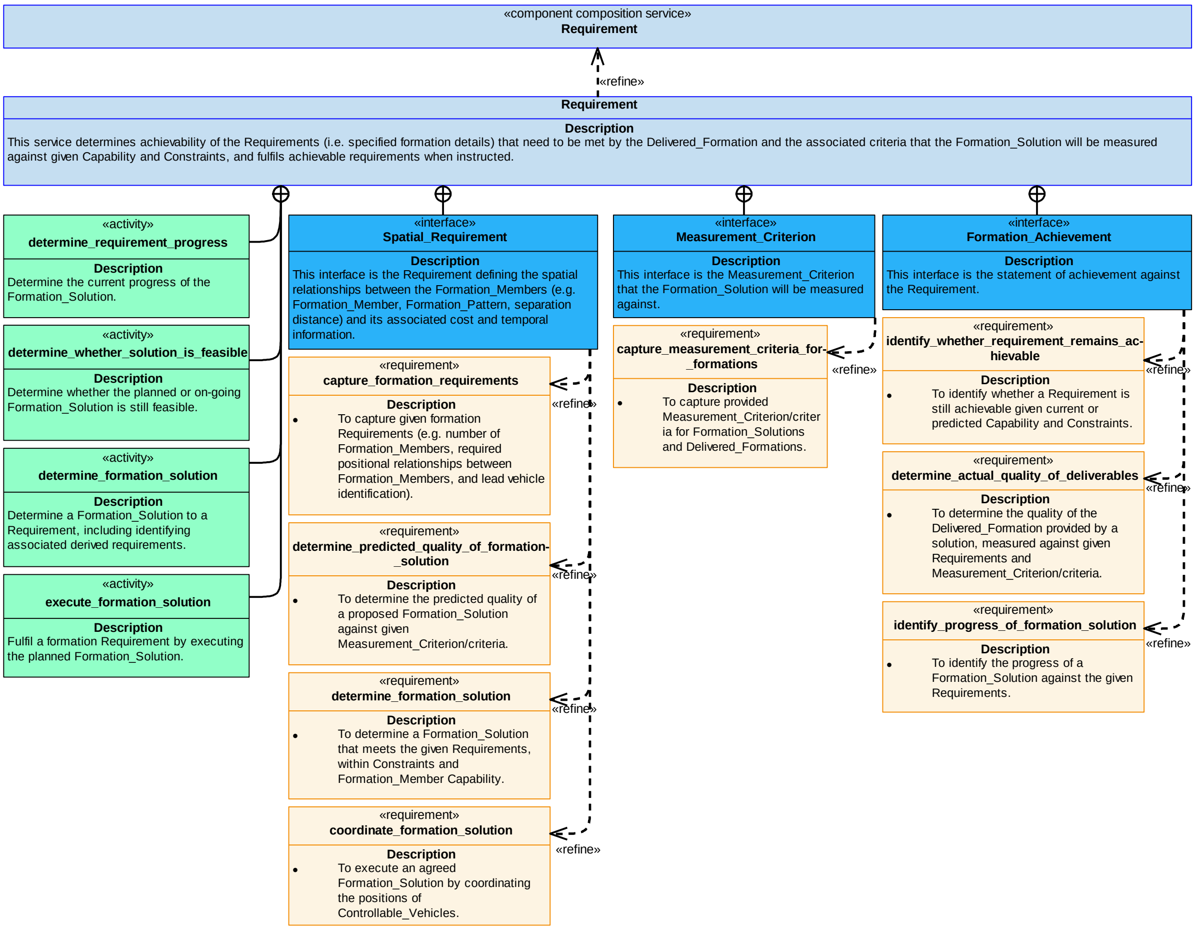

«component composition service»
Requirement
«refine»
Requirement
Description
This service determines achievability of the Requirements (i.e. specified formation details) that need to be met by the Delivered_Formation and the associated criteria that the Formation_Solution will be measured against given Capability and Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_requirement_progress
Description
Determine the current progress of the Formation_Solution.
«activity»
determine_whether_solution_is_feasible
Description
Determine whether the planned or on-going Formation_Solution is still feasible.
«activity»
determine_formation_solution
Description
Determine a Formation_Solution to a Requirement, including identifying associated derived requirements.
«activity»
execute_formation_solution
Description
Fulfil a formation Requirement by executing the planned Formation_Solution.
«interface»
Spatial_Requirement
Description
This interface is the Requirement defining the spatial relationships between the Formation_Members (e.g. Formation_Member, Formation_Pattern, separation distance) and its associated cost and temporal information.
«requirement»
capture_formation_requirements
Description
To capture given formation Requirements (e.g. number of Formation_Members, required positional relationships between Formation_Members, and lead vehicle identification).
«refine»
«requirement»
determine_predicted_quality_of_formation-_solution
Description
To determine the predicted quality of a proposed Formation_Solution against given Measurement_Criterion/criteria.
«refine»
«requirement»
determine_formation_solution
Description
To determine a Formation_Solution that meets the given Requirements, within Constraints and Formation_Member Capability.
«refine»
«requirement»
coordinate_formation_solution
Description
To execute an agreed Formation_Solution by coordinating the positions of Controllable_Vehicles.
«refine»
«interface»
Measurement_Criterion
Description
This interface is the Measurement_Criterion that the Formation_Solution will be measured against.
«requirement»
capture_measurement_criteria_for-_formations
Description
To capture provided Measurement_Criterion/criteria for Formation_Solutions and Delivered_Formations.
«refine»
«interface»
Formation_Achievement
Description
This interface is the statement of achievement against the Requirement.
«requirement»
identify_whether_requirement_remains_ac-hievable
Description
To identify whether a Requirement is still achievable given current or predicted Capability and Constraints.
«refine»
«requirement»
determine_actual_quality_of_deliverables
Description
To determine the quality of the Delivered_Formation provided by a solution, measured against given Requirements and Measurement_Criterion/criteria.
«refine»
«requirement»
identify_progress_of_formation_solution
Description
To identify the progress of a Formation_Solution against the given Requirements.
«refine»

Figure 389: Requirement Service Policy

Requirement

This service determines achievability of the Requirements (i.e. specified formation details) that need to be met by the Delivered_Formation and the associated criteria that the Formation_Solution will be measured against given Capability and Constraints, and fulfils achievable requirements when instructed.

Interfaces

Measurement_Criterion

This interface is the Measurement_Criterion that the Formation_Solution will be measured against.

Attributes

property	The property to be measured, e.g. separation distance.
value	The measured value of the property, e.g. 5 nautical miles.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Spatial_Requirement

This interface is the Requirement defining the spatial relationships between the Formation_Members (e.g. Formation_Member, Formation_Pattern, separation distance) and its associated cost and temporal information.

Attributes

formation_members	The Formation_Members that need to be included in the Delivered_Formation.
formation_pattern	The desired formation pattern (e.g. diamond or V-formation).
relative_position	The required relative distance and angle between Formation_Members.
timing_requirement	Duration of time during which this requirement applies.
cost	The cost of executing the solution, for example: resources used, time taken.
predicted_quality	How well the proposed formation solution is predicted to satisfy the requirement.
formation_type	The high-level purpose of the formation (e.g. air-to-air refuelling) that may be needed to generate a Formation_Solution given an abstract Requirement.

Formation_Achievement

This interface is the statement of achievement against the Requirement.

Activities

determine_requirement_progress

Determine the current progress of the Formation_Solution.

determine_formation_solution

Determine a Formation_Solution to a Requirement, including identifying associated derived requirements.

execute_formation_solution

Fulfil a formation Requirement by executing the planned Formation_Solution.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Formation_Solution is still feasible.

5.4.2.22.7.1.2 Formation_Position

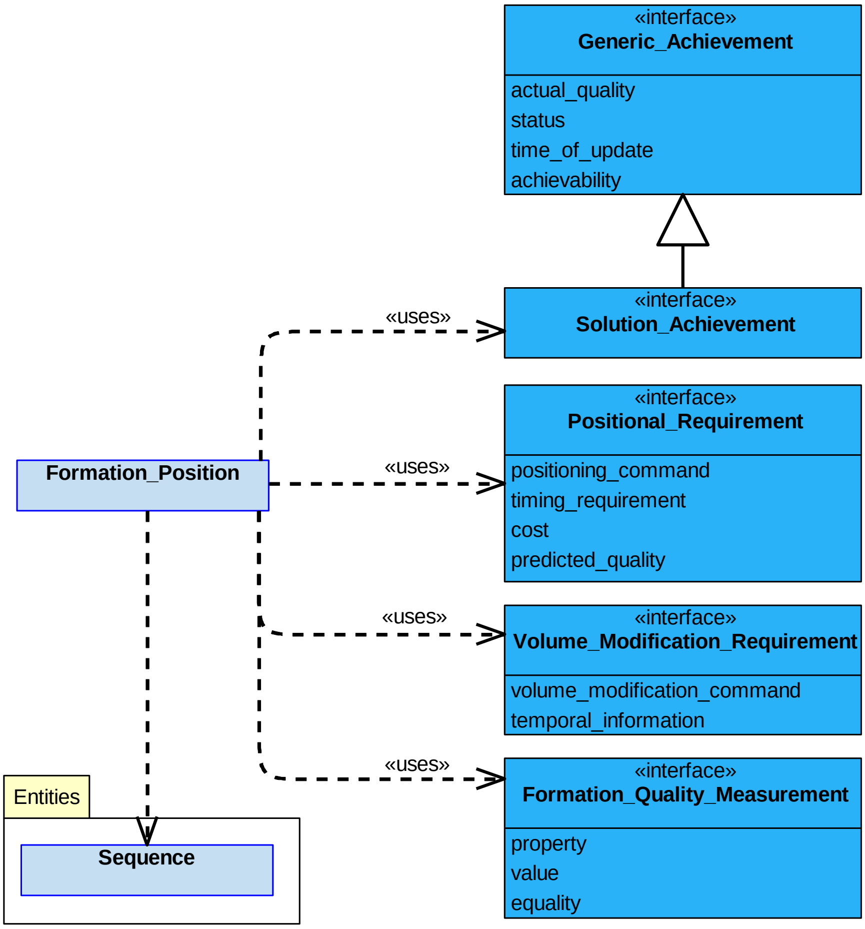
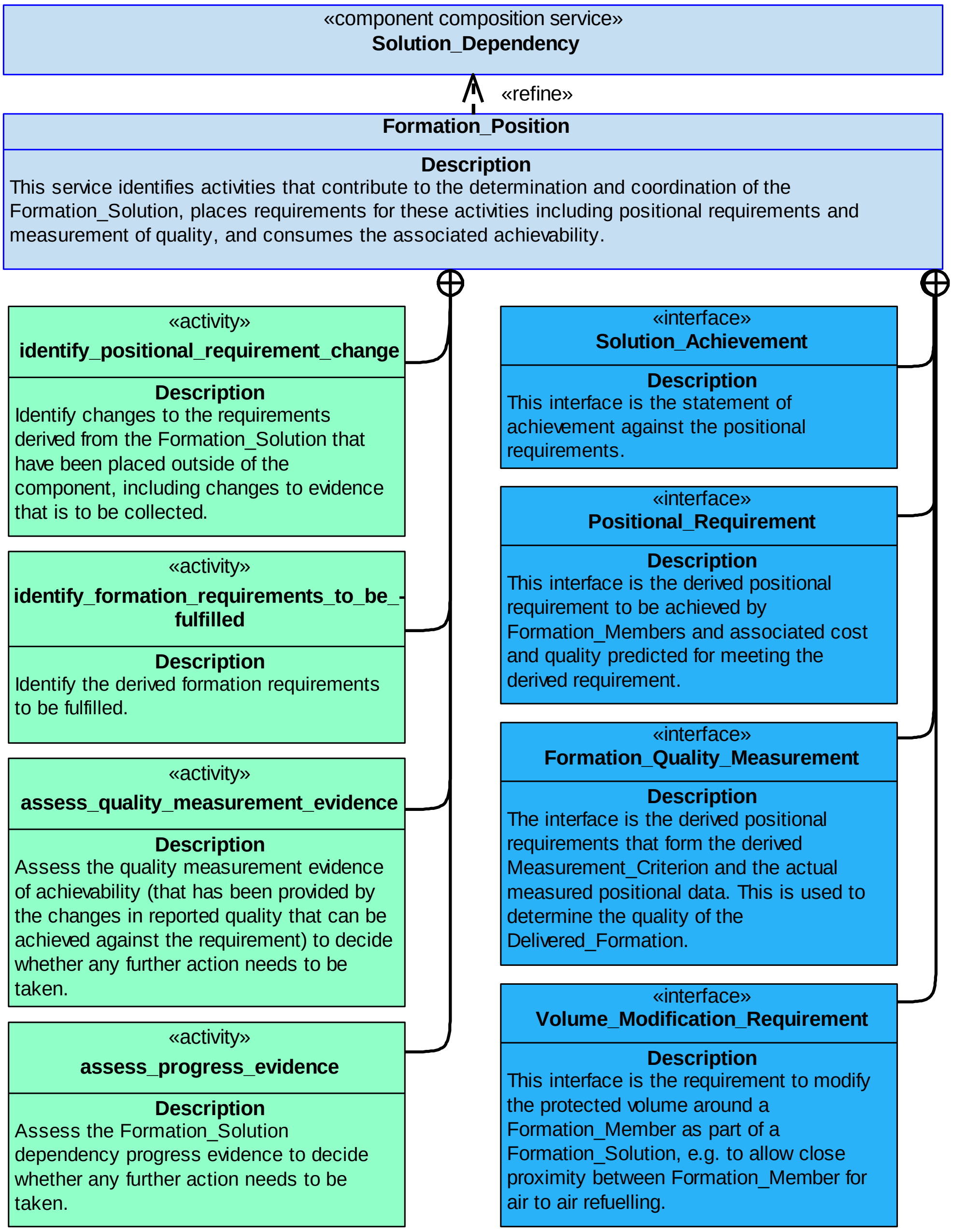


Figure 390: Formation_Position Service Definition



This service identifies activities that contribute to the determination and coordination of the Formation_Solution, places requirements for these activities including positional requirements and measurement of quality, and consumes the associated achievability.

Identify changes to the requirements derived from the Formation_Solution that have been placed outside of the component, including changes to evidence that is to be collected.

Identify the derived formation requirements to be fulfilled.

Assess the quality measurement evidence of achievability (that has been provided by the changes in reported quality that can be achieved against the requirement) to decide whether any further action needs to be taken.

Assess the Formation_Solution dependency progress evidence to decide whether any further action needs to be taken.

This interface is the statement of achievement against the positional requirements.

This interface is the derived positional requirement to be achieved by Formation_Members and associated cost and quality predicted for meeting the derived requirement.

The interface is the derived positional requirements that form the derived Measurement_Criterion and the actual measured positional data. This is used to determine the quality of the Delivered_Formation.

This interface is the requirement to modify the protected volume around a Formation_Member as part of a Formation_Solution, e.g. to allow close proximity between Formation_Member for air to air refuelling.

Figure 391: Formation_Position Service Policy

Formation_Position

This service identifies activities that contribute to the determination and coordination of the Formation_Solution, places requirements for these activities including positional requirements and measurement of quality, and consumes the associated achievability.

Interfaces

Positional_Requirement

This interface is the derived positional requirement to be achieved by Formation_Members and associated cost and quality predicted for meeting the derived requirement.

Attributes

positioning_command	Command for Formation_Member to change position, orientation or velocity.
timing_requirement	Duration of time during which this requirement applies.
cost	The cost of executing the solution, for example: resources used or time taken.
predicted_quality	How well the proposed formation solution is predicted to satisfy the requirement.

Formation_Quality_Measurement

The interface is the derived positional requirements that form the derived Measurement_Criterion and the actual measured positional data. This is used to determine the quality of the Delivered_Formation.

Attributes

property	The property to be measured, e.g. altitude.
value	The measured value of the property, e.g. 25,000 ft.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Solution_Achievement

This interface is the statement of achievement against the positional requirements.

Volume_Modification_Requirement

This interface is the requirement to modify the protected volume around a Formation_Member as part of a Formation_Solution, e.g. to allow close proximity between Formation_Member for air to air refuelling.

Attributes

volume_modification_command	Command to modify the protected volume around a Formation_Member to allow another Formation_Member to approach.
temporal_information	Information covering timing, such as start and end times.

Activities

assess_progress_evidence

Assess the Formation_Solution dependency progress evidence to decide whether any further action needs to be taken.

identify_positional_requirement_change

Identify changes to the requirements derived from the Formation_Solution that have been placed outside of the component, including changes to evidence that is to be collected.

identify_formation_requirements_to_be_fulfilled

Identify the derived formation requirements to be fulfilled.

assess_quality_measurement_evidence

Assess the quality measurement evidence of achievability (that has been provided by the changes in reported quality that can be achieved against the requirement) to decide whether any further action needs to be taken.

5.4.2.22.7.1.3 Formation_Observation

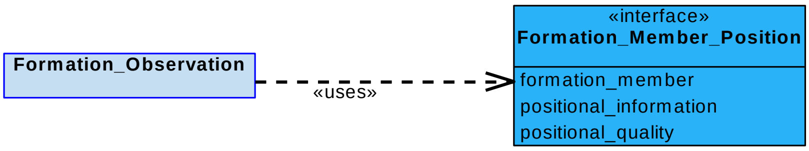


Figure 392: Formation_Observation Service Definition

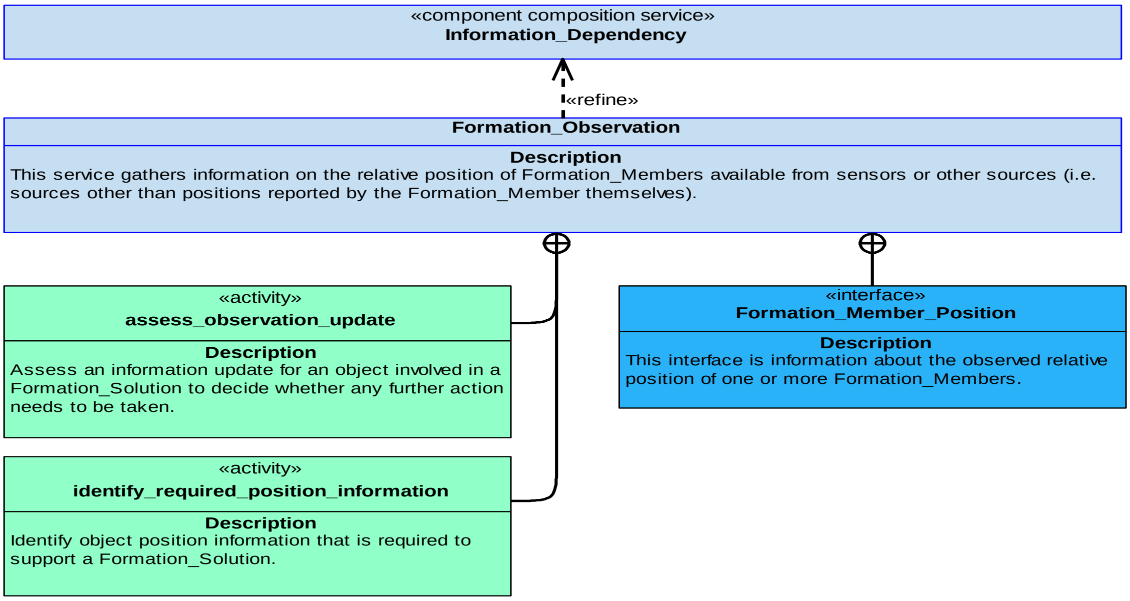

«component composition service»
Information_Dependency
«refine»
Formation_Observation
Description
This service gathers information on the relative position of Formation_Members available from sensors or other sources (i.e. sources other than positions reported by the Formation_Member themselves).
«activity»
assess_observation_update
Description
Assess an information update for an object involved in a Formation_Solution to decide whether any further action needs to be taken.
«activity»
identify_required_position_information
Description
Identify object position information that is required to support a Formation_Solution.
«interface»
Formation_Member_Position
Description
This interface is information about the observed relative position of one or more Formation_Members.

Figure 393: Formation_Observation Service Policy

Formation_Observation

This service gathers information on the relative position of Formation_Members available from sensors or other sources (i.e. sources other than positions reported by the Formation_Member themselves).

Interface

Formation_Member_Position

This interface is information about the observed relative position of one or more Formation_Members.

Attributes

formation_member	The specific Formation_Member to which the observation applies.
positional_information	The observed relative position of the Formation_Member.
positional_quality	The accuracy and certainty of an observed relative location.

Activities

assess_observation_update

Assess an information update for an object involved in a Formation_Solution to decide whether any further action needs to be taken.

identify_required_position_information

Identify object position information that is required to support a Formation_Solution.

5.4.2.22.7.1.4 Formation_Member

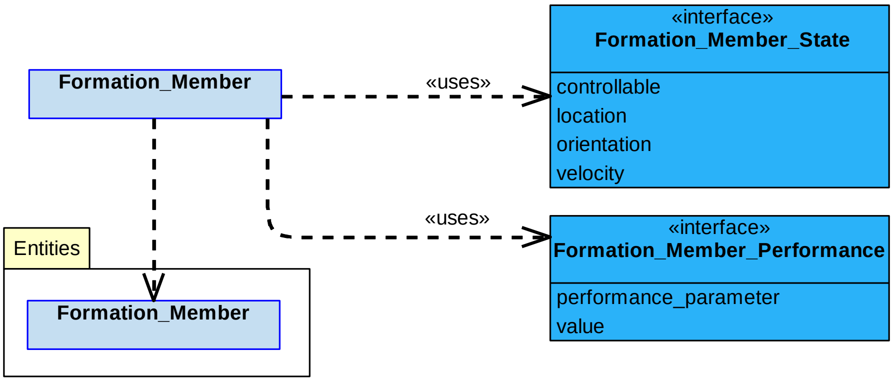


Figure 394: Formation_Member Service Definition

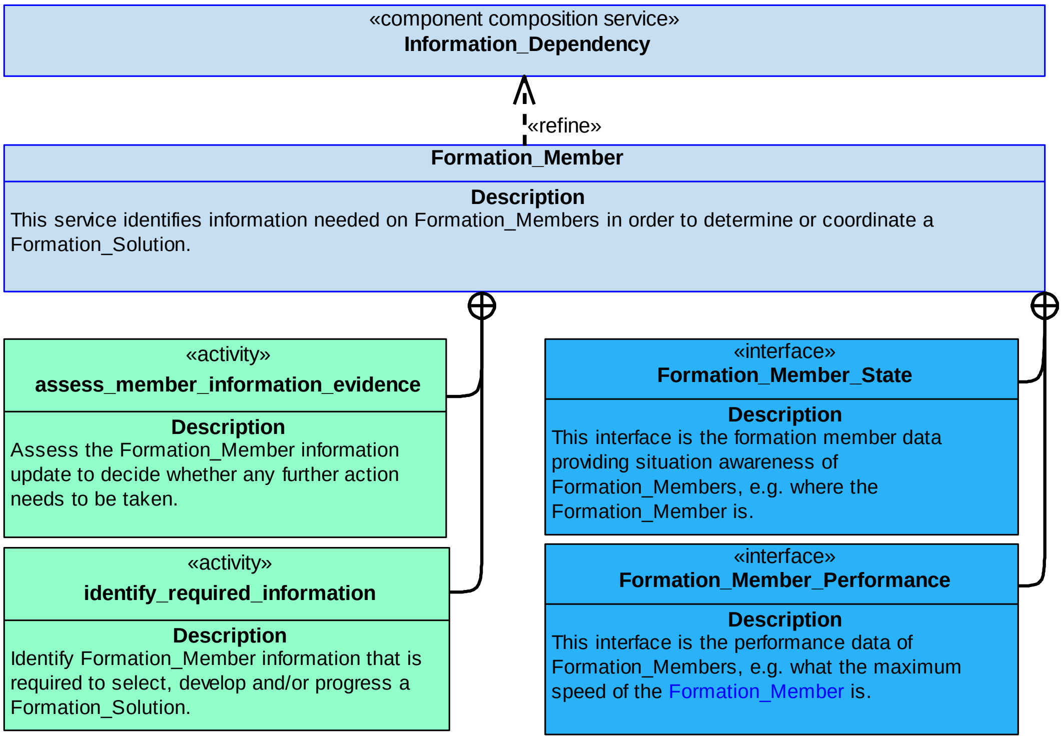

«component composition service»
Information_Dependency
«refine»
Formation_Member
Description
This service identifies information needed on Formation_Members in order to determine or coordinate a Formation_Solution.
«activity»
assess_member_information_evidence
Description
Assess the Formation_Member information update to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify Formation_Member information that is required to select, develop and/or progress a Formation_Solution.
«interface»
Formation_Member_State
Description
This interface is the formation member data providing situation awareness of Formation_Members, e.g. where the Formation_Member is.
«interface»
Formation_Member_Performance
Description
This interface is the performance data of Formation_Members, e.g. what the maximum speed of the Formation_Member is.

Figure 395: Formation_Member Service Policy

Formation_Member

This service identifies information needed on Formation_Members in order to determine or coordinate a Formation_Solution.

Interfaces

Formation_Member_State

This interface is the formation member data providing situation awareness of Formation_Members, e.g. where the Formation_Member is.

Attributes

controllable	A mechanism to distinguish between Controllable_Vehicles and Non-controllable_Vehicles.
location	Current location of the Formation_Member.
orientation	Orientation of the Formation_Member.
velocity	Current velocity of the Formation_Member.

Formation_Member_Performance

This interface is the performance data of Formation_Members, e.g. what the maximum speed of the Formation_Member is.

Attributes

performance_parameter	A property relating to the performance of a Formation_Member, e.g. the maximum speed.
value	The value of the performance_parameter.

Activities

assess_member_information_evidence

Assess the Formation_Member information update to decide whether any further action needs to be taken.

identify_required_information

Identify Formation_Member information that is required to select, develop and/or progress a Formation_Solution.

5.4.2.22.7.1.5 Constraint

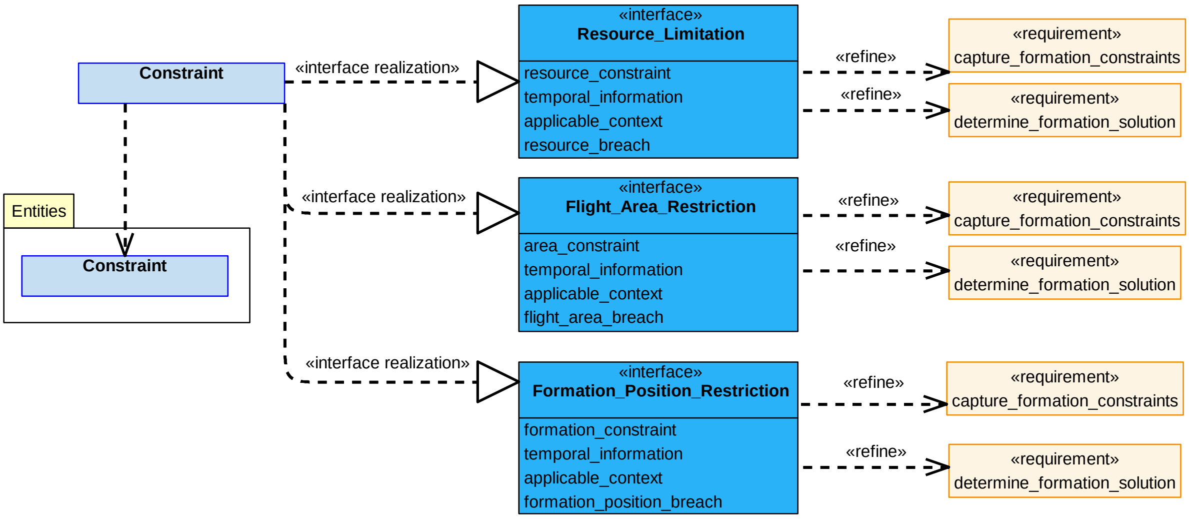
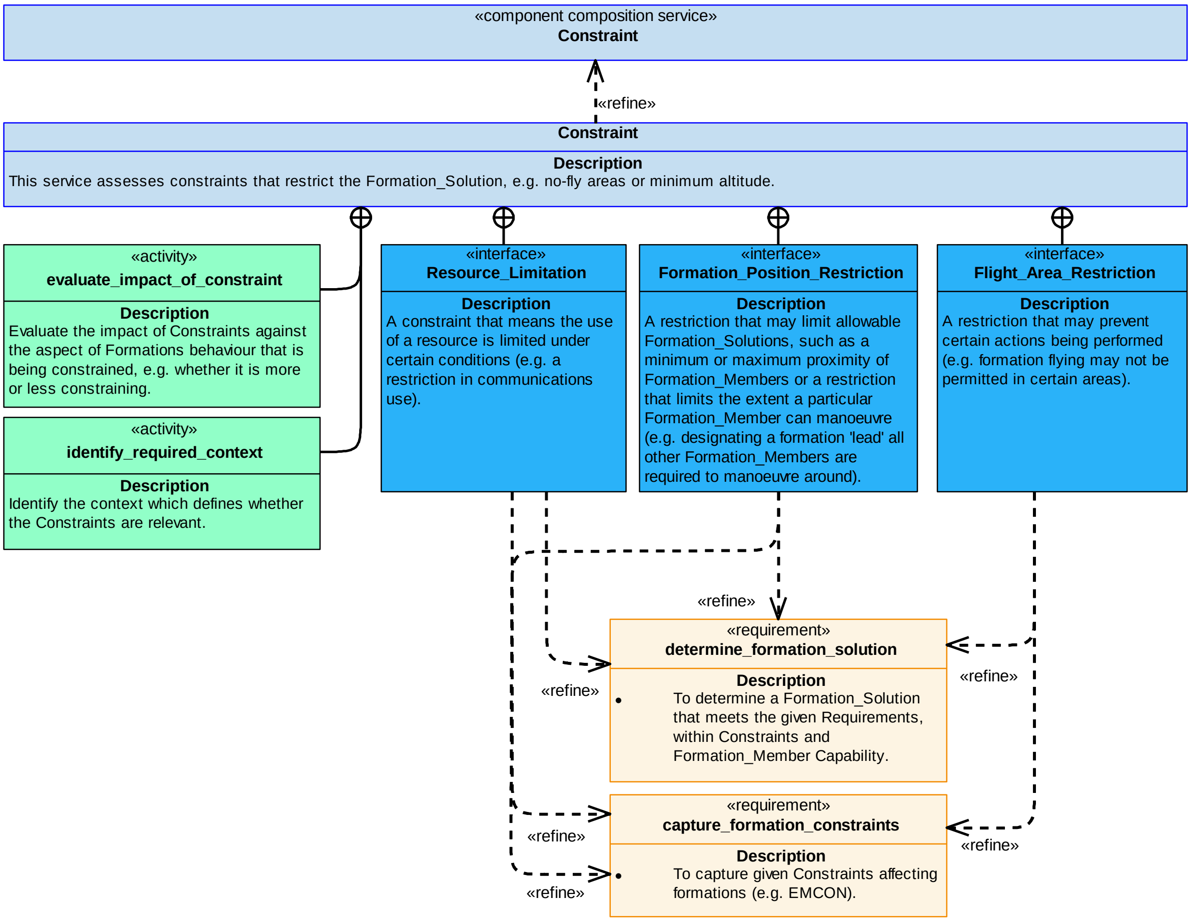


Figure 396: Constraint Service Definition



Constraint — Description: This service assesses constraints that restrict the Formation_Solution, e.g. no-fly areas or minimum altitude.

evaluate_impact_of_constraint — Description: Evaluate the impact of Constraints against the aspect of Formations behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context — Description: Identify the context which defines whether the Constraints are relevant.

Resource_Limitation — Description: A constraint that means the use of a resource is limited under certain conditions (e.g. a restriction in communications use).

Formation_Position_Restriction — Description: A restriction that may limit allowable Formation_Solutions, such as a minimum or maximum proximity of Formation_Members or a restriction that limits the extent a particular Formation_Member can manoeuvre (e.g. designating a formation 'lead' all other Formation_Members are required to manoeuvre around).

Flight_Area_Restriction — Description: A restriction that may prevent certain actions being performed (e.g. formation flying may not be permitted in certain areas).

determine_formation_solution — Description:

- To determine a Formation_Solution that meets the given Requirements, within Constraints and Formation_Member Capability.

capture_formation_constraints — Description:

- To capture given Constraints affecting formations (e.g. EMCON).

Figure 397: Constraint Service Policy

Constraint

This service assesses constraints that restrict the Formation_Solution, e.g. no-fly areas or minimum altitude.

Interfaces

Resource_Limitation

A constraint that means the use of a resource is limited under certain conditions (e.g. a restriction in communications use).

Attributes

resource_constraint	Constraints upon a resource being used.
temporal_information	Timing information pertaining to the periods of time when the constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the constraint is applicable.
resource_breach	A statement that the resource limitation has been breached.

Flight_Area_Restriction

A restriction that may prevent certain actions being performed (e.g. formation flying may not be permitted in certain areas).

Attributes

area_constraint	Area constraints that have been provided, e.g. no fly zones.
temporal_information	Timing information pertaining to the periods of time when the constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the constraint is applicable.
flight_area_breach	A statement that the flight area restriction has been breached.

Formation_Position_Restriction

A restriction that may limit allowable Formation_Solutions, such as a minimum or maximum proximity of Formation_Members or a restriction that limits the extent a particular Formation_Member can manoeuvre (e.g. designating a formation 'lead' all other Formation_Members are required to manoeuvre around).

Attributes

formation_constraint	Formation constraints that have been provided, e.g. a Formation_Member that may not be required to change position when changing formation.
temporal_information	Timing information pertaining to the periods of time when the constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the constraint is applicable.
formation_position_breach	A statement that the formation position restriction has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraints against the aspect of Formations behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.22.7.1.6 Capability

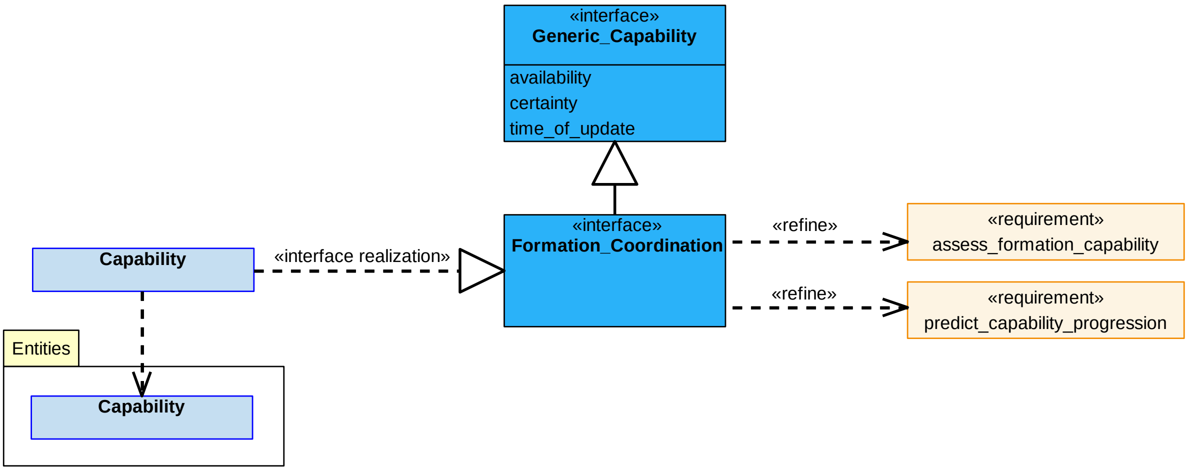


Figure 398: Capability Service Definition

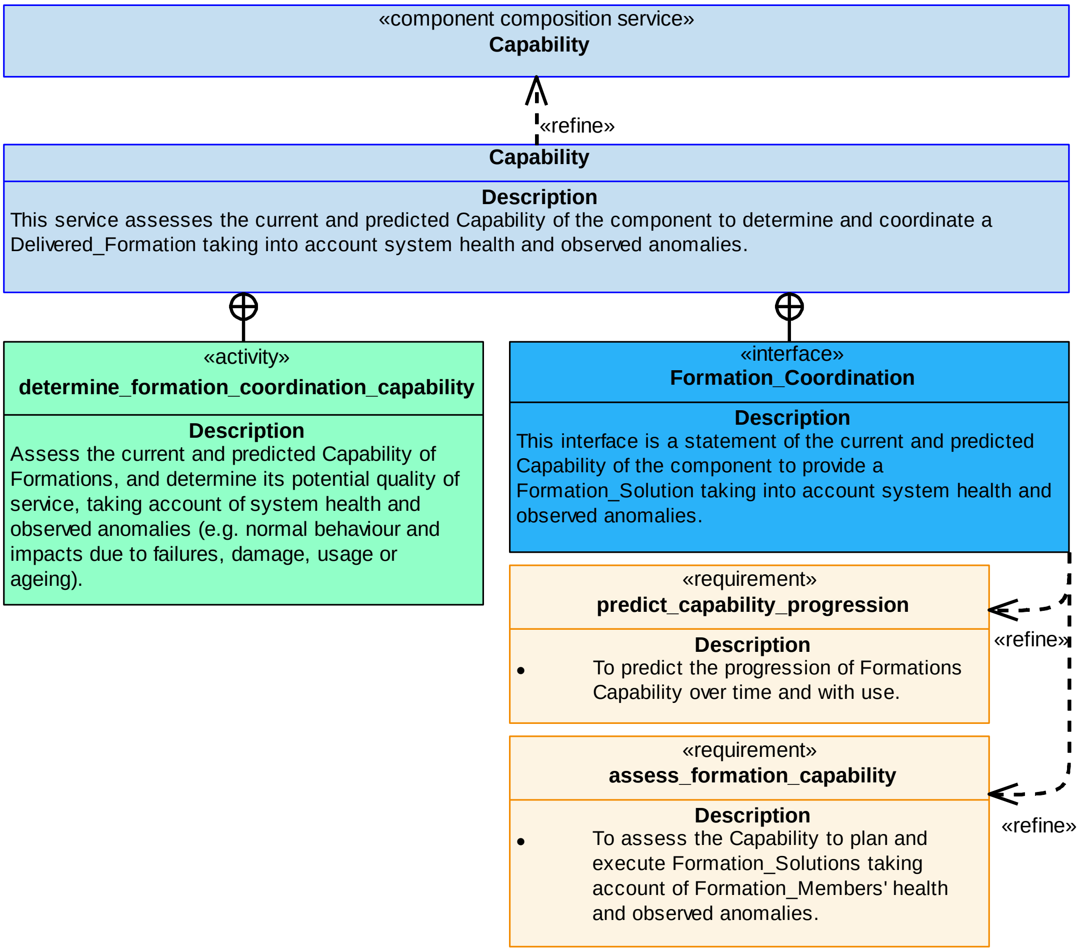

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability of the component to determine and coordinate a Delivered_Formation taking into account system health and observed anomalies.
«activity»
determine_formation_coordination_capability
Description
Assess the current and predicted Capability of Formations, and determine its potential quality of service, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Formation_Coordination
Description
This interface is a statement of the current and predicted Capability of the component to provide a Formation_Solution taking into account system health and observed anomalies.
«requirement»
predict_capability_progression
Description
To predict the progression of Formations Capability over time and with use.
«refine»
«requirement»
assess_formation_capability
Description
To assess the Capability to plan and execute Formation_Solutions taking account of Formation_Members' health and observed anomalies.
«refine»

Figure 399: Capability Service Policy

Capability

This service assesses the current and predicted Capability of the component to determine and coordinate a Delivered_Formation taking into account system health and observed anomalies.

Interface

Formation_Coordination

This interface is a statement of the current and predicted Capability of the component to provide a Formation_Solution taking into account system health and observed anomalies.

Activity

determine_formation_coordination_capability

Assess the current and predicted Capability of Formations, and determine its potential quality of service, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.22.7.1.7 Capability_Evidence

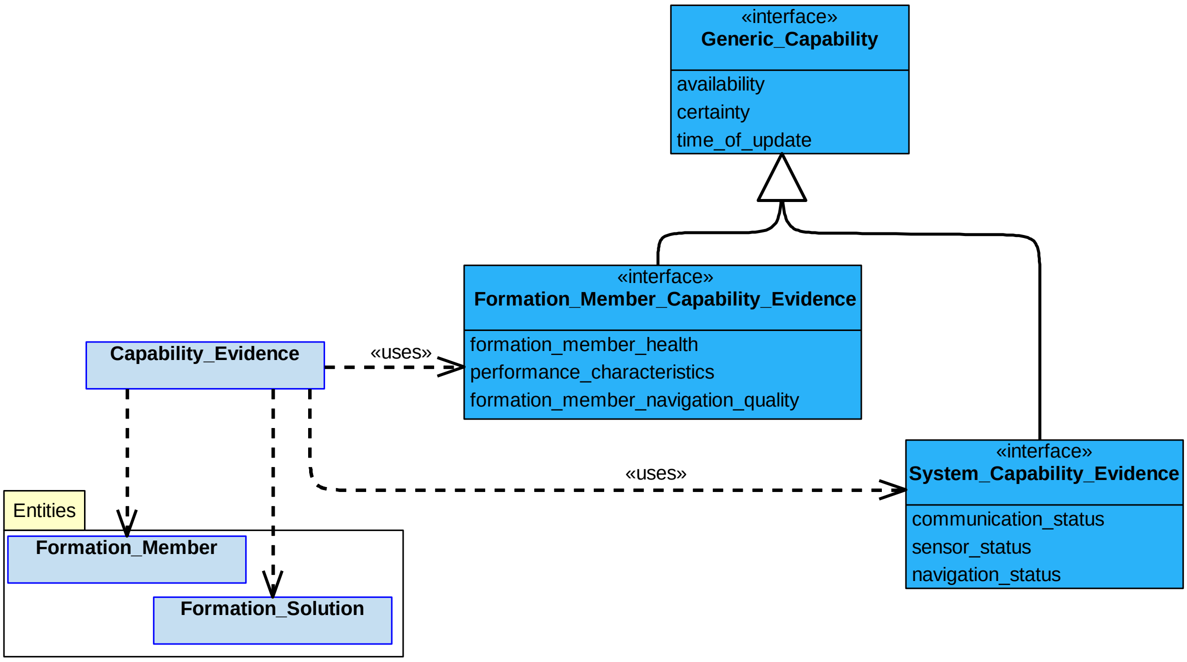
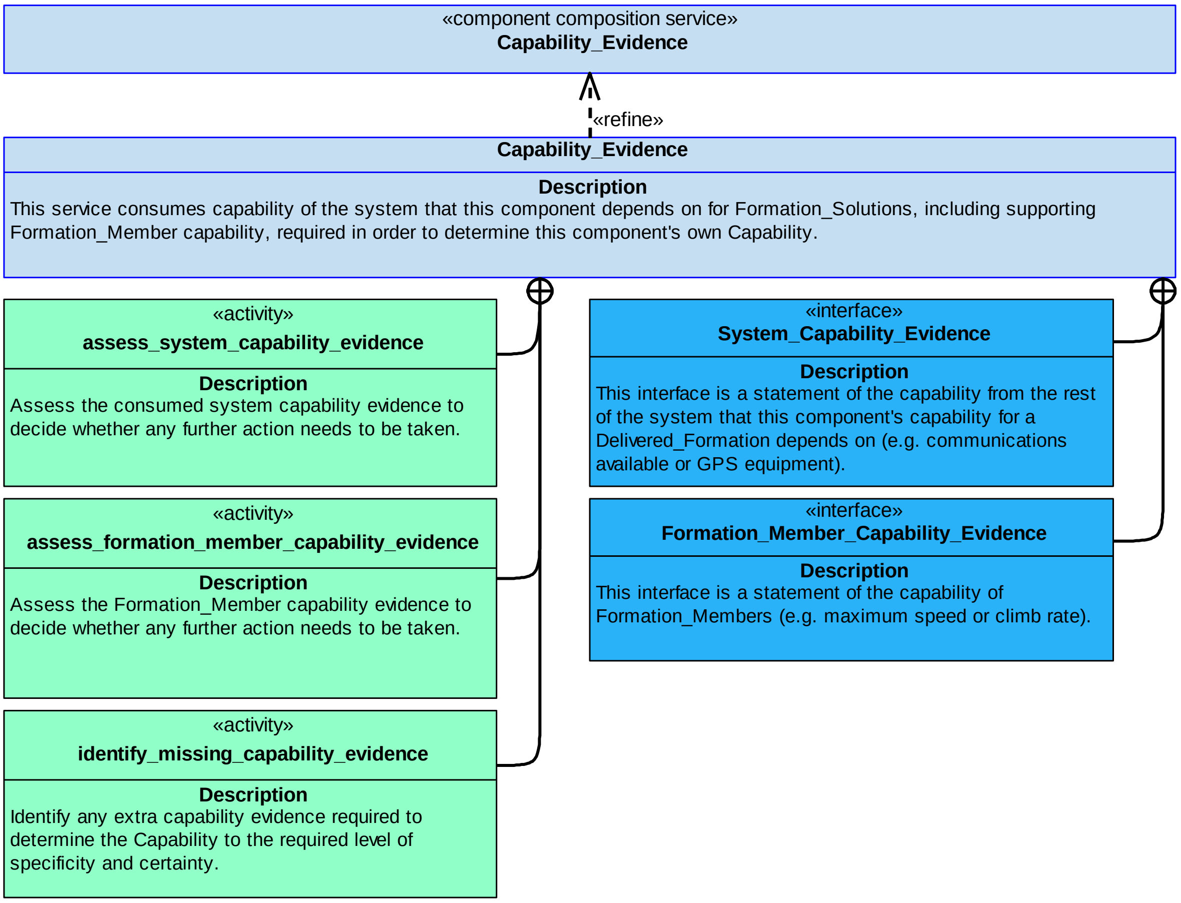


Figure 400: Capability_Evidence Service Definition


«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
«activity»
assess_system_capability_evidence
«activity»
assess_formation_member_capability_evidence
«activity»
identify_missing_capability_evidence
«interface»
System_Capability_Evidence
«interface»
Formation_Member_Capability_Evidence

Description
This service consumes capability of the system that this component depends on for Formation_Solutions, including supporting Formation_Member capability, required in order to determine this component's own Capability.

Description
Assess the consumed system capability evidence to decide whether any further action needs to be taken.

Description
Assess the Formation_Member capability evidence to decide whether any further action needs to be taken.

Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

Description
This interface is a statement of the capability from the rest of the system that this component's capability for a Delivered_Formation depends on (e.g. communications available or GPS equipment).

Description
This interface is a statement of the capability of Formation_Members (e.g. maximum speed or climb rate).

Figure 401: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability of the system that this component depends on for Formation_Solutions, including supporting Formation_Member capability, required in order to determine this component's own Capability.

Interfaces

System_Capability_Evidence

This interface is a statement of the capability from the rest of the system that this component's capability for a Delivered_Formation depends on (e.g. communications available or GPS equipment).

Attributes

communication_status	Status of communications resource (e.g. radio).
sensor_status	Status of sensor resources (e.g. radar).
navigation_status	Status of resources providing navigation capability (e.g. data on local navigation beacons).

Formation_Member_Capability_Evidence

This interface is a statement of the capability of Formation_Members (e.g. maximum speed or climb rate).

Attributes

formation_member_health	Health status of Formation_Member.
performance_characteristics	Performance characteristics of formation member (e.g. maximum climb rate, maximum airspeed, or ceiling).
formation_member_navigation_quality	A Formation_Member's reported ability to navigate and determine its relative position through its sensors.

Activities

assess_system_capability_evidence

Assess the consumed system capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

assess_formation_member_capability_evidence

Assess the Formation_Member capability evidence to decide whether any further action needs to be taken.

5.4.2.22.7.2 Service Dependencies

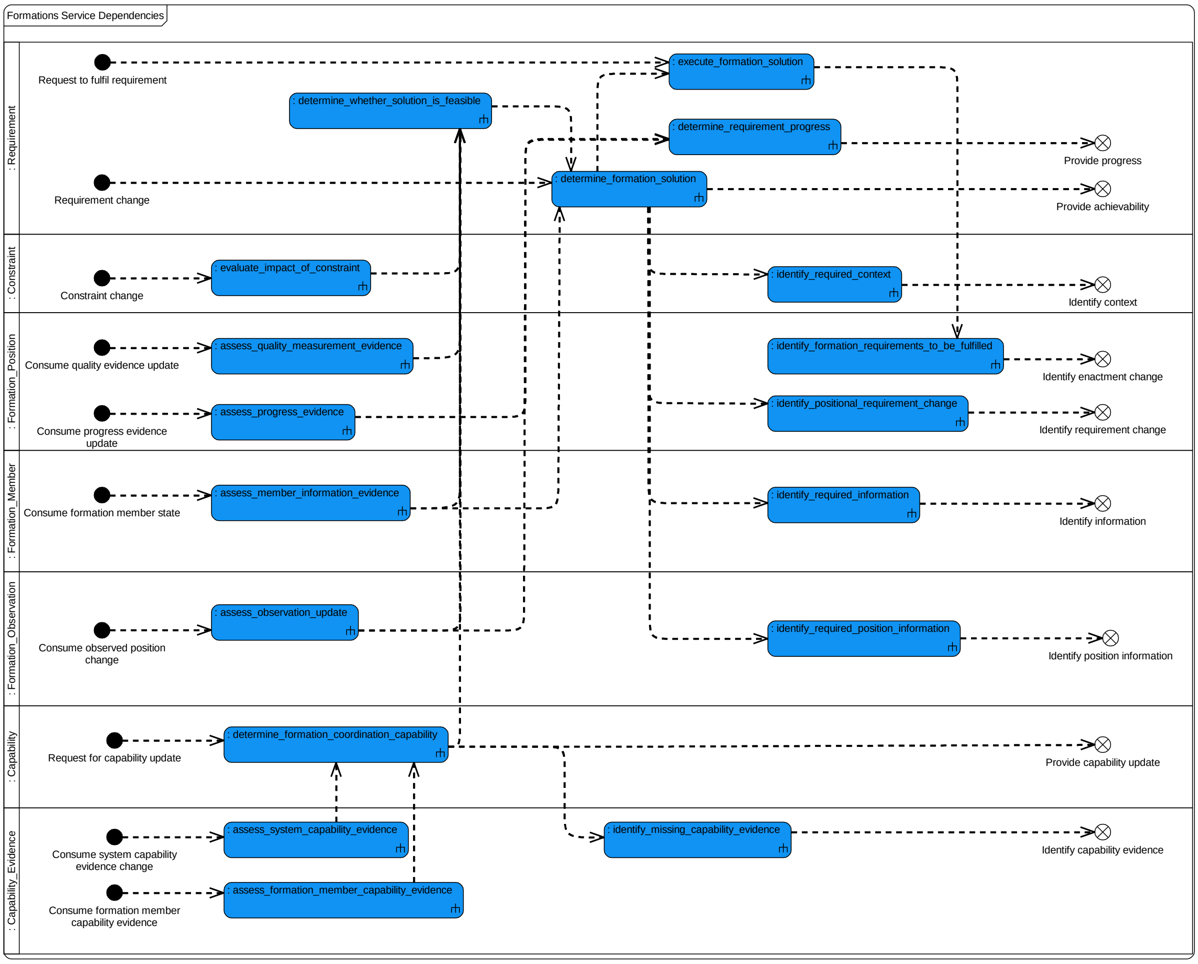

Formations Service Dependencies
: Requirement
Request to fulfil requirement
: execute_formation_solution
: determine_whether_solution_is_feasible
: determine_requirement_progress
Provide progress
Requirement change
: determine_formation_solution
Provide achievability
: Constraint
: evaluate_impact_of_constraint
Constraint change
: identify_required_context
Identify context
: Formation_Position
Consume quality evidence update
: assess_quality_measurement_evidence
: identify_formation_requirements_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_progress_evidence
: identify_positional_requirement_change
Identify requirement change
: Formation_Member
Consume formation member state
: assess_member_information_evidence
: identify_required_information
Identify information
: Formation_Observation
Consume observed position change
: assess_observation_update
: identify_required_position_information
Identify position information
: Capability
Request for capability update
: determine_formation_coordination_capability
Provide capability update
: Capability_Evidence
Consume system capability evidence change
: assess_system_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence
Consume formation member capability evidence
: assess_formation_member_capability_evidence

Figure 402: Formations Service Dependencies

5.4.2.23 Geography

5.4.2.23.1 Role

The role of Geography is to represent information about geographical features, including their location and the relationships between them.

5.4.2.23.2 Overview

Control Architecture

Geography is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In order to meet a request placed upon it, Geography will provide information on Geographical_Features (e.g. what features of a particular type are within a particular region, the height of a building or the type of terrain), information on the relationship between one or more Geographical_Features (such as the range/bearing between two masts), and/or information on the relationship between Geographical_Features and Reference_Items.

Examples of Use

Geography can be used to:

- Identify the geographical region ownship is currently within.
- Determine potential vehicle terrain conflict when planning a route.

5.4.2.23.3 Service Summary

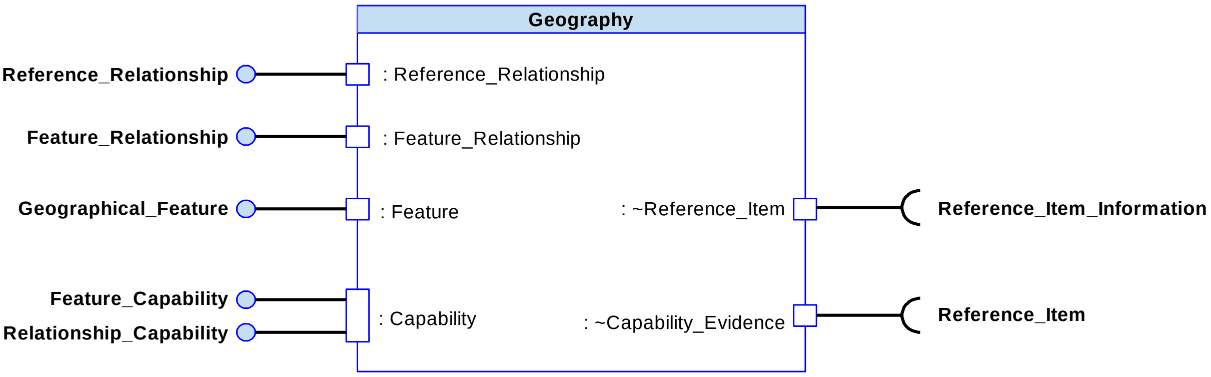


Figure 403: Geography Service Summary

5.4.2.23.4 Responsibilities

capture_geography_information_request

- To capture a request for information on Geographical_Features, the relationship between them, or the relationship between a Reference_Item and Geographical_Feature.

determine_characteristics

- To determine information about a Geographical_Feature (e.g. location, whether a terrain surface is wooded or rocky, or the magnetic variation of a location).

determine_feature_relationships

- To determine information on how Geographical_Features relate to one another (e.g. the distance between two Geographical_Features or what other Geographical_Features exist within a zonal feature).

determine_terrain_conflict

- To determine when a Reference_Item (e.g. ownship's projected route) conflicts with a Geographical_Feature.

determine_reference_relationship

- To determine information on the relationship between a Reference_Item (e.g. ownship position) and Geographical_Features.

assess_geography_service_capability

- To assess the Capability to provide information on Geographical_Features, relationships between Geographical_Features, and relationships between a Reference_Item and a Geographical_Feature.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

5.4.2.23.5 Subject Matter Semantics

The subject matter of Geography is Geographical_Features in the operating environment.

Exclusions

The subject matter of Geography does not include:

- The implications of a conflict between a Reference_Item and a Geographical_Feature.

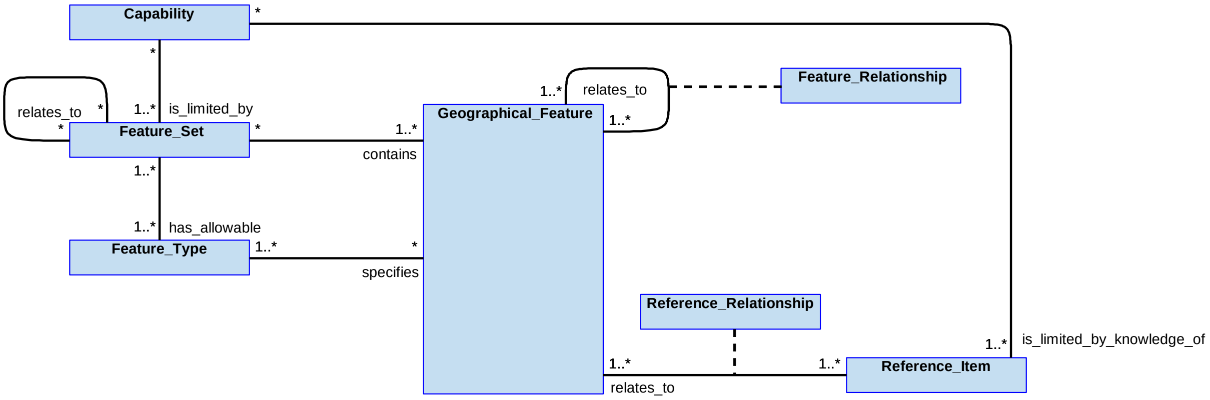


Figure 404: Geography Semantics

5.4.2.23.5.1 Entities

Capability

The range of services that can be performed using the information available, e.g. the ability to provide information on a particular region or Feature_Type of Geographical_Feature.

Feature_Relationship

The relationship between Geographical_Features, such as the distance between two buildings.

Feature_Set

A set of information on Geographical_Features, e.g. a terrain map or a military city map.

Feature_Type

A specific type of Geographical_Feature, e.g. a building, terrain, country border or territorial water border.

Geographical_Feature

A geographical feature of the Earth or a celestial feature that is observable from the Earth, e.g. a specific river, the land border between two countries, or a star.

Reference_Item

A specific item, or a reference to, that is the subject of a request, e.g. ownship or projected route.

Reference_Relationship

The relationship between a Geographical_Feature and a Reference_Item, e.g. the distance between ownship position and a bridge or the conflict between ownship projected route and terrain.

5.4.2.23.6 Design Rationale

5.4.2.23.6.1 Assumptions

- Geography reasons about terrain and obstructions data, and has access to geographical points of mission interest (e.g. intersection of rivers).
- Geography will be supplied with any spatial parameters (e.g. a location, area or volume) needed to constrain requests for information.
- Geographical_Features can be points, areas or volumes.
- Geographical_Features can be tangible (e.g. buildings) and intangible (e.g. a country border).

5.4.2.23.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Geography:

- Data Driving - This PYRAMID concept is applicable as Geographical_Features can be data-driven in order to support variation in the types and formats required for different system capabilities (e.g. a round 4/3 earth model of terrain is required for RF propagation calculations, whereas navigation could require a 1-to-1 flat earth model).

Exploitation Considerations

- Separate instances of the Geography component may be created where a build set is only concerned with specific Geographical Features. For example, a navigation system may only need to use magnetic variation and a ground proximity warning system may only need to use terrain and obstructions data.

5.4.2.23.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component would check that flightpaths do not conflict with terrain or obstructions. This could be performed both when determining the intended flightpath of the air vehicle (see the Vehicle Movement IV) and during execution of the flightpath (see the Avoidance IV). Consequently, failure of this component could result in inadvertent flight into terrain, i.e. an uncontrolled crash resulting in loss of air vehicle and fatalities. The flightpath would normally be planned to follow safe departure/approach profiles (provided by Environment Infrastructure) and for some aircraft would otherwise be above a Minimum Safe Altitude (MSA). Whilst these features will provide additional mitigation against inadvertent flight into terrain, they do not cover all circumstances. For example:
    - Threat, aircraft collision or weather avoidance manoeuvre occurs on approach. The air vehicle may need to deviate from the safe approach profile, but knowledge of the terrain and obstructions data provided by this component is expected to be used to ensure the manoeuvre does not impact the ground.
    - Air vehicles may need to operate below the MSA and the terrain and obstructions data is needed to check the intended path maintains separation from terrain. This case is not intended to cover air vehicles that are designed to follow the terrain at low level.

This rationale applies particularly to UAS. For Exploiting Programmes where more reliance may be placed on other barriers to "inadvertent flight into terrain" (e.g. for manned air vehicles the crew can see the ground directly), then the Exploiting Programme may require a less onerous DAL.

The DAL requirements are not expected to be less onerous when an air vehicle is designed and cleared for prolonged flight at low level.

5.4.2.23.6.4 Security Considerations

The indicative security classification is SNEO.

This component provides information about the Earth including the location of and relationships between geographical features, such information is widely available and considered O. The location of certain geographical features will have mission significance, e.g. the location of a target bridge will be understood, however the mission significance (that is to be destroyed) is not held within this component. This component does determine information about the relationship between a Reference_Item, such as the Exploiting Platform, and Geographical_Features though, and will have some operationally significant data about country borders and no-fly zones, etc. This is considered SNEO. This relationship is also used by other components to avoid conflict with terrain. Due to its use in safety and mission critical functions, the component will need to be of high integrity and availability.

The component may be expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to relationships with terrain during a mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is not expected to directly implement security enforcing functions.

5.4.2.23.7 Services

5.4.2.23.7.1 Service Definitions

5.4.2.23.7.1.1 Reference_Relationship

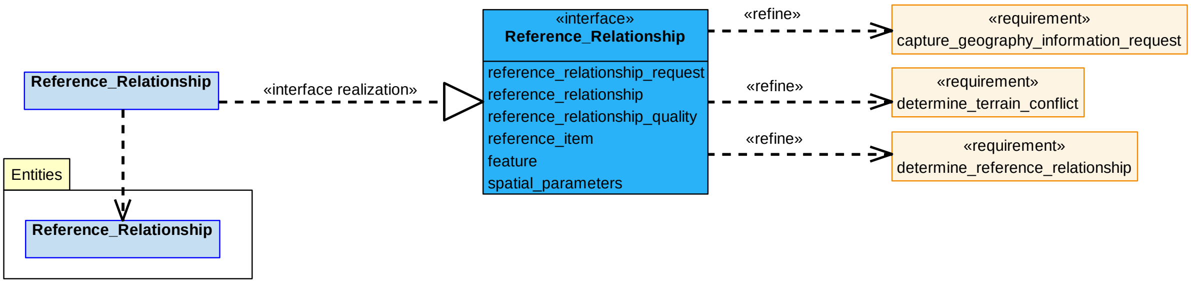


Figure 405: Reference_Relationship Service Definition

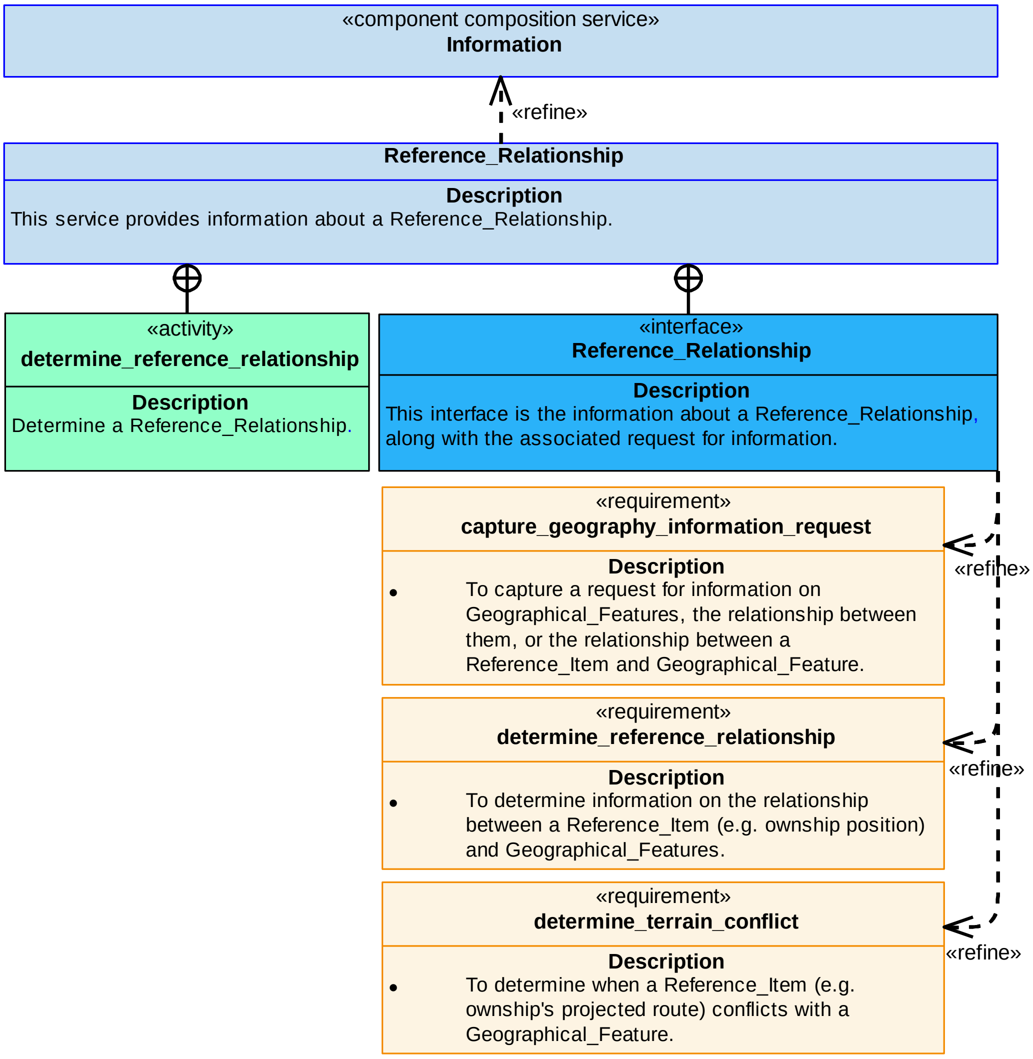

«component composition service»
Information
«refine»
Reference_Relationship
Description
This service provides information about a Reference_Relationship.
«activity»
determine_reference_relationship
Description
Determine a Reference_Relationship.
«interface»
Reference_Relationship
Description
This interface is the information about a Reference_Relationship, along with the associated request for information.
«requirement»
capture_geography_information_request
Description
To capture a request for information on Geographical_Features, the relationship between them, or the relationship between a Reference_Item and Geographical_Feature.
«refine»
«requirement»
determine_reference_relationship
Description
To determine information on the relationship between a Reference_Item (e.g. ownship position) and Geographical_Features.
«refine»
«requirement»
determine_terrain_conflict
Description
To determine when a Reference_Item (e.g. ownship's projected route) conflicts with a Geographical_Feature.
«refine»

Figure 406: Reference_Relationship Service Policy

Reference_Relationship

This service provides information about a Reference_Relationship.

Interface

Reference_Relationship

This interface is the information about a Reference_Relationship, along with the associated request for information.

Attributes

reference_relationship_request	The definition of the request for information about a Reference_Relationship, e.g. to provide the nature of a relationship between a Geographical_Feature and the Reference_Item.
reference_relationship	The details of the relationship between the Geographical_Feature and the Reference_Item needed to satisfy the reference_relationship_request, e.g. a conflict.

reference_relationship_quality	The quality of the reference_relationship, for example the precision of the distance between a Geographical_Feature and a Reference_Item.
reference_item	The Reference_Item which is the subject of a request.
feature	The Geographical_Feature which is the subject of a request.
spatial_parameters	The spatial parameters (e.g. a location, area or volume) that constrain the request.

Activity

determine_reference_relationship

Determine a Reference_Relationship.

5.4.2.23.7.1.2 Feature_Relationship

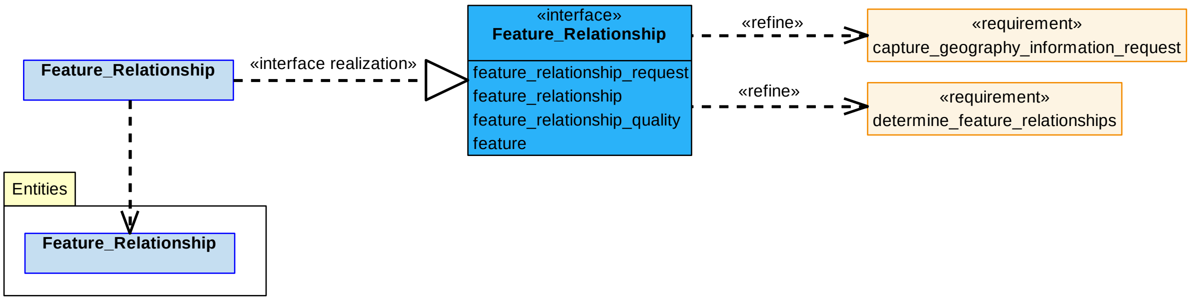


Figure 407: Feature_Relationship Service Definition

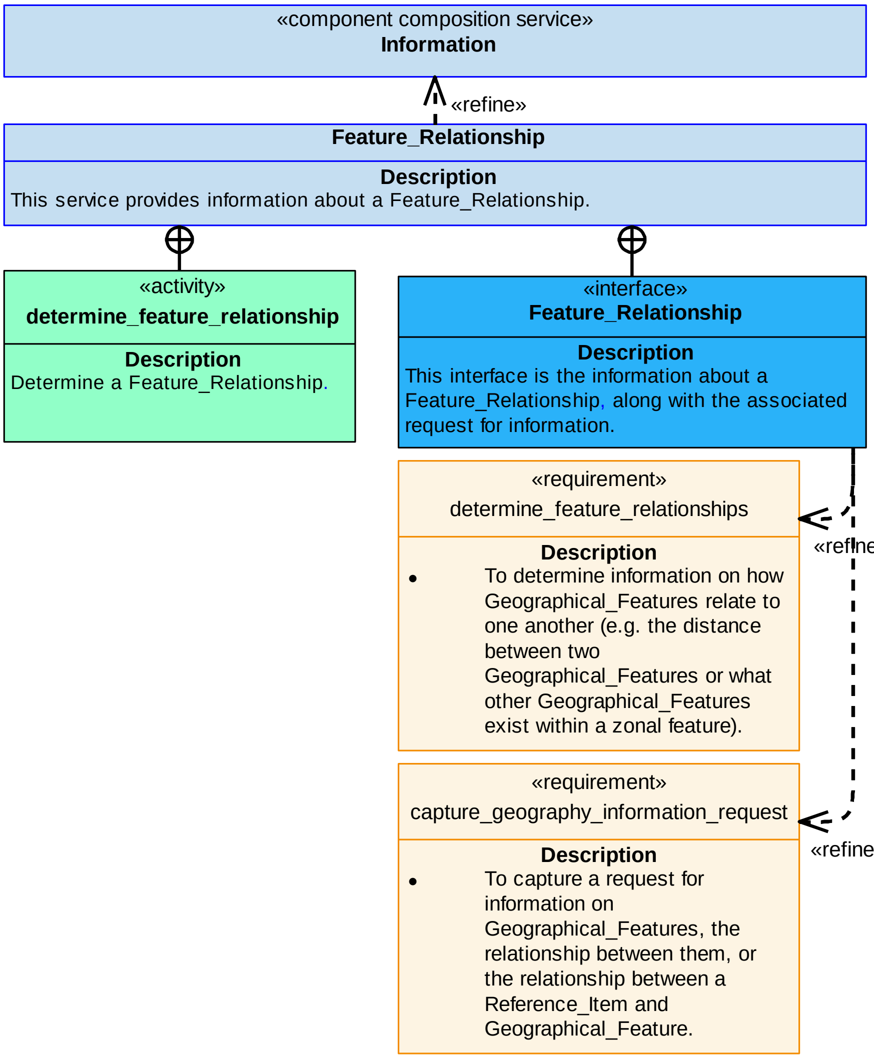

«component composition service»
Information
«refine»
Feature_Relationship
Description
This service provides information about a Feature_Relationship.
«activity»
determine_feature_relationship
Description
Determine a Feature_Relationship.
«interface»
Feature_Relationship
Description
This interface is the information about a Feature_Relationship, along with the associated request for information.
«requirement»
determine_feature_relationships
Description
To determine information on how Geographical_Features relate to one another (e.g. the distance between two Geographical_Features or what other Geographical_Features exist within a zonal feature).
«refine»
«requirement»
capture_geography_information_request
Description
To capture a request for information on Geographical_Features, the relationship between them, or the relationship between a Reference_Item and Geographical_Feature.
«refine»

Figure 408: Feature_Relationship Service Policy

Feature_Relationship

This service provides information about a Feature_Relationship.

Interface

Feature_Relationship

This interface is the information about a Feature_Relationship, along with the associated request for information.

Attributes

feature_relationship_request	The definition of the request for information about a Feature_Relationship, e.g. whether two bridges are over the same river.
feature_relationship	The details of the relationship between the Geographical_Features, e.g. two bridges are over the same river.

feature_relationship_quality	The quality of the feature_relationship, for example the precision of the distance between two Geographical_Features.
feature	The Geographical_Feature which is the subject of a request.

Activity

determine_feature_relationship

Determine a Feature_Relationship.

5.4.2.23.7.1.3 Feature

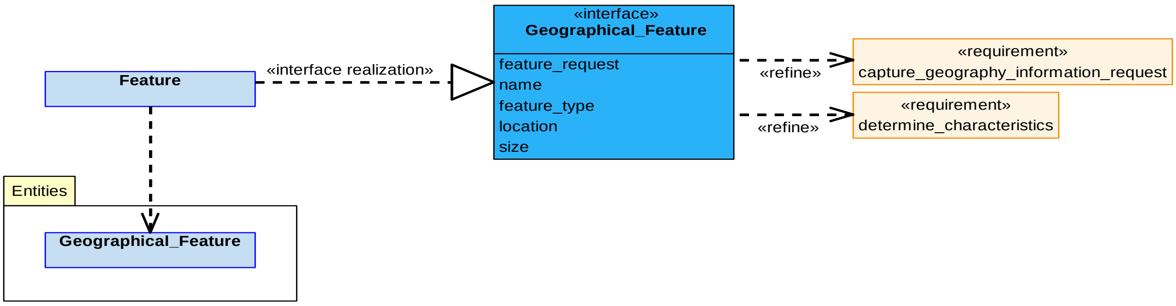


Figure 409: Feature Service Definition

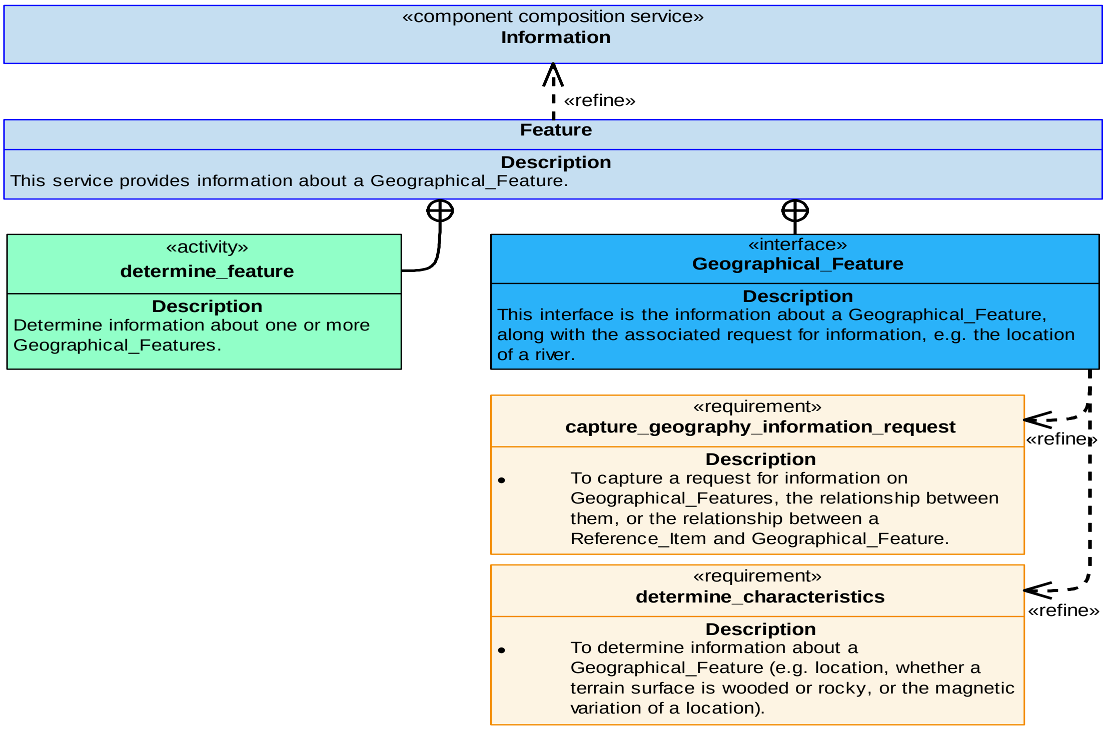

«component composition service»
Information
«refine»
Feature
Description
This service provides information about a Geographical_Feature.
«activity»
determine_feature
Description
Determine information about one or more Geographical_Features.
«interface»
Geographical_Feature
Description
This interface is the information about a Geographical_Feature, along with the associated request for information, e.g. the location of a river.
«requirement»
capture_geography_information_request
Description
• To capture a request for information on Geographical_Features, the relationship between them, or the relationship between a Reference_Item and Geographical_Feature.
«refine»
«requirement»
determine_characteristics
Description
• To determine information about a Geographical_Feature (e.g. location, whether a terrain surface is wooded or rocky, or the magnetic variation of a location).
«refine»

Figure 410: Feature Service Policy

Feature

This service provides information about a Geographical_Feature.

Interface

Geographical_Feature

This interface is the information about a Geographical_Feature, along with the associated request for information, e.g. the location of a river.

Attributes

feature_request	The definition of the request for information about a Geographical_Feature.
name	The identifier of the Geographical_Feature, e.g. the name of a specific river or mountain.
feature_type	The type of Geographical_Feature, e.g. a building, terrain, country border or territorial water border.
location	The location of the Geographical_Feature on the Earth.
size	The size and extent of the Geographical_Feature.

Activity

determine_feature

Determine information about one or more Geographical_Features.

5.4.2.23.7.1.4 Reference_Item

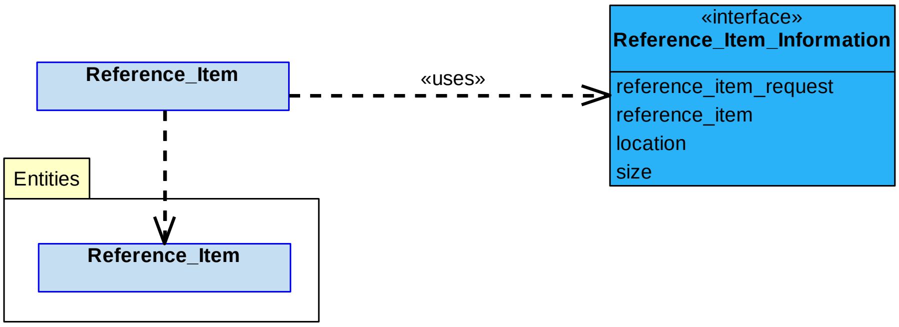


Figure 411: Reference_Item Service Definition

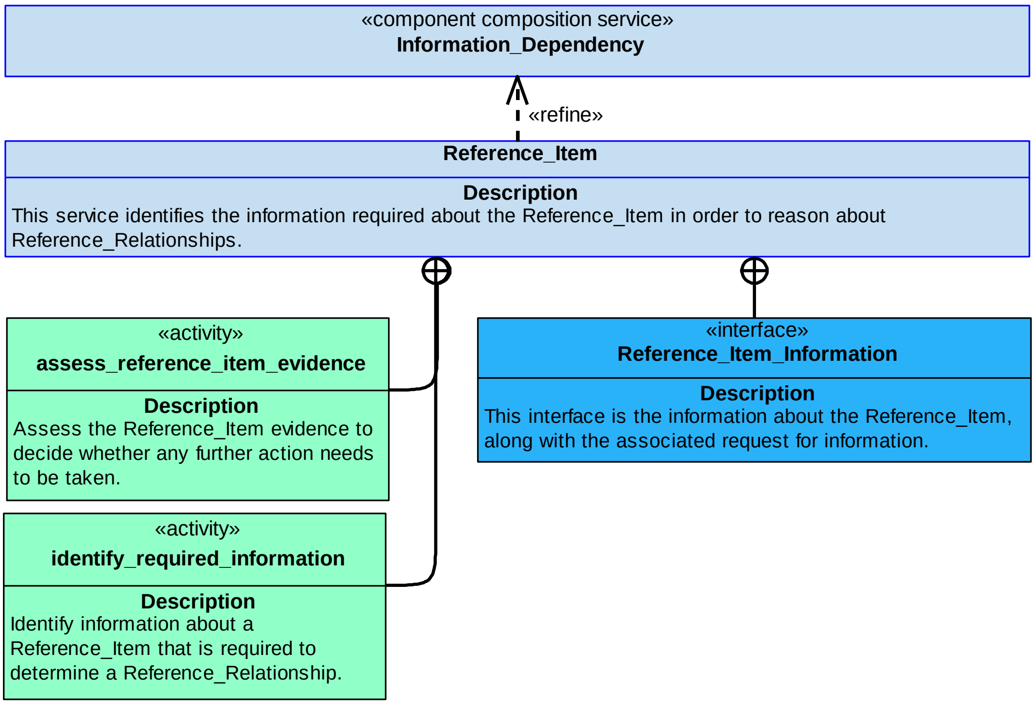

«component composition service»
Information_Dependency
«refine»
Reference_Item
Description
This service identifies the information required about the Reference_Item in order to reason about Reference_Relationships.
«activity»
assess_reference_item_evidence
Description
Assess the Reference_Item evidence to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify information about a Reference_Item that is required to determine a Reference_Relationship.
«interface»
Reference_Item_Information
Description
This interface is the information about the Reference_Item, along with the associated request for information.

Figure 412: Reference_Item Service Policy

Reference_Item

This service identifies the information required about the Reference_Item in order to reason about Reference_Relationships.

Interface

Reference_Item_Information

This interface is the information about the Reference_Item, along with the associated request for information.

Attributes

reference_item_request	The definition of the request for information about a Reference_Item.
reference_item	The Reference_Item which is the subject of a request.
location	The location of the Reference_Item.
size	The size and extent of the Reference_Item.

Activities

assess_reference_item_evidence

Assess the Reference_Item evidence to decide whether any further action needs to be taken.

identify_required_information

Identify information about a Reference_Item that is required to determine a Reference_Relationship.

5.4.2.23.7.1.5 Capability

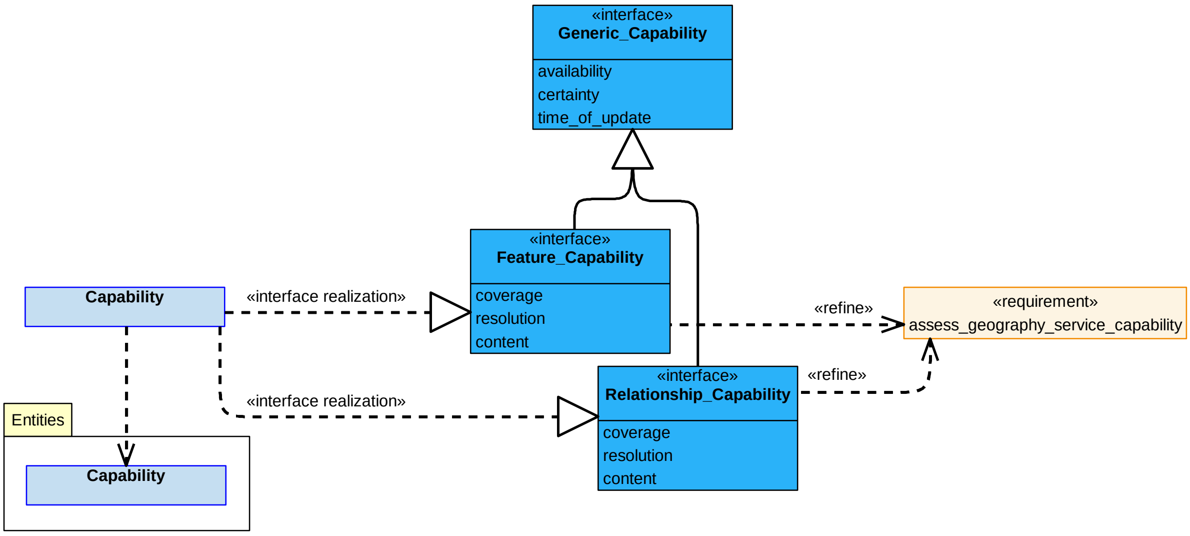
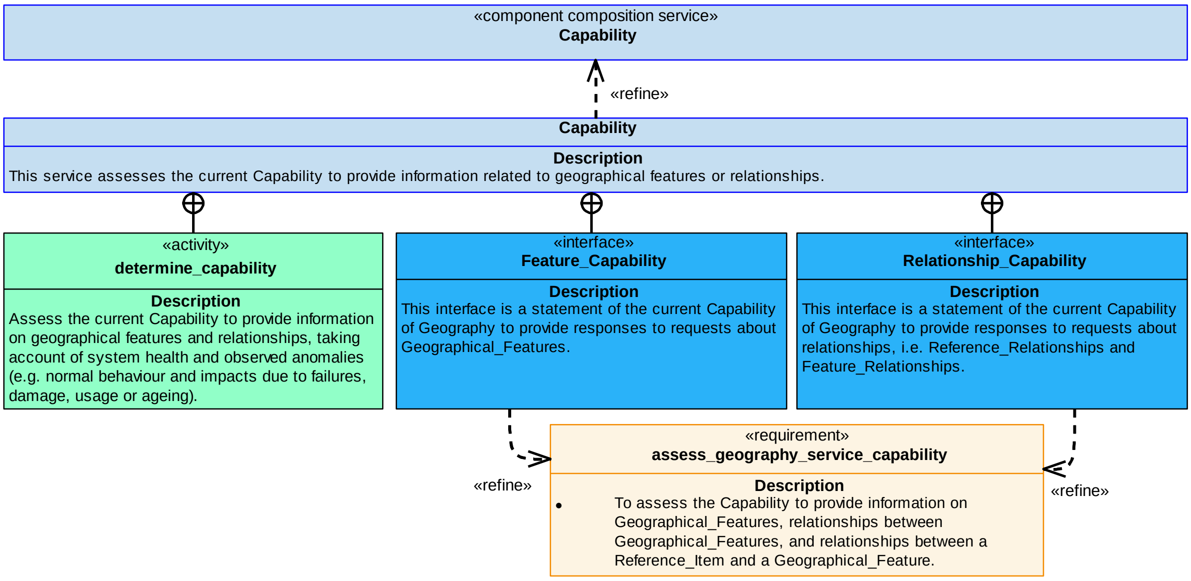


Figure 413: Capability Service Definition



Description
This service assesses the current Capability to provide information related to geographical features or relationships.

Description
Assess the current Capability to provide information on geographical features and relationships, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Description
This interface is a statement of the current Capability of Geography to provide responses to requests about Geographical_Features.

Description
This interface is a statement of the current Capability of Geography to provide responses to requests about relationships, i.e. Reference_Relationships and Feature_Relationships.

Description
- To assess the Capability to provide information on Geographical_Features, relationships between Geographical_Features, and relationships between a Reference_Item and a Geographical_Feature.

Figure 414: Capability Service Policy

Capability

This service assesses the current Capability to provide information related to geographical features or relationships.

Interfaces

Feature_Capability

This interface is a statement of the current Capability of Geography to provide responses to requests about Geographical_Features.

Attributes

coverage	The spatial extent of the Feature_Set data available.
resolution	The resolution of the Feature_Set data available, e.g. terrain data resolution.
content	The range of Feature_Type within the Feature_Set data available.

Relationship_Capability

This interface is a statement of the current Capability of Geography to provide responses to requests about relationships, i.e. Reference_Relationships and Feature_Relationships.

Attributes

coverage	The spatial extent of the Feature_Set data available.
resolution	The resolution of the Feature_Set data available, e.g. terrain data resolution.
content	The range of Feature_Type within the Feature_Set data available.

Activity

determine_capability

Assess the current Capability to provide information on geographical features and relationships, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.23.7.1.6 Capability_Evidence

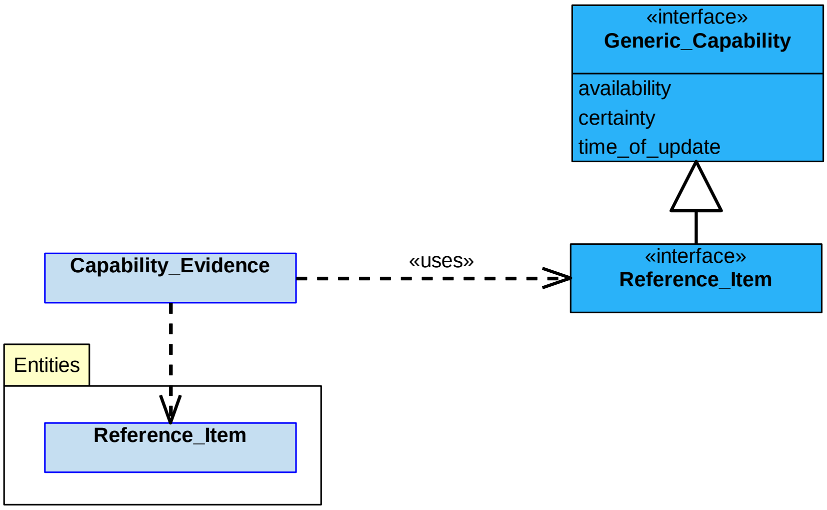


Figure 415: Capability_Evidence Service Definition

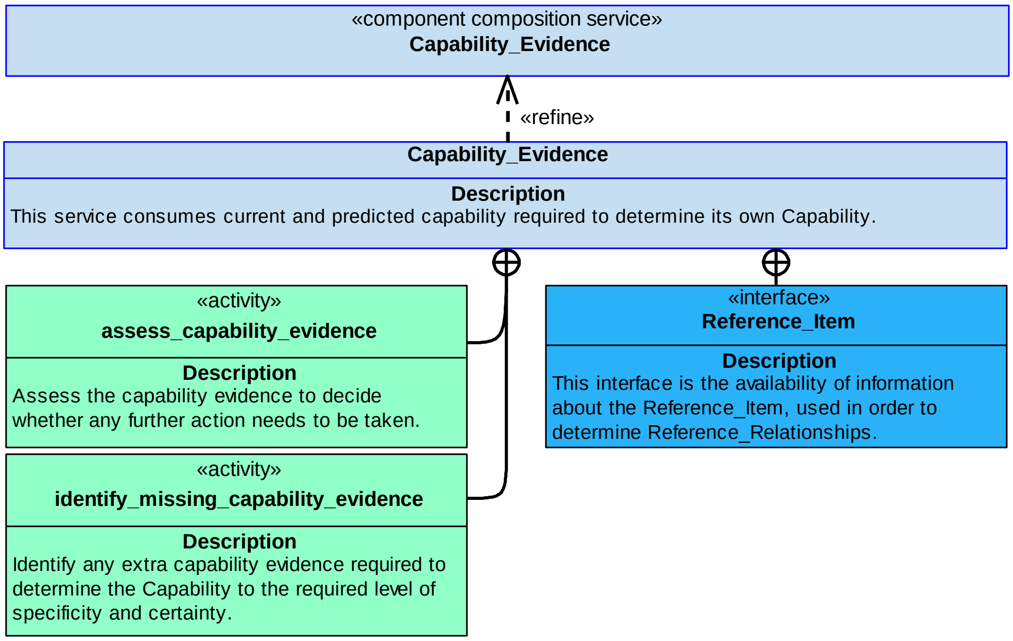

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability required to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Reference_Item
Description
This interface is the availability of information about the Reference_Item, used in order to determine Reference_Relationships.

Figure 416: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability required to determine its own Capability.

Interface

Reference_Item

This interface is the availability of information about the Reference_Item, used in order to determine Reference_Relationships.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.23.7.2 Service Dependencies

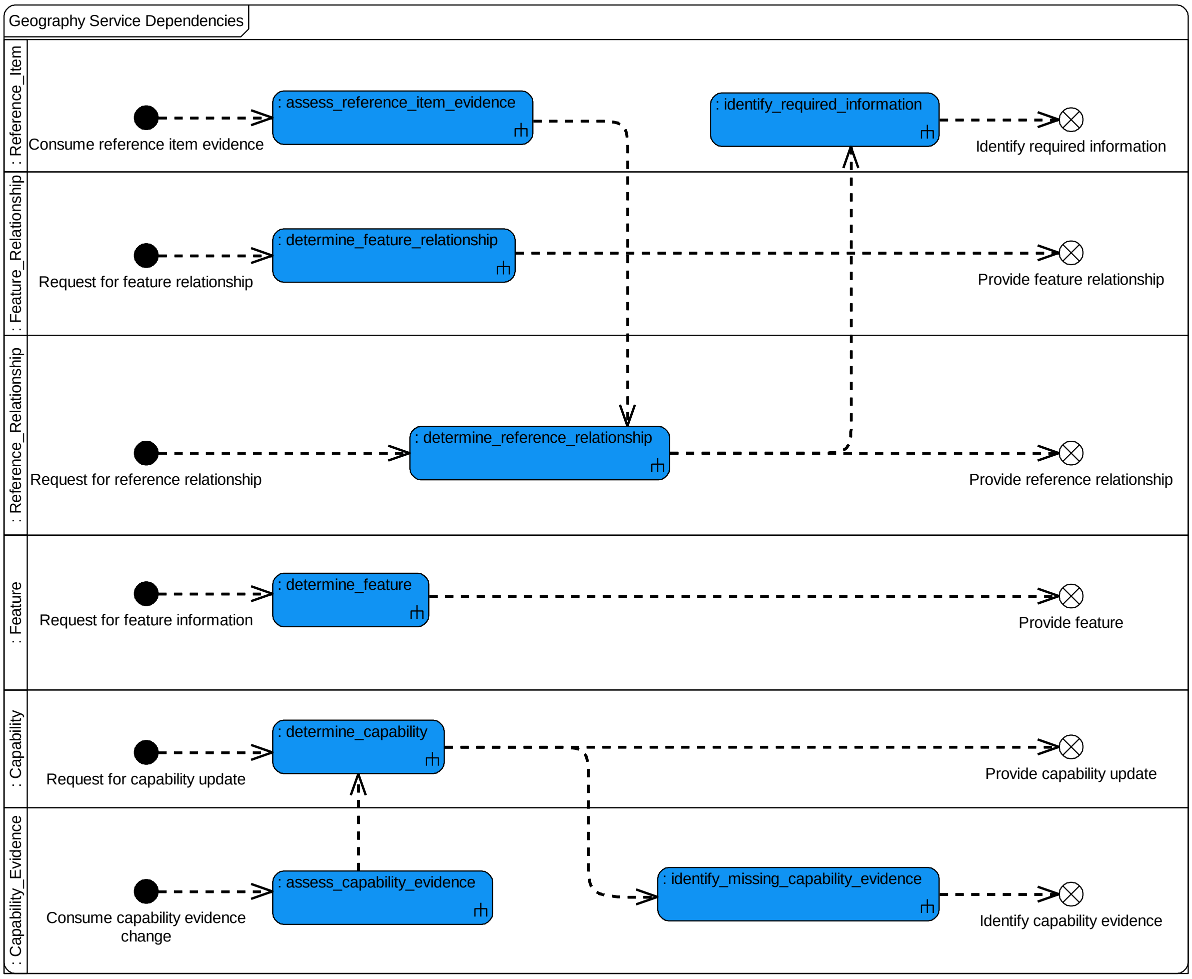

Geography Service Dependencies
: Reference_Item
: Feature_Relationship
: Reference_Relationship
: Feature
: Capability
: Capability_Evidence
Consume reference item evidence
: assess_reference_item_evidence
: identify_required_information
Identify required information
Request for feature relationship
: determine_feature_relationship
Provide feature relationship
Request for reference relationship
: determine_reference_relationship
Provide reference relationship
Request for feature information
: determine_feature
Provide feature
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 417: Geography Service Dependencies

5.4.2.24 Health Assessment

5.4.2.24.1 Role

The role of Health Assessment is to identify when a system hardware item has been degraded due to failure, damage, usage or ageing.

5.4.2.24.2 Overview

Control Architecture

Health Assessment is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

There are two main standard patterns of use:

- An anomaly had been detected and Health Assessment is being requested to determine the health of system hardware and report its health status.
- Health Assessment uses information from health Sensors and/or information about the Exploiting Platform use and environment to determine the life and usage of system hardware.

Examples of Use

Health Assessment can be used when:

- Other components need to know about a health change to aid in assessing their capability.
- It is required to determine likely causes of a recognised change of capability.
- Life and Usage information is required, for example in support of structural health reporting.

5.4.2.24.3 Service Summary

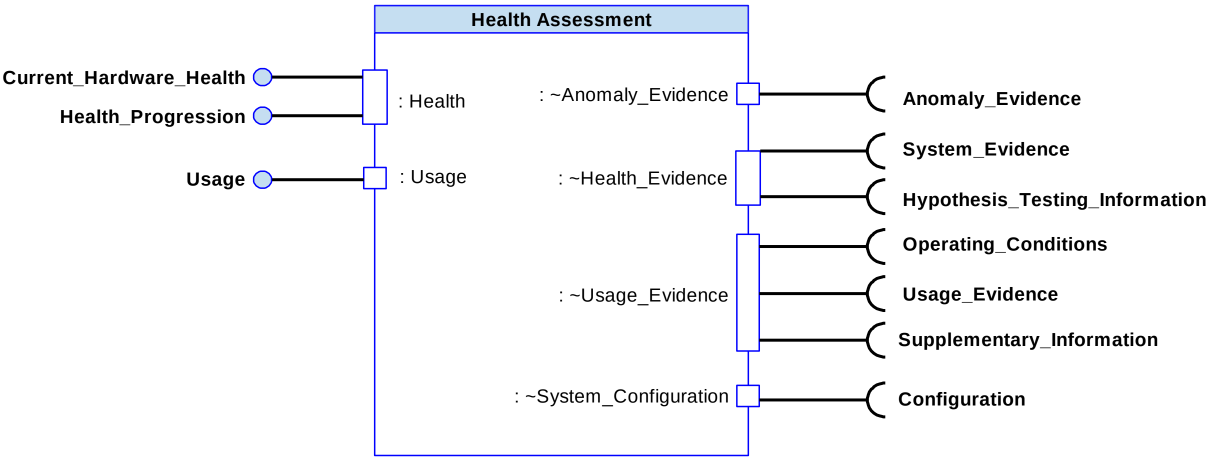


Figure 418: Health Assessment Service Summary

5.4.2.24.4 Responsibilities

predict_degradation

- To predict the progression of Hardware degradation over time and with use.

determine_health_change

- To determine if a health change (degradation or improvement) has caused anomalous system behaviour.

determine_life_consumed

- To identify the extent to which the Hardware's life has been consumed (calendar time remaining to next service, etc.).

determine_usage

- To identify how much the Hardware has been used (flying hours used, hard landings experienced, etc.).

identify_extent_of_degradation

- To identify the extent to which Hardware has been degraded by Failure, Damage, Usage or ageing.

identify_missing_information_to_improve_health_solution

- To identify Missing_Health_Information which could improve the certainty or specificity of the health assessment.

5.4.2.24.5 Subject Matter Semantics

The subject matter of Health Assessment is the health of hardware elements.

Exclusions

The subject matter of Health Assessment does not include:

- Concerns about functional capability as a consequence of a Health_State change.

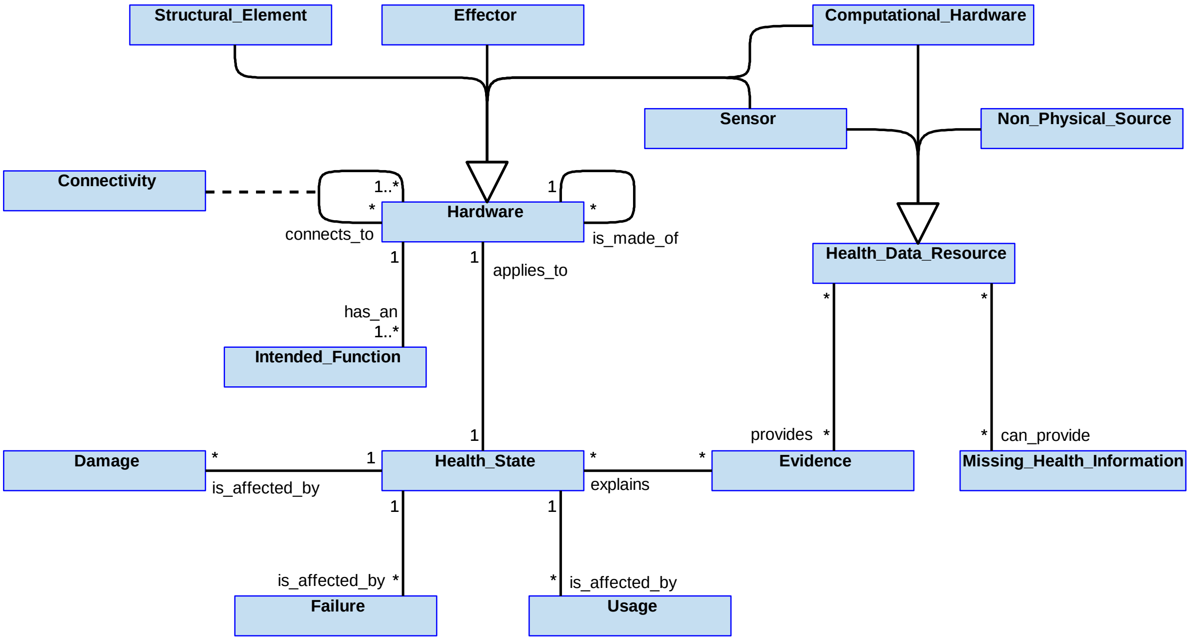


Figure 419: Health Assessment Semantics

5.4.2.24.5.1 Entities

Computational_Hardware

Simple or complex hardware on which computations can be performed.

Connectivity

The arrangement of, and relationship between, hardware elements.

Damage

Physical harm that impacts the hardware's ability to carry out its intended functions.

Effector

A physical element that can directly cause a change to the physical environment. For example, an actuator.

Evidence

Evidence about the health of something.

Failure

The state of being unable to carry out an intended function.

Hardware

Either a specific equipment, part, or structural element, or a group of these.

Health_Data_Resource

A resource used to provide health data.

For example:

- Equipment capable of BIT.

- A diagnostic sensor.

- Something that can provide information that can be used for understanding health issues.

Health_State

An assessment of the state of health of a piece of hardware.

Intended_Function

The purpose(s) for which something has been designed.

Missing_Health_Information

Information that has not currently been obtained but could be used to determine the health or improve the health assessment.

Non_Physical_Source

A non-physical element that provides health data. For example, a software entity that reasons about another part of a system.

Sensor

A physical element that measures the environment or other hardware. For example, a temperature gauge.

Structural_Element

A physical element whose purpose is to contain or transmit loads, contents or electricity. For example, a pipe, cable, or aircraft structure.

Usage

How the hardware has been used.

5.4.2.24.6 Design Rationale

5.4.2.24.6.1 Assumptions

- A consistent source of time data is available so that the order of commands and sensor readings can be determined precisely (see Design Considerations).
- The component is not responsible for the classification of amalgamated data presented to the maintainer.
- Performance data for equipment of higher classification will not need to be assessed for health purposes.
- The Test component will request BIT of other components/hardware.

5.4.2.24.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Health Assessment:

- Health Management - This PYRAMID concept is important in understanding how this component is used.
- Capability Management - This component supports the capability assessment of other components but does not provide an evolving view of its own capabilities.
- Recording and Logging - This PYRAMID concept defines how data retention will be managed, especially for audit purposes.

Extensions

- Health Assessment could utilise extensions to cover different functionalities in a hierarchical manner, with each extension being responsible for assessing the heath of a particular area of the system. The Health Assessment extensions would work together to identify health changes which gave rise to the observed anomalies.

Exploiting Considerations

- Implementations of Health Assessment must take into account any mechanism by which one hardware element can affect another, and hence can propagate the effects of a failure through the system. Incidental and unintended effects must be included. Failure Modes and Effects Analysis identifies such mechanisms, but the more the details can be captured directly from the system design, the better.
- Health Assessment outputs should include an assessment of the certainty of their conclusions. Outputs might not be fully specific: for example, if a power generator has a number of sub-elements, Health Assessment may be certain that the power generator has failed, but not be able to identify a specific sub-element as the cause.
- Health Assessment should recognise the limits of precision of time information, to avoid being confused about the order of very closely-timed events: for example, whether a sensor reading was taken before or after a failure occurred.
- Health Assessment should ignore the data from any sensor that it has identified as having failed.
- Different instances, different variants, or different extensions of this component that are data-driven differently will be required to comply with ISO13374 (Condition Monitoring and Diagnostics of Machines) Ref. [13]. See Health Management PYRAMID concept.

5.4.2.24.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component may result a "large reduction in safety margins", which has a critical severity.

Therefore, the indicative DAL is B.

The rationale behind this is to ensure high level and rigorous safety analysis, which significantly reduces the chance of the Health Assessment component misdiagnosing incorrect data, or no data, on the usage/life of a hardware element, e.g. the airframe. This discards the need for any mitigations which would be required if a lower DAL was specified.

Where this component proposed additional tests to further refine the capability assessment then the Test component would not allow the tests that compromised safety (in the current state of the air system) to be performed.

5.4.2.24.6.4 Security Considerations

The indicative security classification is O-S, however the component(s) with which it is associated will be a significant factor.

This component will require limited information about the system hardware/infrastructure, its connectivity and its functions, etc. in order to be able to identify current health and any future degradation. It is considered unlikely to handle data that is individually above O-S (health data need not contain performance information, for example), however in some cases, data aggregation within the component may be a consideration in raising its classification. It is probable that there will be multiple instances of the component residing in security domains relevant to the system elements being assessed, and these instances may need to communicate in order to help identify root causes, etc.

Hardware that is failing or indeed failed may expose system vulnerabilities to an attacker and therefore preventive or corrective actions will help maintain the security of the system. The confidentiality of information that might divulge these vulnerabilities should be protected.

The component satisfies security related functions relating to:

- **Logging of Security Data**, this will assist with subsequent forensic examination of events such as system shut-down or pauses, which might then point to the presence of a cyber attack or other breach.
- **System Status and Monitoring** of hardware/infrastructure elements.
- Provision of **Warnings and Notifications** that give awareness of unexpected activity.

Is unlikely to directly implement security enforcing functions, but can support **HW Authentication** (including possible tamper detection) and the continued form and fitness integrity of the system.

5.4.2.24.7 Services

5.4.2.24.7.1 Service Definitions

5.4.2.24.7.1.1 Health

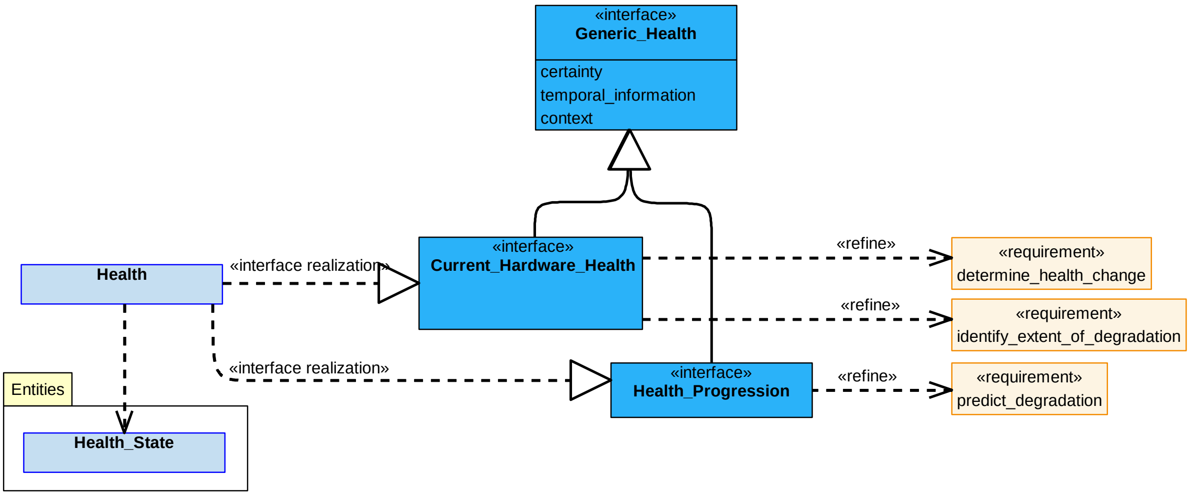


Figure 420: Health Service Definition

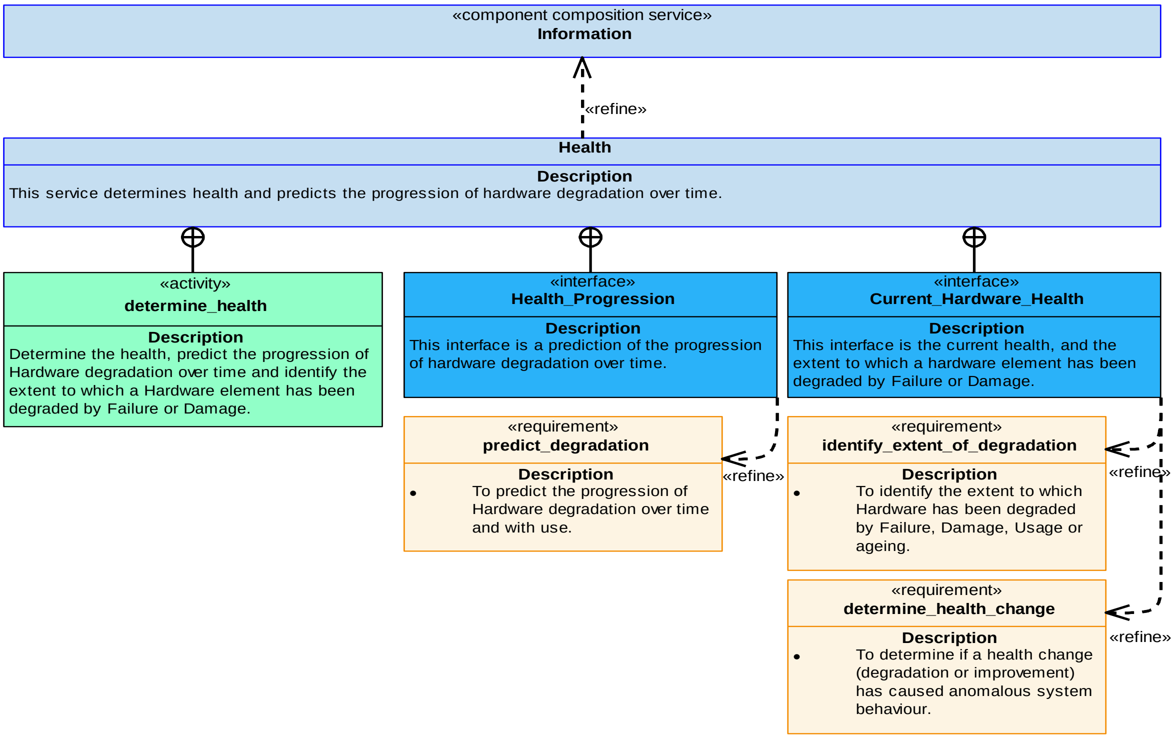

«component composition service»
Information
«refine»
Health
Description
This service determines health and predicts the progression of hardware degradation over time.
«activity»
determine_health
Description
Determine the health, predict the progression of Hardware degradation over time and identify the extent to which a Hardware element has been degraded by Failure or Damage.
«interface»
Health_Progression
Description
This interface is a prediction of the progression of hardware degradation over time.
«interface»
Current_Hardware_Health
Description
This interface is the current health, and the extent to which a hardware element has been degraded by Failure or Damage.
«requirement»
predict_degradation
Description
• To predict the progression of Hardware degradation over time and with use.
«refine»
«requirement»
identify_extent_of_degradation
Description
• To identify the extent to which Hardware has been degraded by Failure, Damage, Usage or ageing.
«refine»
«requirement»
determine_health_change
Description
• To determine if a health change (degradation or improvement) has caused anomalous system behaviour.
«refine»

Figure 421: Health Service Policy

Health

This service determines health and predicts the progression of hardware degradation over time.

Interfaces

Current_Hardware_Health

This interface is the current health, and the extent to which a hardware element has been degraded by Failure or Damage.

Health_Progression

This interface is a prediction of the progression of hardware degradation over time.

Activity

determine_health

Determine the health, predict the progression of Hardware degradation over time and identify the extent to which a Hardware element has been degraded by Failure or Damage.

5.4.2.24.7.1.2 Usage

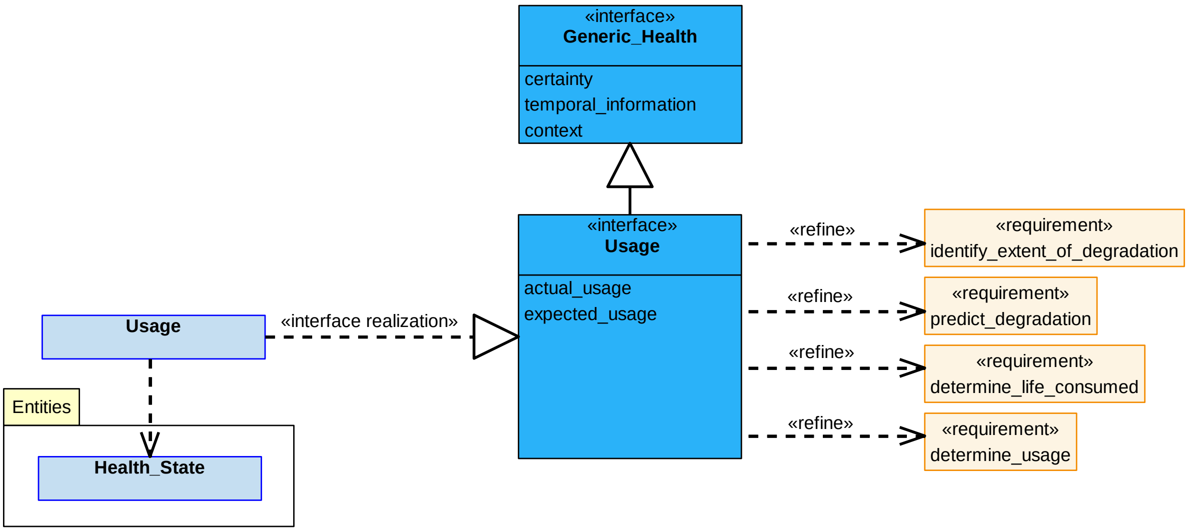


Figure 422: Usage Service Definition

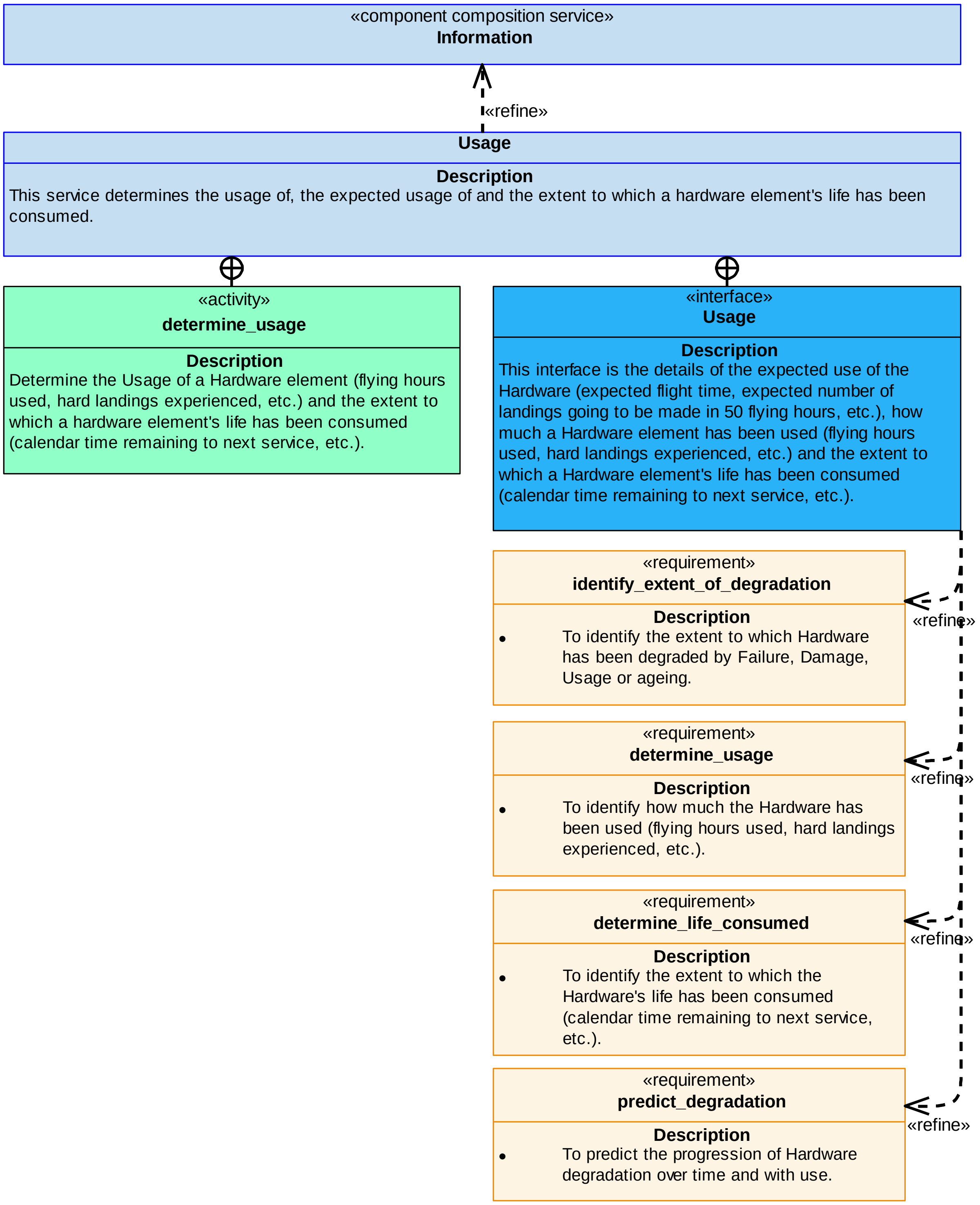

«component composition service»
Information
«refine»
Usage
Description
This service determines the usage of, the expected usage of and the extent to which a hardware element's life has been consumed.
«activity»
determine_usage
Description
Determine the Usage of a Hardware element (flying hours used, hard landings experienced, etc.) and the extent to which a hardware element's life has been consumed (calendar time remaining to next service, etc.).
«interface»
Usage
Description
This interface is the details of the expected use of the Hardware (expected flight time, expected number of landings going to be made in 50 flying hours, etc.), how much a Hardware element has been used (flying hours used, hard landings experienced, etc.) and the extent to which a Hardware element's life has been consumed (calendar time remaining to next service, etc.).
«requirement»
identify_extent_of_degradation
Description
• To identify the extent to which Hardware has been degraded by Failure, Damage, Usage or ageing.
«refine»
«requirement»
determine_usage
Description
• To identify how much the Hardware has been used (flying hours used, hard landings experienced, etc.).
«refine»
«requirement»
determine_life_consumed
Description
• To identify the extent to which the Hardware's life has been consumed (calendar time remaining to next service, etc.).
«refine»
«requirement»
predict_degradation
Description
• To predict the progression of Hardware degradation over time and with use.
«refine»

Figure 423: Usage Service Policy

Usage

This service determines the usage of, the expected usage of and the extent to which a hardware element's life has been consumed.

Interface

Usage

This interface is the details of the expected use of the Hardware (expected flight time, expected number of landings going to be made in 50 flying hours, etc.), how much a Hardware element has been used (flying hours used, hard landings experienced, etc.) and the extent to which a Hardware element's life has been consumed (calendar time remaining to next service, etc.).

Attributes

actual_usage	How a system element has been used, e.g. number of flight hours or number of landings made in 50 flying hours.
expected_usage	The expected Usage of a system element, e.g. expected flight time or expected number of landings going to be made in 50 flying hours.

Activity

determine_usage

Determine the Usage of a Hardware element (flying hours used, hard landings experienced, etc.) and the extent to which a hardware element's life has been consumed (calendar time remaining to next service, etc.).

5.4.2.24.7.1.3 Anomaly_Evidence

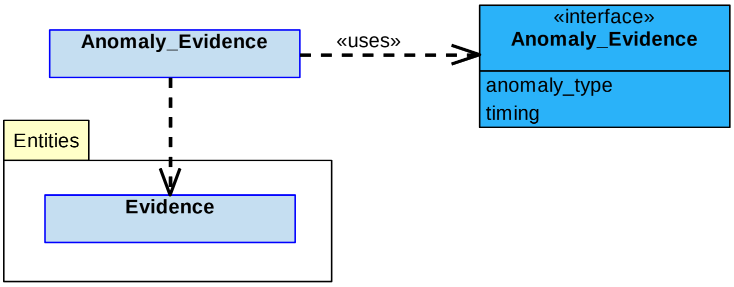


Figure 424: Anomaly_Evidence Service Definition

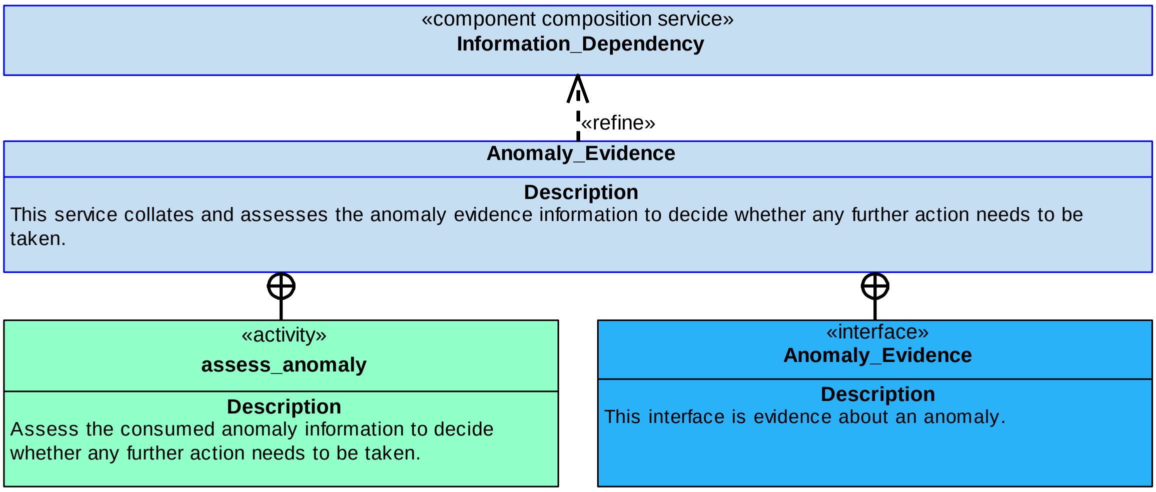

«component composition service»
Information_Dependency
«refine»
Anomaly_Evidence
Description
This service collates and assesses the anomaly evidence information to decide whether any further action needs to be taken.
«activity»
assess_anomaly
Description
Assess the consumed anomaly information to decide whether any further action needs to be taken.
«interface»
Anomaly_Evidence
Description
This interface is evidence about an anomaly.

Figure 425: Anomaly_Evidence Service Policy

Anomaly_Evidence

This service collates and assesses the anomaly evidence information to decide whether any further action needs to be taken.

Interface

Anomaly_Evidence

This interface is evidence about an anomaly.

Attributes

anomaly_type	A description of the anomaly.
timing	Temporal information, such as the persistence of an anomaly (e.g. duration) or time of occurrence.

Activity

assess_anomaly

Assess the consumed anomaly information to decide whether any further action needs to be taken.

5.4.2.24.7.1.4 Health_Evidence

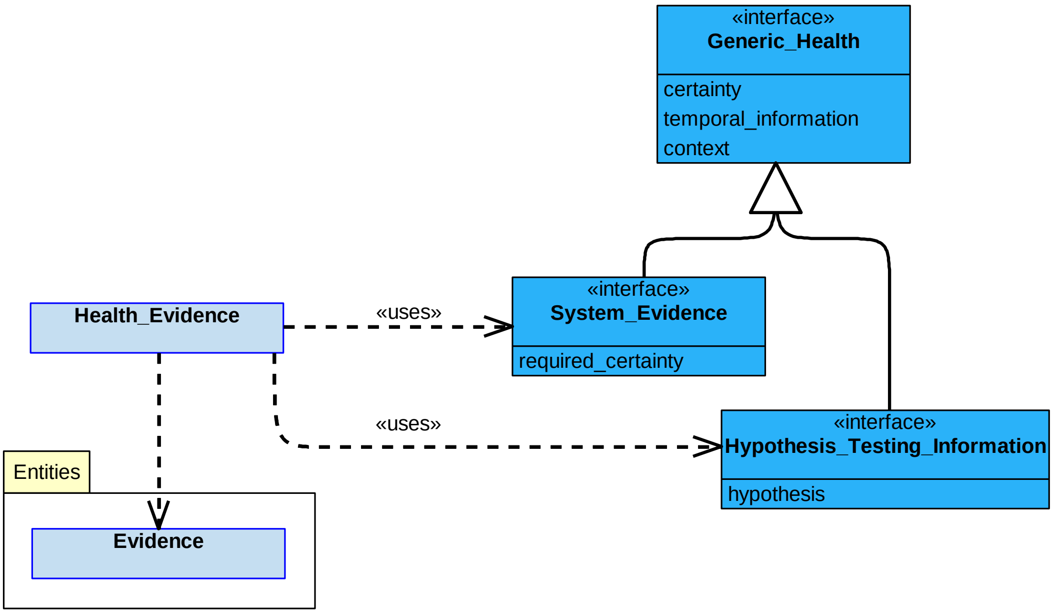


Figure 426: Health_Evidence Service Definition

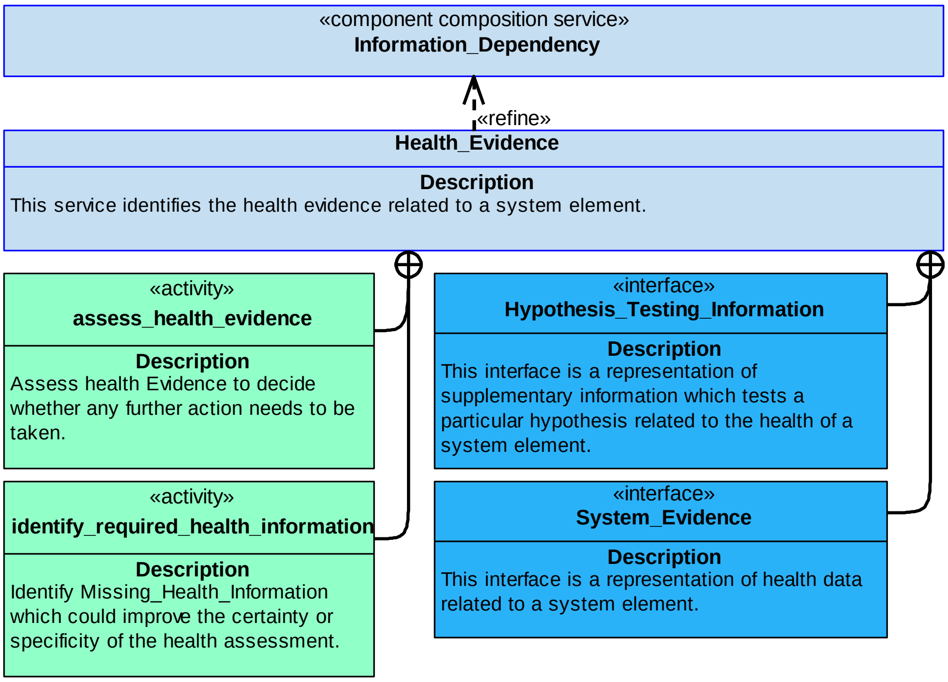

«component composition service»
Information_Dependency
«refine»
Health_Evidence
Description
This service identifies the health evidence related to a system element.
«activity»
assess_health_evidence
Description
Assess health Evidence to decide whether any further action needs to be taken.
«activity»
identify_required_health_information
Description
Identify Missing_Health_Information which could improve the certainty or specificity of the health assessment.
«interface»
Hypothesis_Testing_Information
Description
This interface is a representation of supplementary information which tests a particular hypothesis related to the health of a system element.
«interface»
System_Evidence
Description
This interface is a representation of health data related to a system element.

Figure 427: Health_Evidence Service Policy

Health_Evidence

This service identifies the health evidence related to a system element.

Interfaces

System_Evidence

This interface is a representation of health data related to a system element.

Attribute

required_certainty	The required level of certainty of the health data.

Hypothesis_Testing_Information

This interface is a representation of supplementary information which tests a particular hypothesis related to the health of a system element.

Attribute

hypothesis	The hypothesis which the supplementary information will test. For example, a hypothesis that an element has failed in a specific way.

Activities

identify_required_health_information

Identify Missing_Health_Information which could improve the certainty or specificity of the health assessment.

assess_health_evidence

Assess health Evidence to decide whether any further action needs to be taken.

5.4.2.24.7.1.5 Usage_Evidence

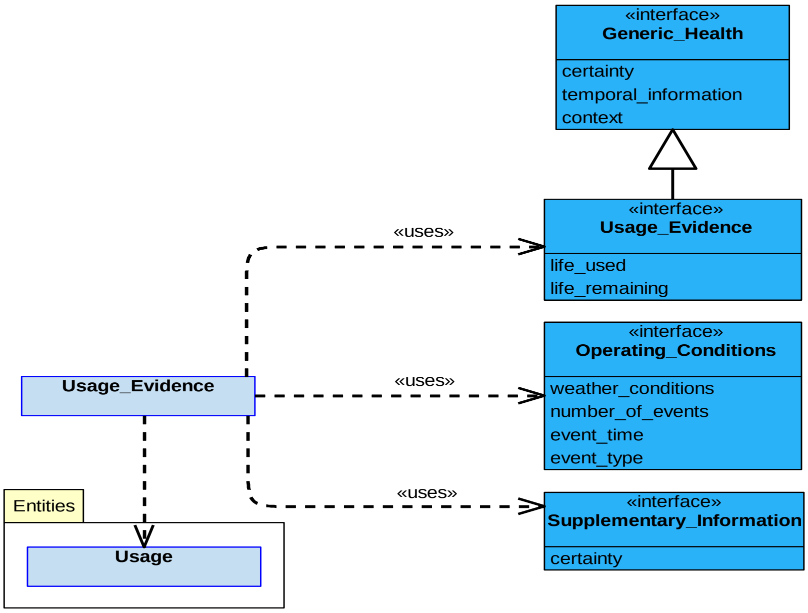


Figure 428: Usage_Evidence Service Definition

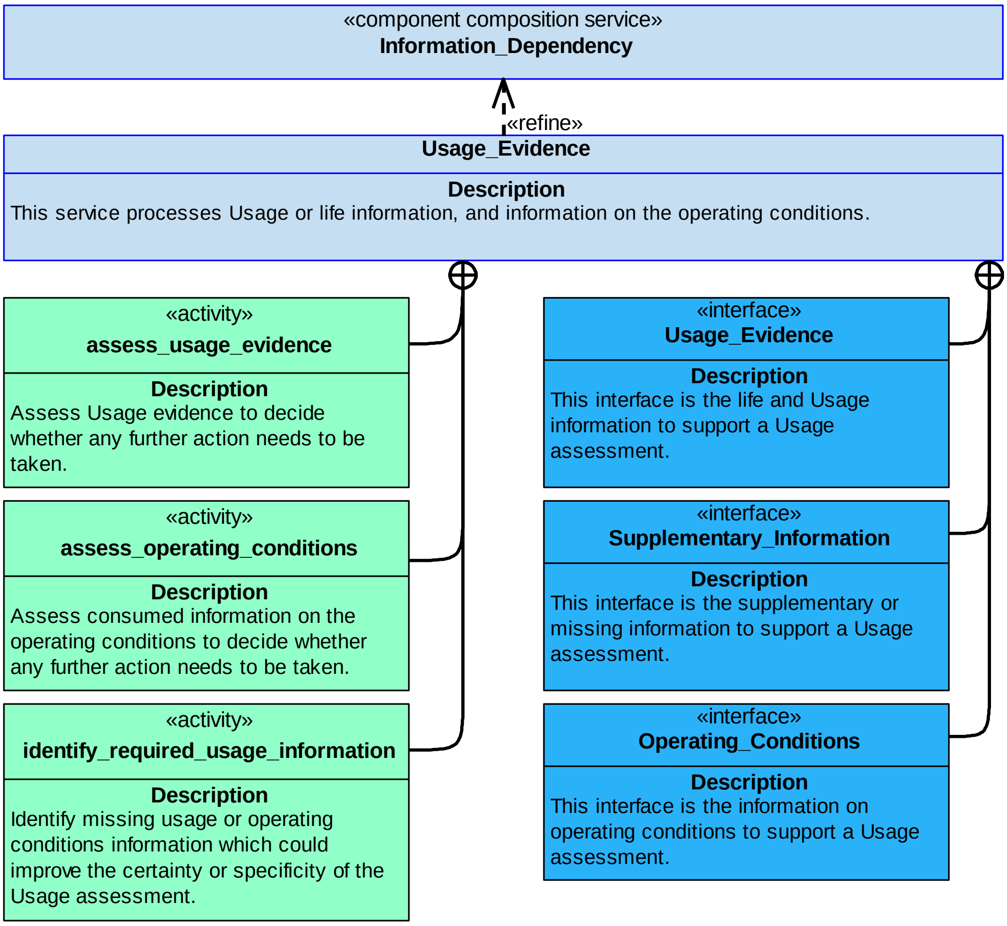

«component composition service»
Information_Dependency
«refine»
Usage_Evidence
Description
This service processes Usage or life information, and information on the operating conditions.
«activity»
assess_usage_evidence
Description
Assess Usage evidence to decide whether any further action needs to be taken.
«activity»
assess_operating_conditions
Description
Assess consumed information on the operating conditions to decide whether any further action needs to be taken.
«activity»
identify_required_usage_information
Description
Identify missing usage or operating conditions information which could improve the certainty or specificity of the Usage assessment.
«interface»
Usage_Evidence
Description
This interface is the life and Usage information to support a Usage assessment.
«interface»
Supplementary_Information
Description
This interface is the supplementary or missing information to support a Usage assessment.
«interface»
Operating_Conditions
Description
This interface is the information on operating conditions to support a Usage assessment.

Figure 429: Usage_Evidence Service Policy

Usage_Evidence

This service processes Usage or life information, and information on the operating conditions.

Interfaces

Usage_Evidence

This interface is the life and Usage information to support a Usage assessment.

Attributes

life_used	The amount of life consumed, e.g. number of flying hours or time since fitted.
life_remaining	The amount of life remaining, e.g. number of flying hours or time until next service.

Operating_Conditions

This interface is the information on operating conditions to support a Usage assessment.

Attributes

weather_conditions	The state of a type of atmospheric condition at a given time and place.
number_of_events	The number of events of a particular type, such as the number of missile firings.
event_time	The duration of an event. For example, flight time or the length of time that a piece of equipment was powered up.
event_type	A type of event experienced by the platform. For example, weight-on-wheels or high-g manoeuvre.

Supplementary_Information

This interface is the supplementary or missing information to support a Usage assessment.

Attribute

certainty	The level of certainty of the reported information.

Activities

assess_usage_evidence

Assess Usage evidence to decide whether any further action needs to be taken.

assess_operating_conditions

Assess consumed information on the operating conditions to decide whether any further action needs to be taken.

identify_required_usage_information

Identify missing usage or operating conditions information which could improve the certainty or specificity of the Usage assessment.

5.4.2.24.7.1.6 System_Configuration

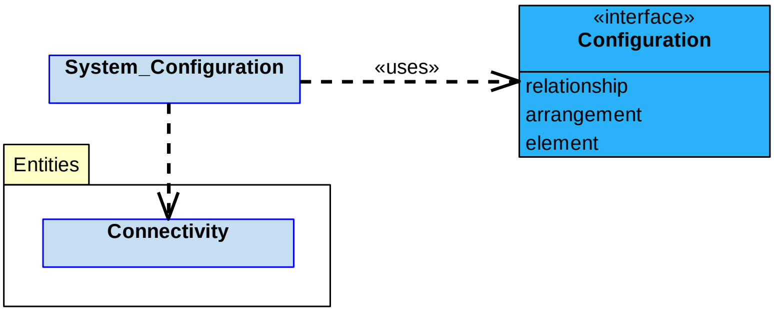
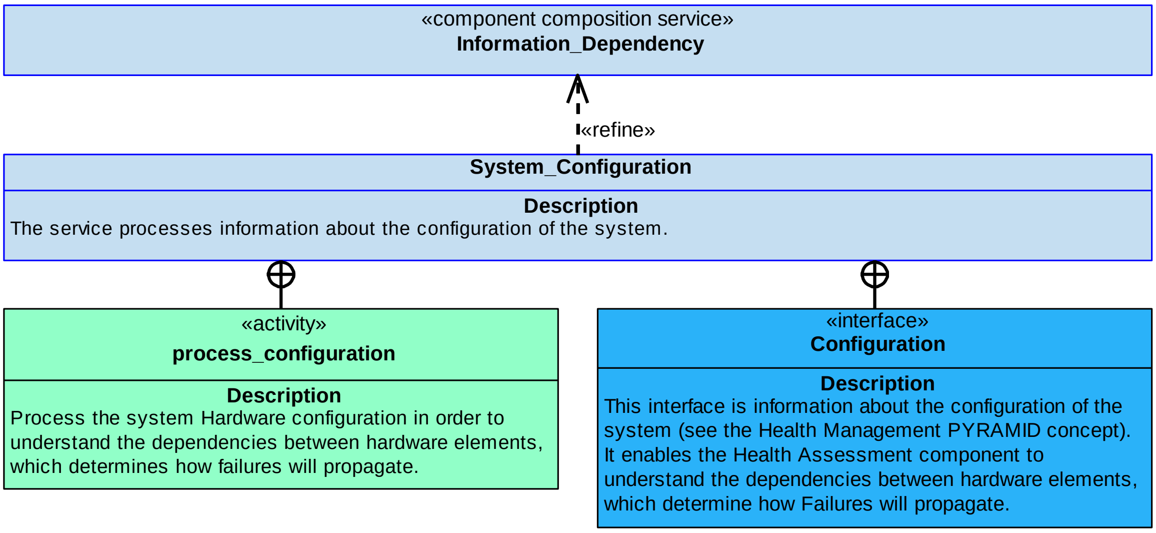


Figure 430: System_Configuration Service Definition



The service processes information about the configuration of the system.

Process the system Hardware configuration in order to understand the dependencies between hardware elements, which determines how failures will propagate.

This interface is information about the configuration of the system (see the Health Management PYRAMID concept). It enables the Health Assessment component to understand the dependencies between hardware elements, which determine how Failures will propagate.

Figure 431: System_Configuration Service Policy

System_Configuration

The service processes information about the configuration of the system.

Interface

Configuration

This interface is information about the configuration of the system (see the Health Management PYRAMID concept). It enables the Health Assessment component to understand the dependencies between hardware elements, which determine how Failures will propagate.

Attributes

relationship	The relationship between Hardware elements.
arrangement	The arrangement of Hardware elements. For example, whether two elements are co-located in an area.
element	A piece of Hardware within the system.

Activity

process_configuration

Process the system Hardware configuration in order to understand the dependencies between hardware elements, which determines how failures will propagate.

5.4.2.24.7.2 Data Model

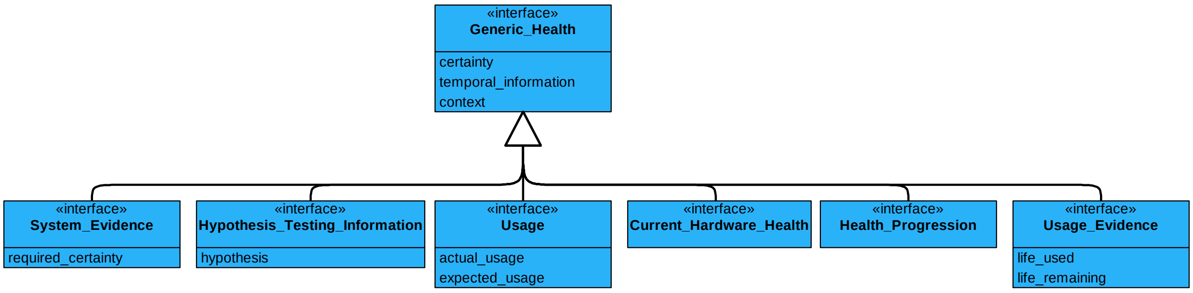


Figure 432: Health Assessment Data Model

The diagram shows a Generic_Health interface which is specialised to show how the Health Assessment component's services can cater for all aspects of health assessment. The generic interface allows the components to communicate essential information without reference to specialisations.

Generic_Health

This interface is a generic expression of health assessment that is specialised for different services within the Health Assessment component definition.

Attributes

certainty	The certainty of the health assessment.
temporal_information	Temporal information, such as the persistence of degradation of a system element (for example glitching) or how the health is expected to progress over time.
context	What aspect of system health the evidence relates to.

5.4.2.24.7.3 Service Dependencies

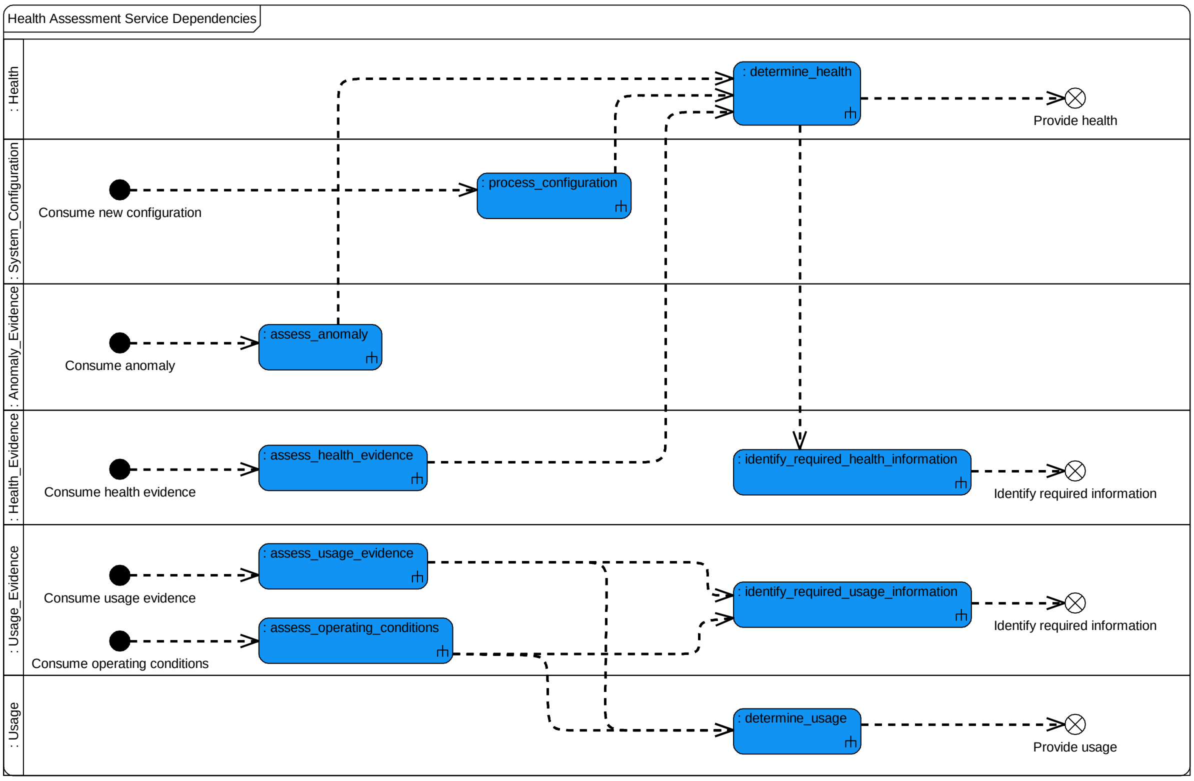


Figure 433: Health Assessment Service Dependencies

5.4.2.25 HMI Dialogue

5.4.2.25.1 Role

The role of HMI Dialogue is to manage and curate the information required for interactions between a system and its users.

5.4.2.25.2 Overview

Control Architecture

HMI Dialogue is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

HMI Dialogue takes Source_Data and makes it presentable to the receiving Participant, taking into account the context of the Dialogue_Interaction and Comprehension_Rules. This includes data being provided by the system from the user, as well as that from the system being presented to the operator.

Examples of Use

HMI Dialogue will be used to curate the information flow to enable the comprehension of Source_Data to the designated Participant, such as:

- Directing commands to the applicable system Participant, including sequence dependent instructions, e.g. to recognise user authentication or to map a user role.
- Enabling the Dialogue_Interactions required for data interpretation and to support decision making, e.g. coordinating the required data for a weight or relative speed value for a requestor or collating a status report.

5.4.2.25.3 Service Summary

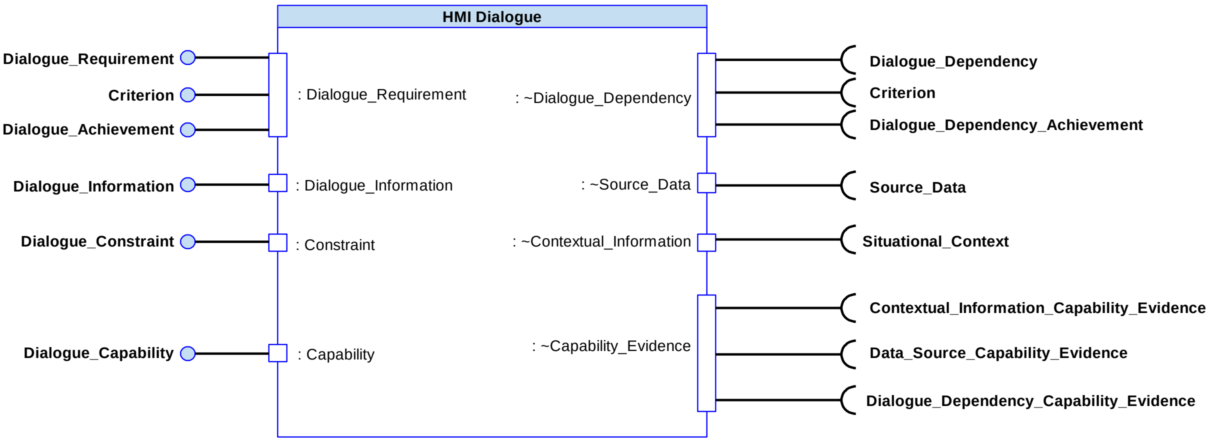


Figure 434: HMI Dialogue Service Summary

5.4.2.25.4 Responsibilities

capture_dialogue_requirements

- To capture Requirements for provision of Presentable_Information to a consumer Participant. For example, a requirement for a user request to be interpreted and information provided to the appropriate system elements; or for information from multiple components to be processed together so as to be understood by a user.

capture_provided_data_constraints

- To capture Constraints on the provided data that limit the extent of possible dialogue.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Capability, Constraints, Context and progress of the Interaction_Sequence.

determine_comprehension_rules

- To determine Comprehension_Rules for Source_Data in the context of the associated dialogue, within any Constraints.

request_participant_interaction

- To request Participants to perform a Dialogue_Interaction which provides additional Source_Data or Context to support a dialogue.

identify_interaction_pre-conditions

- To identify Pre-conditions required for a Dialogue_Interaction.

fulfil_dialogue_requirement

- To derive Presentable_Information in support of dialogue.

identify_dialogue_progress

- To identify progress of an Interaction_Sequence and how it relates to achievement against the Requirement. For example, the effect of a halt in user input or system data provision.

capture_information_dependencies

- To capture information dependencies for Dialogue_Interactions, such as a Context change or the condition of Participants.

capture_participant_relationships

- To capture Participant relationships and associated Comprehension_Rules.

assess_dialogue_capability

- To assess the component's Capability to provide a Dialogue_Interaction, taking account of system health and observed anomalies.

identify_missing_information

- To identify missing information, including Context, which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the component's Capability over time and with use.

5.4.2.25.5 Subject Matter Semantics

The subject matter of HMI Dialogue is the Comprehension_Rules for Dialogue_Interaction between Participants in a given context.

Exclusions

The subject matter of HMI Dialogue does not include:

- The mechanisms of information presentation, only the provision and availability of information to be presented.

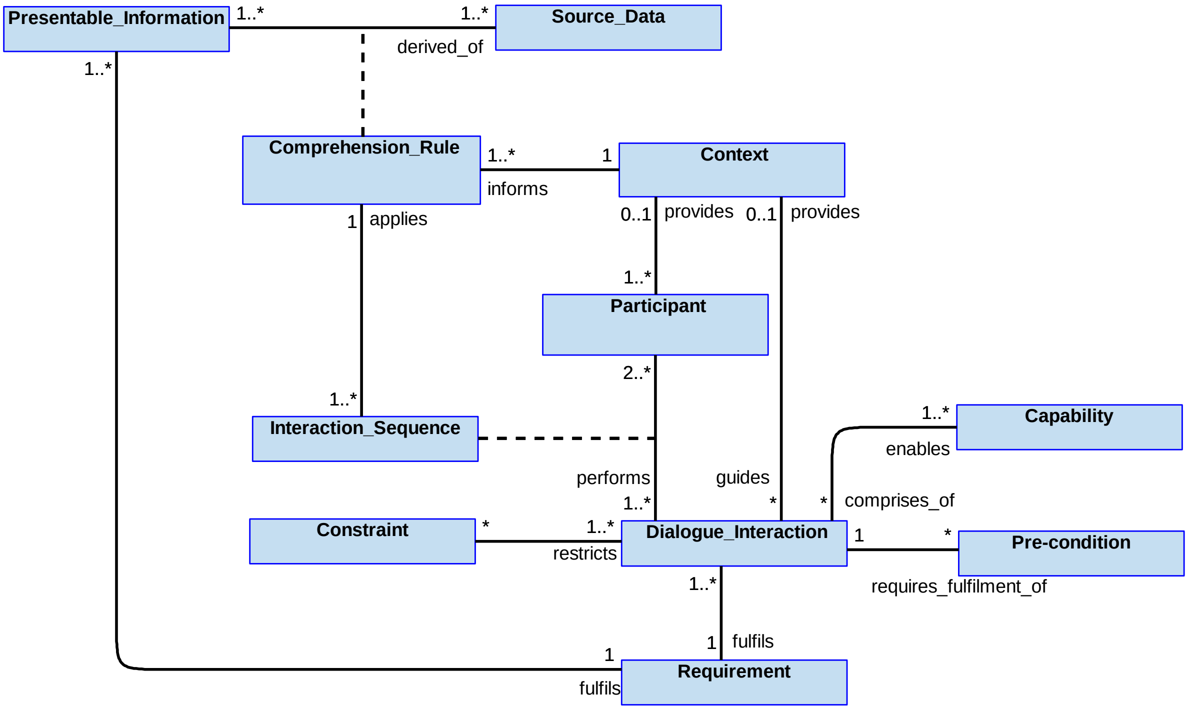


Figure 435: HMI Dialogue Semantics

5.4.2.25.5.1 Entities

Capability

The ability to enable comprehension of HMI information when required.

Comprehension_Rule

A system rule for converting data into information that is comprehensible to its consumer. For example, the rules for how user input can be interpreted and how relevant consumers of the input are chosen; or the rules for preparing system data to make it presentable to a user, such as the need to compare fuel volume data to determine if the current volume is sufficient.

Constraint

A constraint on dialogue which restricts the information that can be delivered. Factors that may cause constraints include the availability, freshness and security level of data.

Context

An aspect of system-level context that may affect a dialogue. For example, a task state, user role, user permissions, or participant history; all of which can affect how data is interpreted in a specific interaction.

Dialogue_Interaction

An individual interaction that is part of a dialogue, including the provision of data that contributes to the dialogue's context. For example, the exchange of signals that establishes a dialogue link, or the provision of data to be compiled from multiple sources.

Interaction_Sequence

The order that dialogue interactions are enacted, which may affect the meaning of the resulting information (as with the order of inputs required for a passcode, or the order of mathematical operations).

Participant

Either a provider of data or consumer of information in a dialogue. For example, a user providing a request or consuming information; a system element consuming a request, providing data or providing context to a dialogue.

Pre-condition

A condition on which the ability to have an interaction is dependant, such as whether the required communication connections are in place.

Presentable_Information

Information that is ready to present to its consumer. For example a command to the system from an operator or information from the system to the operator.

Source_Data

Data that has been provided through the HMI by a user, or system data that may be made available through the HMI. For example, sound data provided by a user, or vehicle status data provided by the system.

Requirement

A requirement for a dialogue that provides appropriate information to a user or system consumer. For example, dialogue may be required so that a system element receives the meaning of a user input; or so that the meaning of a variety of data from multiple system sources is comprehensible for a user.

5.4.2.25.6 Design Rationale

5.4.2.25.6.1 Assumptions

- This component supports user login and role management through managing provision of the login and role assignment processes (see User Management IV).

5.4.2.25.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining HMI Dialogue:

- Human-Machine Interface - The HMI PYRAMID concept is fundamental to this component.
- Data Driving - Configuration data can support the Context of Dialogue_Interactions that constitute a dialogue, and this enables the Presentable_Information derived by this component to be better tailored for the consuming user or system element Participant. This is important where a Participant can only recognise or understand certain forms or types of information.
- Component Connections - This component collates Source_Data and Context from the exploiting system through connections with other components.
- Recording and Logging - Logging of Dialogue_Interactions will be performed in accordance with this PYRAMID concept.

Exploitation Considerations

- A Requirement may specify that a dialogue should enable a certain level of autonomy, and this will require the resulting Presentable_Information to be of a certain level of detail to support automated decision making by the system.
- A Participant may recognise or understand only certain forms or types of information. For example, a system element may only be able to comprehend specific instructions.

5.4.2.25.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component provides the interface between a user and the system - both for control inputs and display. Failure of this component could, as a worst case, cause catastrophic consequences, including erroneous inputs to the flight controls system, inadvertent weapon release commands, or omission of critical information being made presentable.

It is expected that multiple implementations and instances of this component may be created in a PYRAMID deployment. Analysis of the specific uses of each instance, by the Exploiting Programme, may justify a less onerous DAL for some instances.

5.4.2.25.6.4 Security Considerations

The indicative security classification is notionally O.

This component will pass information between the operator and the system and as such the indicative security classification is dependent on the classification of information being processed. Where different classifications of data may be handled by the same instance, it will be at the highest of those classifications. It should be treated as per the confidentiality, integrity and availability needs of the interactions taking place.

It is expected that this component will be required within security domains that interface to a user. Where there are multiple security domains and multiple instances or variants of the component, these may need to communicate with each other. Separation will be enforced by a boundary protection function located outside the component.

The component is expected to at least partially satisfy security related functions by:

- Sharing information based upon the needs of the **Classification of Data** involved and the specific interaction in which it is shared.
- **Maintaining Audit Records** to support non-repudiation of events based on the information shared and when it was shared.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is expected to perform some aspects of security enforcing functions relating to:

- **Restricting Access to Data** through its involvement in determining the Context in which an interaction is carried out, this includes only sharing information suitable for the clearance of the operator, etc.
- **User Login and Authentication** processes; whilst this component does not perform authentication, it is part of the user interface by which authentication is provided, role allocations and handover performed, etc.

5.4.2.25.7 Services

5.4.2.25.7.1 Service Definitions

5.4.2.25.7.1.1 Dialogue_Requirement

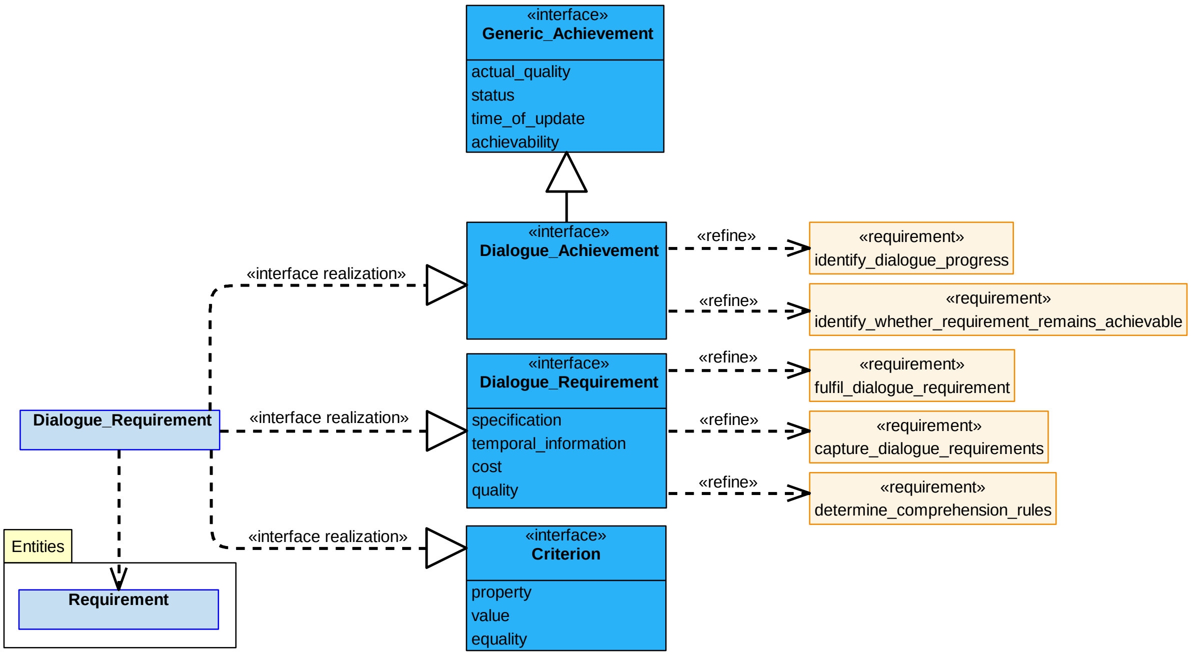
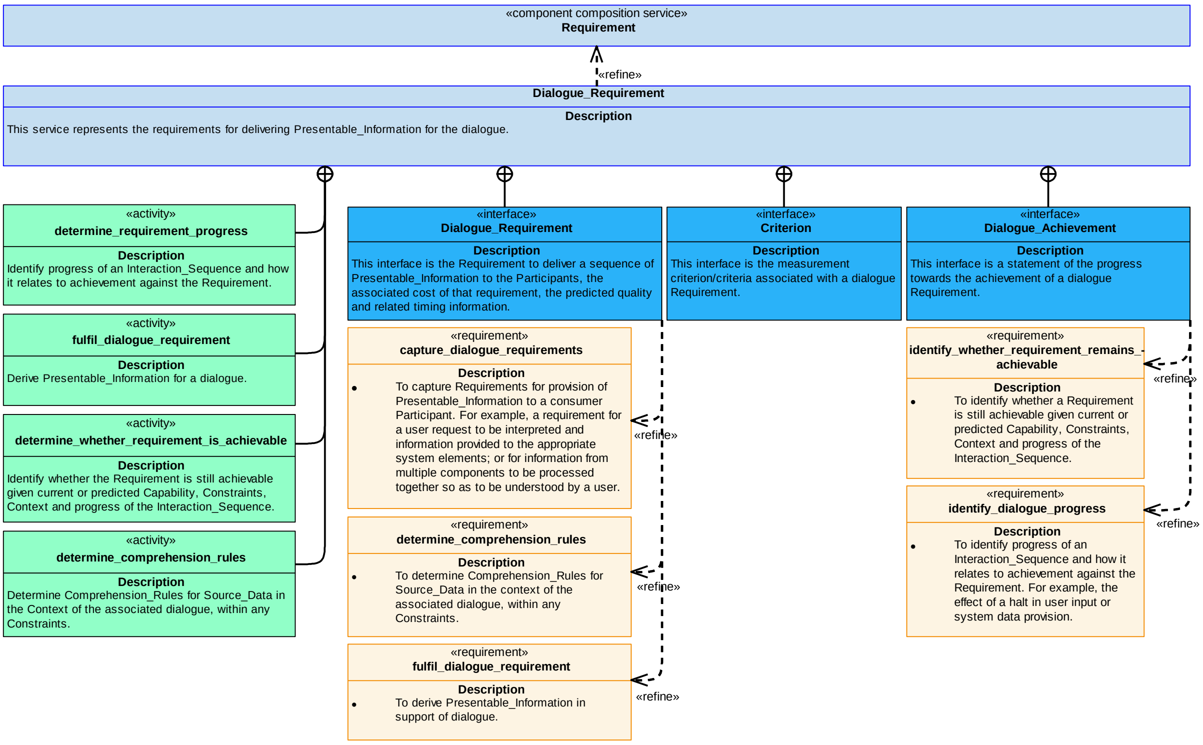


Figure 436: Dialogue_Requirement Service Definition


«component composition service»
Requirement
«refine»
Dialogue_Requirement
«activity»
determine_requirement_progress
«activity»
fulfil_dialogue_requirement
«activity»
determine_whether_requirement_is_achievable
«activity»
determine_comprehension_rules
«interface»
Dialogue_Requirement
«interface»
Criterion
«interface»
Dialogue_Achievement
«requirement»
capture_dialogue_requirements
«requirement»
determine_comprehension_rules
«requirement»
fulfil_dialogue_requirement
«requirement»
identify_whether_requirement_remains_achievable
«requirement»
identify_dialogue_progress
«refine»
«refine»
«refine»
«refine»
«refine»

Dialogue_Requirement

Description

This service represents the requirements for delivering Presentable_Information for the dialogue.

«activity» determine_requirement_progress

Description

Identify progress of an Interaction_Sequence and how it relates to achievement against the Requirement.

«activity» fulfil_dialogue_requirement

Description

Derive Presentable_Information for a dialogue.

«activity» determine_whether_requirement_is_achievable

Description

Identify whether the Requirement is still achievable given current or predicted Capability, Constraints, Context and progress of the Interaction_Sequence.

«activity» determine_comprehension_rules

Description

Determine Comprehension_Rules for Source_Data in the Context of the associated dialogue, within any Constraints.

«interface» Dialogue_Requirement

Description

This interface is the Requirement to deliver a sequence of Presentable_Information to the Participants, the associated cost of that requirement, the predicted quality and related timing information.

«interface» Criterion

Description

This interface is the measurement criterion/criteria associated with a dialogue Requirement.

«interface» Dialogue_Achievement

Description

This interface is a statement of the progress towards the achievement of a dialogue Requirement.

«requirement» capture_dialogue_requirements

Description

- To capture Requirements for provision of Presentable_Information to a consumer Participant. For example, a requirement for a user request to be interpreted and information provided to the appropriate system elements; or for information from multiple components to be processed together so as to be understood by a user.

«requirement» determine_comprehension_rules

Description

- To determine Comprehension_Rules for Source_Data in the context of the associated dialogue, within any Constraints.

«requirement» fulfil_dialogue_requirement

Description

- To derive Presentable_Information in support of dialogue.

«requirement» identify_whether_requirement_remains_achievable

Description

- To identify whether a Requirement is still achievable given current or predicted Capability, Constraints, Context and progress of the Interaction_Sequence.

«requirement» identify_dialogue_progress

Description

- To identify progress of an Interaction_Sequence and how it relates to achievement against the Requirement. For example, the effect of a halt in user input or system data provision.

Figure 437: Dialogue_Requirement Service Policy

Dialogue_Requirement

This service represents the requirements for delivering Presentable_Information for the dialogue.

Interfaces

Dialogue_Requirement

This interface is the Requirement to deliver a sequence of Presentable_Information to the Participants, the associated cost of that requirement, the predicted quality and related timing information.

Attributes

specification	The definition of the information delivery requirement, e.g. process and deliver information to a Participant.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used or time taken.
quality	The required information of the delivery solution to satisfy the requirement.

Criterion

This interface is the measurement criterion/criteria associated with a dialogue Requirement.

Attributes

property	The property to be measured, e.g. time of data delivery.
value	The measured value of the property, e.g. 3 milliseconds.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Dialogue_Achievement

This interface is a statement of the progress towards the achievement of a dialogue Requirement.

Activities

determine_requirement_progress

Identify progress of an Interaction_Sequence and how it relates to achievement against the Requirement.

fulfil_dialogue_requirement

Derive Presentable_Information for a dialogue.

determine_whether_requirement_is_achievable

Identify whether the Requirement is still achievable given current or predicted Capability, Constraints, Context and progress of the Interaction_Sequence.

determine_comprehension_rules

Determine Comprehension_Rules for Source_Data in the Context of the associated dialogue, within any Constraints.

5.4.2.25.7.1.2 Dialogue_Information

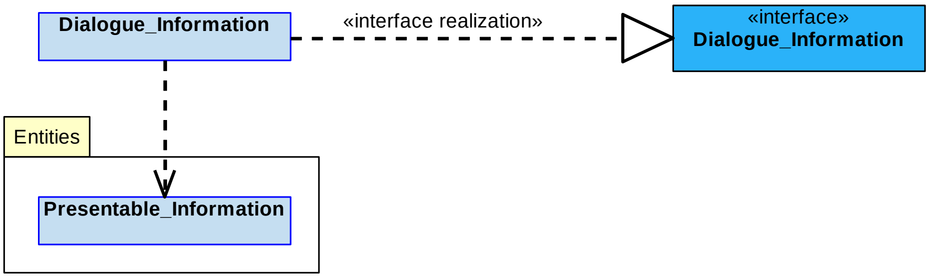


Figure 438: Dialogue_Information Service Definition

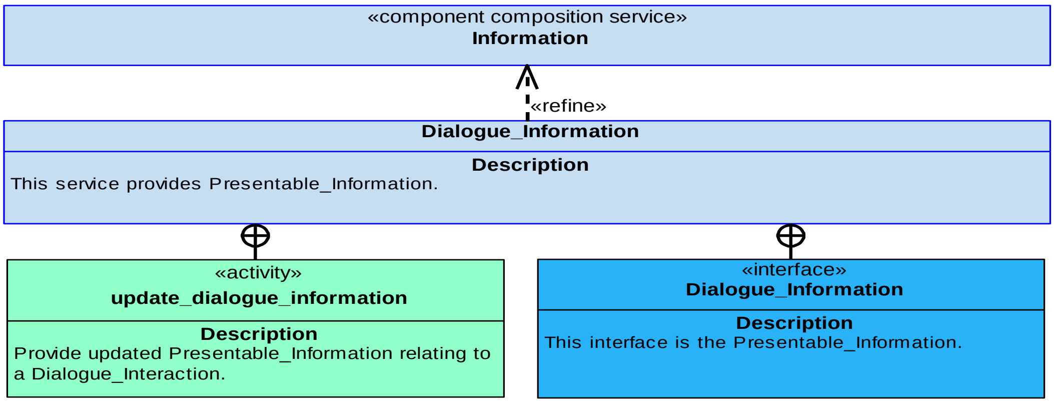

«component composition service»
Information
«refine»
Dialogue_Information
Description
This service provides Presentable_Information.
«activity»
update_dialogue_information
Description
Provide updated Presentable_Information relating to a Dialogue_Interaction.
«interface»
Dialogue_Information
Description
This interface is the Presentable_Information.

Figure 439: Dialogue_Information Service Policy

Dialogue_Information

This service provides Presentable_Information.

Interface

Dialogue_Information

This interface is the Presentable_Information.

Activity

update_dialogue_information

Provide updated Presentable_Information relating to a Dialogue_Interaction.

5.4.2.25.7.1.3 Dialogue_Dependency

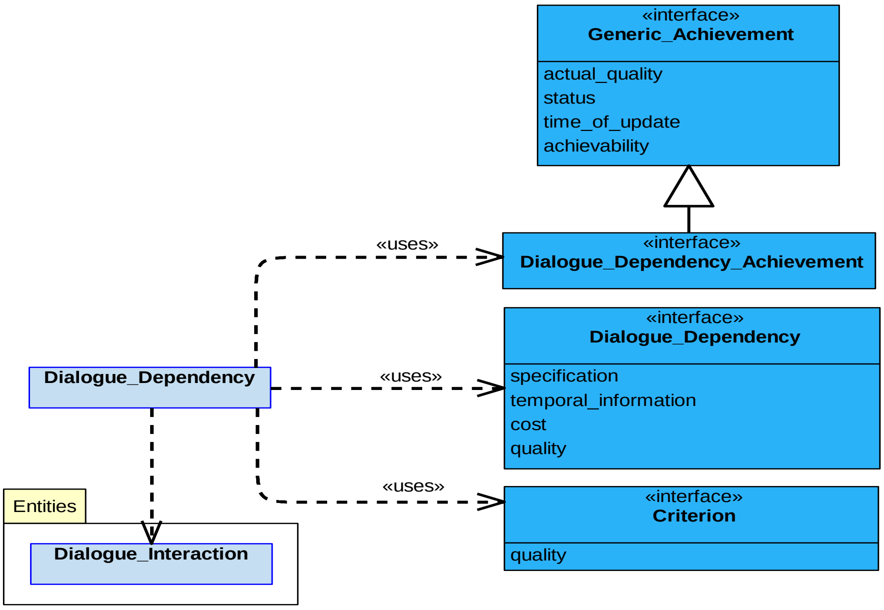
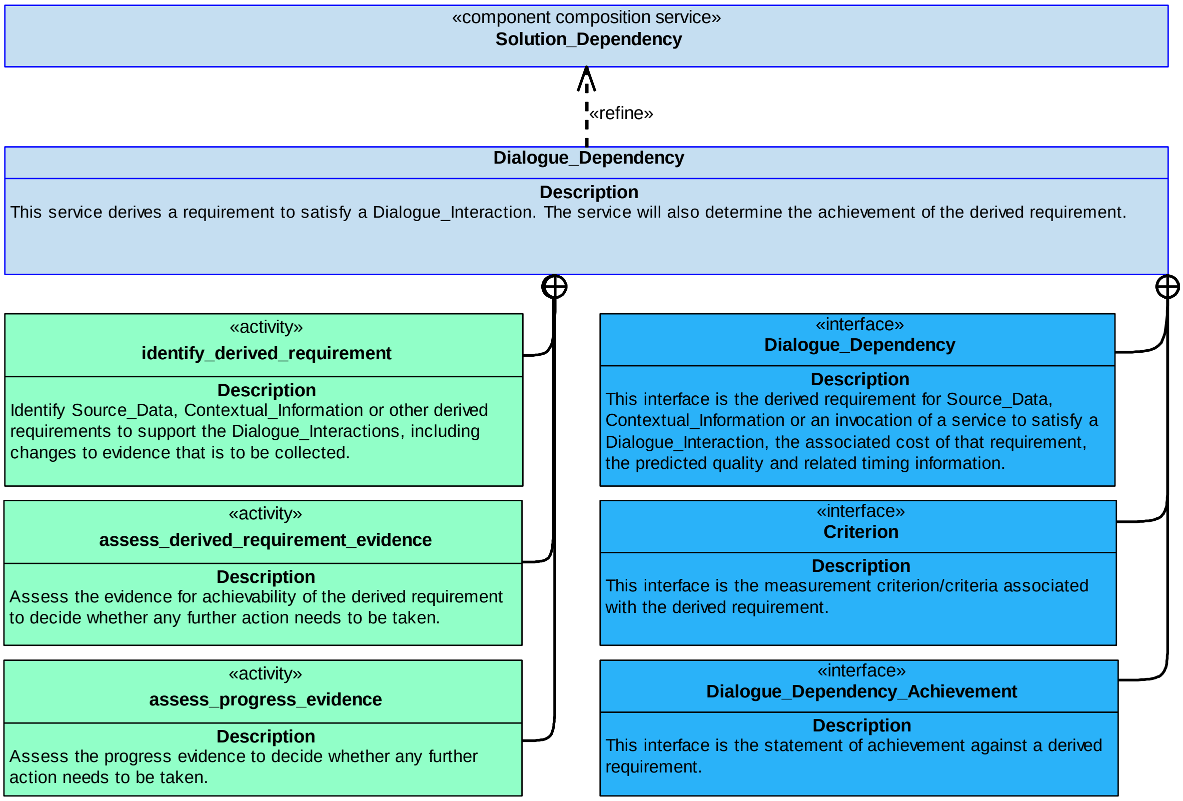


Figure 440: Dialogue_Dependency Service Definition



This service derives a requirement to satisfy a Dialogue_Interaction. The service will also determine the achievement of the derived requirement.

Identify Source_Data, Contextual_Information or other derived requirements to support the Dialogue_Interactions, including changes to evidence that is to be collected.

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

Assess the progress evidence to decide whether any further action needs to be taken.

This interface is the derived requirement for Source_Data, Contextual_Information or an invocation of a service to satisfy a Dialogue_Interaction, the associated cost of that requirement, the predicted quality and related timing information.

This interface is the measurement criterion/criteria associated with the derived requirement.

This interface is the statement of achievement against a derived requirement.

Figure 441: Dialogue_Dependency Service Policy

Dialogue_Dependency

This service derives a requirement to satisfy a Dialogue_Interaction. The service will also determine the achievement of the derived requirement.

Interfaces

Criterion

This interface is the measurement criterion/criteria associated with the derived requirement.

Attribute

quality	The required quality of the data, e.g. accuracy, timeliness, precision and trustworthiness.

Dialogue_Dependency_Achievement

This interface is the statement of achievement against a derived requirement.

Dialogue_Dependency

This interface is the derived requirement for Source_Data, Contextual_Information or an invocation of a service to satisfy a Dialogue_Interaction, the associated cost of that requirement, the predicted quality and related timing information.

Attributes

specification	The definition of the derived requirement.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used or time taken.
quality	How well the solution satisfies the requirement.

Activities

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_derived_requirement

Identify Source_Data, Contextual_Information or other derived requirements to support the Dialogue_Interactions, including changes to evidence that is to be collected.

assess_derived_requirement_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

5.4.2.25.7.1.4 Source_Data

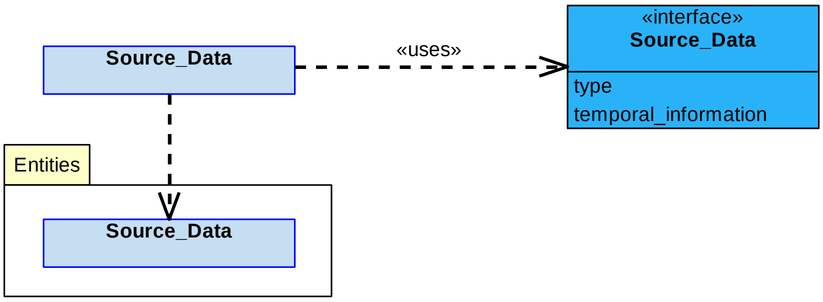


Figure 442: Source_Data Service Definition

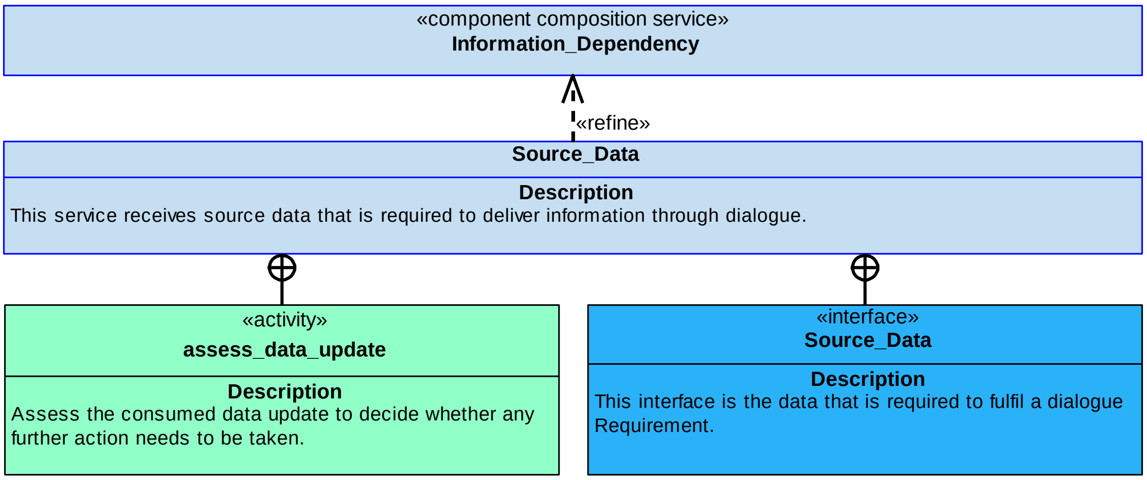

«component composition service»
Information_Dependency
«refine»
Source_Data
Description
This service receives source data that is required to deliver information through dialogue.
«activity»
assess_data_update
Description
Assess the consumed data update to decide whether any further action needs to be taken.
«interface»
Source_Data
Description
This interface is the data that is required to fulfil a dialogue Requirement.

Figure 443: Source_Data Service Policy

Source_Data

This service receives source data that is required to deliver information through dialogue.

Interface

Source_Data

This interface is the data that is required to fulfil a dialogue Requirement.

Attributes

type	The type of data that is received.
temporal_information	Information covering timing, such as data update rate.

Activity

assess_data_update

Assess the consumed data update to decide whether any further action needs to be taken.

5.4.2.25.7.1.5 Contextual_Information

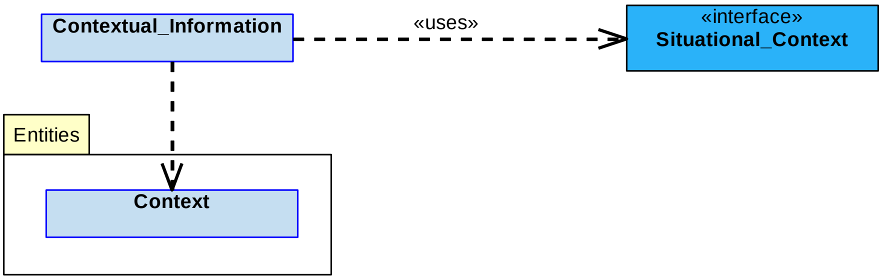


Figure 444: Contextual_Information Service Definition

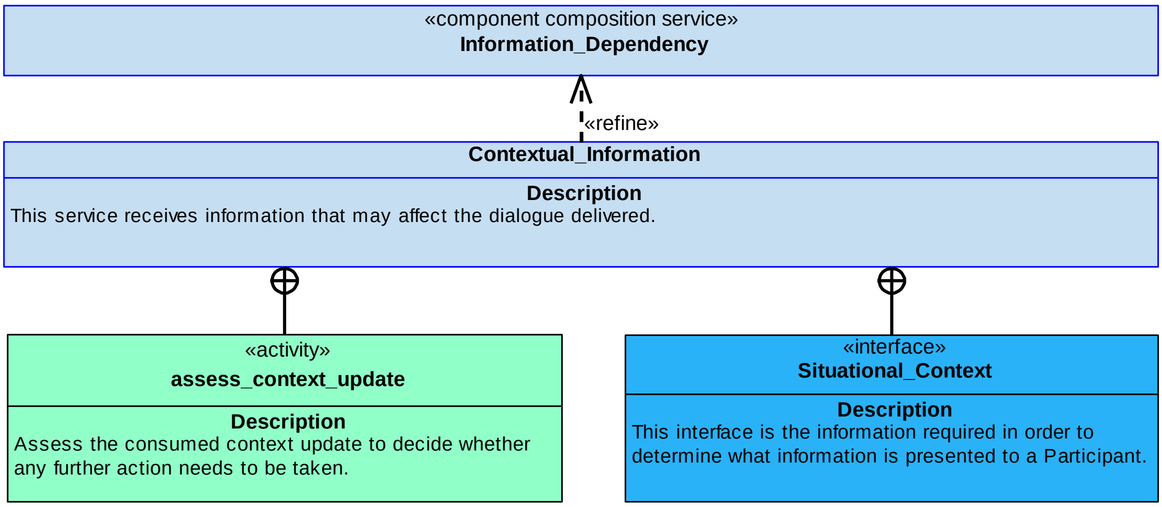

«component composition service»
Information_Dependency
«refine»
Contextual_Information
Description
This service receives information that may affect the dialogue delivered.
«activity»
assess_context_update
Description
Assess the consumed context update to decide whether any further action needs to be taken.
«interface»
Situational_Context
Description
This interface is the information required in order to determine what information is presented to a Participant.

Figure 445: Contextual_Information Service Policy

Contextual_Information

This service receives information that may affect the dialogue delivered.

Interface

Situational_Context

This interface is the information required in order to determine what information is presented to a Participant.

Activity

assess_context_update

Assess the consumed context update to decide whether any further action needs to be taken.

5.4.2.25.7.1.6 Constraint

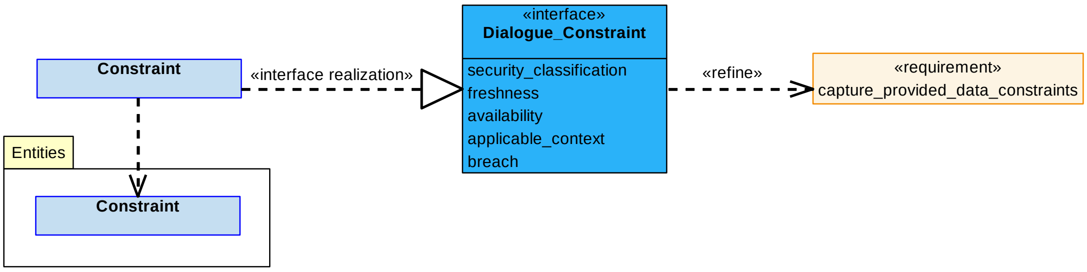


Figure 446: Constraint Service Definition

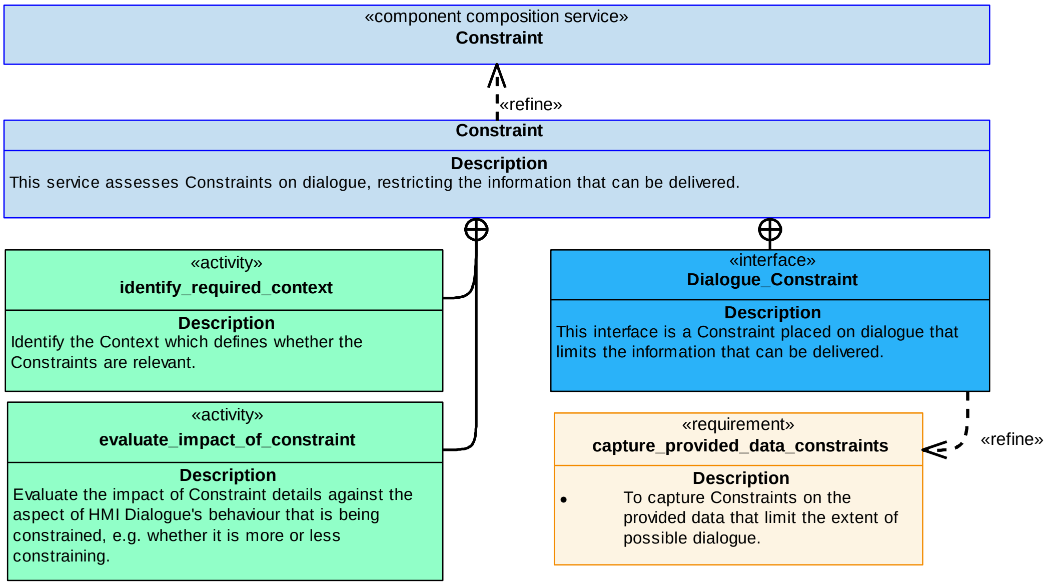

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints on dialogue, restricting the information that can be delivered.
«activity»
identify_required_context
Description
Identify the Context which defines whether the Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of HMI Dialogue's behaviour that is being constrained, e.g. whether it is more or less constraining.
«interface»
Dialogue_Constraint
Description
This interface is a Constraint placed on dialogue that limits the information that can be delivered.
«requirement»
capture_provided_data_constraints
Description
To capture Constraints on the provided data that limit the extent of possible dialogue.
«refine»

Figure 447: Constraint Service Policy

Constraint

This service assesses Constraints on dialogue, restricting the information that can be delivered.

Interface

Dialogue_Constraint

This interface is a Constraint placed on dialogue that limits the information that can be delivered.

Attributes

security_classification	Security classification indicating the sensitivity of the data.
freshness	How current the data is.
availability	Timeliness and reliability of access to the data.
applicable_context	The context in which the constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of HMI Dialogue's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the Context which defines whether the Constraints are relevant.

5.4.2.25.7.1.7 Capability

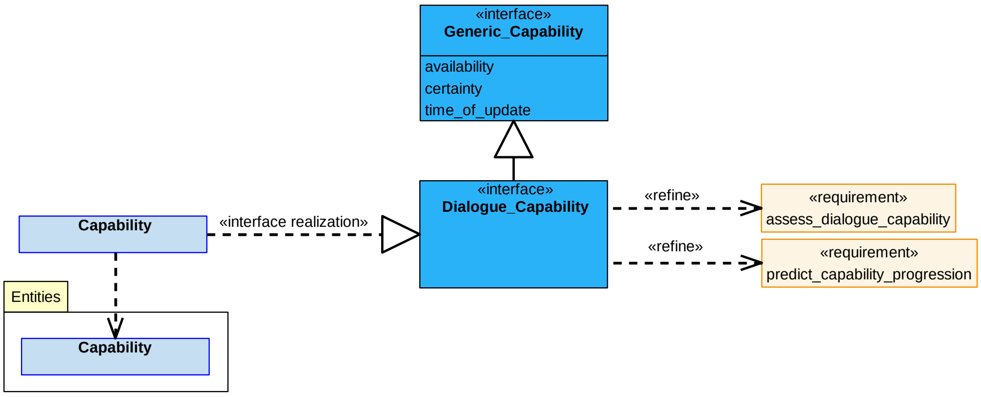


Figure 448: Capability Service Definition

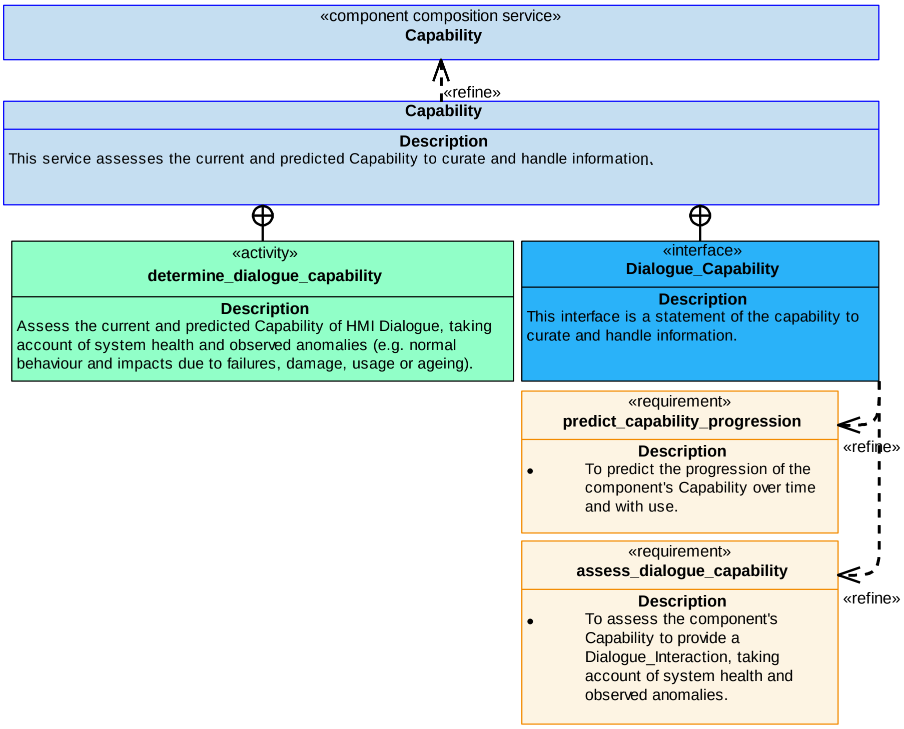

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to curate and handle information.
«activity»
determine_dialogue_capability
Description
Assess the current and predicted Capability of HMI Dialogue, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Dialogue_Capability
Description
This interface is a statement of the capability to curate and handle information.
«requirement»
predict_capability_progression
Description
To predict the progression of the component's Capability over time and with use.
«refine»
«requirement»
assess_dialogue_capability
Description
To assess the component's Capability to provide a Dialogue_Interaction, taking account of system health and observed anomalies.
«refine»

Figure 449: Capability Service Policy

Capability

This service assesses the current and predicted Capability to curate and handle information.

Interface

Dialogue_Capability

This interface is a statement of the capability to curate and handle information.

Activity

determine_dialogue_capability

Assess the current and predicted Capability of HMI Dialogue, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.25.7.1.8 Capability_Evidence

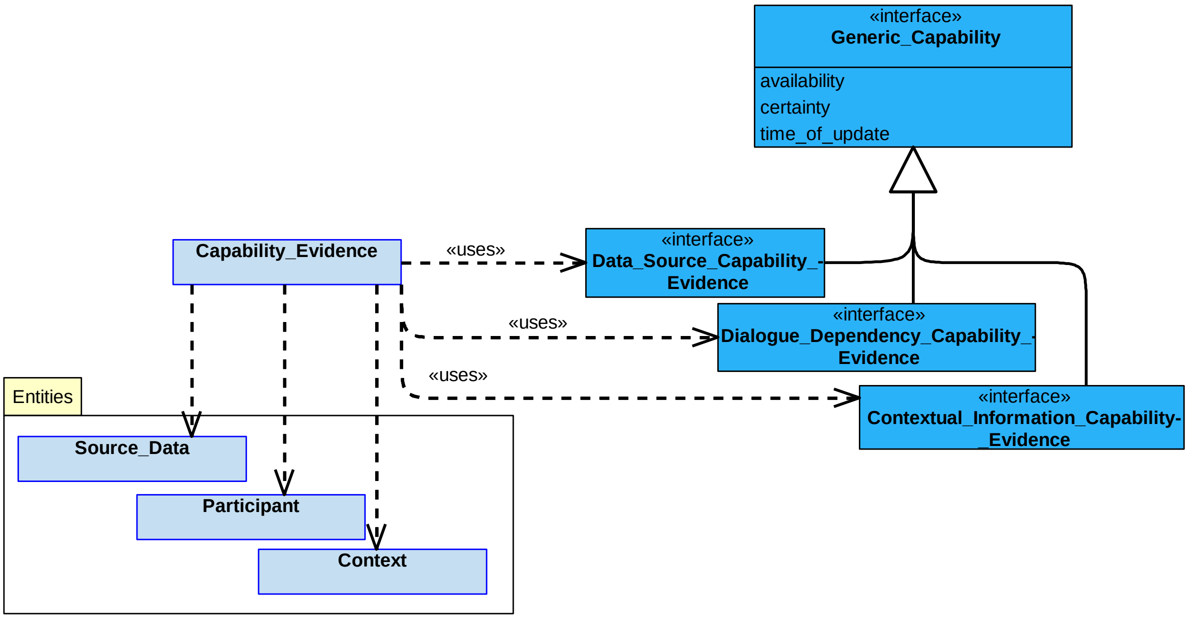


Figure 450: Capability_Evidence Service Definition

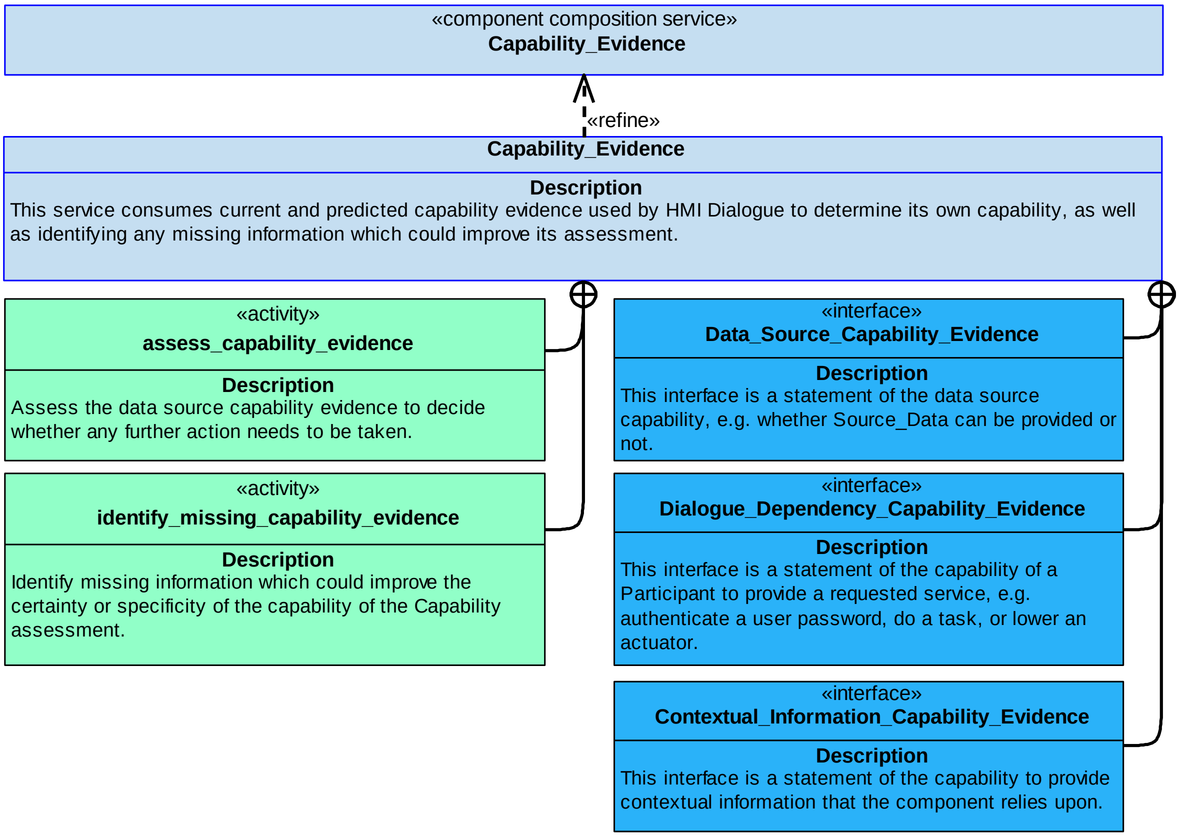

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability evidence used by HMI Dialogue to determine its own capability, as well as identifying any missing information which could improve its assessment.
«activity»
assess_capability_evidence
Description
Assess the data source capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify missing information which could improve the certainty or specificity of the capability of the Capability assessment.
«interface»
Data_Source_Capability_Evidence
Description
This interface is a statement of the data source capability, e.g. whether Source_Data can be provided or not.
«interface»
Dialogue_Dependency_Capability_Evidence
Description
This interface is a statement of the capability of a Participant to provide a requested service, e.g. authenticate a user password, do a task, or lower an actuator.
«interface»
Contextual_Information_Capability_Evidence
Description
This interface is a statement of the capability to provide contextual information that the component relies upon.

Figure 451: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability evidence used by HMI Dialogue to determine its own capability, as well as identifying any missing information which could improve its assessment.

Interfaces

Data_Source_Capability_Evidence

This interface is a statement of the data source capability, e.g. whether Source_Data can be provided or not.

Dialogue_Dependency_Capability_Evidence

This interface is a statement of the capability of a Participant to provide a requested service, e.g. authenticate a user password, do a task, or lower an actuator.

Contextual_Information_Capability_Evidence

This interface is a statement of the capability to provide contextual information that the component relies upon.

Activities

assess_capability_evidence

Assess the data source capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify missing information which could improve the certainty or specificity of the capability of the Capability assessment.

5.4.2.25.7.2 Service Dependencies

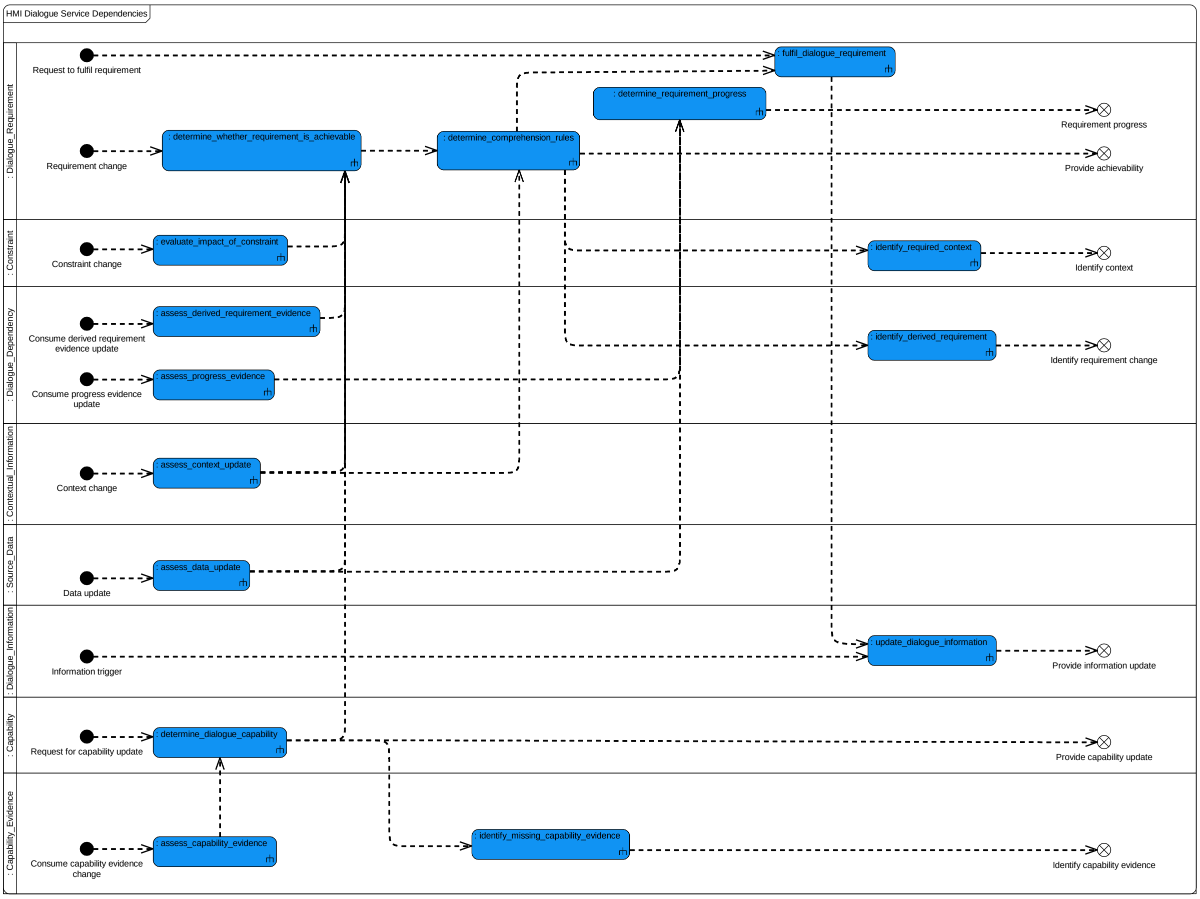

HMI Dialogue Service Dependencies
: Dialogue_Requirement
: Constraint
: Dialogue_Dependency
: Contextual_Information
: Source_Data
: Dialogue_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
: fulfil_dialogue_requirement
: determine_requirement_progress
Requirement progress
Requirement change
: determine_whether_requirement_is_achievable
: determine_comprehension_rules
Provide achievability
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume derived requirement evidence update
: assess_derived_requirement_evidence
: identify_derived_requirement
Identify requirement change
Consume progress evidence update
: assess_progress_evidence
Context change
: assess_context_update
Data update
: assess_data_update
Information trigger
: update_dialogue_information
Provide information update
Request for capability update
: determine_dialogue_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 452: HMI Dialogue Service Dependencies

5.4.2.26 Human Interaction

5.4.2.26.1 Role

The role of Human Interaction is to enable communication between humans.

5.4.2.26.2 Overview

Control Architecture

Human Interaction is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a Requirement for human interaction is received, the component will propose an Interaction_Solution to achieve the Requirement. This will involve coordinated control of Interaction_Devices, to enable human interaction via a suitable Conduit. The possible Interaction_Medias and Endpoints may be limited by Constraints. The proposed Interaction_Solution will result in an Interaction that will enable the required human interaction to take place.

Examples of Use

- Human Interaction will be required when a UAV operator communicates with the mission commander who is located at a different physical location.

5.4.2.26.3 Service Summary

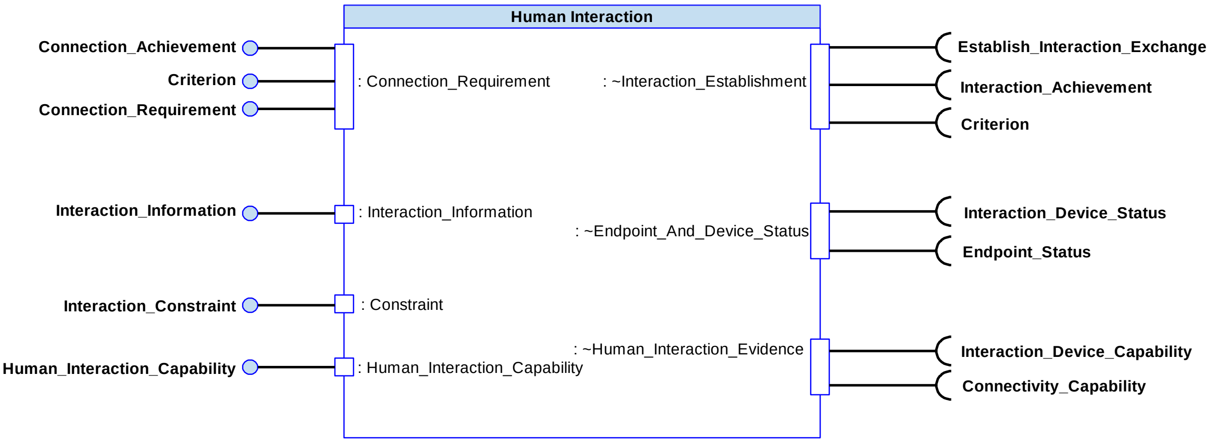

Human Interaction
Connection_Achievement
Criterion
Connection_Requirement
: Connection_Requirement
: ~Interaction_Establishment
Establish_Interaction_Exchange
Interaction_Achievement
Criterion
Interaction_Information
: Interaction_Information
: ~Endpoint_And_Device_Status
Interaction_Device_Status
Endpoint_Status
Interaction_Constraint
: Constraint
Human_Interaction_Capability
: Human_Interaction_Capability
: ~Human_Interaction_Evidence
Interaction_Device_Capability
Connectivity_Capability

Figure 453: Human Interaction Service Summary

5.4.2.26.4 Responsibilities

capture_interaction_requirements

- To capture Requirements for Interactions (e.g. set-up interactions ahead of time to enable push-to-talk without any delay due to establishing the connection).

capture_interaction_constraints

- To capture any Constraints that may be applied to Participants' possible Interactions (e.g. constrain access to specific contact(s) or interaction types).

determine_possible_interactions

- To determine how Participants can interact (e.g. radio, text message, or voice).

determine_available_endpoints

- To determine the available human interaction Endpoints (e.g. phone number or radio channel).

determine_if_solution_remains_feasible

- To determine if a planned or on-going Interaction_Solution remains feasible given current Constraints and Capability.

identify_contacts

- To identify contacts for possible Interactions (e.g. users, user-aliases and groups of users that can interact).

coordinate_interaction

- To setup, start and end an Interaction.

determine_status_of_interaction

- To determine the status of an Interaction.

determine_quality_of_interaction

- To determine the quality of an Interaction against given Measurement_Criterion/criteria.

assess_interaction_support_capability

- To assess the Capability of the component to support human interaction taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Human Interaction Capability assessment.

predict_capability_progression

- To predict the progression of the Human Interaction Capability over time and with use.

5.4.2.26.5 Subject Matter Semantics

The subject matter of Human Interaction is the resources that can be used to enable communication between humans.

Exclusions

The subject matter of Human Interaction does not include:

- The provision and management of the connections between devices.
- User devices and conversion of the signals between the system and the user.

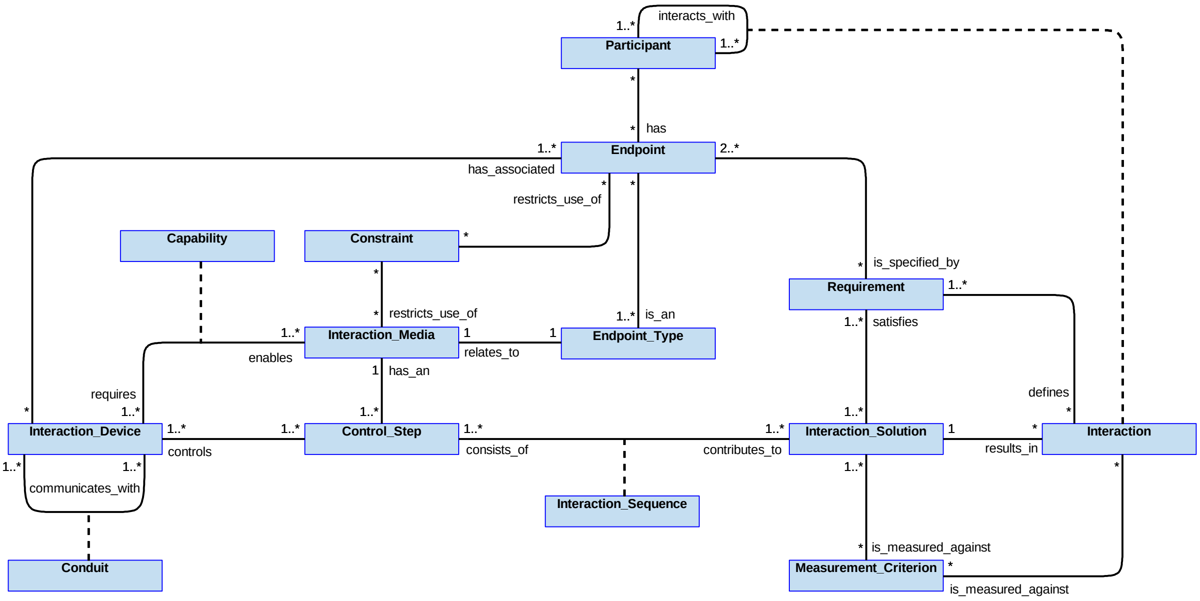


Figure 454: Human Interaction Semantics

5.4.2.26.5.1 Entities

Capability

The range of Interaction_Medias given the available Interaction_Devices.

Interaction_Device

A system interface device that enables human interactions to occur, e.g. telephone, radio, or mobile phone application.

Conduit

The connectivity between two or more Interaction_Devices through which human interaction can occur (e.g. a network).

Constraint

An externally imposed restriction that limits the behaviour of the component.

Control_Step

An individual step involved in setting up, starting and ending an Interaction.

Endpoint

The terminus of an Interaction (e.g. the number of a phone or participant, the channel reference for a radio broadcast, or the machine address for an instant message).

Endpoint_Type

The specific functionality supported by an Endpoint, e.g. Skype audio call, radio broadcast, or text message.

Interaction

A specific instance of human to human connectivity enabled by the system (e.g. a specific telephone call or an instant message).

Interaction_Sequence

The order in which Control_Steps must be performed to implement an Interaction_Solution.

Interaction_Solution

The mechanisms to enable an interaction between two or more humans.

Interaction_Media

The media through which a human interaction is conducted (e.g. audio, visual, or textual (or any combination thereof)).

Measurement_Criterion

A criterion that the quality of an Interaction will be measured against (e.g. the call quality).

Participant

A human involved in a human interaction.

Requirement

A requirement to enable a human interaction, e.g. to connect two Participants for a voice call.

5.4.2.26.6 Design Rationale

5.4.2.26.6.1 Assumptions

- The overall planning of human communications, which will involve multiple components, will be done at a tasking level and not just in this component.
- This component will be notified by other components when logon status changes or interaction status changes.
- This component can be involved in setting up pre-planned communications and then inform a user that it is ready (e.g. setting up a connection to ATC).
- The setting up of connections and determining the messaging protocols to be used is the responsibility of other components.

5.4.2.26.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Human Interaction:

- Use of Communications - This PYRAMID concept specifies how communications are managed by components such as this one.

- Data Driving - There are a variety of different means of human interaction which this component may need to support. The use of data driving to support this should be considered.

Other Factors that were Taken into Account

- The Human Interaction component is responsible for the management and monitoring of an interaction.
- The Human Interaction component facilitates capability such as call groups, call forwarding, call holding, pick-up groups and redirect.
- The Human Interaction component is not a directory or directory service, it only defines Endpoints sufficiently to facilitate and manage human interactions. Other information required to deliver user interaction services is held by those components that directly require it. Data is defined, understood and utilised only by the component at the point of use, and never by a component that does not act on the data. Reference to contact information between components may be realised through the use of counterpart relationships.

5.4.2.26.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- This component is involved in the set-up of an interaction (e.g. voice, email or text message) between humans. Failure of this component would prevent the interaction occurring - i.e. a loss of communications. Communication, particularly with ATC is used to mitigate hazards- e.g. mid-air collision. However, as a loss of communication (particularly via radio or satellite) can occur for many reasons external to the air system (e.g. atmospheric interference) then other safety barriers are designed into the air vehicle (e.g. ACAS). Therefore, it is judged that failure of this component would be no more severe than a ‘significant reduction in safety margins’ (severity major) which requires an indicative IDAL of DAL C.

5.4.2.26.6.4 Security Considerations

The indicative security classification is O-S.

This component is responsible for accessing and establishing means of Interaction between Participants, with access to a level of information about those users (including their node) necessary in order to perform that function. This component may assist in facilitating transfer of classified data higher than O-S when located in appropriate security domains but as it is only involved in set-up of the interaction it does not have access to that data. It may also be involved in cross-security domain communications, but the separation of the domains will be handled outside this component.

Although it is possible to implement a reduced participant “whitelist”, this is unlikely to be considered security “enforcing” in nature; the enforcement role would be performed outside this component. This component is reliant on the integrity of the information provided to identify participants.

The component is expected to at least partially satisfy security related functions by:

- **Identification of Data Sources** representing the Endpoints of an interaction.
- **Logging of Security Data** relating to the nature of connections made and released, etc.
- **Maintaining Audit Data** relating to interactions during the course of the mission. The logging of certain interactions (e.g. with ATC) is a legal requirement.

Note: security and audit data from this component will not include interaction content.

Is unlikely to directly implement security enforcing functions, although it relies on those involved in:

- **User Login and Authentication** as it will be notified when logon or availability status changes, etc.

5.4.2.26.7 Services

5.4.2.26.7.1 Service Definitions

5.4.2.26.7.1.1 Connection_Requirement

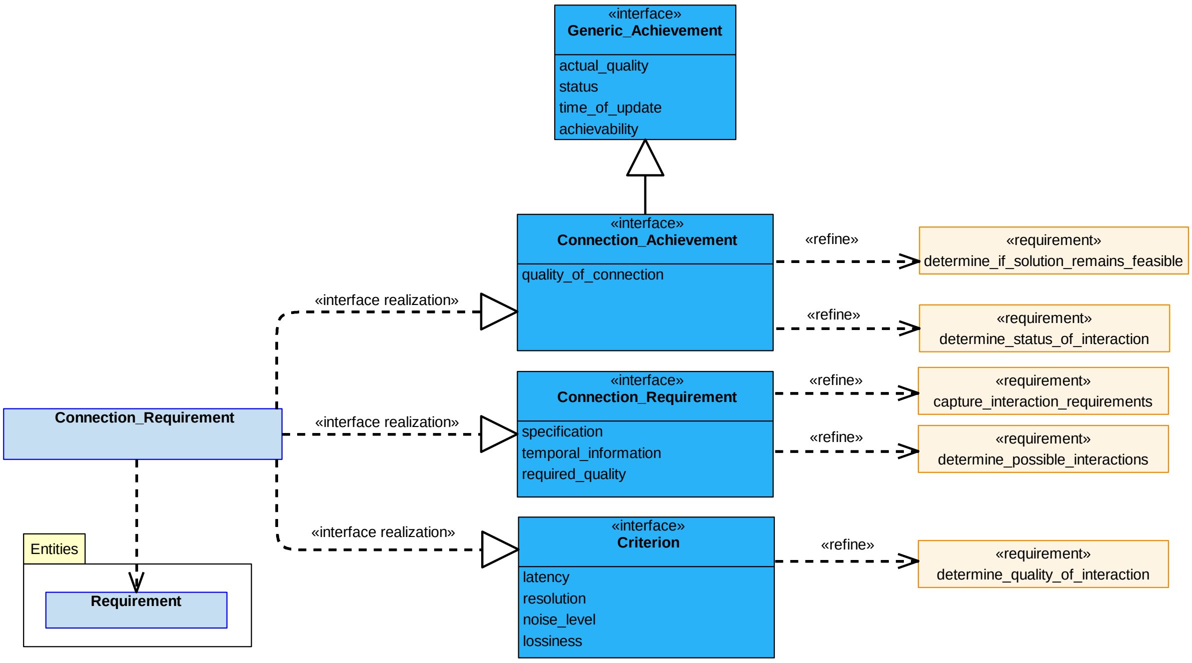


Figure 455: Connection_Requirement Service Definition

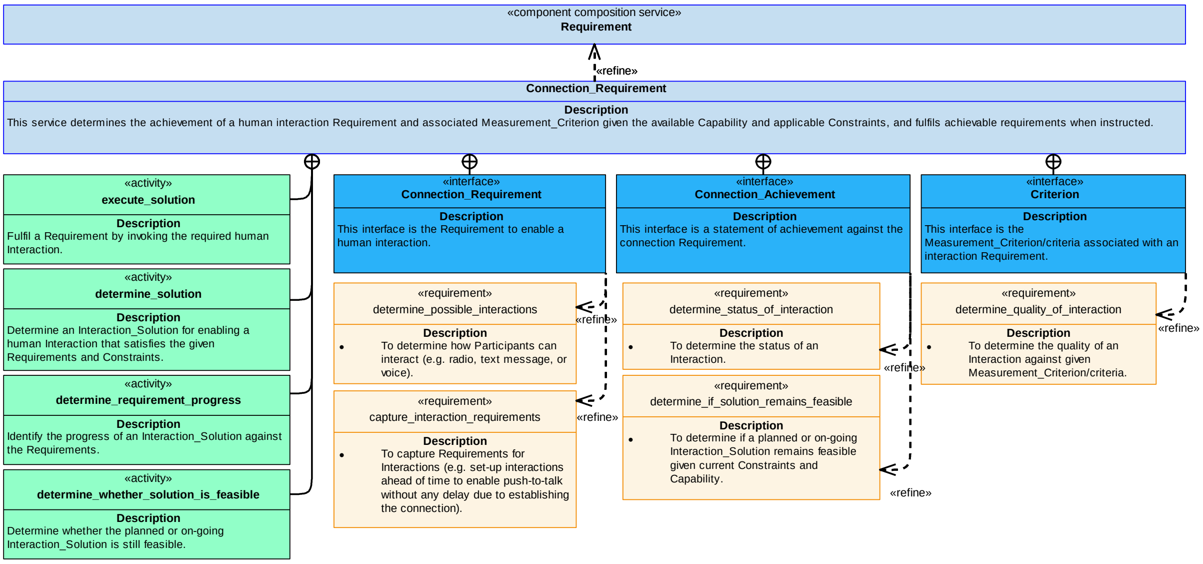

«component composition service»
Requirement
«refine»
Connection_Requirement
Description
This service determines the achievement of a human interaction Requirement and associated Measurement_Criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.
«activity»
execute_solution
Description
Fulfil a Requirement by invoking the required human Interaction.
«activity»
determine_solution
Description
Determine an Interaction_Solution for enabling a human Interaction that satisfies the given Requirements and Constraints.
«activity»
determine_requirement_progress
Description
Identify the progress of an Interaction_Solution against the Requirements.
«activity»
determine_whether_solution_is_feasible
Description
Determine whether the planned or on-going Interaction_Solution is still feasible.
«interface»
Connection_Requirement
Description
This interface is the Requirement to enable a human interaction.
«requirement»
determine_possible_interactions
Description
To determine how Participants can interact (e.g. radio, text message, or voice).
«refine»
«requirement»
capture_interaction_requirements
Description
To capture Requirements for Interactions (e.g. set-up interactions ahead of time to enable push-to-talk without any delay due to establishing the connection).
«refine»
«interface»
Connection_Achievement
Description
This interface is a statement of achievement against the connection Requirement.
«requirement»
determine_status_of_interaction
Description
To determine the status of an Interaction.
«refine»
«requirement»
determine_if_solution_remains_feasible
Description
To determine if a planned or on-going Interaction_Solution remains feasible given current Constraints and Capability.
«refine»
«interface»
Criterion
Description
This interface is the Measurement_Criterion/criteria associated with an interaction Requirement.
«requirement»
determine_quality_of_interaction
Description
To determine the quality of an Interaction against given Measurement_Criterion/criteria.
«refine»

Figure 456: Connection_Requirement Service Policy

Connection_Requirement

This service determines the achievement of a human interaction Requirement and associated Measurement_Criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Connection_Requirement

This interface is the Requirement to enable a human interaction.

Attributes

specification	The definition of the human interaction Requirement. For example, a phone call to a specific Endpoint or Participant.
temporal_information	Information covering timing, such as start and end times or length of an interaction.
required_quality	The required quality of the interaction. For example, audibility.

Criterion

This interface is the Measurement_Criterion/criteria associated with an interaction Requirement.

Attributes

latency	The level of delay associated with each interaction.
resolution	The accuracy at which the interaction is reproduced.
noise_level	The level of interference in the interaction.
lossiness	The characteristic or quality of being lossy experienced by the interaction.

Connection_Achievement

This interface is a statement of achievement against the connection Requirement.

Attribute

quality_of_connection	The achieved quality of communication connection e.g. video resolution, audio bitrate, or peak distortion level.

Activities

determine_solution

Determine an Interaction_Solution for enabling a human Interaction that satisfies the given Requirements and Constraints.

determine_requirement_progress

Identify the progress of an Interaction_Solution against the Requirements.

execute_solution

Fulfil a Requirement by invoking the required human Interaction.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Interaction_Solution is still feasible.

5.4.2.26.7.1.2 Interaction_Establishment

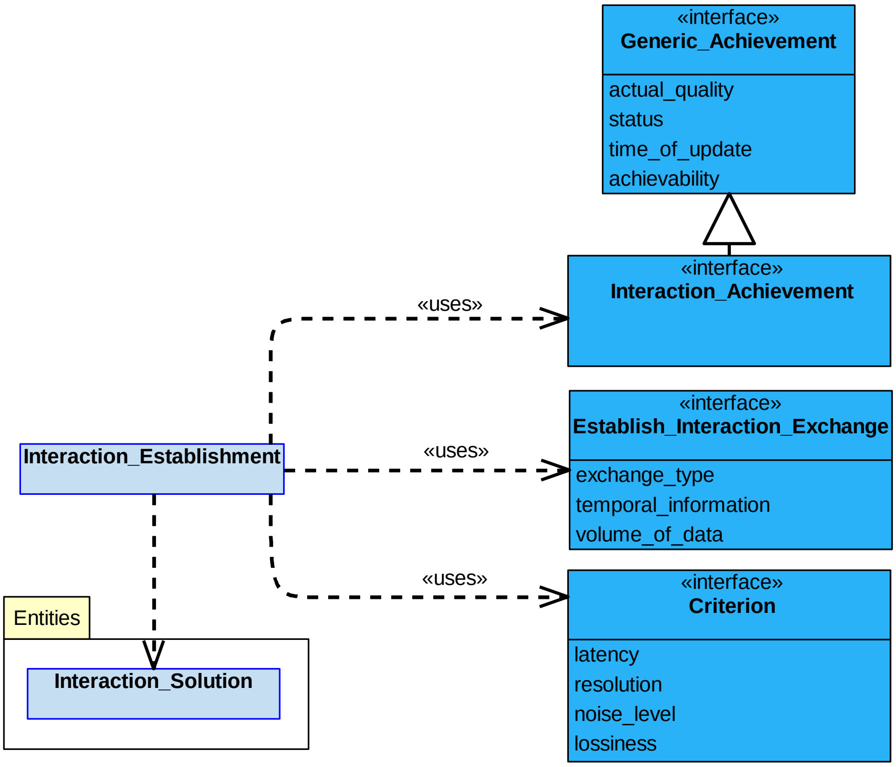
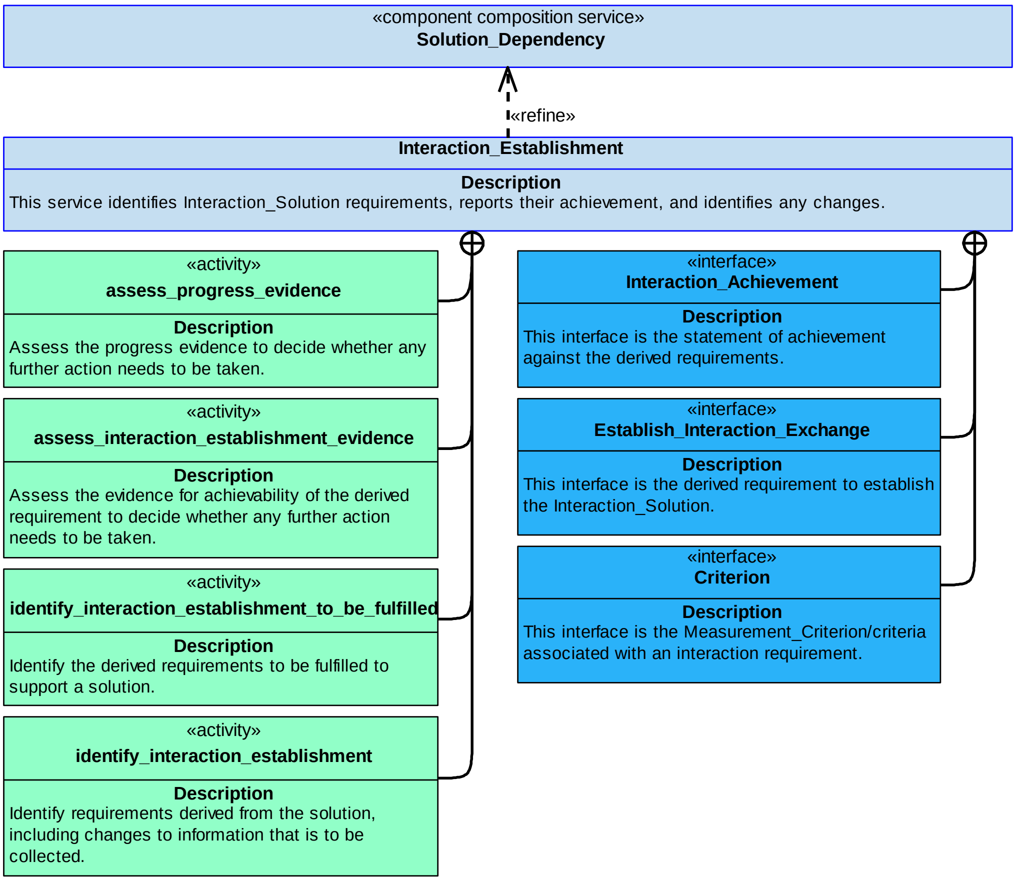


Figure 457: Interaction_Establishment Service Definition


«component composition service»
Solution_Dependency
«refine»
Interaction_Establishment
«activity»
assess_progress_evidence
«activity»
assess_interaction_establishment_evidence
«activity»
identify_interaction_establishment_to_be_fulfilled
«activity»
identify_interaction_establishment
«interface»
Interaction_Achievement
«interface»
Establish_Interaction_Exchange
«interface»
Criterion

Description
This service identifies Interaction_Solution requirements, reports their achievement, and identifies any changes.

Description
Assess the progress evidence to decide whether any further action needs to be taken.

Description
Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

Description
Identify the derived requirements to be fulfilled to support a solution.

Description
Identify requirements derived from the solution, including changes to information that is to be collected.

Description
This interface is the statement of achievement against the derived requirements.

Description
This interface is the derived requirement to establish the Interaction_Solution.

Description
This interface is the Measurement_Criterion/criteria associated with an interaction requirement.

Figure 458: Interaction_Establishment Service Policy

Interaction_Establishment

This service identifies Interaction_Solution requirements, reports their achievement, and identifies any changes.

Interfaces

Establish_Interaction_Exchange

This interface is the derived requirement to establish the Interaction_Solution.

Attributes

exchange_type	The type of exchange required for the interaction. For example, a specific telecommunications protocol.
temporal_information	Information covering timing, such as start and end times.
volume_of_data	The levels of connectivity required. For example, bands indicative of volumes of data to support the interaction.

Interaction_Achievement

This interface is the statement of achievement against the derived requirements.

Criterion

This interface is the Measurement_Criterion/criteria associated with an interaction requirement.

Attributes

latency	The level of delay associated with each interaction.
resolution	The accuracy at which the data has been reproduced.
noise_level	The level of interference in the signal.
lossiness	The characteristic or quality of being lossy.

Activities

assess_interaction_establishment_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_interaction_establishment

Identify requirements derived from the solution, including changes to information that is to be collected.

identify_interaction_establishment_to_be_fulfilled

Identify the derived requirements to be fulfilled to support a solution.

5.4.2.26.7.1.3 Interaction_Information

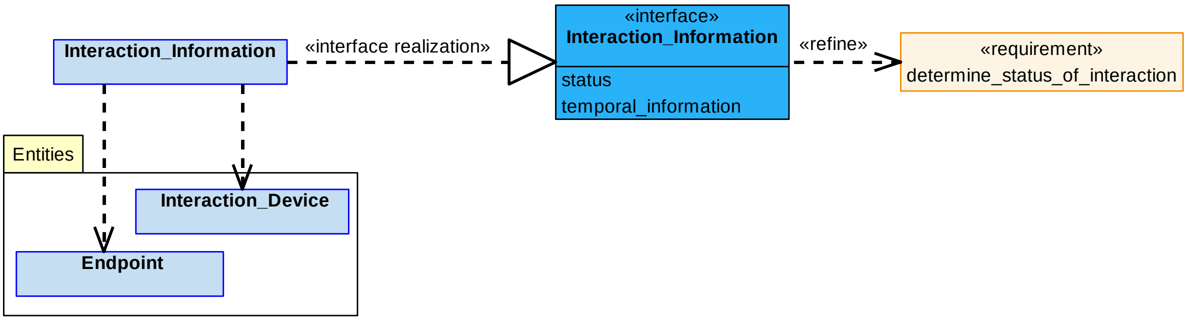


Figure 459: Interaction_Information Service Definition

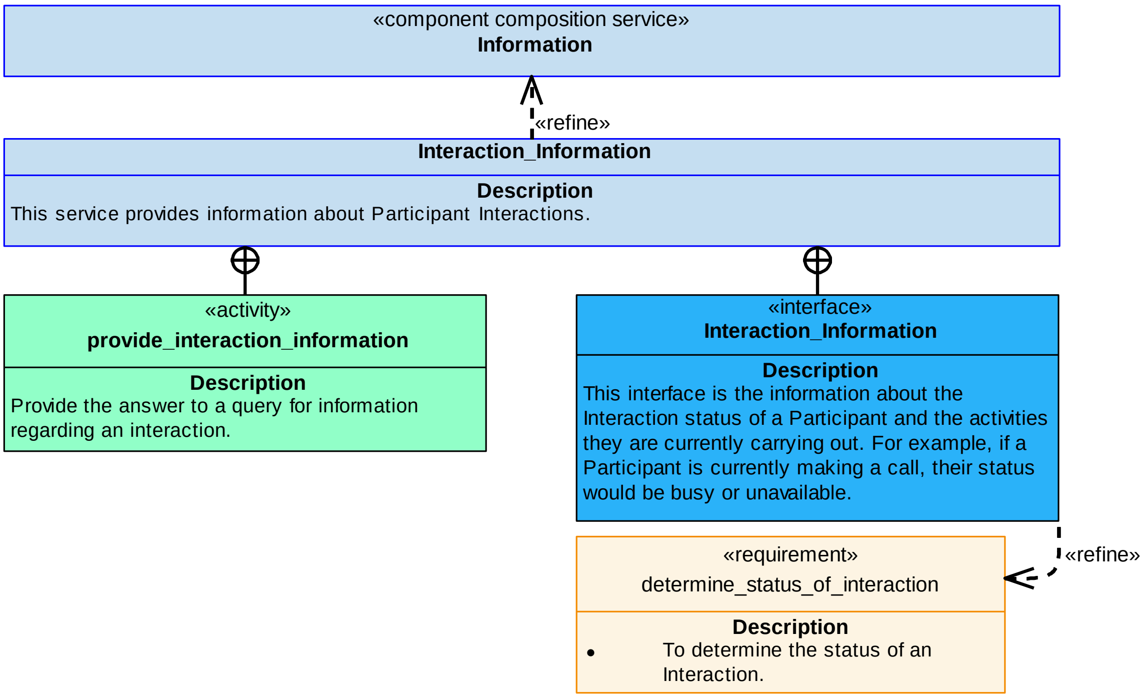

«component composition service»
Information
«refine»
Interaction_Information
Description
This service provides information about Participant Interactions.
«activity»
provide_interaction_information
Description
Provide the answer to a query for information regarding an interaction.
«interface»
Interaction_Information
Description
This interface is the information about the Interaction status of a Participant and the activities they are currently carrying out. For example, if a Participant is currently making a call, their status would be busy or unavailable.
«requirement»
determine_status_of_interaction
«refine»
Description
• To determine the status of an Interaction.

Figure 460: Interaction_Information Service Policy

Interaction_Information

This service provides information about Participant Interactions.

Interface

Interaction_Information

This interface is the information about the Interaction status of a Participant and the activities they are currently carrying out. For example, if a Participant is currently making a call, their status would be busy or unavailable.

Attributes

status	The status of a Participant Interaction. For example, active or inactive.
temporal_information	Information regarding timings, such as Participant time since last available, or time and length of an interaction.

Activity

provide_interaction_information

Provide the answer to a query for information regarding an interaction.

5.4.2.26.7.1.4 Endpoint_And_Device_Status

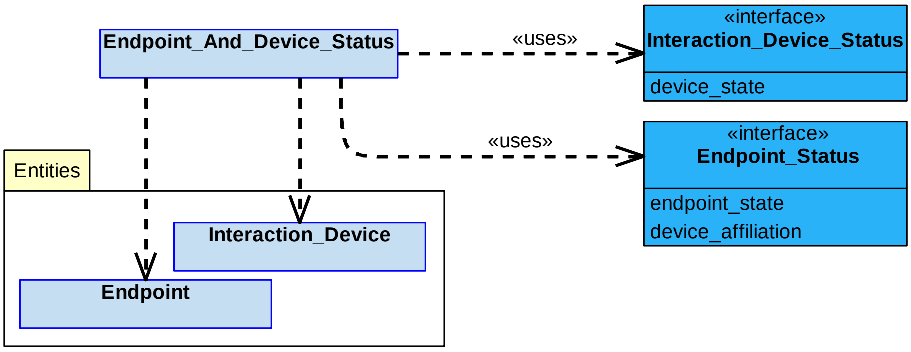


Figure 461: Endpoint_and_Device_Status Service Definition

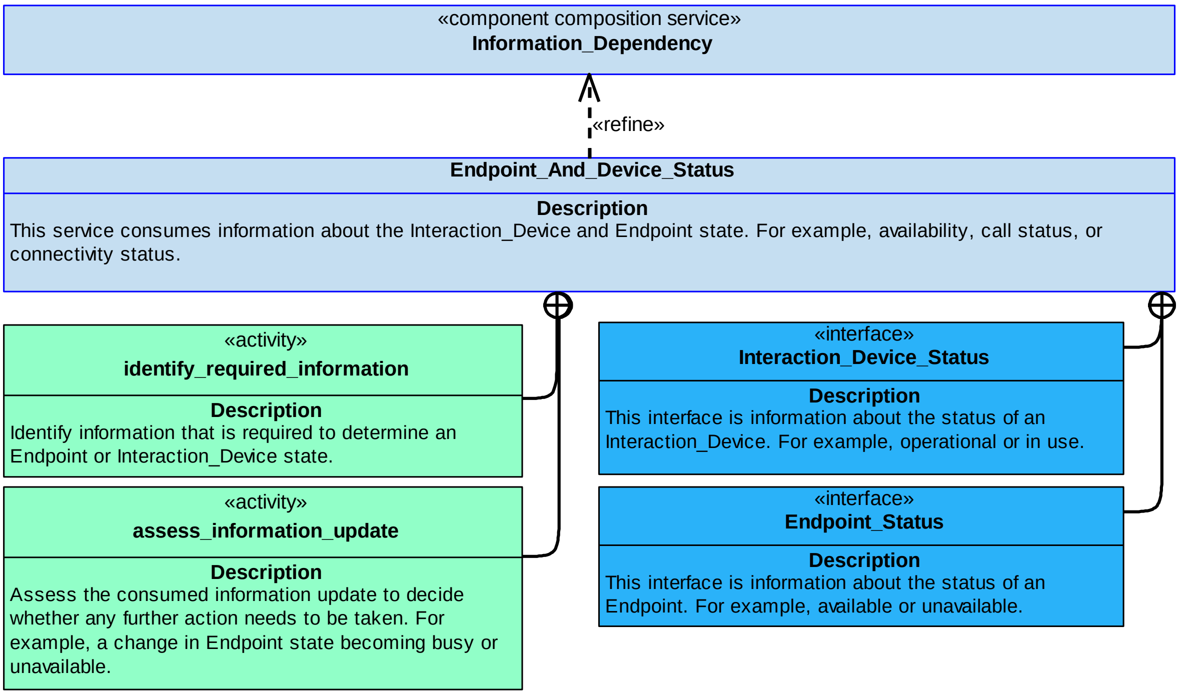

«component composition service»
Information_Dependency
«refine»
Endpoint_And_Device_Status
Description
This service consumes information about the Interaction_Device and Endpoint state. For example, availability, call status, or connectivity status.
«activity»
identify_required_information
Description
Identify information that is required to determine an Endpoint or Interaction_Device state.
«activity»
assess_information_update
Description
Assess the consumed information update to decide whether any further action needs to be taken. For example, a change in Endpoint state becoming busy or unavailable.
«interface»
Interaction_Device_Status
Description
This interface is information about the status of an Interaction_Device. For example, operational or in use.
«interface»
Endpoint_Status
Description
This interface is information about the status of an Endpoint. For example, available or unavailable.

Figure 462: Endpoint_and_Device_Status Service Policy

Endpoint_And_Device_Status

This service consumes information about the Interaction_Device and Endpoint state. For example, availability, call status, or connectivity status.

Interfaces

Interaction_Device_Status

This interface is information about the status of an Interaction_Device. For example, operational or in use.

Attribute

device_state	The state of an Interaction_Device to take part in an interaction. For example, radio channel open.

Endpoint_Status

This interface is information about the status of an Endpoint. For example, available or unavailable.

Attributes

endpoint_state	The state of an Endpoint affecting its ability to take part in an interaction. For example, busy or unavailable.
device_affiliation	An Endpoint's association to an Interaction_Device.

Activities

assess_information_update

Assess the consumed information update to decide whether any further action needs to be taken. For example, a change in Endpoint state becoming busy or unavailable.

identify_required_information

Identify information that is required to determine an Endpoint or Interaction_Device state.

5.4.2.26.7.1.5 Constraint

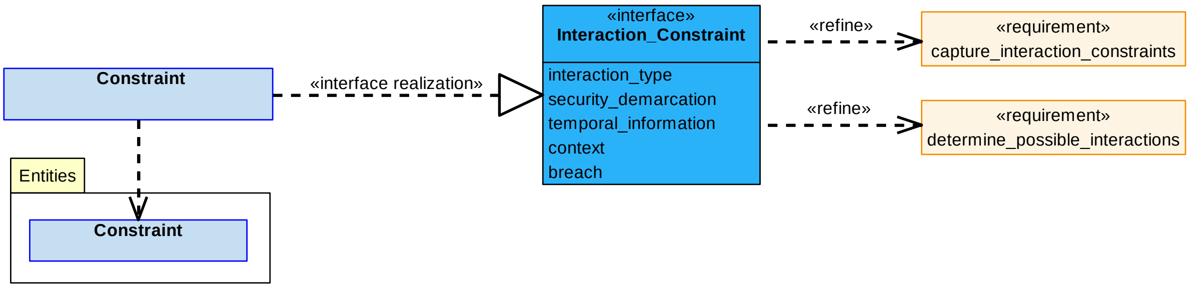
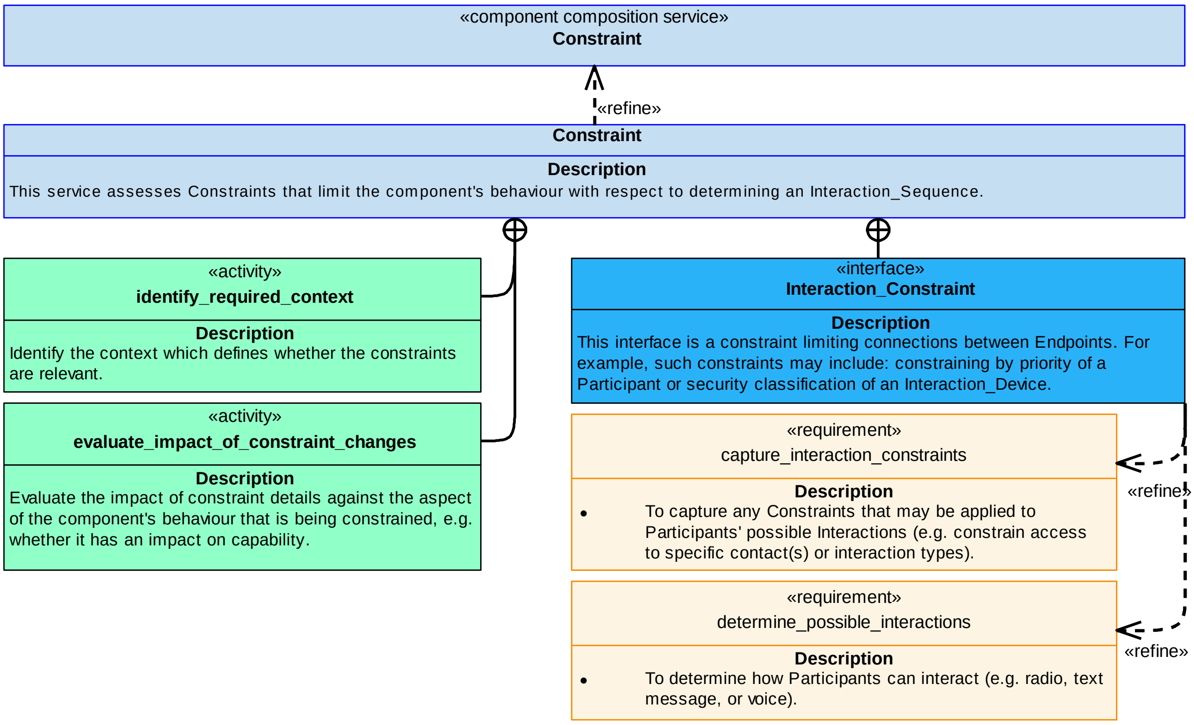


Figure 463: Constraint Service Definition



This service assesses Constraints that limit the component's behaviour with respect to determining an Interaction_Sequence.

Identify the context which defines whether the constraints are relevant.

Evaluate the impact of constraint details against the aspect of the component's behaviour that is being constrained, e.g. whether it has an impact on capability.

This interface is a constraint limiting connections between Endpoints. For example, such constraints may include: constraining by priority of a Participant or security classification of an Interaction_Device.

- To capture any Constraints that may be applied to Participants' possible Interactions (e.g. constrain access to specific contact(s) or interaction types).

- To determine how Participants can interact (e.g. radio, text message, or voice).

Figure 464: Constraint Service Policy

Constraint

This service assesses Constraints that limit the component's behaviour with respect to determining an Interaction_Sequence.

Interface

Interaction_Constraint

This interface is a constraint limiting connections between Endpoints. For example, such constraints may include: constraining by priority of a Participant or security classification of an Interaction_Device.

Attributes

interaction_type	A type of interaction that is prohibited between Endpoints, e.g. not allowing video calls.
security_demarcation	The allowed security level(s) at which an interaction can occur. For example, the required security clearance of an Endpoint.
temporal_information	Information covering constraints on timing such as, when a call is allowed to take place, or how long the call has to connect before failure.
context	The context in which the constraint is applicable.
breach	A statement that the constraint has been breached.

Activities

evaluate_impact_of_constraint_changes

Evaluate the impact of constraint details against the aspect of the component's behaviour that is being constrained, e.g. whether it has an impact on capability.

identify_required_context

Identify the context which defines whether the constraints are relevant.

5.4.2.26.7.1.6 Human_Interaction_Capability

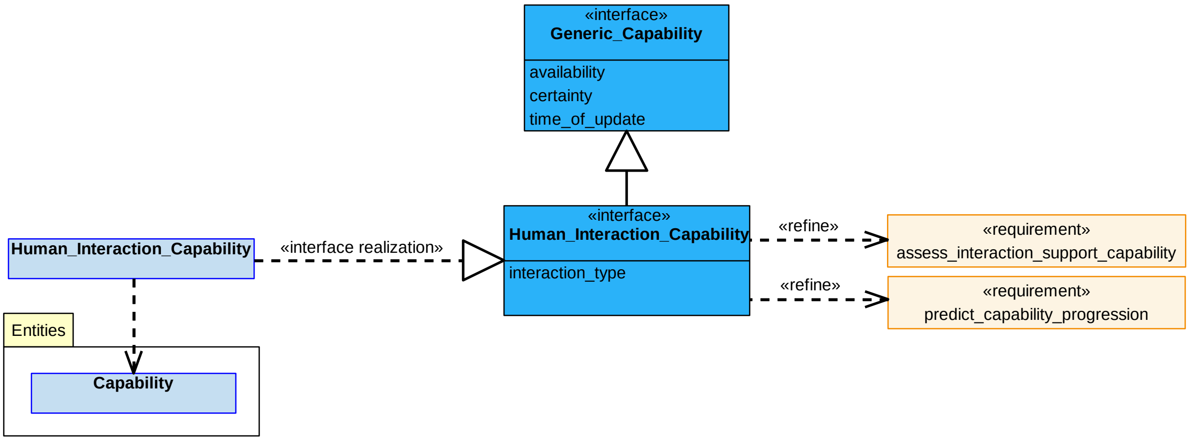
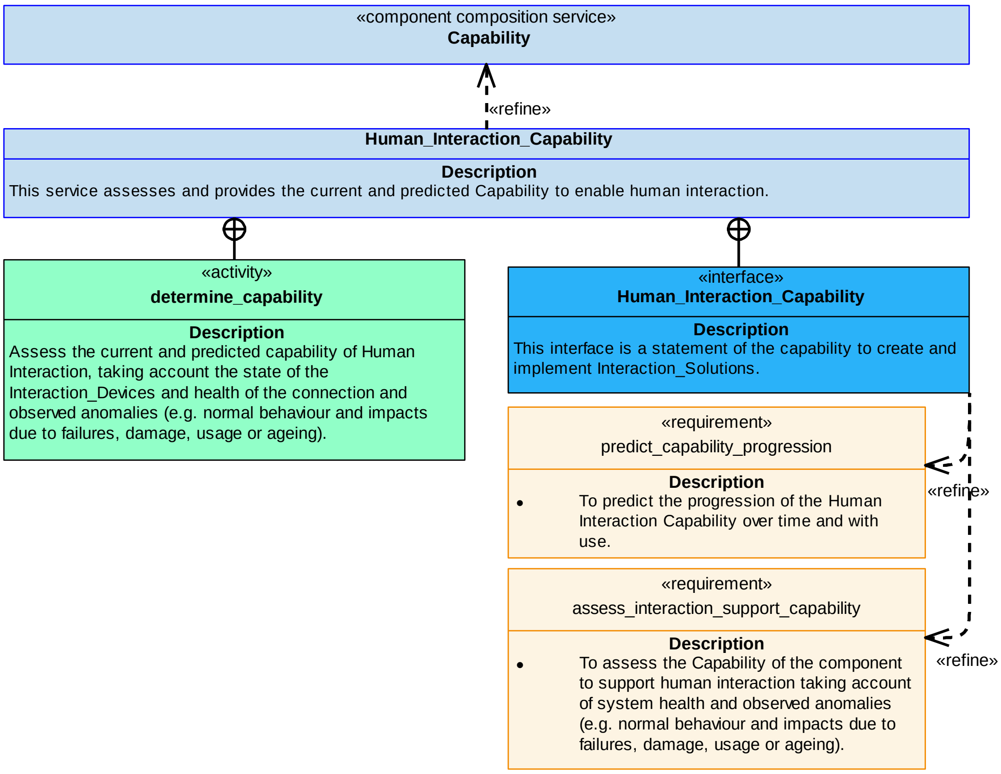


Figure 465: Human_Interaction Capability Service Definition


«component composition service»
Capability
«refine»
Human_Interaction_Capability
«activity»
determine_capability
«interface»
Human_Interaction_Capability
«requirement»
predict_capability_progression
«refine»
«requirement»
assess_interaction_support_capability
«refine»

Human_Interaction_Capability

Description
This service assesses and provides the current and predicted Capability to enable human interaction.

determine_capability

Description
Assess the current and predicted capability of Human Interaction, taking account the state of the Interaction_Devices and health of the connection and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Human_Interaction_Capability

Description
This interface is a statement of the capability to create and implement Interaction_Solutions.

predict_capability_progression

Description
- To predict the progression of the Human Interaction Capability over time and with use.

assess_interaction_support_capability

Description
- To assess the Capability of the component to support human interaction taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Figure 466: Human_Interaction_Capability Service Policy

Human_Interaction_Capability

This service assesses and provides the current and predicted Capability to enable human interaction.

Interface

Human_Interaction_Capability

This interface is a statement of the capability to create and implement Interaction_Solutions.

Attribute

interaction_type	The types of supported interaction. For example an audio or video call.

Activity

determine_capability

Assess the current and predicted capability of Human Interaction, taking account the state of the Interaction_Devices and health of the connection and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.26.7.1.7 Human_Interaction_Evidence

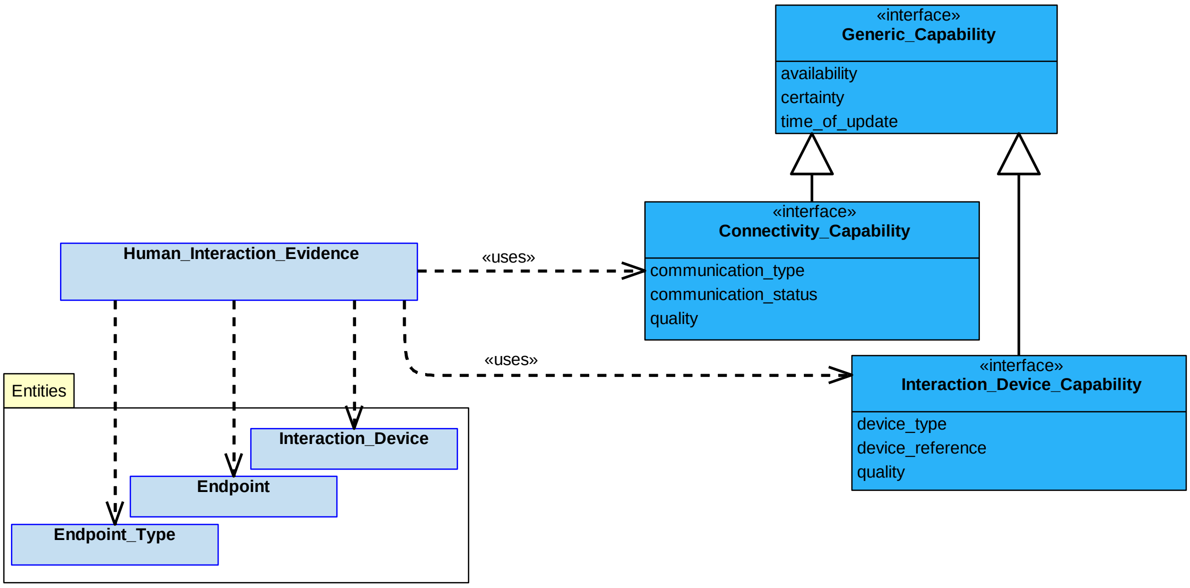


Figure 467: Human_Interaction_Evidence Service Definition

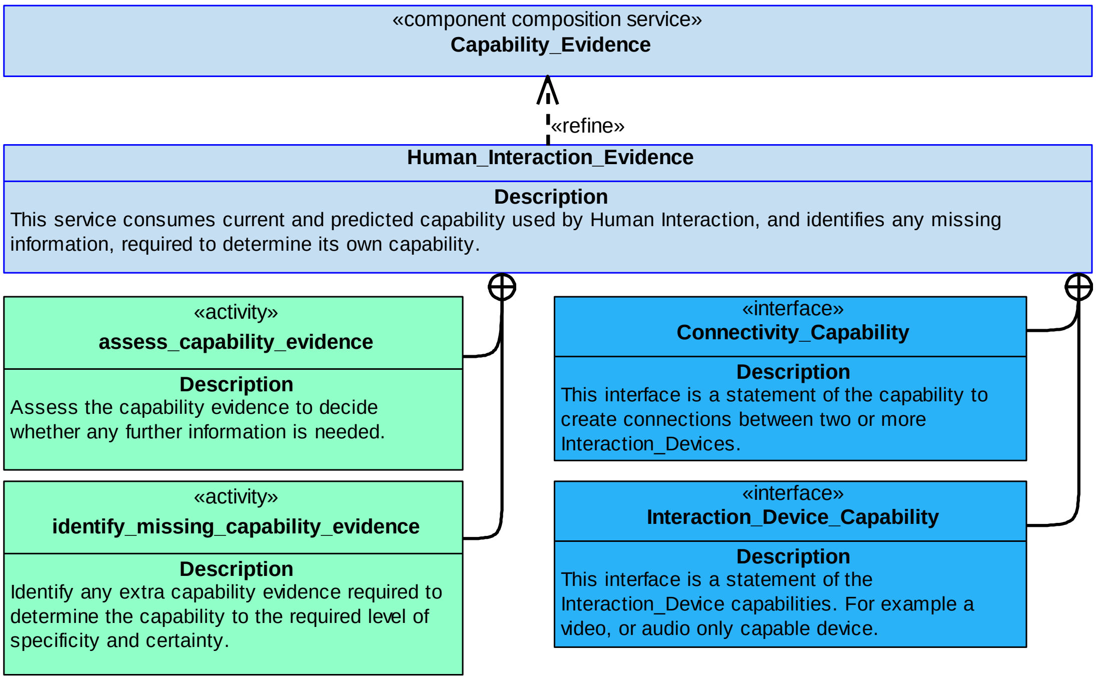

«component composition service»
Capability_Evidence
«refine»
Human_Interaction_Evidence
Description
This service consumes current and predicted capability used by Human Interaction, and identifies any missing information, required to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further information is needed.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.
«interface»
Connectivity_Capability
Description
This interface is a statement of the capability to create connections between two or more Interaction_Devices.
«interface»
Interaction_Device_Capability
Description
This interface is a statement of the Interaction_Device capabilities. For example a video, or audio only capable device.

Figure 468: Human_Interaction_Evidence Service Policy

Human_Interaction_Evidence

This service consumes current and predicted capability used by Human Interaction, and identifies any missing information, required to determine its own capability.

Interfaces

Connectivity_Capability

This interface is a statement of the capability to create connections between two or more Interaction_Devices.

Attributes

communication_type	The type of communication interaction mechanism. For example, support for streamed or static communications, and broadcast or peer to peer connections.
communication_status	The status of a connection, e.g. connection availability.
quality	The quality of a connection.

Interaction_Device_Capability

This interface is a statement of the Interaction_Device capabilities. For example a video, or audio only capable device.

Attributes

device_type	The specific functionality supported by an Interaction_Device, e.g. skype audio call, radio broadcast, or text message.
device_reference	The unique identifier for an Interaction_Device. For example, media access control address, IP address, or phone number and its associated Endpoint.

quality	The available quality from a device, e.g. video resolution and audio fidelity levels.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further information is needed.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.26.7.2 Service Dependencies

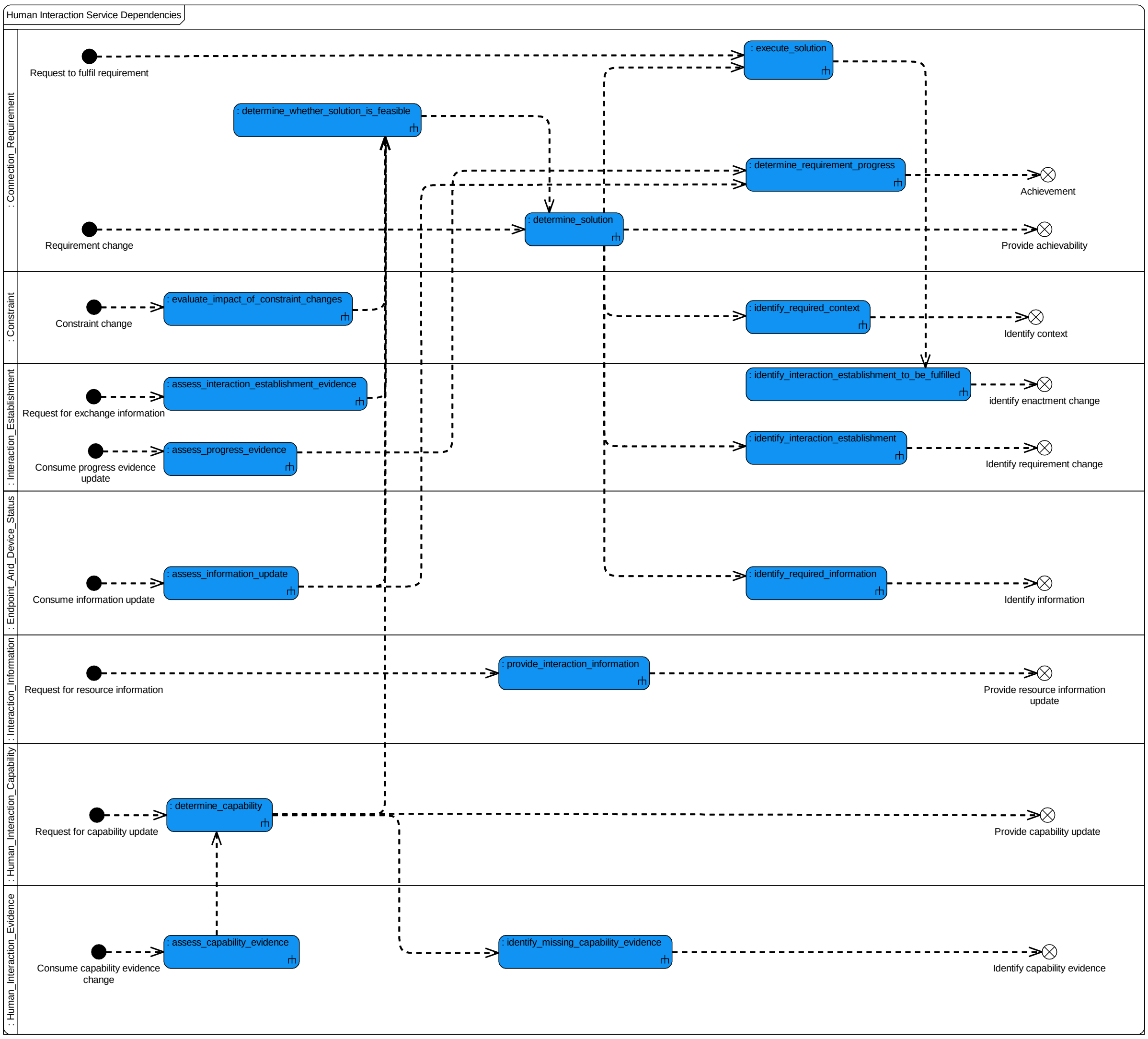

Human Interaction Service Dependencies
: Connection_Requirement
: Constraint
: Interaction_Establishment
: Endpoint_And_Device_Status
: Interaction_Information
: Human_Interaction_Capability
: Human_Interaction_Evidence
Request to fulfil requirement
: execute_solution
: determine_whether_solution_is_feasible
: determine_requirement_progress
Achievement
Requirement change
: determine_solution
Provide achievability
Constraint change
: evaluate_impact_of_constraint_changes
: identify_required_context
Identify context
Request for exchange information
: assess_interaction_establishment_evidence
: identify_interaction_establishment_to_be_fulfilled
identify enactment change
Consume progress evidence update
: assess_progress_evidence
: identify_interaction_establishment
Identify requirement change
Consume information update
: assess_information_update
: identify_required_information
Identify information
Request for resource information
: provide_interaction_information
Provide resource information update
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 469: Human Interaction Service Dependencies

5.4.2.27 Information Brokerage

5.4.2.27.1 Role

The role of Information Brokerage is to instigate and oversee information exchanges.

5.4.2.27.2 Overview

Control Architecture

Information Brokerage is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Information Brokerage will be notified when a Distributable_Item of information becomes available, it will determine if it is required by a third party, and it will determine a permitted Exchange between the Participants.

Examples of Use

Information Brokerage can be used to:

- Manage prioritisation of the exchange of information via internal communications (e.g. over a ground link).
- Manage the sharing and receiving information from 3rd parties.
- Coordinate the collation of information for the creation of post mission analysis reports.

5.4.2.27.3 Service Summary

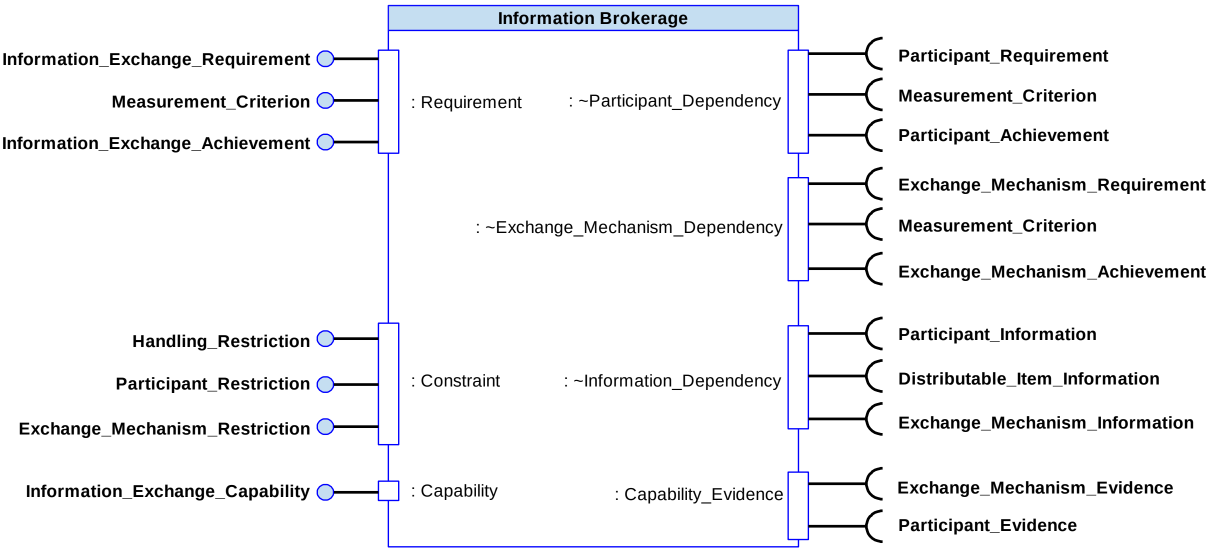

Information Brokerage
Information_Exchange_Requirement
Measurement_Criterion
Information_Exchange_Achievement
: Requirement
: ~Participant_Dependency
Participant_Requirement
Measurement_Criterion
Participant_Achievement
: ~Exchange_Mechanism_Dependency
Exchange_Mechanism_Requirement
Measurement_Criterion
Exchange_Mechanism_Achievement
Handling_Restriction
Participant_Restriction
Exchange_Mechanism_Restriction
: Constraint
: ~Information_Dependency
Participant_Information
Distributable_Item_Information
Exchange_Mechanism_Information
Information_Exchange_Capability
: Capability
: Capability_Evidence
Exchange_Mechanism_Evidence
Participant_Evidence

Figure 470: Information Brokerage Service Summary

5.4.2.27.4 Responsibilities

capture_requirements_for_information_exchange

- To capture the information Exchange requirement (e.g. the information that is to be exchanged, between whom and when).

capture_measurement_criteria_for_exchange

- To capture the Delivery_Characteristic required for an information exchange.

capture_exchange_constraints

- To capture the constraints associated with an Exchange.

determine_exchange_solution

- To determine an Exchange solution that meets the given Delivery_Characteristic requirements and Participant constraints for a Distributable_Item using available Exchange_Mechanism resources.

determine_allowable_exchange

- To determine whether an Exchange using a specific combination of Distributable_Items, Participants and Exchange_Mechanisms is allowable.

identify_exchange_in_progress_remains_feasible

- To identify if an Exchange in progress remains feasible, taking account of current resource constraints.

instigate_information_configuration

- To instigate the transformation of a Distributable_Item to meet an Information_Configuration required by a Participant (including the combination of Distributable_Items from multiple sources).

instigate_information_exchange

- To place the requirements for information Exchange onto Participants and Exchange_Mechanisms.

determine_quality_of_exchange_delivery

- To determine the quality of an information Exchange, measured against given requirements and measurement criteria.

capture_exchange_mechanism_of_participants

- To capture the information Exchange_Mechanisms of Participants.

assess_information_exchange_capability

- To assess the capability to provide information Exchange taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify what information is missing that could improve the certainty or specificity of the Exchange capability assessment.

predict_capability_progression

- To predict the progression of an Exchange capability over time and with use.

5.4.2.27.5 Subject Matter Semantics

The subject matter of Information Brokerage is the information Exchange needs of Participants.

Exclusions

The subject matter of Information Brokerage does not include:

- Authorisation of human users (e.g. operators) of the system.
- The actual storage and transfer of the information that Information Brokerage reasons about.
- The details of how different variations of exchange mechanism are achieved.
- The details of cryptography and data protection.
- The transformation of a Distributable_Item purely for the purposes of data transmission.

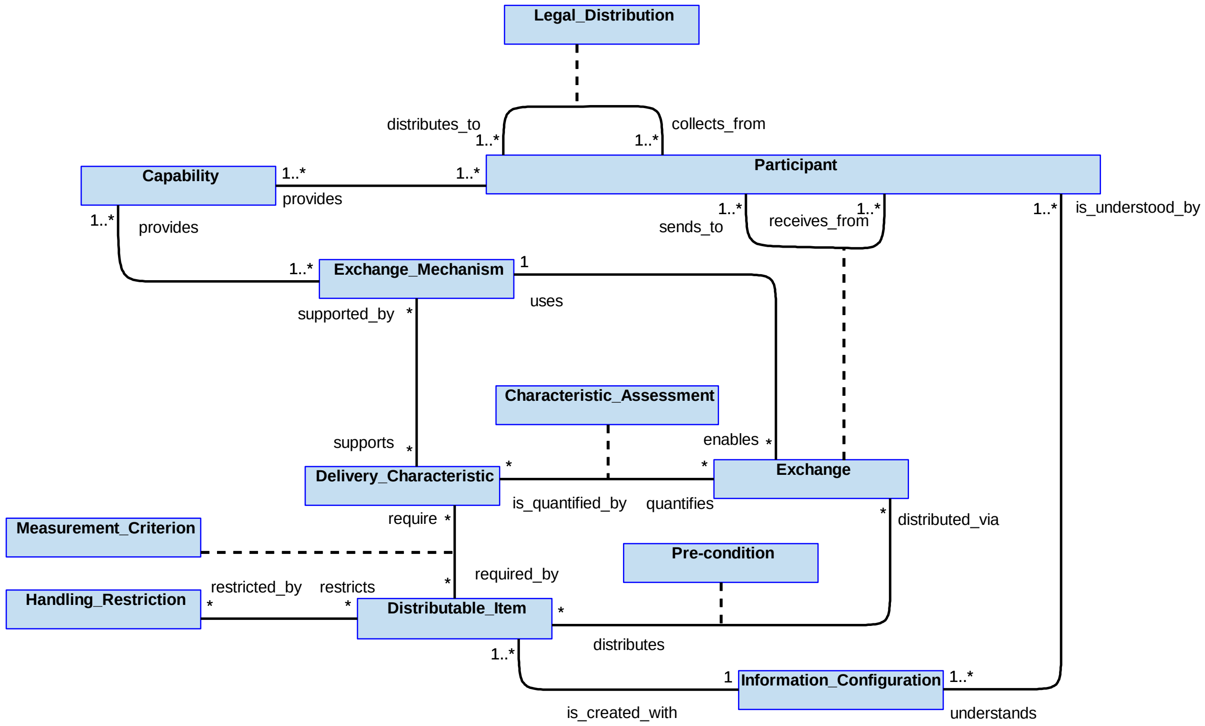


Figure 471: Information Brokerage Semantics

5.4.2.27.5.1 Entities

Capability

An assessment of the capability to perform Exchanges between Participants.

Delivery_Characteristic

A characteristic required for data delivery, e.g. priority or timeliness.

Distributable_Item

The item of information that is the subject of the exchange.

Exchange

The actual exchange between Participants and its properties, for example: When is it to happen? Did it happen? Is it happening? Is it possible?

Exchange_Mechanism

The actual exchange mechanism (resource) for the exchange, considering the type of exchange, the required Information_Configuration and a definition of where responsibility for carrying out the actual exchange lies.

Handling_Restriction

A restriction (constraint) on movement of data, examples include security classification and safety assurance level required.

Legal_Distribution

An allowed combination of Participants.

Characteristic_Assessment

An assessment of the Delivery_Characteristics achieved by a particular Exchange.

Measurement_Criterion

The quality measures required for an exchange of a particular type of data, e.g. sensor control commands need to be sent within a number of seconds.

Participant

A component, node, partition or other system element that can provide or request information. It is not intended that a participant will be a human operator.

Pre-condition

A condition that must be satisfied in order for the transfer of information to begin, e.g. the collation of data or the triggers for the start of transfer.

Information_Configuration

The way items of information that are required in an exchange are structured. This does not represent the way information will be formatted for distribution but only how the information should be structured for the participants, e.g. there are multiple ways to define the location of an object.

5.4.2.27.6 Design Rationale

5.4.2.27.6.1 Assumptions

- There are no specific assumptions associated with this component.

5.4.2.27.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Information Brokerage:

- Recording and Logging - Report generation is coordinated by this component (it is just another form of information exchange).
- Use of Communications - This component will coordinate the exchange of information (but not management of infrastructure).
- Cyber Defence - A key responsibility of this component is the determination of what information will be received from or sent to a third party (or even another node).
- Interfacing with Deployable Assets - A deployed asset is off platform so this component is likely to be involved in coordinating information exchanges.

Other Factors that were Taken into Account

- This component will be used to understand the information requirements and also where the local node's information requirements can be met. It will manage information distribution within a system, e.g. node to node distribution, as well as to external systems.
- The allowed Participants, including their information requirements and classifications, can change on a mission to mission basis.
- This component will only handle metadata about information and Participants; it will not handle any tangible data item (e.g. sensor product or detection), nor be directly involved in handling data transfers.
- The identification of information to be collated for later use is covered by this component and extends to post mission report generation.

Exploitation Considerations

- Combination of information from multiple sources can be accommodated by different instances of initiating condition.
- Differing mission to mission and/or platform-to-platform permissions for information exchange can be accommodated by different instances of Participants and Legal_Distributions.
- Differing standards for Exchange can be accommodated by different instances of Exchange_Mechanism, as this component is not involved in the actual transfer.
- Differing data formats for use by other Participants can be accommodated by different instances of Distributable_Item.

- Differing security, safety and control authorisation groups can be accommodated by different instances of Handling_Restriction or Legal_Distribution.
- Multiple instances or variants of Information Brokerage may be used to fulfil the needs of a whole system, with the distributed components coordinating with each other on multiple platforms, to determine the exchange capabilities.
- This component is intended to manage information exchange where there is the possibility of contention, loss, security issues or a time delay between data being sent to when it is required (typically over communication links). However if managed information exchange is not required (e.g. due to highly trusted, high bandwidth, high reliability links and clear definition of required exchanges) neither is this component.

5.4.2.27.6.3 Safety Considerations

The indicative IDAL is DAL C*.*

The rationale behind this is:

- Failure of this component may result in the inability to transfer data between, for example, a ground based control station and the air vehicle. This is primarily a concern for UAVs, but may apply to manned air vehicles where some functions are controlled by external users. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) then the air vehicle will have been designed to mitigate a loss of communications. For a UAS this would be achieved by relying on pre-determined automatic / autonomous behaviour. For this failure mode it is concluded that failure of this component may result a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.

This component does not handle the data being transferred. Therefore, this component cannot corrupt data.

5.4.2.27.6.4 Security Considerations

The indicative security classification is O-S.

This component is involved in planning information exchanges and will handle metadata about destinations and data to be exchanged with third parties, and determining the Exchange_Mechanism based upon the Handling_Restrictions and Legal_Distribution in place. Tactical information will have higher confidentiality requirements, and the classification of the metadata etc. will be increased. This component is cognisant of the security classification and confidentiality of information being brokered. Instances of this component within different security domains may need to be able to coordinate. This component is expected to have an understanding of the trustworthiness of Participant services, and may prevent an exchange based on this (e.g. determining whether Exchanges between these Participants is a "whitelisted" Legal_Distribution).

Due to its role in the security of information exchange, this component is a likely target for attack and will benefit from increased levels of security protection.

The component will satisfy security related functions by:

- **Identifying Data Sources** and their associated trustworthiness.
- **Logging of Security Information** relating to the use of specific exchanges, access brokered to high-value data, etc.
- **Maintaining Audit Records** for information exchanges made during the mission, including source and recipient, etc.
- **Supporting Secure Remote Operation** through the handling of control messages and handover messages, etc.

The component will satisfy security enforcing functions relating to:

- Identifying the classification of an Exchange.
- **Ensuring Separation of Security Domains** by placing requirements for controls to prevent inappropriate cross-domain communication.
- **Restricting Access to Data** by determining whether a specific service or platform exchange is permissible, e.g. according to the domain of the provider and recipient, or due to the available Exchange_Mechanisms.

5.4.2.27.7 Services

5.4.2.27.7.1 Service Definitions

5.4.2.27.7.1.1 Requirement

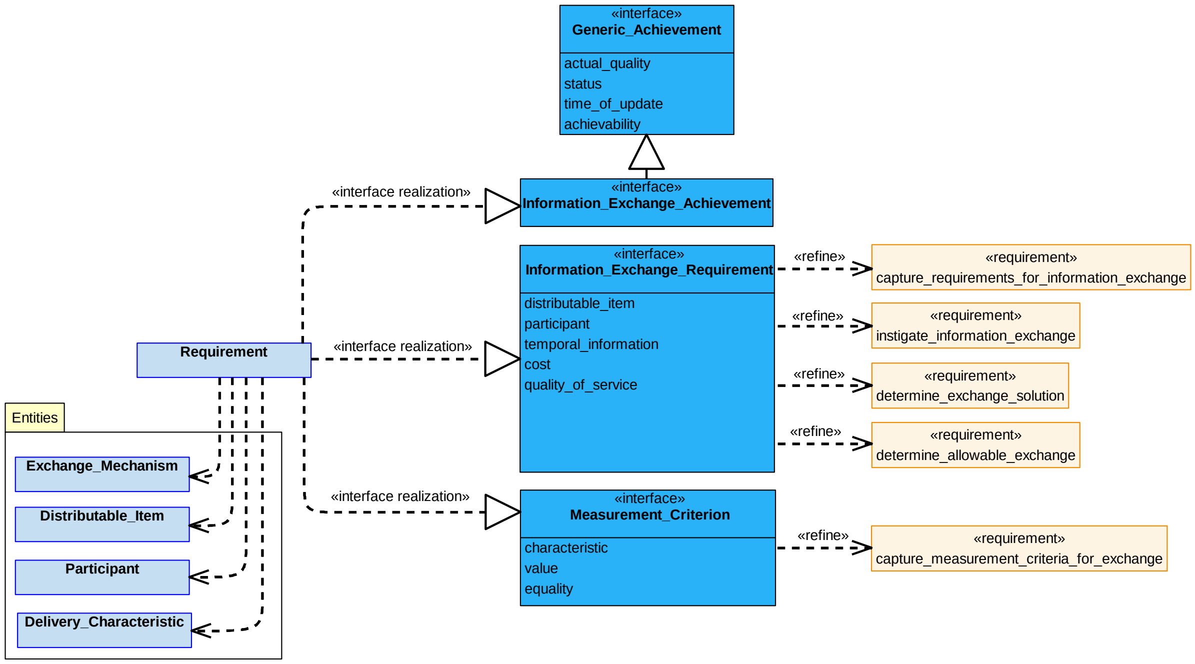


Figure 472: Requirement Service Definition

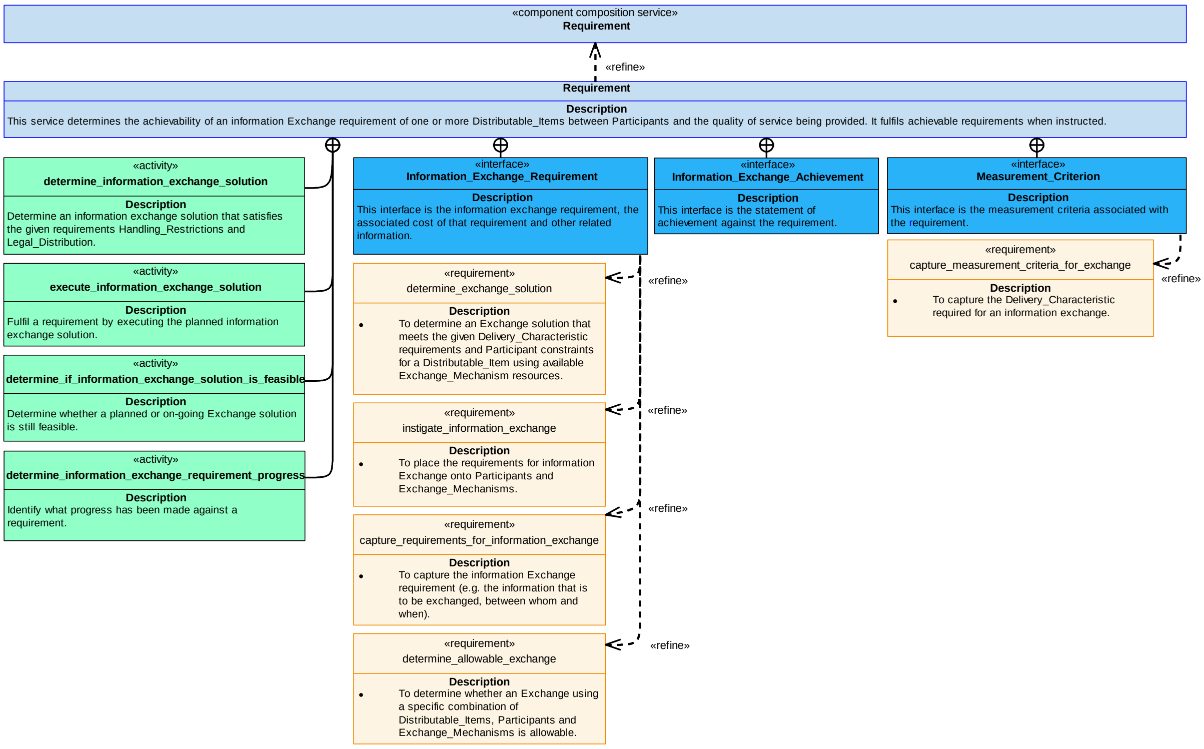

«component composition service»
Requirement
«refine»
Requirement
Description
This service determines the achievability of an information Exchange requirement of one or more Distributable_Items between Participants and the quality of service being provided. It fulfils achievable requirements when instructed.
«activity»
determine_information_exchange_solution
Description
Determine an information exchange solution that satisfies the given requirements Handling_Restrictions and Legal_Distribution.
«activity»
execute_information_exchange_solution
Description
Fulfil a requirement by executing the planned information exchange solution.
«activity»
determine_if_information_exchange_solution_is_feasible
Description
Determine whether a planned or on-going Exchange solution is still feasible.
«activity»
determine_information_exchange_requirement_progress
Description
Identify what progress has been made against a requirement.
«interface»
Information_Exchange_Requirement
Description
This interface is the information exchange requirement, the associated cost of that requirement and other related information.
«interface»
Information_Exchange_Achievement
Description
This interface is the statement of achievement against the requirement.
«interface»
Measurement_Criterion
Description
This interface is the measurement criteria associated with the requirement.
«requirement»
capture_measurement_criteria_for_exchange
Description
To capture the Delivery_Characteristic required for an information exchange.
«refine»
«requirement»
determine_exchange_solution
Description
To determine an Exchange solution that meets the given Delivery_Characteristic requirements and Participant constraints for a Distributable_Item using available Exchange_Mechanism resources.
«refine»
«requirement»
instigate_information_exchange
Description
To place the requirements for information Exchange onto Participants and Exchange_Mechanisms.
«refine»
«requirement»
capture_requirements_for_information_exchange
Description
To capture the information Exchange requirement (e.g. the information that is to be exchanged, between whom and when).
«refine»
«requirement»
determine_allowable_exchange
Description
To determine whether an Exchange using a specific combination of Distributable_Items, Participants and Exchange_Mechanisms is allowable.
«refine»

Figure 473: Requirement Service Policy

Requirement

This service determines the achievability of an information Exchange requirement of one or more Distributable_Items between Participants and the quality of service being provided. It fulfils achievable requirements when instructed.

Interfaces

Information_Exchange_Requirement

This interface is the information exchange requirement, the associated cost of that requirement and other related information.

Attributes

distributable_item	The Distributable_Item required and any Information_Configuration requirements associated with the Distributable_Item.
participant	The Participant involved in the Exchange.
temporal_information	Information covering timing, such as start and end times of the requirement.
cost	The cost of executing the solution, e.g. resources used or time taken.
quality_of_service	A measure of required or proposed Exchange solution quality.

Measurement_Criterion

This interface is the measurement criteria associated with the requirement.

Attributes

characteristic	The Delivery_Characteristic to be measured.

value	The value of the Delivery_Characteristic.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Information_Exchange_Achievement

This interface is the statement of achievement against the requirement.

Activities

determine_information_exchange_solution

Determine an information exchange solution that satisfies the given requirements Handling_Restrictions and Legal_Distribution.

execute_information_exchange_solution

Fulfil a requirement by executing the planned information exchange solution.

determine_if_information_exchange_solution_is_feasible

Determine whether a planned or on-going Exchange solution is still feasible.

determine_information_exchange_requirement_progress

Identify what progress has been made against a requirement.

5.4.2.27.7.1.2 Participant_Dependency

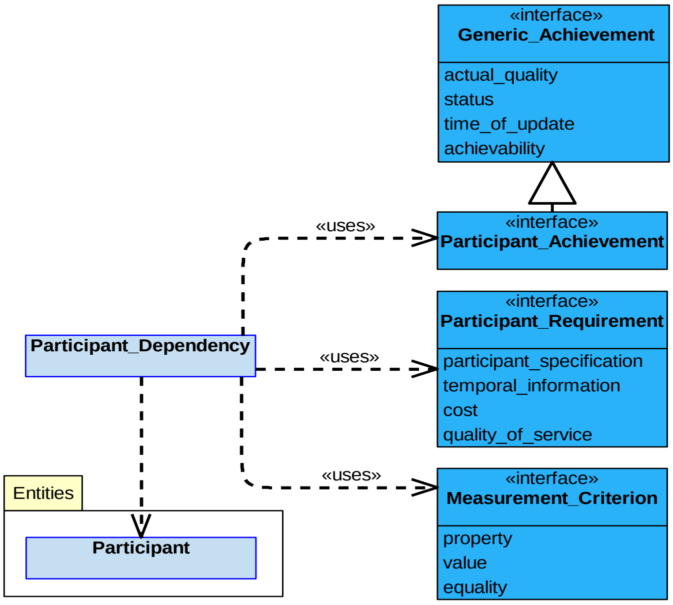


Figure 474: Participant_Dependency Service Definition

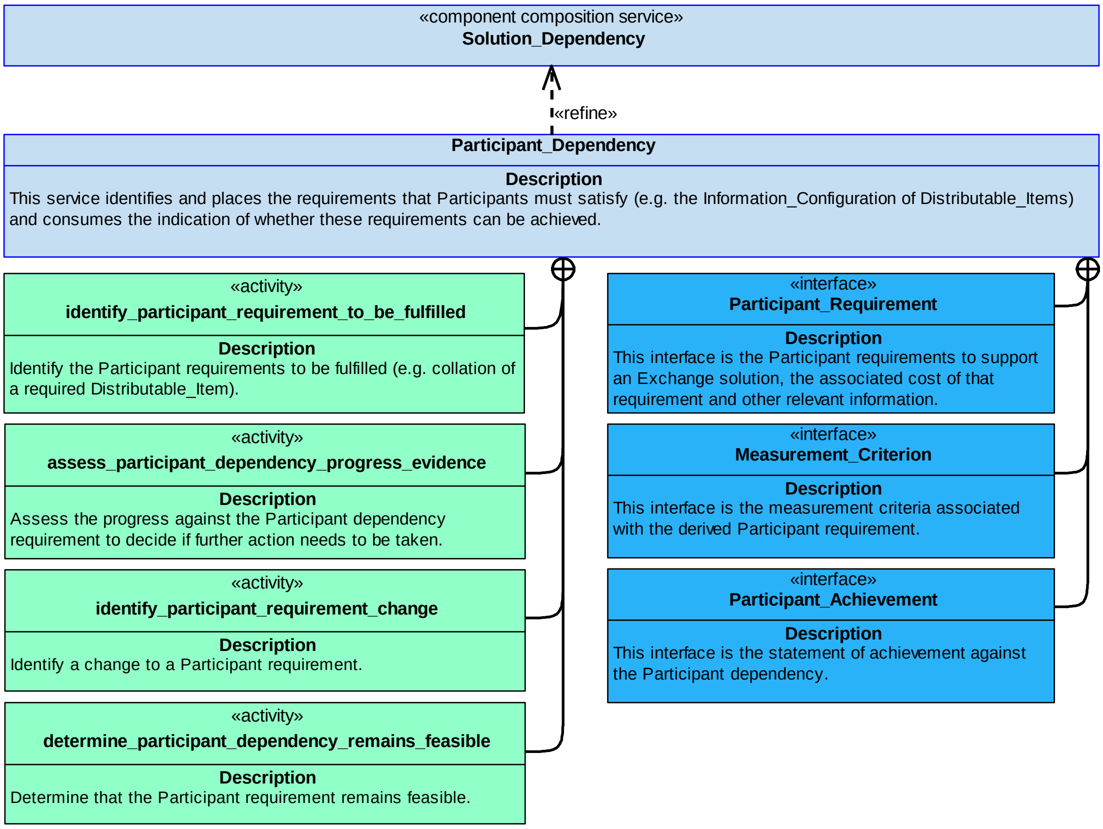

«component composition service»
Solution_Dependency
«refine»
Participant_Dependency
Description
This service identifies and places the requirements that Participants must satisfy (e.g. the Information_Configuration of Distributable_Items) and consumes the indication of whether these requirements can be achieved.
«activity»
identify_participant_requirement_to_be_fulfilled
Description
Identify the Participant requirements to be fulfilled (e.g. collation of a required Distributable_Item).
«activity»
assess_participant_dependency_progress_evidence
Description
Assess the progress against the Participant dependency requirement to decide if further action needs to be taken.
«activity»
identify_participant_requirement_change
Description
Identify a change to a Participant requirement.
«activity»
determine_participant_dependency_remains_feasible
Description
Determine that the Participant requirement remains feasible.
«interface»
Participant_Requirement
Description
This interface is the Participant requirements to support an Exchange solution, the associated cost of that requirement and other relevant information.
«interface»
Measurement_Criterion
Description
This interface is the measurement criteria associated with the derived Participant requirement.
«interface»
Participant_Achievement
Description
This interface is the statement of achievement against the Participant dependency.

Figure 475: Participant_Dependency Service Policy

Participant_Dependency

This service identifies and places the requirements that Participants must satisfy (e.g. the Information_Configuration of Distributable_Items) and consumes the indication of whether these requirements can be achieved.

Interfaces

Participant_Requirement

This interface is the Participant requirements to support an Exchange solution, the associated cost of that requirement and other relevant information.

Attributes

participant_specification	The specification of an Exchange of Distributable_Item(s) to be provided to an Exchange_Mechanism.
temporal_information	Information covering timing, such as start and end times of the derived requirement.
cost	The cost of executing the solution, e.g. resources used or time taken.
quality_of_service	A measure of required or proposed solution quality.

Measurement_Criterion

This interface is the measurement criteria associated with the derived Participant requirement.

Attributes

property	The Participant property to be measured.
value	The value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Participant_Achievement

This interface is the statement of achievement against the Participant dependency.

Activities

identify_participant_requirement_to_be_fulfilled

Identify the Participant requirements to be fulfilled (e.g. collation of a required Distributable_Item).

assess_participant_dependency_progress_evidence

Assess the progress against the Participant dependency requirement to decide if further action needs to be taken.

identify_participant_requirement_change

Identify a change to a Participant requirement.

determine_participant_dependency_remains_feasible

Determine that the Participant requirement remains feasible.

5.4.2.27.7.1.3 Exchange_Mechanism_Dependency

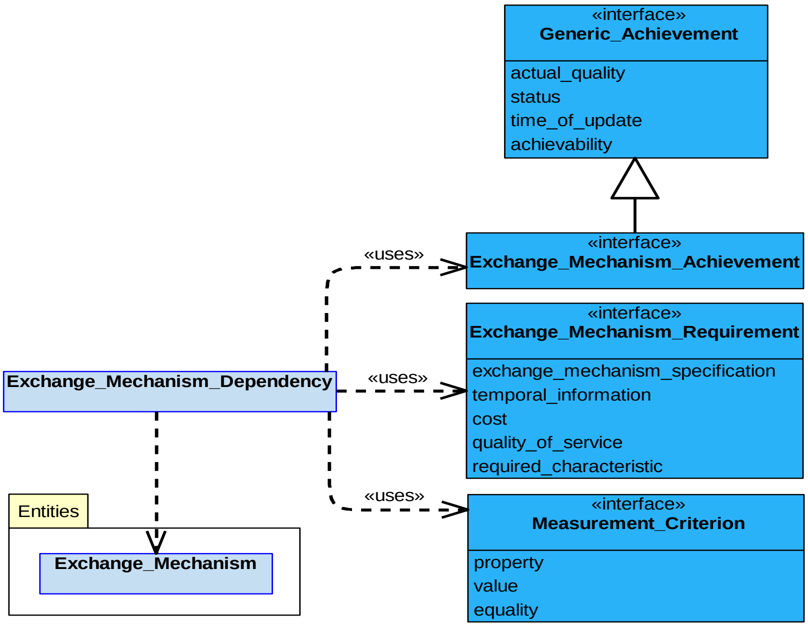
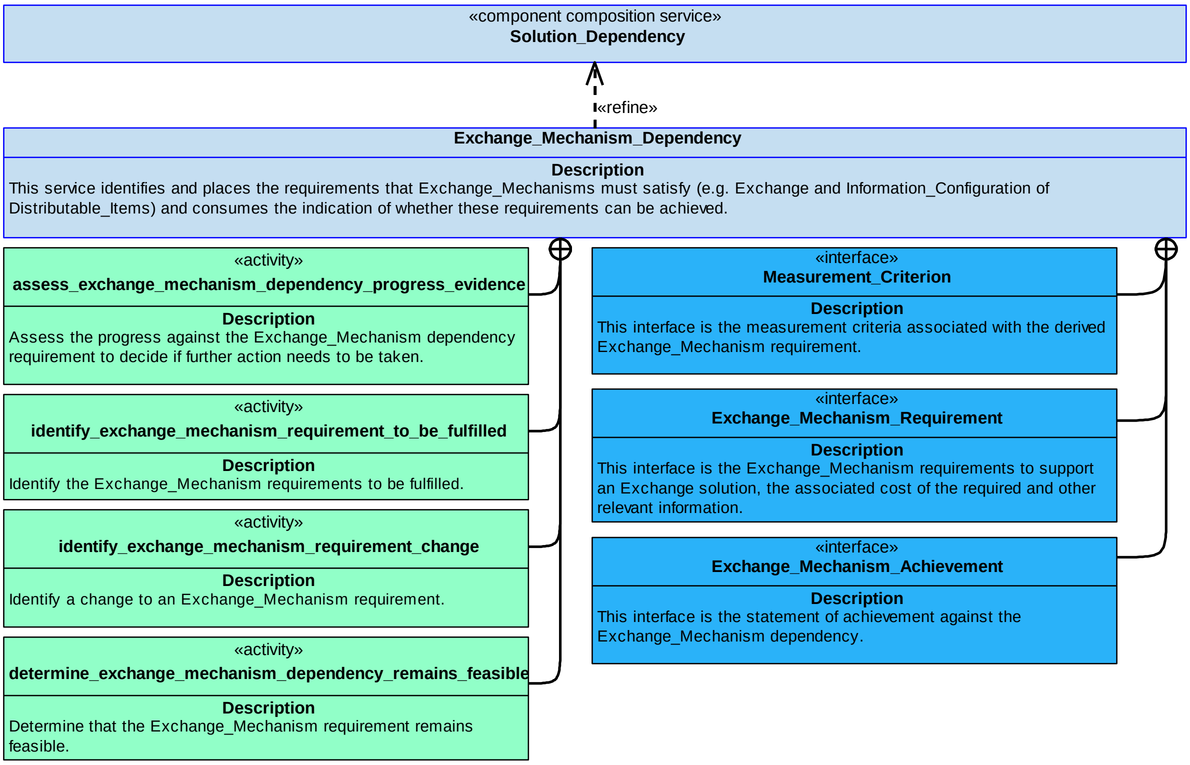


Figure 476: Exchange_Mechanism_Dependency Service Definition


«component composition service»
Solution_Dependency
«refine»
Exchange_Mechanism_Dependency

Description
This service identifies and places the requirements that Exchange_Mechanisms must satisfy (e.g. Exchange and Information_Configuration of Distributable_Items) and consumes the indication of whether these requirements can be achieved.

«activity»
assess_exchange_mechanism_dependency_progress_evidence
Description
Assess the progress against the Exchange_Mechanism dependency requirement to decide if further action needs to be taken.

«activity»
identify_exchange_mechanism_requirement_to_be_fulfilled
Description
Identify the Exchange_Mechanism requirements to be fulfilled.

«activity»
identify_exchange_mechanism_requirement_change
Description
Identify a change to an Exchange_Mechanism requirement.

«activity»
determine_exchange_mechanism_dependency_remains_feasible
Description
Determine that the Exchange_Mechanism requirement remains feasible.

«interface»
Measurement_Criterion
Description
This interface is the measurement criteria associated with the derived Exchange_Mechanism requirement.

«interface»
Exchange_Mechanism_Requirement
Description
This interface is the Exchange_Mechanism requirements to support an Exchange solution, the associated cost of the required and other relevant information.

«interface»
Exchange_Mechanism_Achievement
Description
This interface is the statement of achievement against the Exchange_Mechanism dependency.

Figure 477: Exchange_Mechanism_Dependency Service Policy

Exchange_Mechanism_Dependency

This service identifies and places the requirements that Exchange_Mechanisms must satisfy (e.g. Exchange and Information_Configuration of Distributable_Items) and consumes the indication of whether these requirements can be achieved.

Interfaces

Exchange_Mechanism_Requirement

This interface is the Exchange_Mechanism requirements to support an Exchange solution, the associated cost of the required and other relevant information.

Attributes

exchange_mechanism_specification	The specification of an Exchange of Distributable_Item(s) between Participants using an Exchange_Mechanism.
temporal_information	Information covering timing, such as start and end times of the derived requirement.
cost	The cost of executing the solution, e.g. resources used or time taken.
quality_of_service	A measure of required or proposed solution quality.
required_characteristic	The required Delivery_Characteristic(s) of the Exchange, such as classification.

Measurement_Criterion

This interface is the measurement criteria associated with the derived Exchange_Mechanism requirement.

Attributes

property	The Exchange_Mechanism property to be measured.
value	The value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Exchange_Mechanism_Achievement

This interface is the statement of achievement against the Exchange_Mechanism dependency.

Activities

assess_exchange_mechanism_dependency_progress_evidence

Assess the progress against the Exchange_Mechanism dependency requirement to decide if further action needs to be taken.

identify_exchange_mechanism_requirement_change

Identify a change to an Exchange_Mechanism requirement.

identify_exchange_mechanism_requirement_to_be_fulfilled

Identify the Exchange_Mechanism requirements to be fulfilled.

determine_exchange_mechanism_dependency_remains_feasible

Determine that the Exchange_Mechanism requirement remains feasible.

5.4.2.27.7.1.4 Information_Dependency

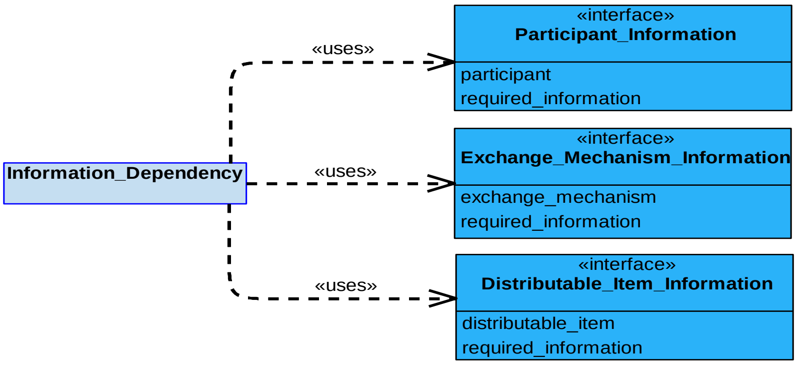
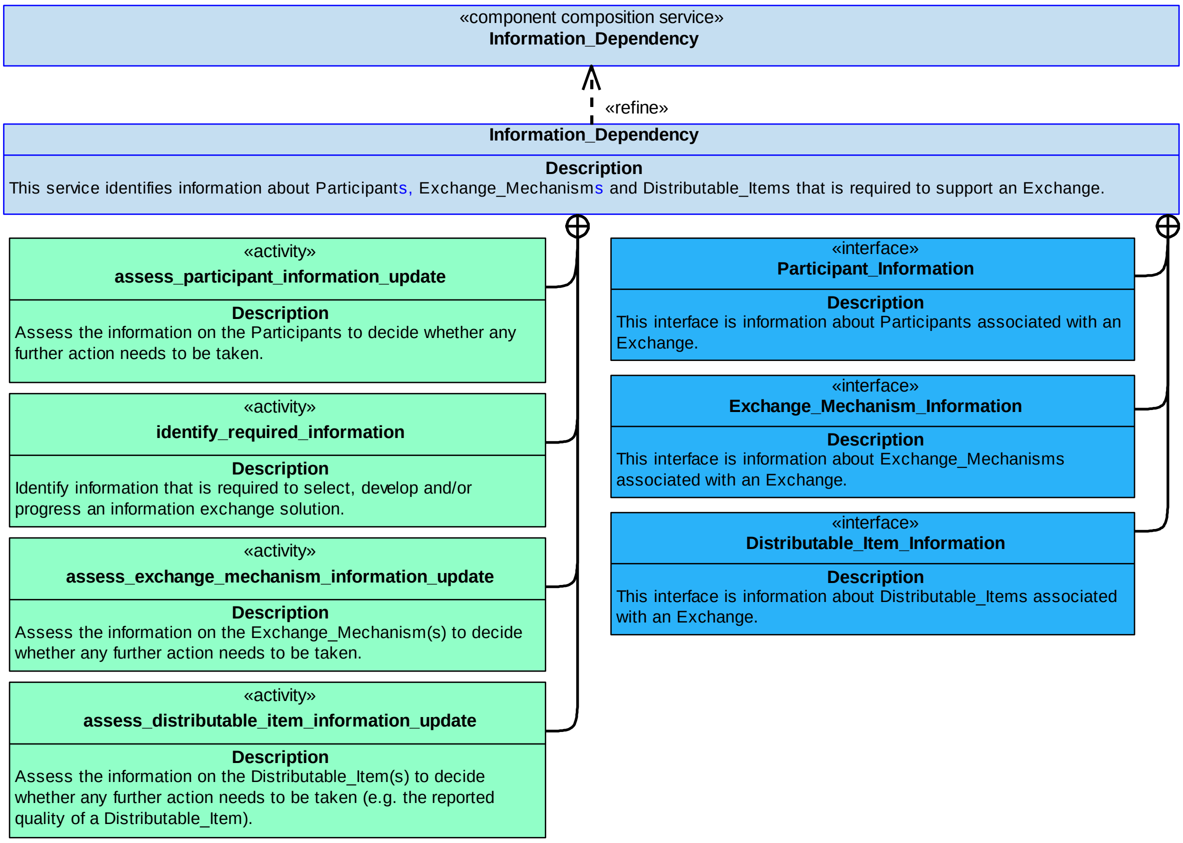


Figure 478: Information_Dependency Service Definition


«component composition service»
Information_Dependency
«refine»
Information_Dependency
«activity»
assess_participant_information_update
«activity»
identify_required_information
«activity»
assess_exchange_mechanism_information_update
«activity»
assess_distributable_item_information_update
«interface»
Participant_Information
«interface»
Exchange_Mechanism_Information
«interface»
Distributable_Item_Information

Description
This service identifies information about Participants, Exchange_Mechanisms and Distributable_Items that is required to support an Exchange.

Description
Assess the information on the Participants to decide whether any further action needs to be taken.

Description
Identify information that is required to select, develop and/or progress an information exchange solution.

Description
Assess the information on the Exchange_Mechanism(s) to decide whether any further action needs to be taken.

Description
Assess the information on the Distributable_Item(s) to decide whether any further action needs to be taken (e.g. the reported quality of a Distributable_Item).

Description
This interface is information about Participants associated with an Exchange.

Description
This interface is information about Exchange_Mechanisms associated with an Exchange.

Description
This interface is information about Distributable_Items associated with an Exchange.

Figure 479: Information_Dependency Service Policy

Information_Dependency

This service identifies information about Participants, Exchange_Mechanisms and Distributable_Items that is required to support an Exchange.

Interfaces

Participant_Information

This interface is information about Participants associated with an Exchange.

Attributes

participant	A Participant associated with an Exchange.
required_information	The information required about a Participant, e.g. Distributable_Items being handled by that Participant outside the Exchange under consideration.

Exchange_Mechanism_Information

This interface is information about Exchange_Mechanisms associated with an Exchange.

Attributes

exchange_mechanism	An Exchange_Mechanism associated with an Exchange.
required_information	The information needed about an Exchange_Mechanism; e.g. other Participants currently using that Exchange_Mechanism outside the Exchange under consideration.

Distributable_Item_Information

This interface is information about Distributable_Items associated with an Exchange.

Attributes

distributable_item	A Distributable_Item associated with an Exchange.
required_information	The information needed about a Distributable_Item; e.g. whether it is currently collated and ready for distribution and the Information_Configuration of the Distributable_Item.

Activities

assess_participant_information_update

Assess the information on the Participants to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to select, develop and/or progress an information exchange solution.

assess_exchange_mechanism_information_update

Assess the information on the Exchange_Mechanism(s) to decide whether any further action needs to be taken.

assess_distributable_item_information_update

Assess the information on the Distributable_Item(s) to decide whether any further action needs to be taken (e.g. the reported quality of a Distributable_Item).

5.4.2.27.7.1.5 Constraint

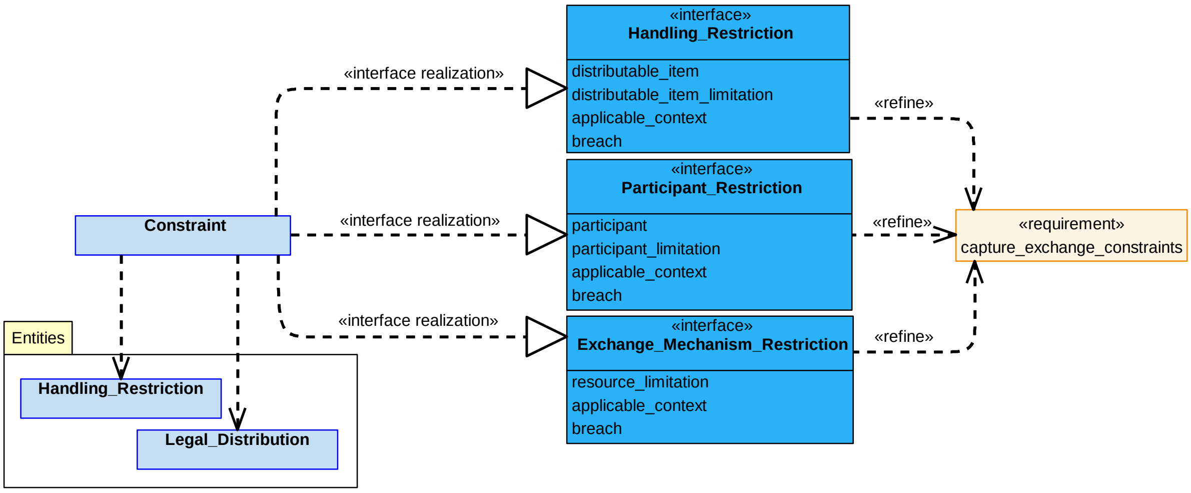


Figure 480: Constraint Service Definition

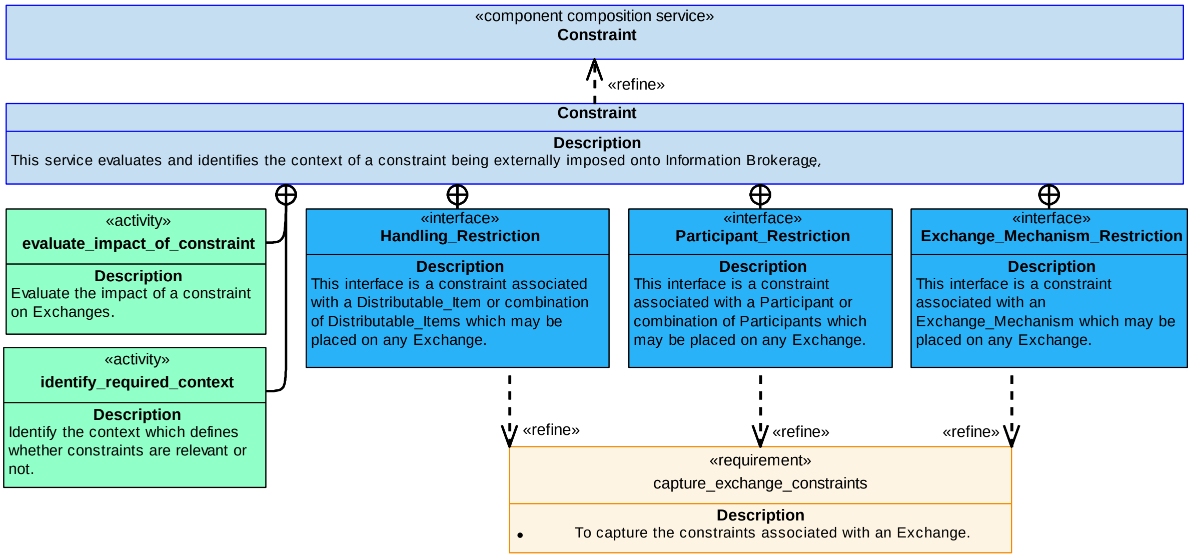

«component composition service»
Constraint
«refine»
Constraint
Description
This service evaluates and identifies the context of a constraint being externally imposed onto Information Brokerage.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of a constraint on Exchanges.
«activity»
identify_required_context
Description
Identify the context which defines whether constraints are relevant or not.
«interface»
Handling_Restriction
Description
This interface is a constraint associated with a Distributable_Item or combination of Distributable_Items which may be placed on any Exchange.
«interface»
Participant_Restriction
Description
This interface is a constraint associated with a Participant or combination of Participants which may be placed on any Exchange.
«interface»
Exchange_Mechanism_Restriction
Description
This interface is a constraint associated with an Exchange_Mechanism which may be placed on any Exchange.
«refine»
«refine»
«refine»
«requirement»
capture_exchange_constraints
Description
To capture the constraints associated with an Exchange.

Figure 481: Constraint Service Policy

Constraint

This service evaluates and identifies the context of a constraint being externally imposed onto Information Brokerage.

Interfaces

Handling_Restriction

This interface is a constraint associated with a Distributable_Item or combination of Distributable_Items which may be placed on any Exchange.

Attributes

distributable_item	The Distributable_Item(s) to which the constraint applies.

distributable_item_limitation	The handling limitation specified by the constraint (e.g. classification of any exchange of one or more Distributable_Items).
applicable_context	The context in which the constraint is applicable.
breach	A statement that a restriction has been breached.

Participant_Restriction

This interface is a constraint associated with a Participant or combination of Participants which may be placed on any Exchange.

Attributes

participant	The Participant(s) to which a constraint applies.
participant_limitation	The participant limitation specified by the constraint.
applicable_context	The context in which the constraint is applicable.
breach	A statement that a restriction has been breached.

Exchange_Mechanism_Restriction

This interface is a constraint associated with an Exchange_Mechanism which may be placed on any Exchange.

Attributes

resource_limitation	The exchange mechanism limitation defined by the constraint (e.g. a time-bound restriction on the use of a particular Exchange_Mechanism).
applicable_context	The context in which the constraint is applicable.
breach	A statement that a restriction has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of a constraint on Exchanges.

identify_required_context

Identify the context which defines whether constraints are relevant or not.

5.4.2.27.7.1.6 Capability

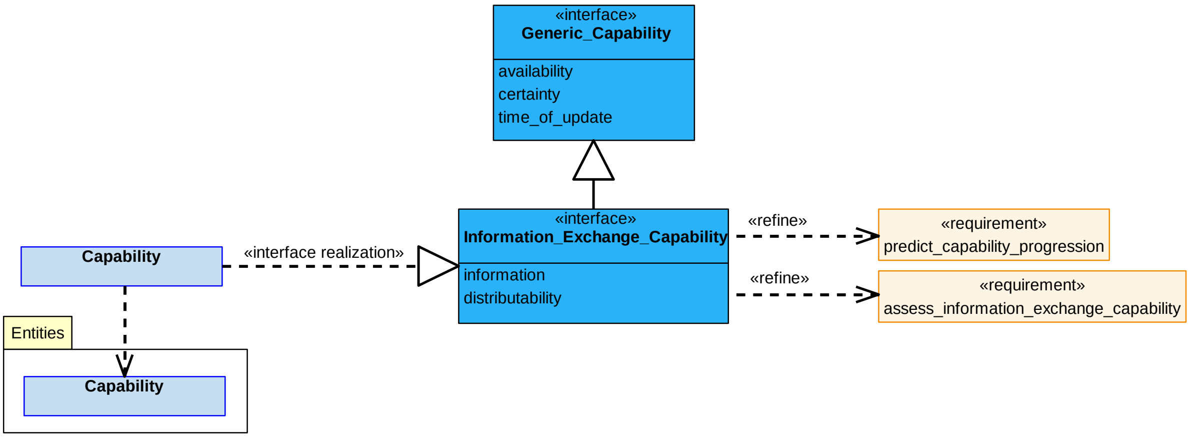


Figure 482: Capability Service Definition

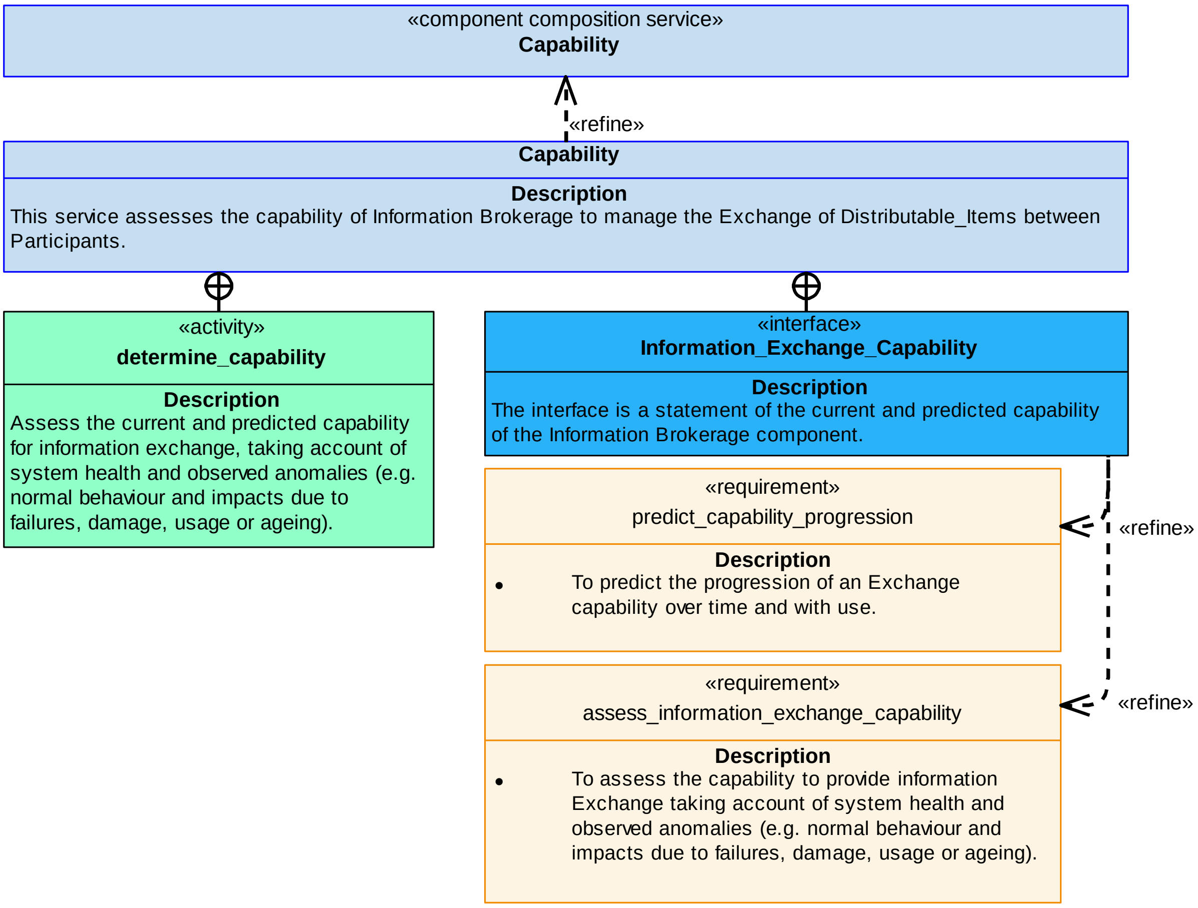

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the capability of Information Brokerage to manage the Exchange of Distributable_Items between Participants.
«activity»
determine_capability
Description
Assess the current and predicted capability for information exchange, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Information_Exchange_Capability
Description
The interface is a statement of the current and predicted capability of the Information Brokerage component.
«requirement»
predict_capability_progression
Description
• To predict the progression of an Exchange capability over time and with use.
«refine»
«requirement»
assess_information_exchange_capability
Description
• To assess the capability to provide information Exchange taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 483: Capability Service Policy

Capability

This service assesses the capability of Information Brokerage to manage the Exchange of Distributable_Items between Participants.

Interface

Information_Exchange_Capability

The interface is a statement of the current and predicted capability of the Information Brokerage component.

Attributes

information	Distributable_Items that can be exchanged (e.g. Information_Configurations that can currently be generated for a given Distributable_Item).
distributability	Participants that can be supported (e.g. Legal_Distributions where a currently allowable Exchange_Mechanism exists).

Activity

determine_capability

Assess the current and predicted capability for information exchange, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.27.7.1.7 Capability_Evidence

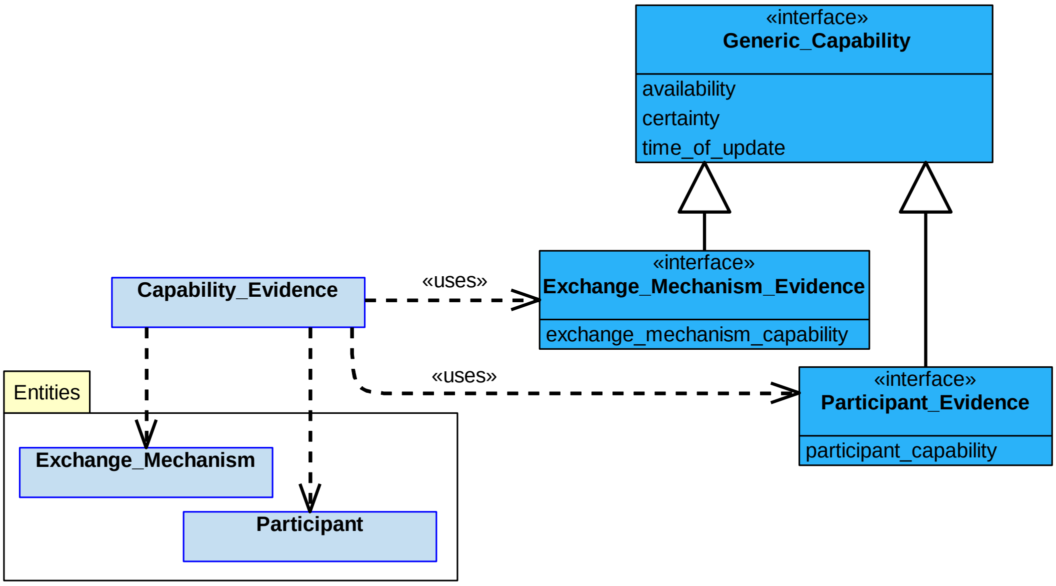


Figure 484: Capability_Evidence Service Definition

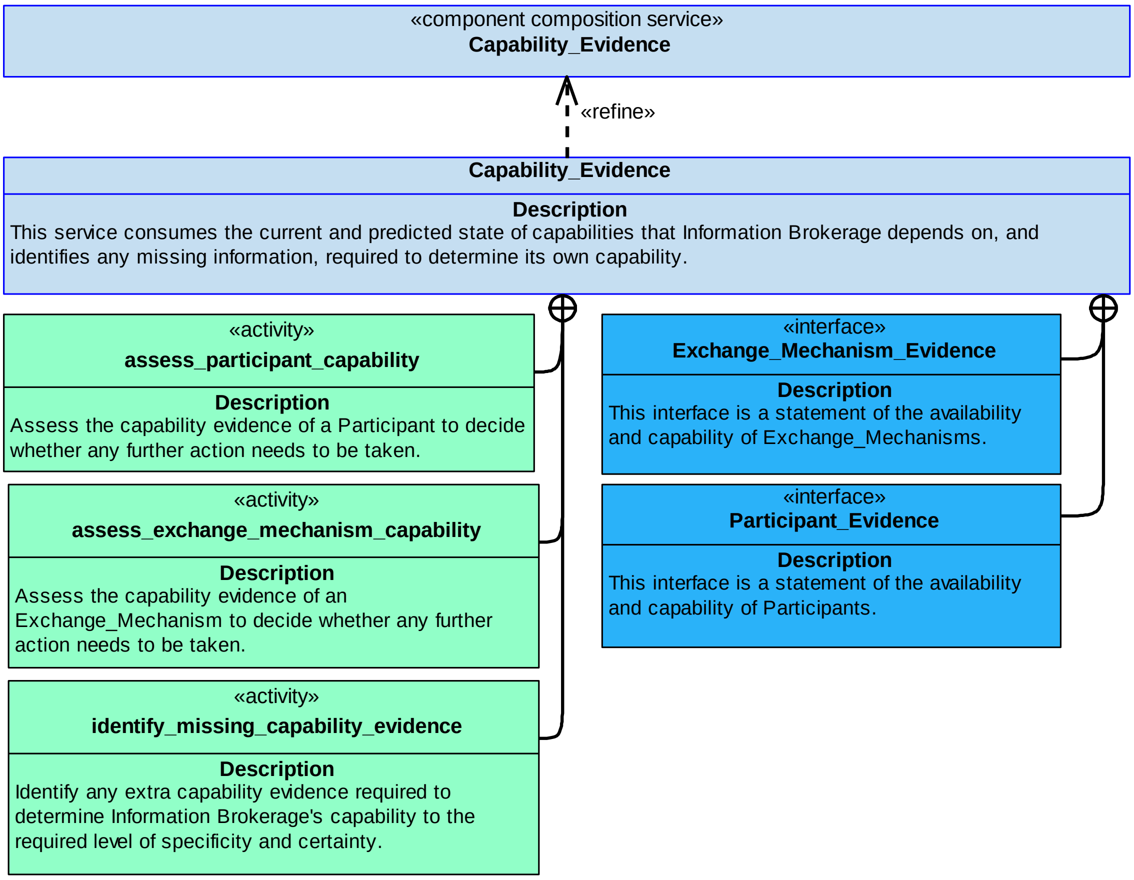

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted state of capabilities that Information Brokerage depends on, and identifies any missing information, required to determine its own capability.
«activity»
assess_participant_capability
Description
Assess the capability evidence of a Participant to decide whether any further action needs to be taken.
«activity»
assess_exchange_mechanism_capability
Description
Assess the capability evidence of an Exchange_Mechanism to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine Information Brokerage's capability to the required level of specificity and certainty.
«interface»
Exchange_Mechanism_Evidence
Description
This interface is a statement of the availability and capability of Exchange_Mechanisms.
«interface»
Participant_Evidence
Description
This interface is a statement of the availability and capability of Participants.

Figure 485: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted state of capabilities that Information Brokerage depends on, and identifies any missing information, required to determine its own capability.

Interfaces

Exchange_Mechanism_Evidence

This interface is a statement of the availability and capability of Exchange_Mechanisms.

Attribute

exchange_mechanism_capability	The capability which an Exchange_Mechanism can provide, in terms of types of Exchange that can be supported.

Participant_Evidence

This interface is a statement of the availability and capability of Participants.

Attribute

participant_capability	The capability which a Participant can provide, in terms of types of Distributable_Item and Information_Configurations that can be supported.

Activities

assess_participant_capability

Assess the capability evidence of a Participant to decide whether any further action needs to be taken.

assess_exchange_mechanism_capability

Assess the capability evidence of an Exchange_Mechanism to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine Information Brokerage's capability to the required level of specificity and certainty.

5.4.2.27.7.2 Service Dependencies

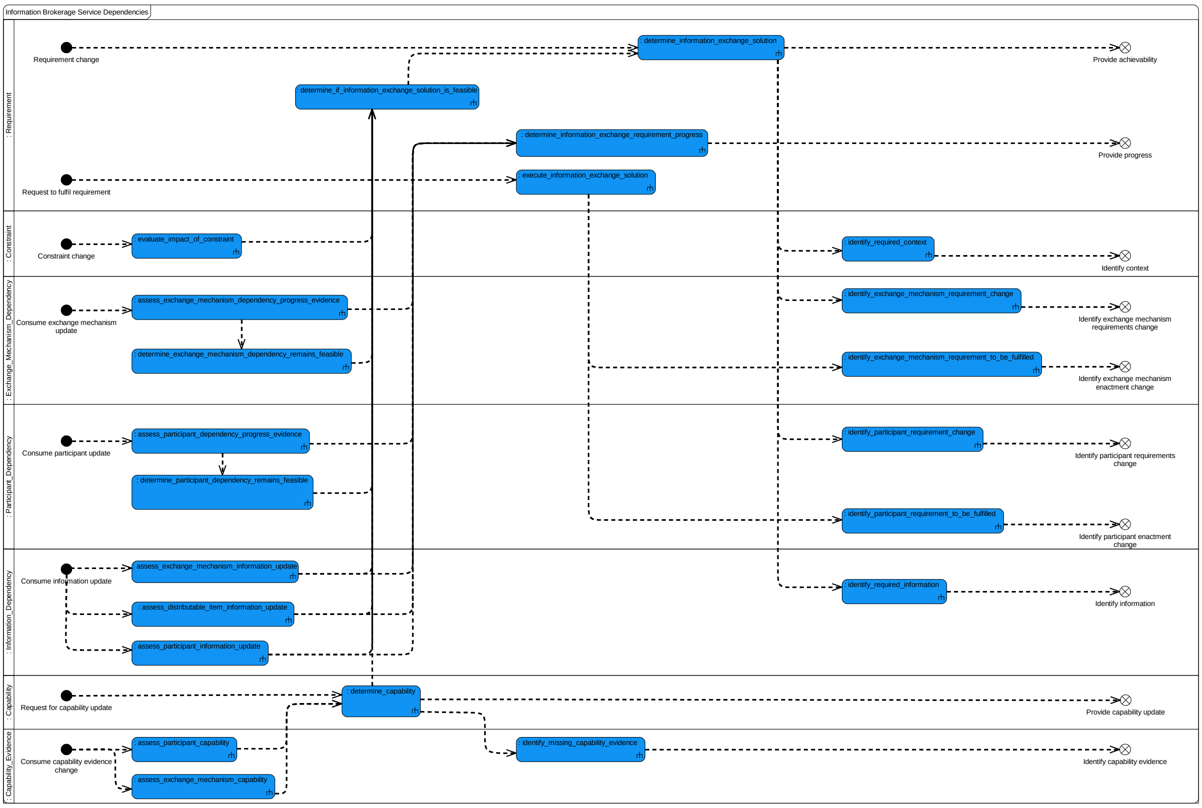

Information Brokerage Service Dependencies
: Requirement
Requirement change
: determine_information_exchange_solution
Provide achievability
: determine_if_information_exchange_solution_is_feasible
: determine_information_exchange_requirement_progress
Provide progress
Request to fulfil requirement
: execute_information_exchange_solution
: Constraint
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
: Exchange_Mechanism_Dependency
Consume exchange mechanism update
: assess_exchange_mechanism_dependency_progress_evidence
: determine_exchange_mechanism_dependency_remains_feasible
: identify_exchange_mechanism_requirement_change
Identify exchange mechanism requirements change
: identify_exchange_mechanism_requirement_to_be_fulfilled
Identify exchange mechanism enactment change
: Participant_Dependency
Consume participant update
: assess_participant_dependency_progress_evidence
: determine_participant_dependency_remains_feasible
: identify_participant_requirement_change
Identify participant requirements change
: identify_participant_requirement_to_be_fulfilled
Identify participant enactment change
: Information_Dependency
Consume information update
: assess_exchange_mechanism_information_update
: assess_distributable_item_information_update
: assess_participant_information_update
: identify_required_information
Identify information
: Capability
Request for capability update
: determine_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_participant_capability
: assess_exchange_mechanism_capability
: identify_missing_capability_evidence
Identify capability evidence

Figure 486: Information Brokerage Service Dependencies

5.4.2.28 Information Presentation

5.4.2.28.1 Role

The role of Information Presentation is to represent the conveyance and perception of information between the system and an HMI user.

5.4.2.28.2 Overview

Control Architecture

Information Presentation is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Following the receipt of a Requirement to provide an HMI user with information, Information Presentation determines the most appropriate manner in which to present the Presentation_Interaction by delivering a cohesive composition of Presentation_Elements appropriate for the Presentation_Resources available. Any such presentation shall be in accordance with the applicable policies and Context.

Information Presentation also captures user input (e.g. button presses, selections or pointer movements) that may be required to be passed to the system.

The presentation is updated as appropriate with new information from the system or the user as it is received (including any HMI-related feedback).

Examples of Use

A deployment will use Information Presentation when it needs to:

- Convey information, input by an HMI user, to the system, e.g. interpreting keyboard input, pointer movements, eye tracking or voice commands, as a selection or instruction for the system.
- Convey information, provided by the system, to the user in a meaningful way, e.g. on a display screen, or as haptics, aural alerts and messages.
- Provide indications as to the timeliness or freshness of information where appropriate, e.g. to occult flight reference data that is considered stale.

5.4.2.28.3 Service Summary

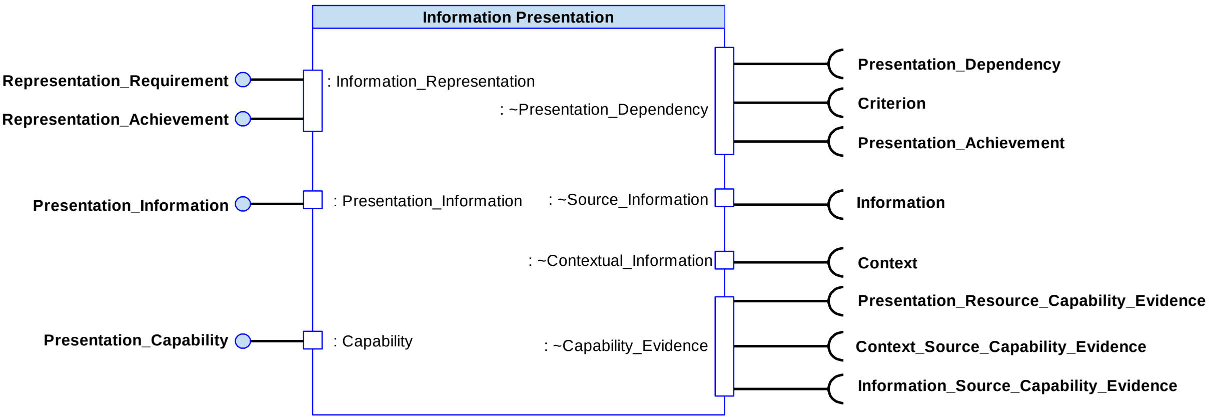

Information Presentation
Representation_Requirement
Representation_Achievement
Presentation_Information
Presentation_Capability
: Information_Representation
: ~Presentation_Dependency
: Presentation_Information
: ~Source_Information
: ~Contextual_Information
: Capability
: ~Capability_Evidence
Presentation_Dependency
Criterion
Presentation_Achievement
Information
Context
Presentation_Resource_Capability_Evidence
Context_Source_Capability_Evidence
Information_Source_Capability_Evidence

Figure 487: Information Presentation Service Summary

5.4.2.28.4 Responsibilities

capture_conveyance_requirement

- To capture Requirements for Presentation_Interactions. This can be for information coming into the system or from the system.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is achievable given current or predicted Presentation_Capability and any Context.

determine_presentation_solution

- To determine the method of Presentation_Interaction that meets the Requirement.

identify_pre-conditions

- To identify Pre-conditions required to permit the conveyance of information.

fulfil_requirement

- To convey the information coming into the system or from the system.

capture_external_factors

- To capture Contexts that will influence the chosen mode of Presentation_Interaction.

capture_perception_efficacy

- To capture the Perception_Efficacy for a mode of Presentation_Interaction.

assess_information_presentation_capability

- To assess the Presentation_Capability taking account the health of resources and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Presentation_Capability assessment.

predict_capability_progression

- To predict the progression of Presentation_Capability over time and with use.

5.4.2.28.5 Subject Matter Semantics

The subject matter of Information Presentation is system information that is conveyed to an HMI user (how it is received, seen, heard, etc.) and user instructions, including selections, that are provided to the system.

Exclusions

The subject matter of Information Presentation does not include:

- The interpretation of the data by the user or the system.
- The quality of the information that is presented.
- The classification and security clearance.

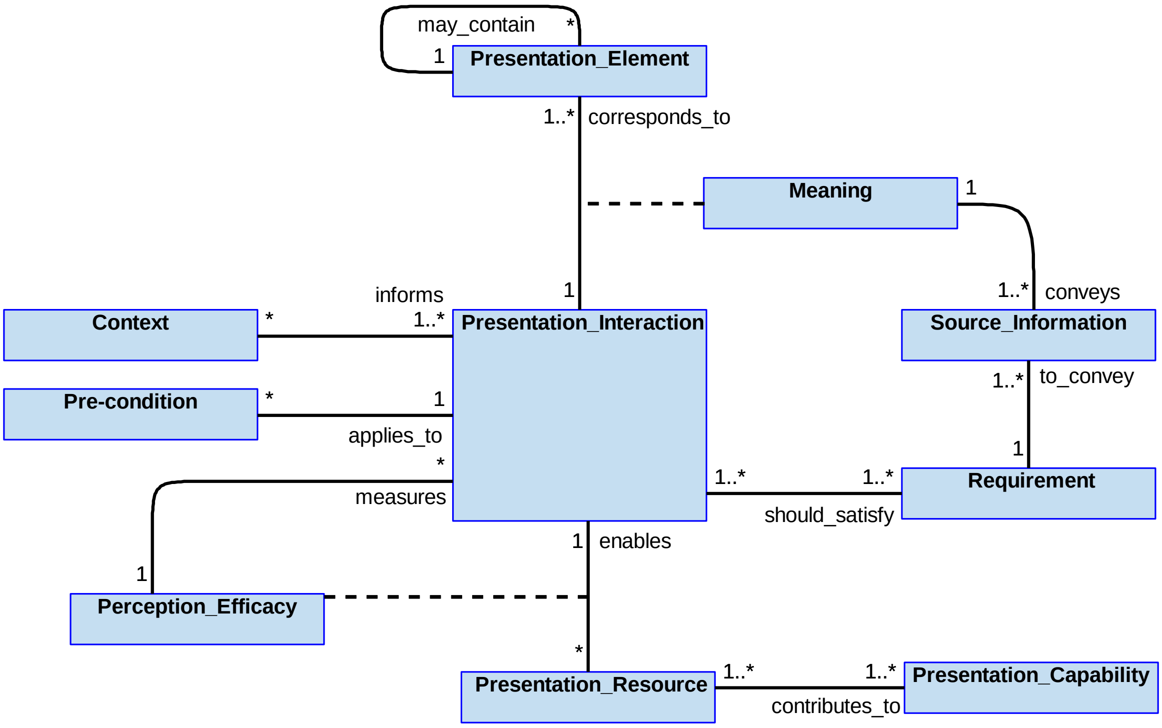


Figure 488: Information Presentation Semantics

5.4.2.28.5.1 Entities

Context

Information required in order to determine how information should be conveyed, e.g. the role of the person receiving the information, the phase of flight, lighting conditions or noise levels in a cockpit.

Meaning

The nature of a Presentation_Interaction with a Presentation_Element that provides the meaning given to the information conveyed, e.g. moving a pointer over an icon may result in the icon changing colour (which remains within this component), but selecting it may initiate a dialogue to send an input to the system.

Perception_Efficacy

The effectiveness of the solution in communicating between system and HMI user, e.g. the speed of drawing attention to information against the risk of overloading the user's senses.

Pre-condition

A condition upon which the conveyance of information is dependant, e.g. that a certain display format must be selected to present the information.

Presentation_Capability

The ability to provide or accept information.

Presentation_Element

An item of content that makes up part of the presented information or a group of items structured or layered to make a more meaningful whole, e.g. an individual icon from a graphics library, a sound file, the enumeration of a single word string from a voice command, or a Morse code message.

Presentation_Interaction

An exchange carried out to convey the required information, or part thereof. This can be to present system information to the HMI user (visually, aurally, etc.) or to receive information from the user (through button presses, voice commands, etc.) to pass to the system.

Presentation_Resource

Something that can be used to interact with one (or more) of the senses or modes of input, e.g. a touch-screen display surface, keyboard or force-feedback device.

Requirement

A requirement to convey information, e.g. to capture an HMI user input or to provide a user with information.

Source_Information

The information being conveyed through the use of HMI, e.g. that the fuel levels are low, or the latitude/longitude of a waypoint.

5.4.2.28.6 Design Rationale

5.4.2.28.6.1 Assumptions

- Where user authentication processes are in place prior to authentication by the system it is assumed the available presentation will only cover relevant aspects of the authentication process, e.g. by providing the login screen.
- The component may perform some level of checking of data input, e.g. the data is in the required form (e.g. alpha or numerical) and within expected limits. If so, it would not necessarily be expected to consider the integrity, quality or impact of the input data.

5.4.2.28.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Information Presentation:

- Human-Machine Interface - The HMI PYRAMID concept is fundamental to this component.
- Data Driving - This component is likely to be data-driven. Such data driving could typically include composition rules, policy content and symbol libraries, etc. which change infrequently, or personalisation files that may change on a regular basis, e.g. user preferences.
- Interaction with Equipment - depending on complexity of interface, Information Presentation may interact directly with HMI devices (keyboards, display screens, etc.) or they may be considered as a form of sensor or effector should there be a need for a Resource Layer-type interaction.
- Recording and Logging - logging of HMI Presentation_Interactions and presentations as required by the Exploiting Programme.

Extensions

- It is possible that extension components will be utilised for different presentation modalities (visual, audio and tactile).

Exploitation Considerations

- This component should be adaptable to emerging technologies (e.g. augmented and virtual reality) as well as for traditional presentation technologies.
- Graphical Presentation_Elements will be drawn/rendered - their visual representation should consider layout aspects such as their position, size, transformation (scale, rotate, translate), clipping and dynamic layout behaviour (e.g. animation, size to fit), and type-specific properties (e.g. enabled/disabled, label text, colour, visibility, interaction state).
- Audio Presentation_Elements may be generated at a position relative to the listener (for stereo or 3D audio).
- A Presentation_Element may respond to user Presentation_Interaction by updating its state (e.g. focusing, scrolling or selecting) and generating events to signal the nature of the interaction (e.g. cursor over or selection). Some user inputs may be ignored depending on the Presentation_Element's type and state (e.g. disabled).
- A presentation, e.g. a display format, could be generated by a single instance of Information Presentation or by a number of instances or variants, depending on the requirements of the Exploiting Programme.
- A composition of Presentation_Elements may need to be kept updated, even when not required (e.g. format not selected) to ensure it is available in a timely manner when requested.
- If this component allows low and high integrity (safety or security) information to be presented together, then it needs to be of high integrity itself.

- Due to the Context and Meaning it is possible that the same Source_Information is presented in different ways to different users. For example, a single set of route information could be provided to both a pilot and a navigator and be different for both users, with the presentation to the pilot being presented in one scale setting with the next leg highlighted in order for it to be flown, whereas the navigator has a different scale setting and a future leg selected with additional information available to support the planning of a contingency update to that leg of the route. Personalisation/customisation settings may further change how information is presented.
- The utilisation of standard interfaces such as ARINC 661 Ref. [14], OpenGL (e.g. embedded and safety critical profiles), and Vulkan should be considered; this would allow the component implementation to be generated using commercial tooling and provide compatibility with graphics drivers.

5.4.2.28.6.3 Safety Considerations

The indicative IDAL is DAL A*.*

The rationale behind this is:

- This component provides the interface between a user and the system - both for control inputs and display. Failure of this component could, as a worst case, cause catastrophic consequences, e.g. erroneous inputs to the flight controls system or inadvertent weapon release commands.
- It is expected that multiple implementations and instances of this component may be created in a PYRAMID deployment. Analysis of the specific uses of each instance, by the Exploiting Programme, may justify a less onerous DAL for some instances.

5.4.2.28.6.4 Security Considerations

The indicative security classification is notionally O.

This component is part of the interface between the operator and the system and as such the indicative security classification is dependent on the classification of information being processed. Where different classifications of data may be handled by the same instance, it will be at the highest of those classifications. It should be treated as per the confidentiality, integrity and availability needs of the information being presented.

It is expected that this component will be required within security domains that interface to a user. Where there are multiple security domains and multiple instances of the component, these may need to communicate with each other. Permitted types of information and participants will be managed by a boundary protection function located outside the component.

The component is expected to at least partially satisfy security related functions through:

- Presenting information based upon the needs of the **Classification of Data** involved and of the other Contexts in which it is shared. This component may also support the introduction of classification information by the operator - the operator will be responsible for ensuring this is correct.

- **Identifying Data Sources** for information to be shared between the system and operators.
- **Maintaining Audit Records** to support non-repudiation of events based on the information presented and when it was presented.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **Warnings and Notifications** are conveyed through this component, thus supporting awareness of unexpected activity that may be a result of cyber attack.

The component is expected to perform some aspects of security enforcing functions contributing to:

- **Restricting Access to Data** through consideration of the appropriate Context, this includes only providing information suitable for the clearance or role of the operator, etc.
- **User Login and Authentication** processes; whilst this component does not perform authentication, it is part of the user interface by which authentication is provided, role allocations and handover are performed, etc.

5.4.2.28.7 Services

5.4.2.28.7.1 Service Definitions

5.4.2.28.7.1.1 Information_Representation

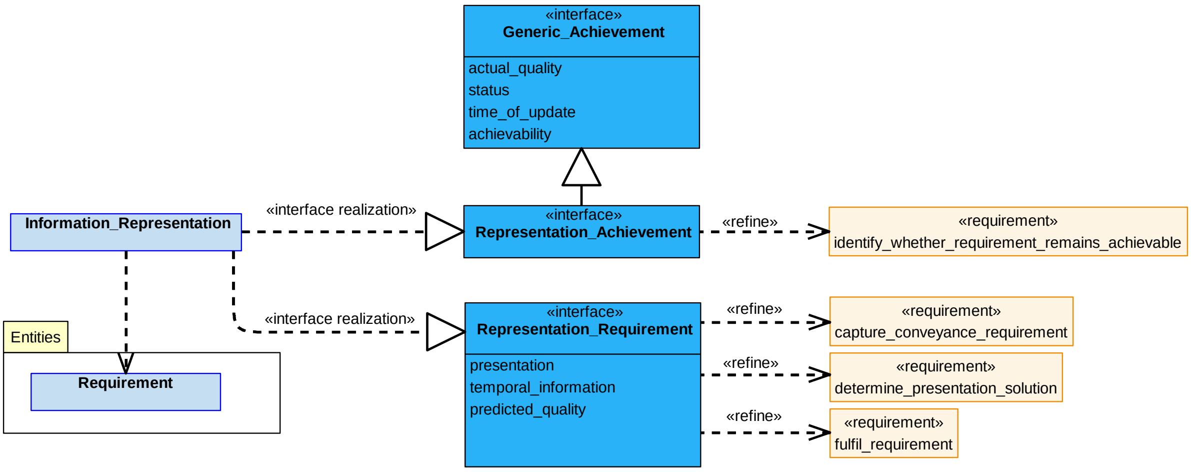


Figure 489: Information_Representation Service Definition

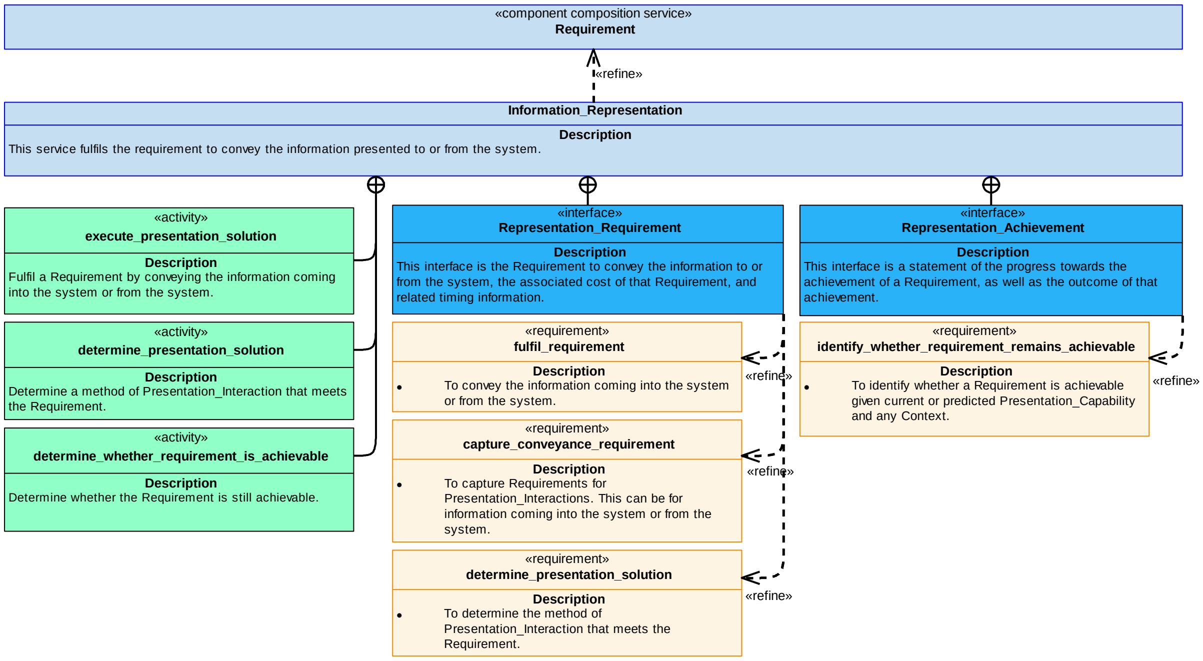

«component composition service»
Requirement
«refine»
Information_Representation
Description
This service fulfils the requirement to convey the information presented to or from the system.
«activity»
execute_presentation_solution
Description
Fulfil a Requirement by conveying the information coming into the system or from the system.
«activity»
determine_presentation_solution
Description
Determine a method of Presentation_Interaction that meets the Requirement.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether the Requirement is still achievable.
«interface»
Representation_Requirement
Description
This interface is the Requirement to convey the information to or from the system, the associated cost of that Requirement, and related timing information.
«requirement»
fulfil_requirement
Description
To convey the information coming into the system or from the system.
«refine»
«requirement»
capture_conveyance_requirement
Description
To capture Requirements for Presentation_Interactions. This can be for information coming into the system or from the system.
«refine»
«requirement»
determine_presentation_solution
Description
To determine the method of Presentation_Interaction that meets the Requirement.
«refine»
«interface»
Representation_Achievement
Description
This interface is a statement of the progress towards the achievement of a Requirement, as well as the outcome of that achievement.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Requirement is achievable given current or predicted Presentation_Capability and any Context.
«refine»

Figure 490: Information_Representation Service Policy

Information_Representation

This service fulfils the requirement to convey the information presented to or from the system.

Interfaces

Representation_Requirement

This interface is the Requirement to convey the information to or from the system, the associated cost of that Requirement, and related timing information.

Attributes

presentation	The Requirement to convey the information, e.g. the need for an image on a screen, a sound alert, or the intent of a button press.
temporal_information	Information covering timing, such as start and end times, e.g. the length of time an image appears on a screen, duration of alarm, response time.
predicted_quality	How well the planned presentation solution is predicted to satisfy the Requirement.

Representation_Achievement

This interface is a statement of the progress towards the achievement of a Requirement, as well as the outcome of that achievement.

Activities

determine_presentation_solution

Determine a method of Presentation_Interaction that meets the Requirement.

execute_presentation_solution

Fulfil a Requirement by conveying the information coming into the system or from the system.

determine_whether_requirement_is_achievable

Determine whether the Requirement is still achievable.

5.4.2.28.7.1.2 Presentation_Dependency

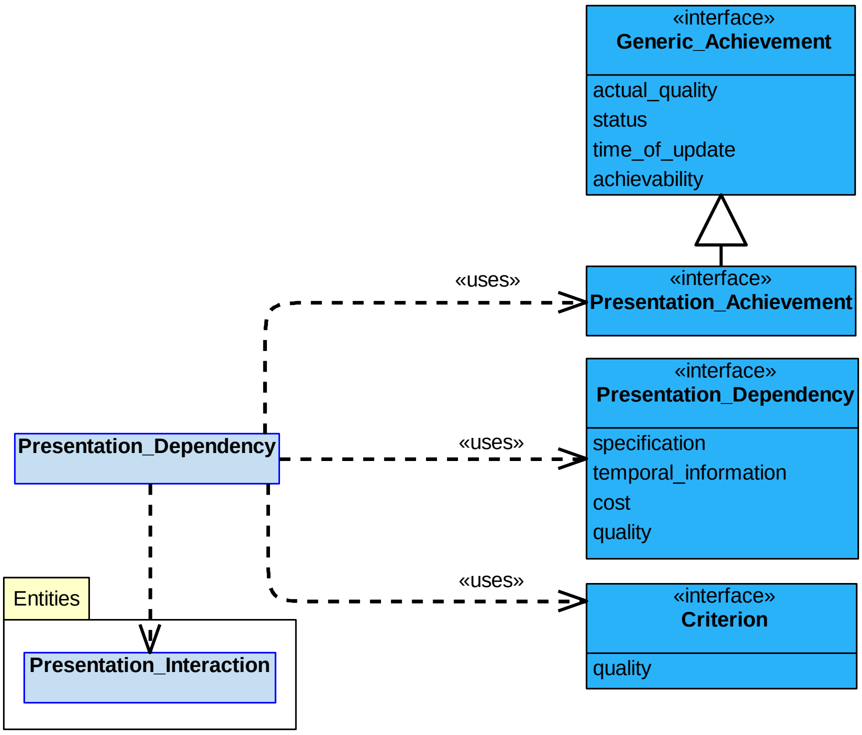


Figure 491: Presentation_Dependency Service Definition

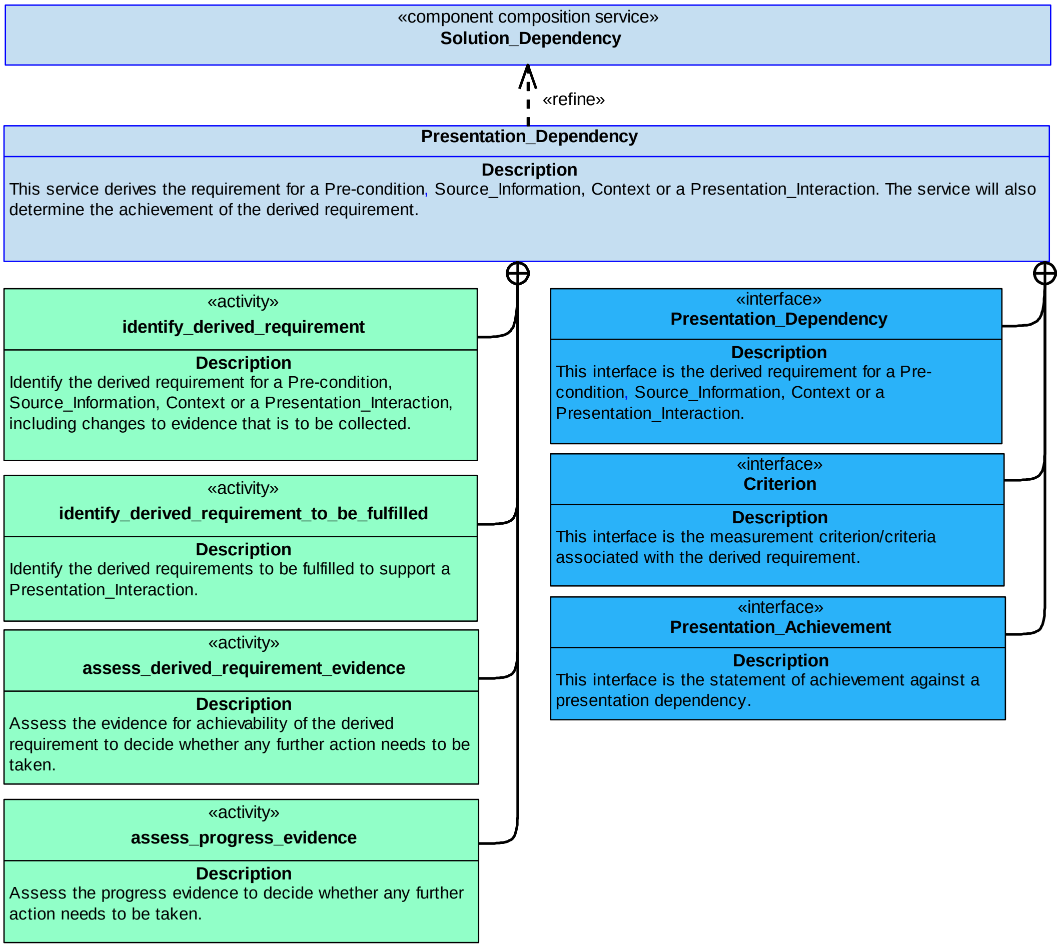

«component composition service»
Solution_Dependency
«refine»
Presentation_Dependency
Description
This service derives the requirement for a Pre-condition, Source_Information, Context or a Presentation_Interaction. The service will also determine the achievement of the derived requirement.
«activity»
identify_derived_requirement
Description
Identify the derived requirement for a Pre-condition, Source_Information, Context or a Presentation_Interaction, including changes to evidence that is to be collected.
«activity»
identify_derived_requirement_to_be_fulfilled
Description
Identify the derived requirements to be fulfilled to support a Presentation_Interaction.
«activity»
assess_derived_requirement_evidence
Description
Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«interface»
Presentation_Dependency
Description
This interface is the derived requirement for a Pre-condition, Source_Information, Context or a Presentation_Interaction.
«interface»
Criterion
Description
This interface is the measurement criterion/criteria associated with the derived requirement.
«interface»
Presentation_Achievement
Description
This interface is the statement of achievement against a presentation dependency.

Figure 492: Presentation_Dependency Service Policy

Presentation_Dependency

This service derives the requirement for a Pre-condition, Source_Information, Context or a Presentation_Interaction. The service will also determine the achievement of the derived requirement.

Interfaces

Presentation_Dependency

This interface is the derived requirement for a Pre-condition, Source_Information, Context or a Presentation_Interaction.

Attributes

specification	The definition of the derived requirement.
temporal_information	Information covering timing, such as start and end timing.
cost	The cost of executing the solution, for example: resources used or time taken.
quality	How well the solution satisfies the requirement.

Presentation_Achievement

This interface is the statement of achievement against a presentation dependency.

Criterion

This interface is the measurement criterion/criteria associated with the derived requirement.

Attribute

quality	The required quality of the data, e.g. timeliness, precision and trustworthiness.

Activities

identify_derived_requirement

Identify the derived requirement for a Pre-condition, Source_Information, Context or a Presentation_Interaction, including changes to evidence that is to be collected.

identify_derived_requirement_to_be_fulfilled

Identify the derived requirements to be fulfilled to support a Presentation_Interaction.

assess_derived_requirement_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

5.4.2.28.7.1.3 Presentation_Information

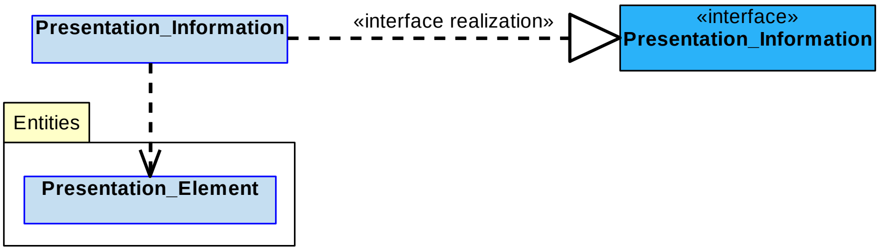


Figure 493: Presentation_Information Service Definition

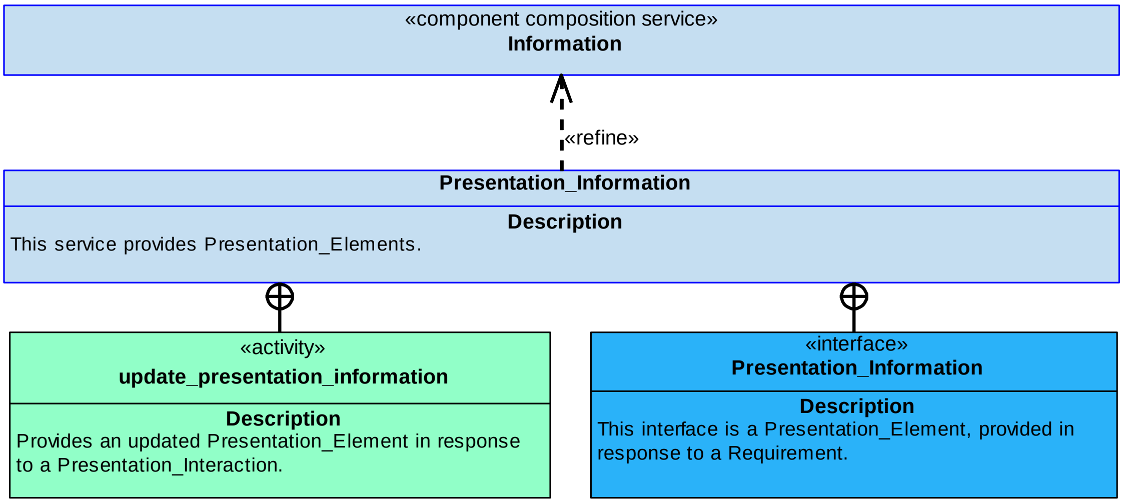

«component composition service»
Information
«refine»
Presentation_Information
Description
This service provides Presentation_Elements.
«activity»
update_presentation_information
Description
Provides an updated Presentation_Element in response to a Presentation_Interaction.
«interface»
Presentation_Information
Description
This interface is a Presentation_Element, provided in response to a Requirement.

Figure 494: Presentation_Information Service Policy

Presentation_Information

This service provides Presentation_Elements.

<u>Interface</u>

Presentation_Information

This interface is a Presentation_Element, provided in response to a Requirement.

<u>Activity</u>

update_presentation_information

Provides an updated Presentation_Element in response to a Presentation_Interaction.

5.4.2.28.7.1.4 Source_Information

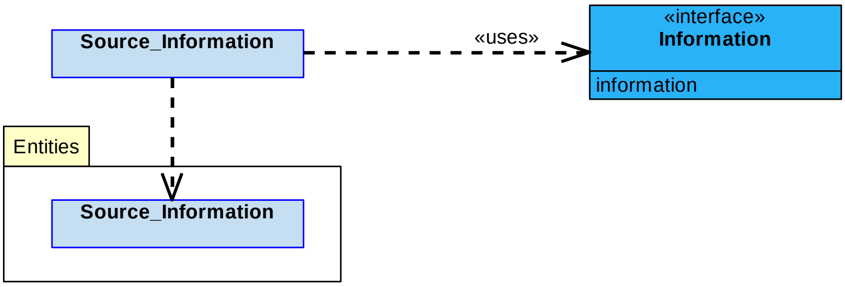


Figure 495: Source_Information Service Definition

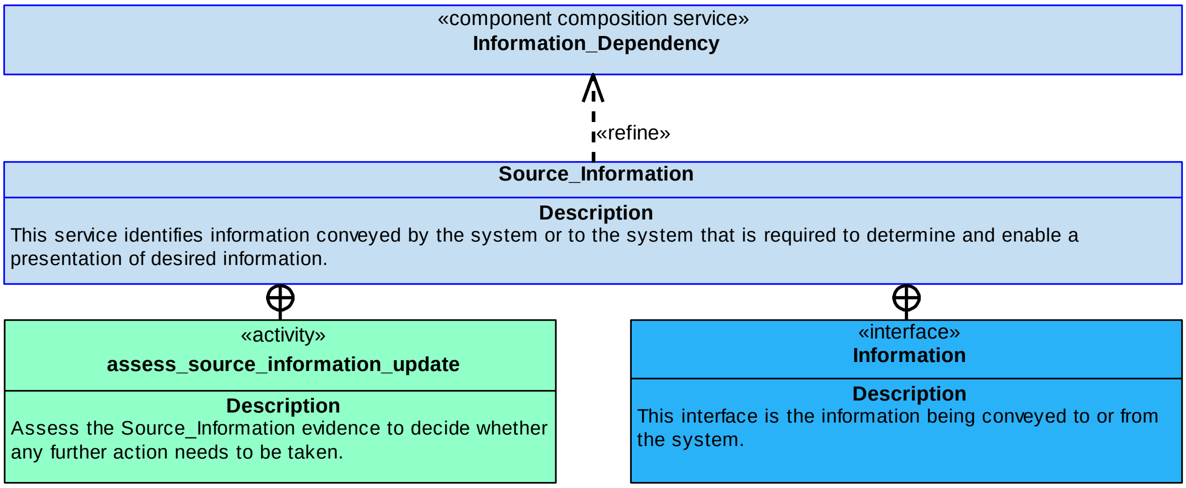

«component composition service»
Information_Dependency
«refine»
Source_Information
Description
This service identifies information conveyed by the system or to the system that is required to determine and enable a presentation of desired information.
«activity»
assess_source_information_update
Description
Assess the Source_Information evidence to decide whether any further action needs to be taken.
«interface»
Information
Description
This interface is the information being conveyed to or from the system.

Figure 496: Source_Information Service Policy

Source_Information

This service identifies information conveyed by the system or to the system that is required to determine and enable a presentation of desired information.

Interface

Information

This interface is the information being conveyed to or from the system.

Attribute

information	Information received to be conveyed to or from the system, e.g. a warning, a button being pressed.

Activity

assess_source_information_update

Assess the Source_Information evidence to decide whether any further action needs to be taken.

5.4.2.28.7.1.5 Contextual_Information

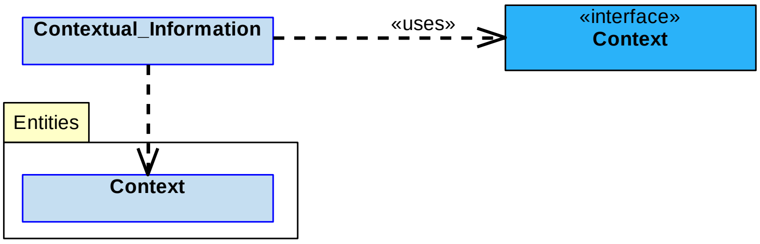


Figure 497: Contextual_Information Service Definition

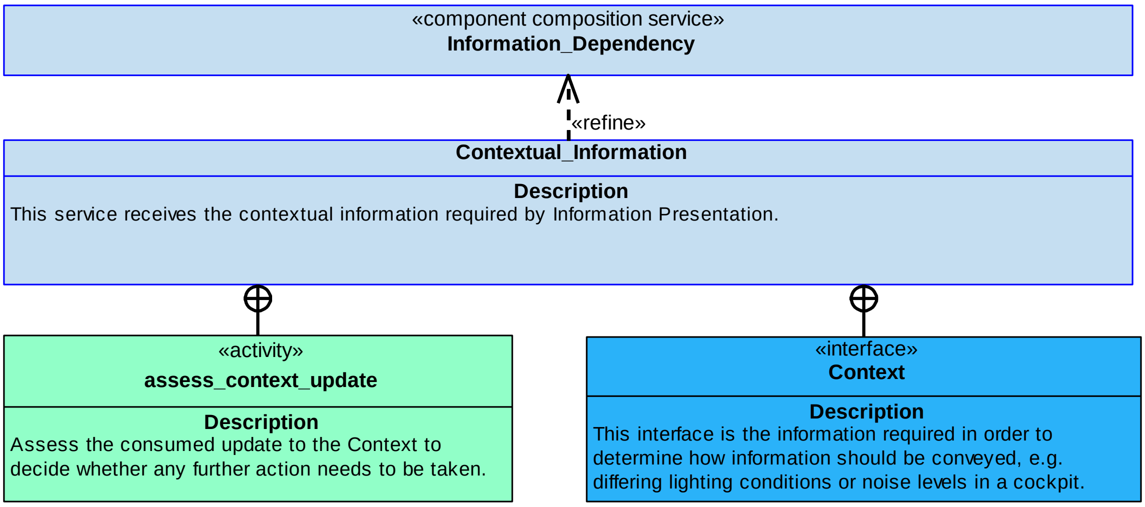

«component composition service»
Information_Dependency
«refine»
Contextual_Information
Description
This service receives the contextual information required by Information Presentation.
«activity»
assess_context_update
Description
Assess the consumed update to the Context to decide whether any further action needs to be taken.
«interface»
Context
Description
This interface is the information required in order to determine how information should be conveyed, e.g. differing lighting conditions or noise levels in a cockpit.

Figure 498: Contextual_Information Service Policy

Contextual_Information

This service receives the contextual information required by Information Presentation.

Interface

Context

This interface is the information required in order to determine how information should be conveyed, e.g. differing lighting conditions or noise levels in a cockpit.

Activity

assess_context_update

Assess the consumed update to the Context to decide whether any further action needs to be taken.

5.4.2.28.7.1.6 Capability

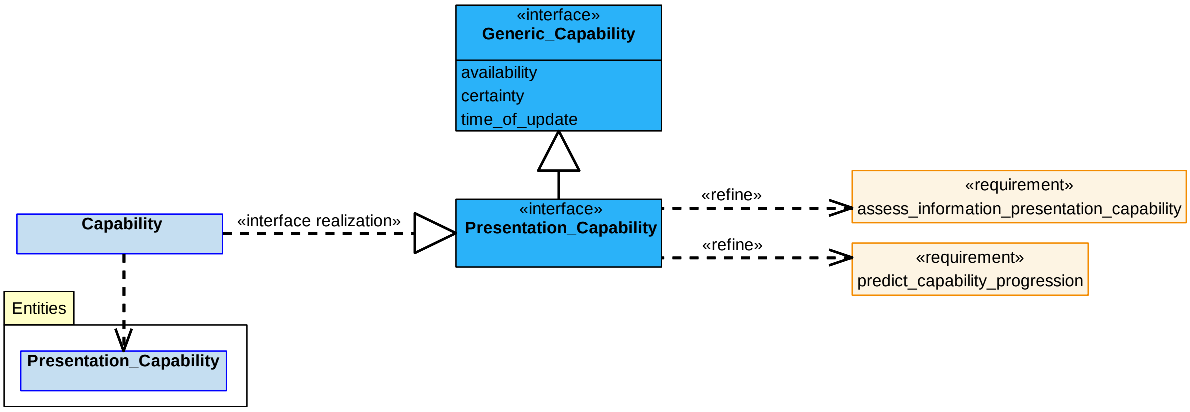


Figure 499: Capability Service Definition

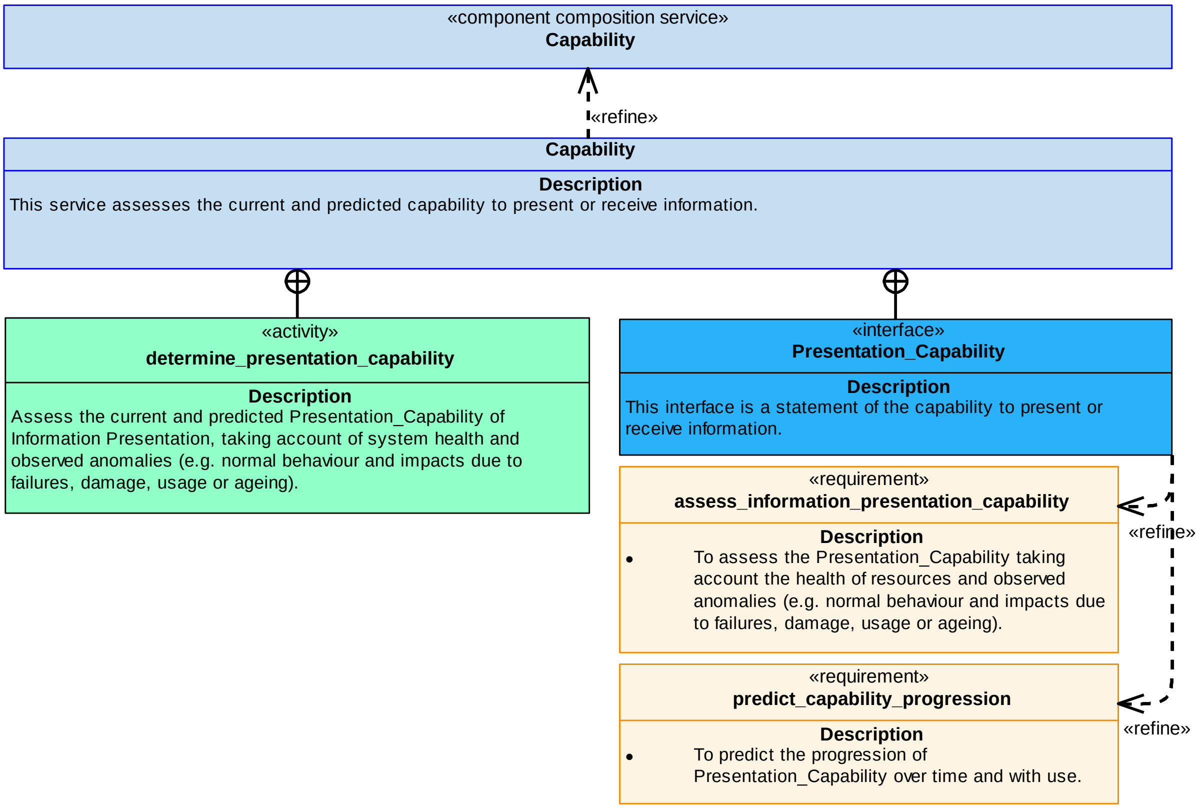

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability to present or receive information.
«activity»
determine_presentation_capability
Description
Assess the current and predicted Presentation_Capability of Information Presentation, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Presentation_Capability
Description
This interface is a statement of the capability to present or receive information.
«requirement»
assess_information_presentation_capability
Description
To assess the Presentation_Capability taking account the health of resources and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of Presentation_Capability over time and with use.
«refine»

Figure 500: Capability Service Policy

Capability

This service assesses the current and predicted capability to present or receive information.

Interface

Presentation_Capability

This interface is a statement of the capability to present or receive information.

Activity

determine_presentation_capability

Assess the current and predicted Presentation_Capability of Information Presentation, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.28.7.1.7 Capability_Evidence

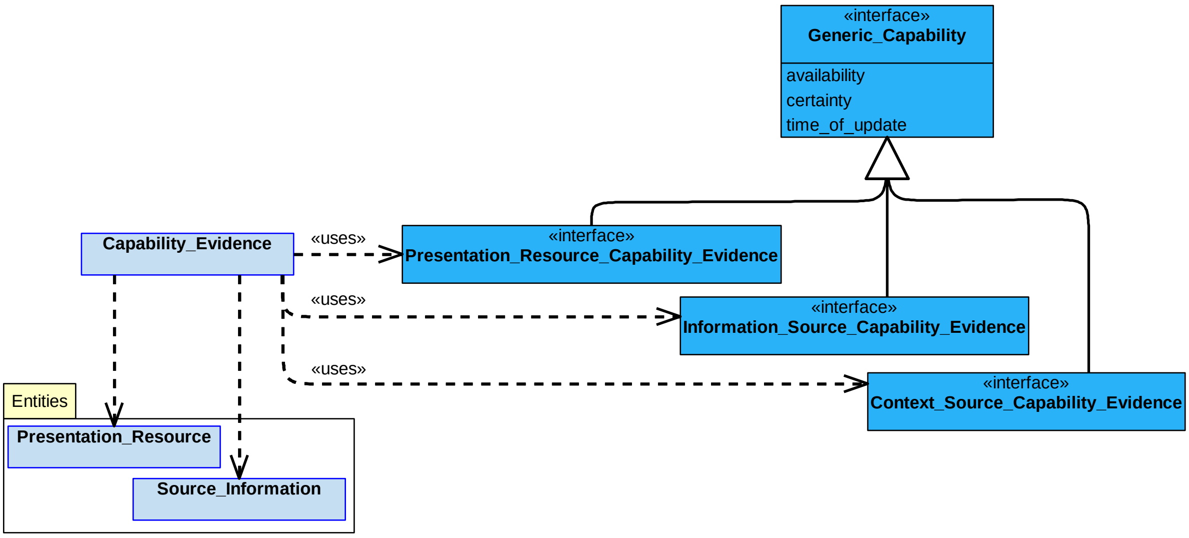
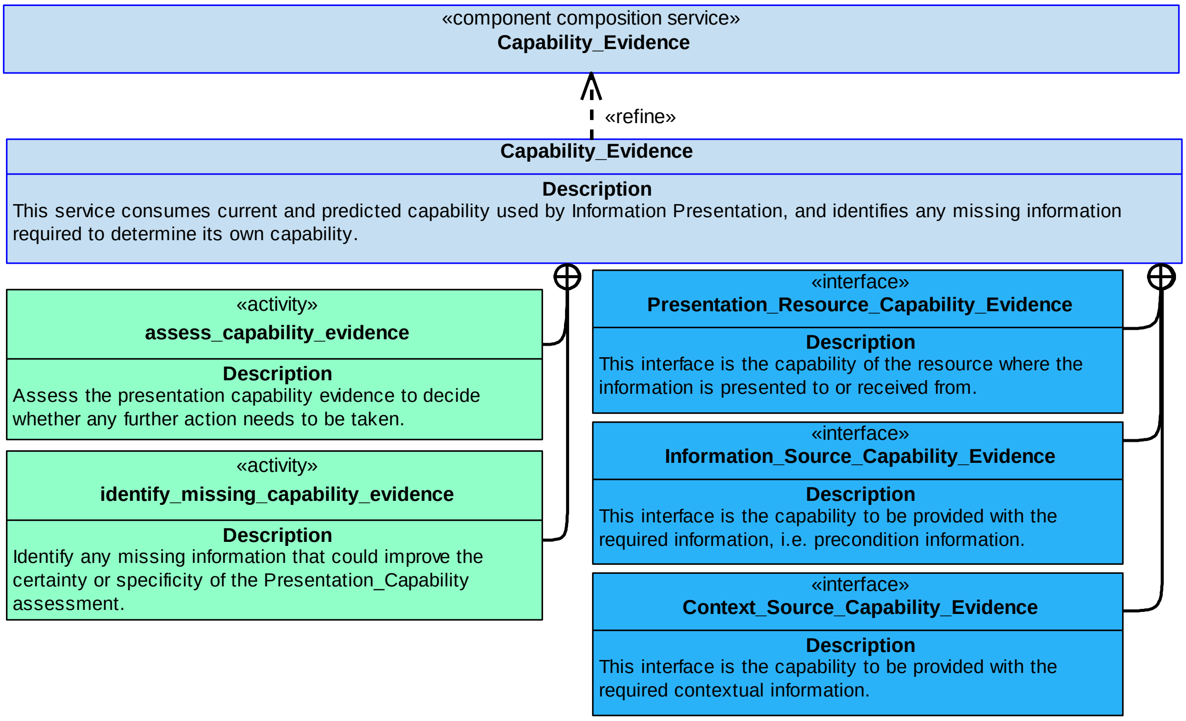


Figure 501: Capability_Evidence Service Definition



Description
This service consumes current and predicted capability used by Information Presentation, and identifies any missing information required to determine its own capability.

«activity»
assess_capability_evidence

Description
Assess the presentation capability evidence to decide whether any further action needs to be taken.

«activity»
identify_missing_capability_evidence

Description
Identify any missing information that could improve the certainty or specificity of the Presentation_Capability assessment.

«interface»
Presentation_Resource_Capability_Evidence

Description
This interface is the capability of the resource where the information is presented to or received from.

«interface»
Information_Source_Capability_Evidence

Description
This interface is the capability to be provided with the required information, i.e. precondition information.

«interface»
Context_Source_Capability_Evidence

Description
This interface is the capability to be provided with the required contextual information.

Figure 502: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability used by Information Presentation, and identifies any missing information required to determine its own capability.

Interfaces

Presentation_Resource_Capability_Evidence

This interface is the capability of the resource where the information is presented to or received from.

Information_Source_Capability_Evidence

This interface is the capability to be provided with the required information, i.e. precondition information.

Context_Source_Capability_Evidence

This interface is the capability to be provided with the required contextual information.

Activities

assess_capability_evidence

Assess the presentation capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any missing information that could improve the certainty or specificity of the Presentation_Capability assessment.

5.4.2.28.7.2 Service Dependencies

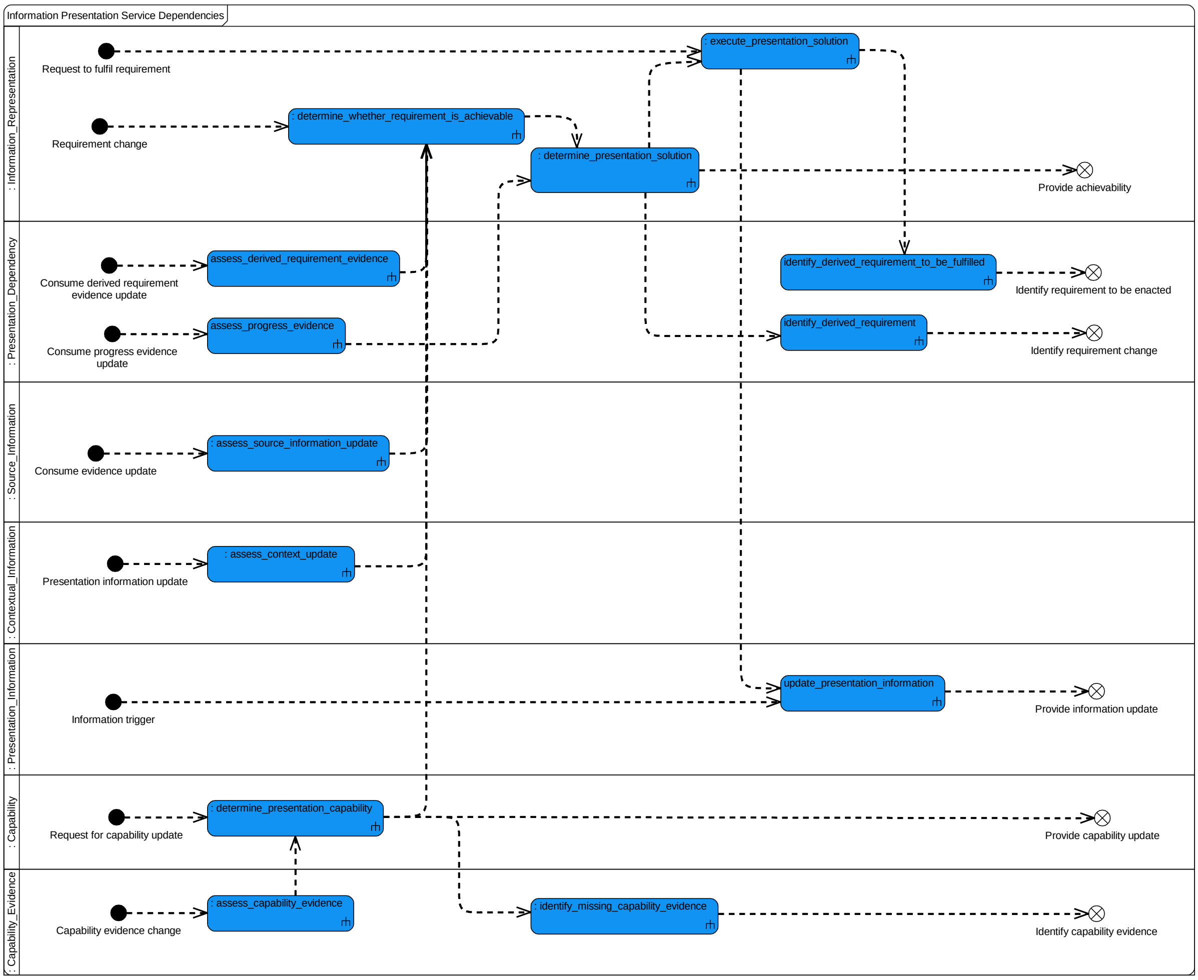

Information Presentation Service Dependencies
: Information_Representation
: Presentation_Dependency
: Source_Information
: Contextual_Information
: Presentation_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_presentation_solution
Requirement change
: determine_whether_requirement_is_achievable
: determine_presentation_solution
Provide achievability
Consume derived requirement evidence update
assess_derived_requirement_evidence
identify_derived_requirement_to_be_fulfilled
Identify requirement to be enacted
Consume progress evidence update
assess_progress_evidence
identify_derived_requirement
Identify requirement change
Consume evidence update
: assess_source_information_update
Presentation information update
: assess_context_update
Information trigger
update_presentation_information
Provide information update
Request for capability update
determine_presentation_capability
Provide capability update
Capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 503: Information Presentation Service Dependencies

5.4.2.29 Interlocks

5.4.2.29.1 Role

The role of Interlocks is to enable and disable functions by checking when defined condition values have been met.

5.4.2.29.2 Overview

Control Architecture

Interlocks is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Interlocks determines whether a Conditional_Check (i.e. a defined set of conditions) is satisfied and enables or disables Function_Interlocks accordingly.

Examples of Use

Interlocks will be required to:

- Enable the arming of a warhead 10 seconds after separation from the launch platform, provided that the launch platform has enabled arming.
- Turn the power on to store release circuits when the Master Armaments Safety Switch (MASS) is live, weight is off wheels, the weapon bay doors are open, store release is authorised, the vehicle is within the appropriate release envelope and store release power has been requested.

5.4.2.29.3 Service Summary

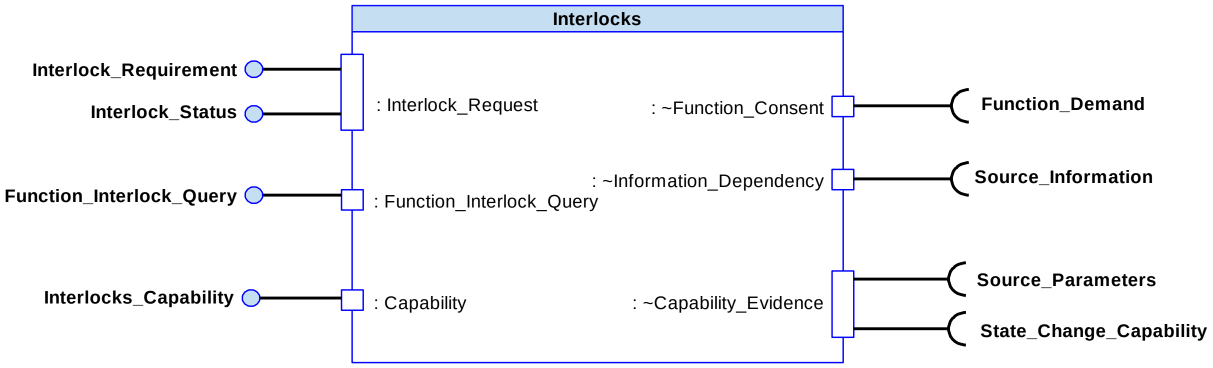


Figure 504: Interlocks Service Summary

5.4.2.29.4 Responsibilities

capture_function_requests

- To capture Requests for enabling or disabling a given Function_Interlock.

determine_interlock

- To determine whether the Conditional_Check for a Function_Interlock is satisfied.

determine_if_conditional_check_remains_feasible

- To determine if a planned or on-going Conditional_Check remains feasible.

identify_current_states

- To maintain a view of the current states of Function_Interlocks.

control_interlock

- To control Function_Interlocks to enable or disable functions.

identify_interlock_request_progress

- To identify the progress of an implementation of a Conditional_Check in response to a Request to enable or disable the operation of a function.

capture_conditional_check

- To capture the Conditional_Checks (required conditions) that define when Function_Interlocks can be enabled or disabled.

capture_source_parameters

- To capture currently provided Source_Parameters used to determine whether Function_Interlocks should be enabled or disabled.

assess_interlocks_capability

- To assess the Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.29.5 Subject Matter Semantics

The subject matter of Interlocks is Conditional_Checks and whether they have been satisfied.

Exclusions

The subject matter of Interlocks does not include:

- The context of a particular Conditional_Check.

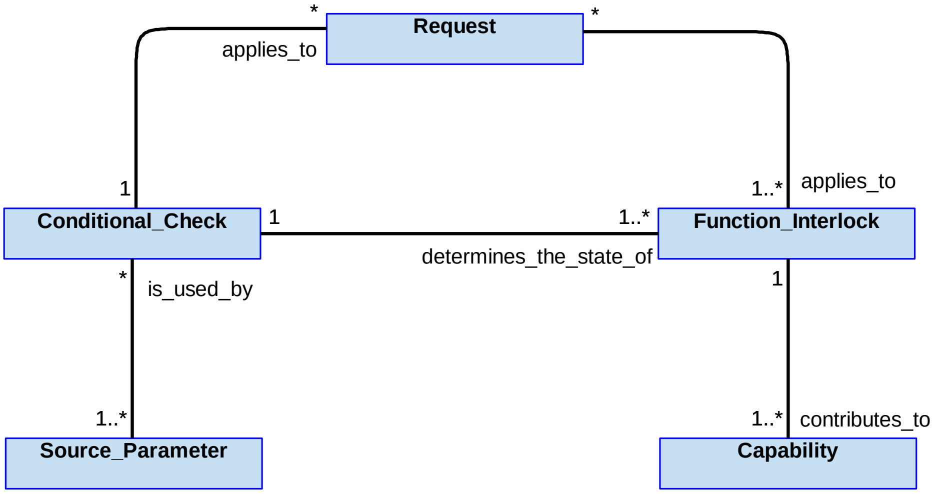


Figure 505: Interlocks Semantics

5.4.2.29.5.1 Entities

Conditional_Check

The conditional check defines the required conditions to enable or inhibit the operation of a function. For example: (weight_off_wheels = TRUE OR airspeed is greater than 200kts) AND altitude is greater than 1000ft.

Function_Interlock

A mechanism (software or physical) that enables or disables the operation of a function.

Capability

The capability to control Function_Interlocks.

Request

A request to enable or disable the operation of a function.

Source_Parameter

The source parameters (or data) used to determine whether to enable or inhibit the operation of a function. For example, undercarriage position or airspeed.

5.4.2.29.6 Design Rationale

5.4.2.29.6.1 Assumptions

None.

5.4.2.29.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Interlocks:

- Data Driving - The Source_Parameters and Conditional_Check associated with each Function_Interlock will vary per deployment and could be based on data.
- Recording and Logging - Describes the mechanism for how audit records relating to the interlocks will be handled.

Other Factors that were Taken into Account:

- Whilst Interlocks provides a view of its current capability, it is not required to predict future capability or identify missing information that affects the capability assessment. This is because this would involve predicting how often an interlock may be required which is outside the scope of this component and any testing required to determine capability of any interlock would be triggered by other components (e.g. Anomaly Detection), covered by continuous or power-up Built In Test (BIT) or covered by maintenance schedules (e.g. periodic checks to detect dormant faults).

Exploitation Considerations

- Interlocks is intended to provide an independent Conditional_Check to be used to enable or disable high criticality functionality.
- Interlocks may control hardware devices (e.g. relays) but will not have any direct physical interfaces with them. Alternatively or in addition to physical devices Interlocks may control elements of software.
- To reduce the time at risk of hazards occurring one of the Conditional_Checks may include the function being requested to be enabled/disabled.
- Interlocks may choose to determine for itself that a required level of confidence has been attained in the signals and or events it receives, e.g. by cross monitoring of dual channel signals from a mechanical switch. Alternatively, it may require that it is provided with assurances on the signals or events it receives, e.g. validity information is provided on received signals and or events.
- It is likely that due to the critical nature of this component that it would be specialised in a deployment, and or multiple instances used, to satisfy the needs of the separation of concerns typically due to safety analyses and or hardware architectural constraints.
- It is likely that this component would not necessarily respond to just the requests from external service requirements (i.e. other components) to enable or disable high criticality functionality (e.g. Asset Transitions request to enable engine ignition), but also to internal requirements to Interlocks (e.g. to disable all safety critical power supplies for stores release if the MASS switch is ever set to a Safe position).

5.4.2.29.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

This component will be used to, for example, prevent stores release when:

- Not authorised.

- The air vehicle is outside the safe release envelope.
- The air vehicle is not in the correct configuration for release (e.g. weapon bay doors are not open).

That is, this component prevents functions being activated based on a simple set of rules. If activated at the wrong time the functions could cause accidents with severity up to and including catastrophic. To minimise the safety reliance on other, more complex, components then DAL A is appropriate.

Where instances of this component are used to prevent hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.29.6.4 Security Considerations

The indicative security classification is O.

This component performs conditional checks prior to enabling or disabling high criticality functions. The rules are considered unlikely to be confidential, but this may vary depending on the function. The component will be expected to reside within the same security domain as the component requesting the interlock in order to avoid possible loss of availability due to boundary protection actions. This component will need to be in receipt of high integrity inputs and to be of high integrity itself.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to any changes made to the configured rules and interlock request successes and failures.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is expected to at least partially satisfy security enforcing functions by:

- **Verifying Integrity of Data** as the correct and authentic source of information to be used in the conditional checks performed.

5.4.2.29.7 Services

5.4.2.29.7.1 Service Definitions

5.4.2.29.7.1.1 Interlock_Request

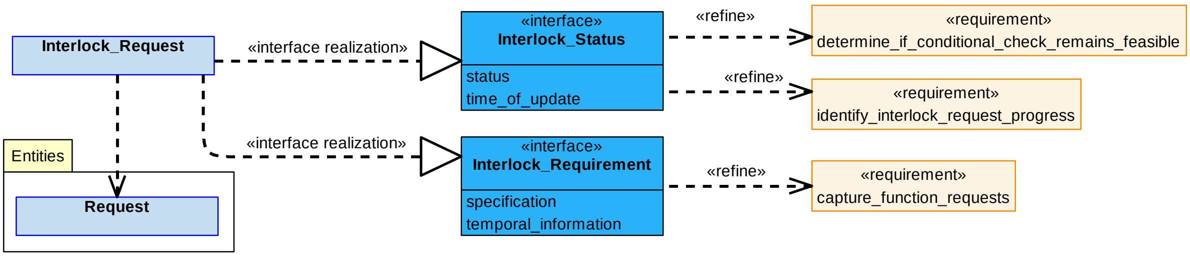


Figure 506: Interlock_Request Service Definition

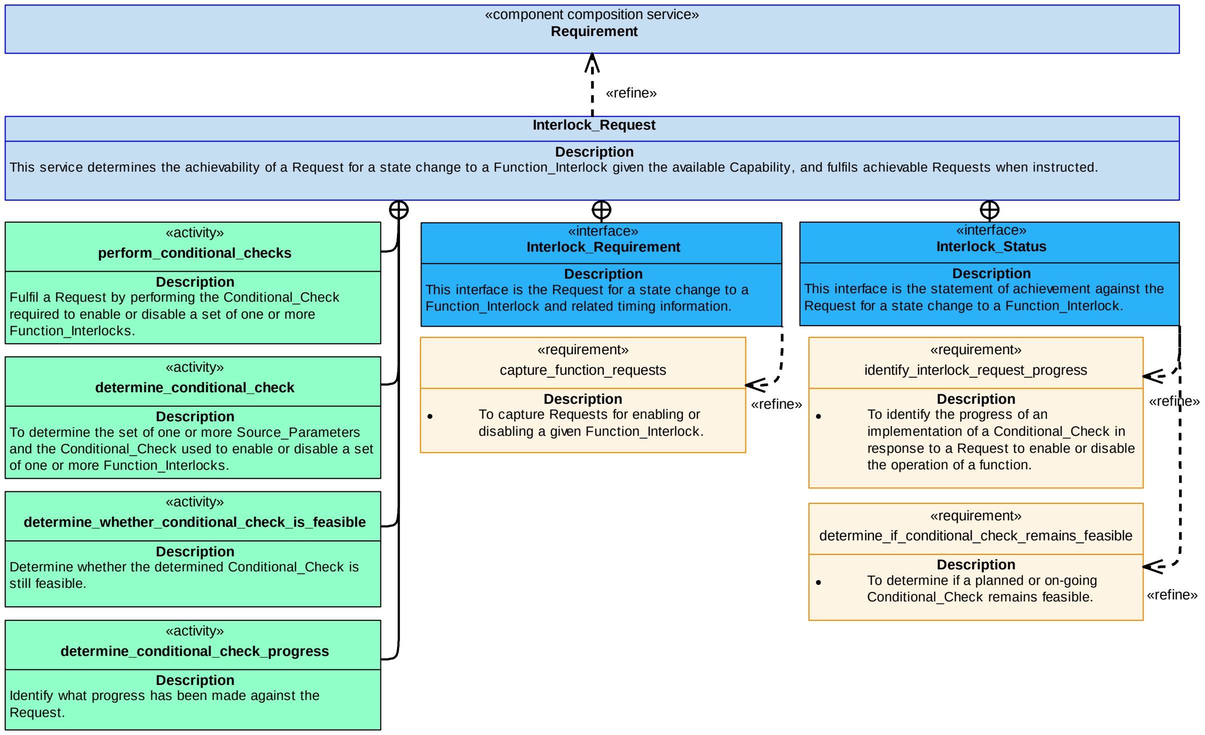

«component composition service»
Requirement
«refine»
Interlock_Request
Description
This service determines the achievability of a Request for a state change to a Function_Interlock given the available Capability, and fulfils achievable Requests when instructed.
«activity»
perform_conditional_checks
Description
Fulfil a Request by performing the Conditional_Check required to enable or disable a set of one or more Function_Interlocks.
«activity»
determine_conditional_check
Description
To determine the set of one or more Source_Parameters and the Conditional_Check used to enable or disable a set of one or more Function_Interlocks.
«activity»
determine_whether_conditional_check_is_feasible
Description
Determine whether the determined Conditional_Check is still feasible.
«activity»
determine_conditional_check_progress
Description
Identify what progress has been made against the Request.
«interface»
Interlock_Requirement
Description
This interface is the Request for a state change to a Function_Interlock and related timing information.
«requirement»
capture_function_requests
Description
To capture Requests for enabling or disabling a given Function_Interlock.
«refine»
«interface»
Interlock_Status
Description
This interface is the statement of achievement against the Request for a state change to a Function_Interlock.
«requirement»
identify_interlock_request_progress
Description
To identify the progress of an implementation of a Conditional_Check in response to a Request to enable or disable the operation of a function.
«refine»
«requirement»
determine_if_conditional_check_remains_feasible
Description
To determine if a planned or on-going Conditional_Check remains feasible.
«refine»

Figure 507: Interlock_Request Service Policy

Interlock_Request

This service determines the achievability of a Request for a state change to a Function_Interlock given the available Capability, and fulfils achievable Requests when instructed.

Interfaces

Interlock_Requirement

This interface is the Request for a state change to a Function_Interlock and related timing information.

Attributes

specification	The definition of the Request, e.g. enable the operation of a switch or enable software control of power supplies.
temporal_information	Information covering timing, such as start and end times.

Interlock_Status

This interface is the statement of achievement against the Request for a state change to a Function_Interlock.

Attributes

status	A high-level representation of achievement in relation to the Request (e.g. not started, in progress, complete).
time_of_update	The time at which a status update occurred.

Activities

perform_conditional_checks

Fulfil a Request by performing the Conditional_Check required to enable or disable a set of one or more Function_Interlocks.

determine_conditional_check

To determine the set of one or more Source_Parameters and the Conditional_Check used to enable or disable a set of one or more Function_Interlocks.

determine_whether_conditional_check_is_feasible

Determine whether the determined Conditional_Check is still feasible.

determine_conditional_check_progress

Identify what progress has been made against the Request.

5.4.2.29.7.1.2 Function_Consent

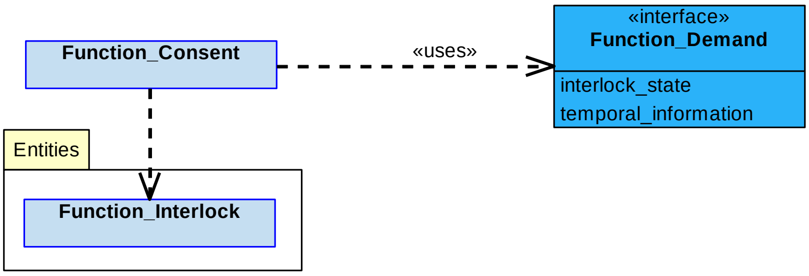


Figure 508: Function_Consent Service Definition

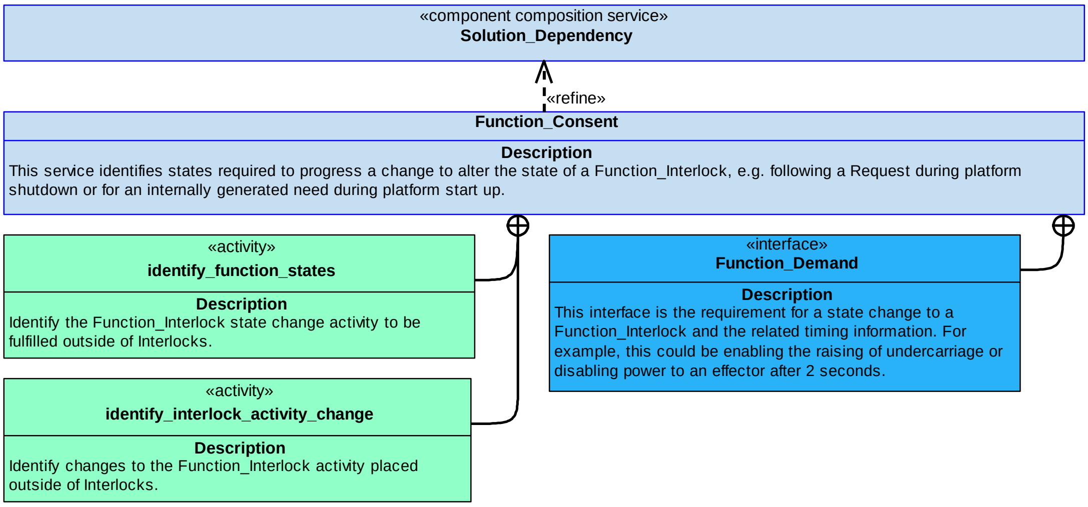

«component composition service»
Solution_Dependency
«refine»
Function_Consent
Description
This service identifies states required to progress a change to alter the state of a Function_Interlock, e.g. following a Request during platform shutdown or for an internally generated need during platform start up.
«activity»
identify_function_states
Description
Identify the Function_Interlock state change activity to be fulfilled outside of Interlocks.
«activity»
identify_interlock_activity_change
Description
Identify changes to the Function_Interlock activity placed outside of Interlocks.
«interface»
Function_Demand
Description
This interface is the requirement for a state change to a Function_Interlock and the related timing information. For example, this could be enabling the raising of undercarriage or disabling power to an effector after 2 seconds.

Figure 509: Function_Consent Service Policy

Function_Consent

This service identifies states required to progress a change to alter the state of a Function_Interlock, e.g. following a Request during platform shutdown or for an internally generated need during platform start up.

Interface

Function_Demand

This interface is the requirement for a state change to a Function_Interlock and the related timing information. For example, this could be enabling the raising of undercarriage or disabling power to an effector after 2 seconds.

Attributes

interlock_state	The required state to be used in order to enable or disable a desired Function_Interlock, e.g. enabling the provision of power or the ignition of engines.
temporal_information	Information covering timing, such as start and end times.

Activities

identify_function_states

Identify the Function_Interlock state change activity to be fulfilled outside of Interlocks.

identify_interlock_activity_change

Identify changes to the Function_Interlock activity placed outside of Interlocks.

5.4.2.29.7.1.3 Function_Interlock_Query

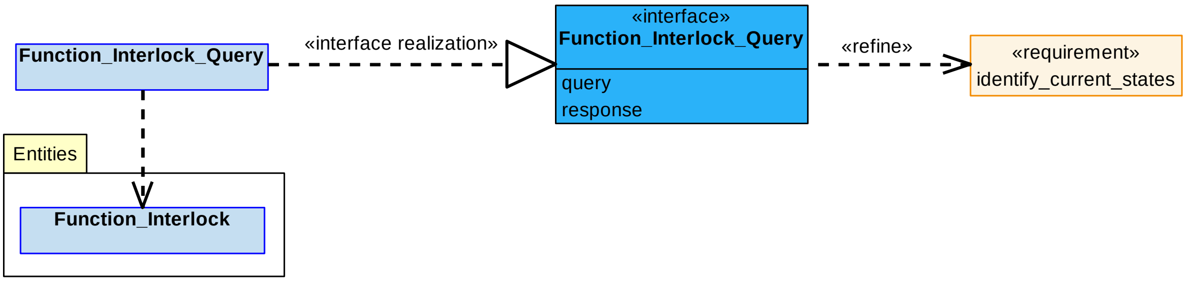


Figure 510: Function_Interlock_Query Service Definition

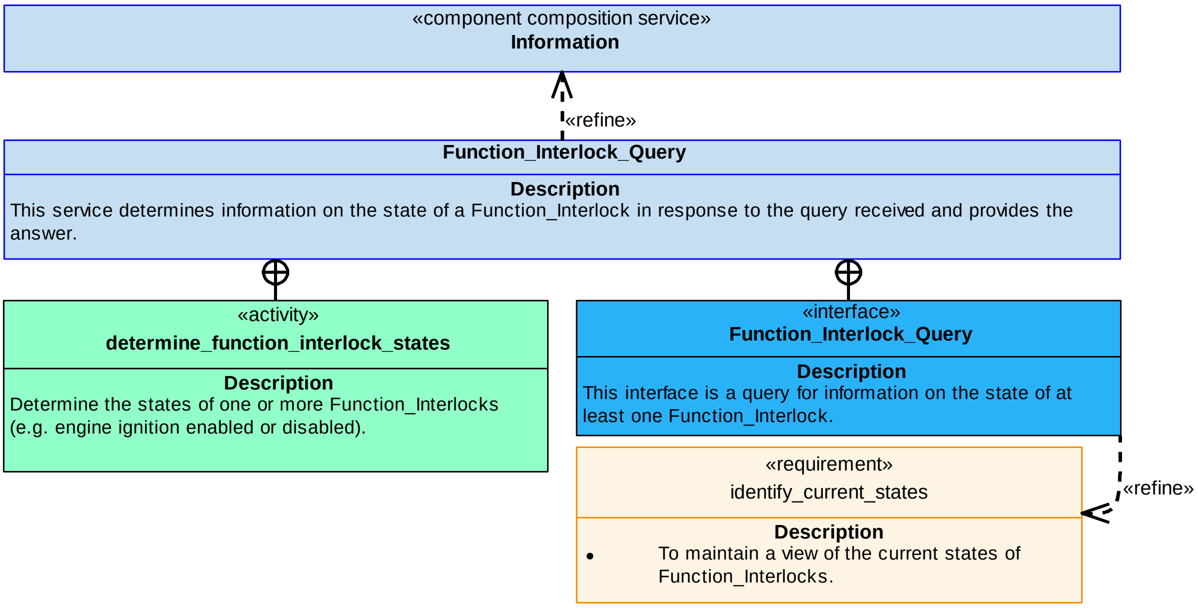

«component composition service»
Information
«refine»
Function_Interlock_Query
Description
This service determines information on the state of a Function_Interlock in response to the query received and provides the answer.
«activity»
determine_function_interlock_states
Description
Determine the states of one or more Function_Interlocks (e.g. engine ignition enabled or disabled).
«interface»
Function_Interlock_Query
Description
This interface is a query for information on the state of at least one Function_Interlock.
«requirement»
identify_current_states
Description
To maintain a view of the current states of Function_Interlocks.
«refine»

Figure 511: Function_Interlock_Query Service Policy

Function_Interlock_Query

This service determines information on the state of a Function_Interlock in response to the query received and provides the answer.

Interface

Function_Interlock_Query

This interface is a query for information on the state of at least one Function_Interlock.

Attributes

query	The request for state information on a Function_Interlock.
response	The state of the Function_Interlock returned in response to the query, e.g. store release power supplies are enabled.

Activity

determine_function_interlock_states

Determine the states of one or more Function_Interlocks (e.g. engine ignition enabled or disabled).

5.4.2.29.7.1.4 Information_Dependency

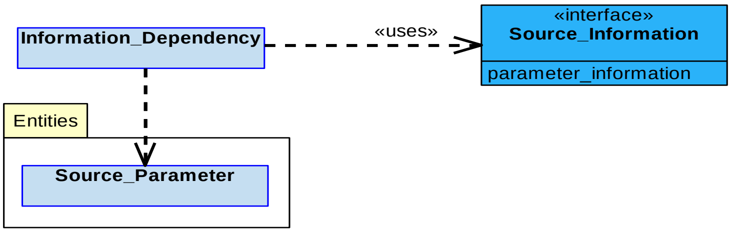


Figure 512: Information_Dependency Service Definition

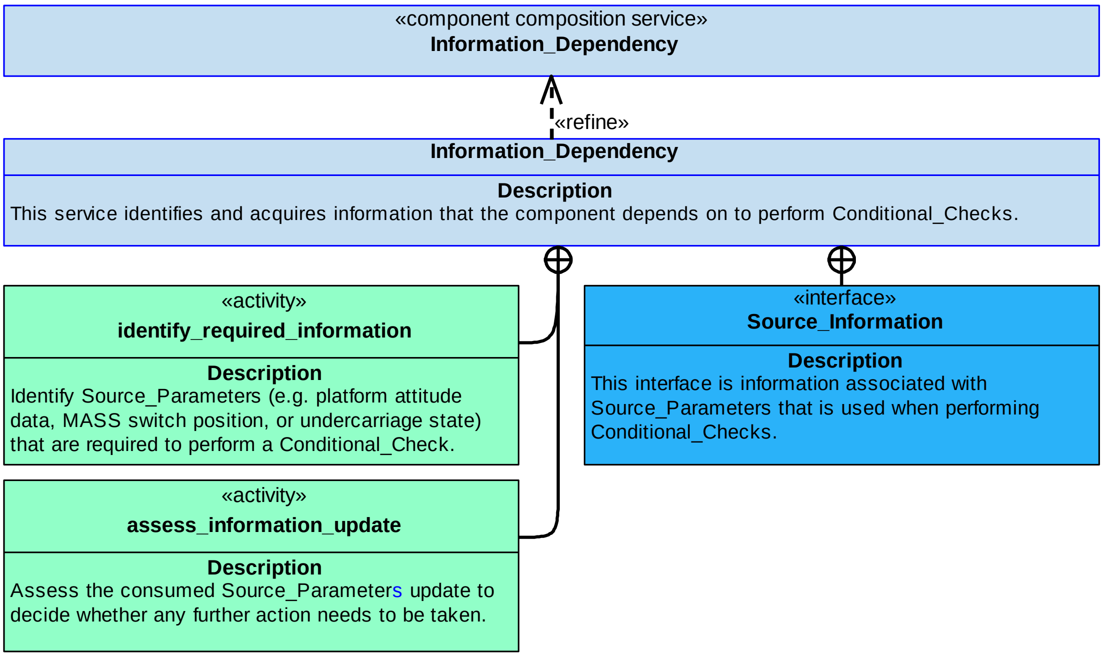

«component composition service»
Information_Dependency
«refine»
Information_Dependency
Description
This service identifies and acquires information that the component depends on to perform Conditional_Checks.
«activity»
identify_required_information
Description
Identify Source_Parameters (e.g. platform attitude data, MASS switch position, or undercarriage state) that are required to perform a Conditional_Check.
«activity»
assess_information_update
Description
Assess the consumed Source_Parameters update to decide whether any further action needs to be taken.
«interface»
Source_Information
Description
This interface is information associated with Source_Parameters that is used when performing Conditional_Checks.

Figure 513: Information_Dependency Service Policy

Information_Dependency

This service identifies and acquires information that the component depends on to perform Conditional_Checks.

Interface

Source_Information

This interface is information associated with Source_Parameters that is used when performing Conditional_Checks.

Attribute

parameter_information	The definition of the Source_Parameter, e.g. current platform altitude above the ground.

Activities

identify_required_information

Identify Source_Parameters (e.g. platform attitude data, MASS switch position, or undercarriage state) that are required to perform a Conditional_Check.

assess_information_update

Assess the consumed Source_Parameters update to decide whether any further action needs to be taken.

5.4.2.29.7.1.5 Capability

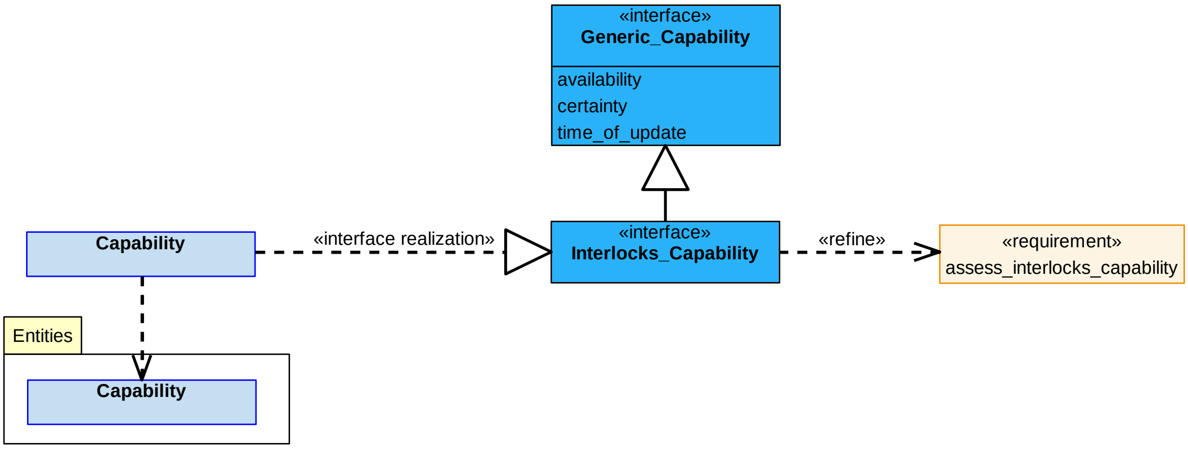


Figure 514: Capability Service Definition

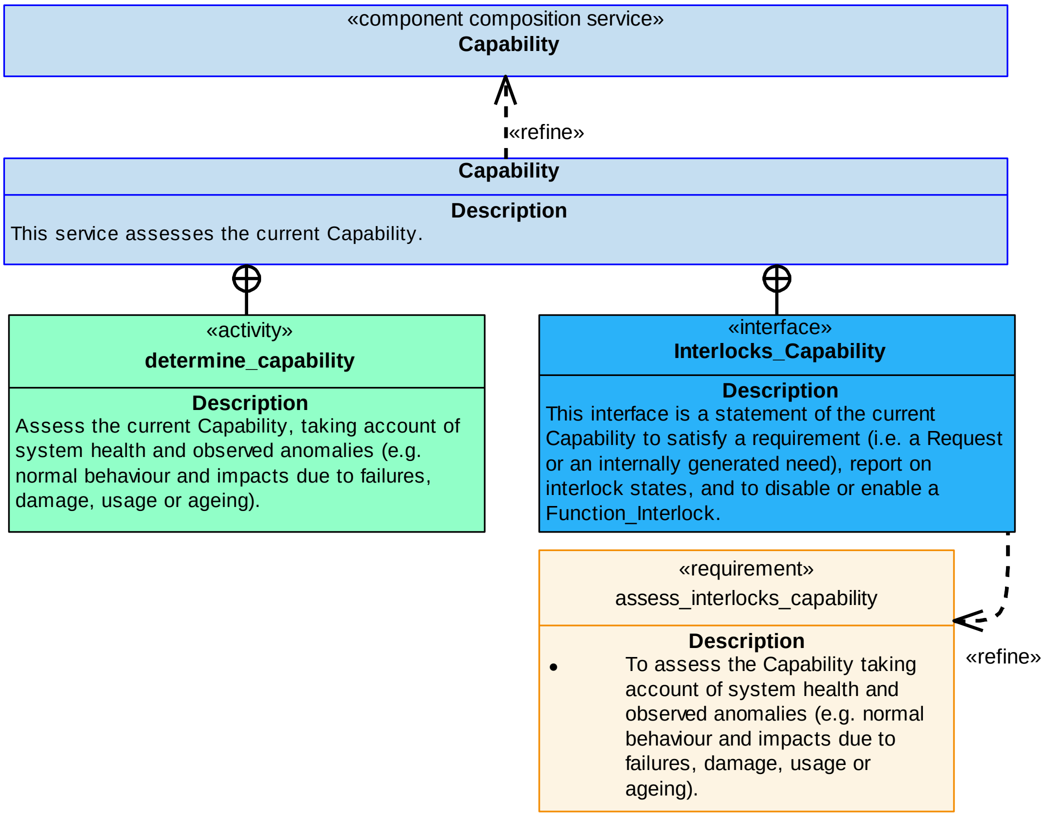

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current Capability.
«activity»
determine_capability
Description
Assess the current Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Interlocks_Capability
Description
This interface is a statement of the current Capability to satisfy a requirement (i.e. a Request or an internally generated need), report on interlock states, and to disable or enable a Function_Interlock.
«requirement»
assess_interlocks_capability
Description
To assess the Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 515: Capability Service Policy

Capability

This service assesses the current Capability.

Interface

Interlocks_Capability

This interface is a statement of the current Capability to satisfy a requirement (i.e. a Request or an internally generated need), report on interlock states, and to disable or enable a Function_Interlock.

Activity

determine_capability

Assess the current Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.29.7.1.6 Capability_Evidence

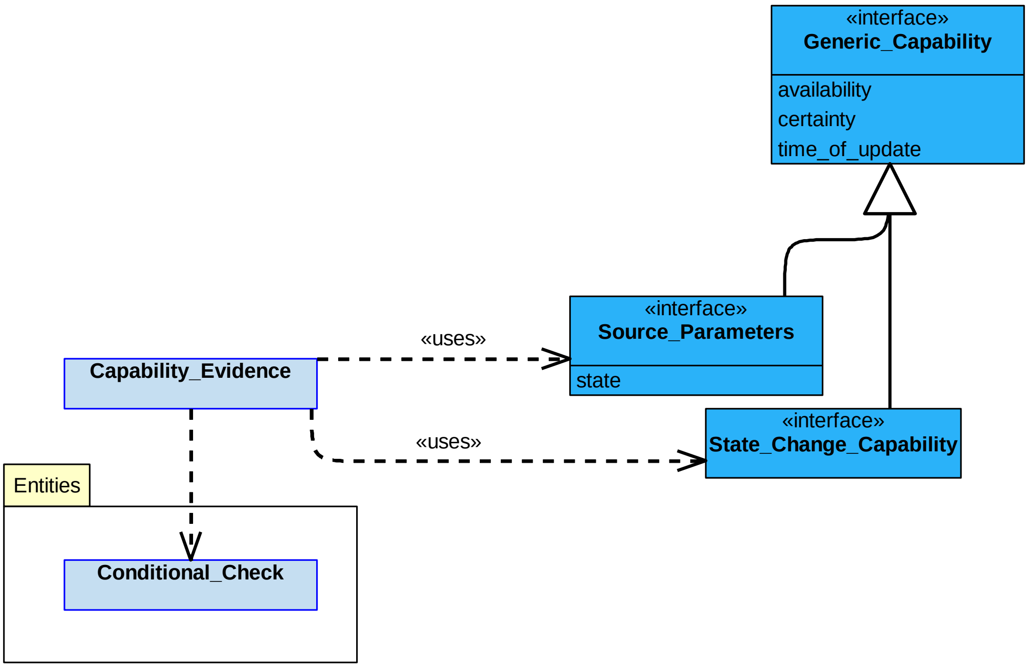


Figure 516: Capability_Evidence Service Definition

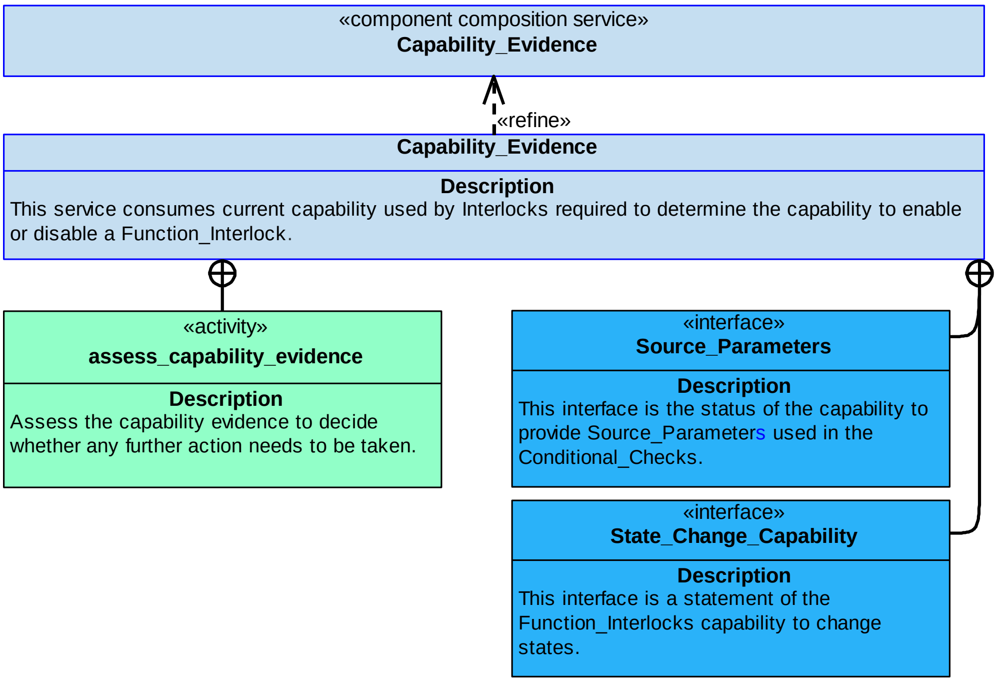

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current capability used by Interlocks required to determine the capability to enable or disable a Function_Interlock.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«interface»
Source_Parameters
Description
This interface is the status of the capability to provide Source_Parameters used in the Conditional_Checks.
«interface»
State_Change_Capability
Description
This interface is a statement of the Function_Interlocks capability to change states.

Figure 517: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current capability used by Interlocks required to determine the capability to enable or disable a Function_Interlock.

Interfaces

Source_Parameters

This interface is the status of the capability to provide Source_Parameters used in the Conditional_Checks.

Attribute

state	The current state of a Source_Parameter which may drive capability evidence, e.g. conflicting states.

State_Change_Capability

This interface is a statement of the Function_Interlocks capability to change states.

Activity

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

5.4.2.29.7.2 Service Dependencies

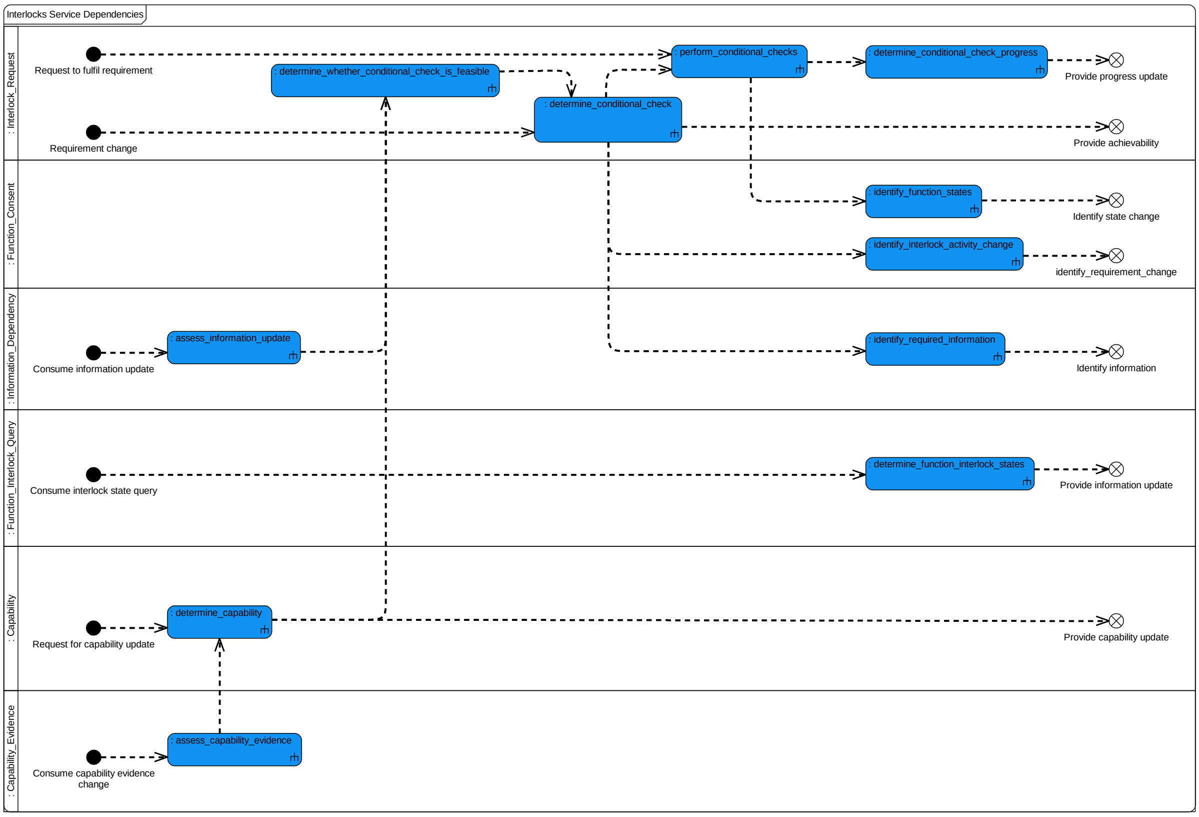

Interlocks Service Dependencies
: Interlock_Request
: Function_Consent
: Information_Dependency
: Function_Interlock_Query
: Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
: determine_whether_conditional_check_is_feasible
: determine_conditional_check
: perform_conditional_checks
: determine_conditional_check_progress
Provide progress update
Provide achievability
: identify_function_states
Identify state change
: identify_interlock_activity_change
identify_requirement_change
Consume information update
: assess_information_update
: identify_required_information
Identify information
Consume interlock state query
: determine_function_interlock_states
Provide information update
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence

Figure 518: Interlocks Service Dependencies

5.4.2.30 Inventory

5.4.2.30.1 Role

The role of Inventory is to determine and verify the system inventory.

5.4.2.30.2 Overview

Control Architecture

Inventory is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Inventory checks whether Items are present at Locations and verifies that each Item_at_Location is allowable (cleared for use). It also checks that the Inventory as a whole is allowable. It updates the Inventory when Items are released.

Examples of Use

Inventory will be used:

- When a system needs to know whether optionally loaded items are present.

5.4.2.30.3 Service Summary

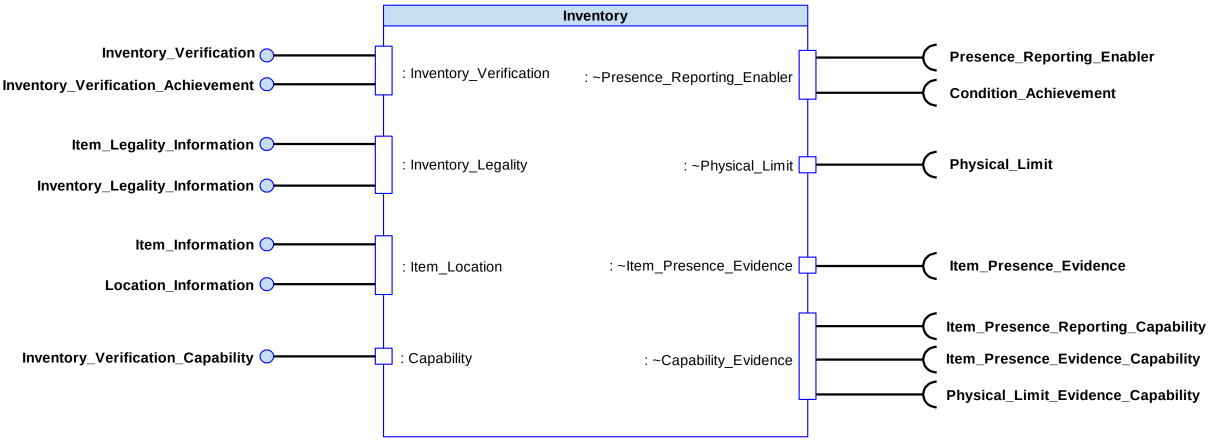

Inventory
Inventory_Verification
Inventory_Verification_Achievement
Item_Legality_Information
Inventory_Legality_Information
Item_Information
Location_Information
Inventory_Verification_Capability
: Inventory_Verification
: Inventory_Legality
: Item_Location
: Capability
: ~Presence_Reporting_Enabler
: ~Physical_Limit
: ~Item_Presence_Evidence
: ~Capability_Evidence
Presence_Reporting_Enabler
Condition_Achievement
Physical_Limit
Item_Presence_Evidence
Item_Presence_Reporting_Capability
Item_Presence_Evidence_Capability
Physical_Limit_Evidence_Capability

Figure 519: Inventory Service Summary

5.4.2.30.4 Responsibilities

verify_inventory

- To verify the current Inventory against the planned Inventory and to verify that an Inventory is a Legal_Inventory.

identify_whether_inventory_verification_remains_achievable

- To identify whether the requirement to determine and verify an Inventory is still achievable given current or predicted capability and conditions.

determine_inventory

- To determine the presence and identity of physical Items within the system Inventory.

identify_progress_of_inventory_verification

- To identify the progress of Inventory determination and verification against the requirement.

assess_capability

- To assess Inventory Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Inventory Capability assessment.

predict_inventory_verification_capability_progression

- To predict the progression of Inventory Capability over time and with use.

5.4.2.30.5 Subject Matter Semantics

The subject matter of Inventory is the defined Locations of the Items on the Exploiting Platform.

Exclusions

The subject matter of Inventory does not include:

- The actual and possible states of physical items (e.g. powered on / powered off) on the Exploiting Platform and how to transition between them in order to meet a capability need.

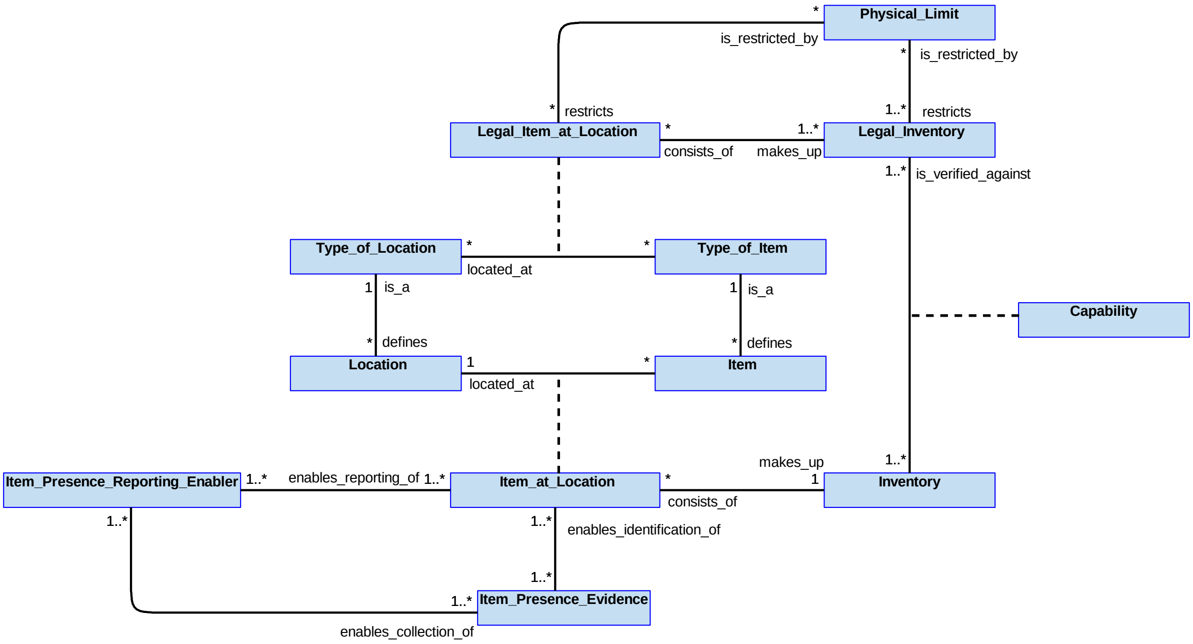


Figure 520: Inventory Semantics

5.4.2.30.5.1 Entities

Legal_Inventory

A set of permissible combinations of Legal_Item_at_Locations.

Legal_Item_at_Location

An allowable combination of a Type_of_Item in a Type_of_Location, e.g. a Storm Shadow is allowed on station 3 as station 3 is a heavy duty station, therefore they are compatible.

Inventory

The set of Item_at_Locations of the system.

Item

A particular object that is part of the system, e.g. a deployable asset such as a specific Paveway IV.

Item_at_Location

An Item at a specific Location, e.g. a Storm Shadow is present on station 3.

Location

A defined position on an Exploiting Platform to which an Item can be attached, e.g. station 3 or avionics bay 2.

Type_of_Item

A type of Item, e.g. a Paveway IV, a 500 litre fuel tank or a Storm Shadow.

Type_of_Location

A type of Location, e.g. a light duty station or a heavy duty station.

Physical_Limit

A physical restriction on a Legal_Inventory or a Legal_Item_at_Location. For example, an inventory legality restriction related to centre of gravity limits.

Item_Presence_Evidence

Evidence for the presence of an Item_at_Location. For example, umbilical sensor connections.

Item_Presence_Reporting_Enabler

A resource that is required to determine the presence of an Item. For example, power for role fit discovery.

Capability

The capability of the Inventory component to determine and verify the Inventory.

5.4.2.30.6 Design Rationale

5.4.2.30.6.1 Assumptions

None.

5.4.2.30.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Inventory:

- Data Driving - Legal system inventories will change following the introduction of new Item types, to accommodate this the legal system inventories should be data drivable using build time data in accordance with the Data Driving PYRAMID concept.
- Interfacing with Deployable Assets - Inventory will determine the location of and verify the legality of deployable assets attached to the Exploiting Platform and therefore any communication with deployable assets should be in accordance with the Interfacing with Deployable Assets PYRAMID concept.

Extensions

- It is not expected that extension components will be needed.

Exploitation Considerations

- Inventory may be updated during the mission if Items are released or jettisoned. During a release or jettison the presence of stores may need to be updated frequently to allow other components to control the release sequence.

5.4.2.30.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

Failure of this component could cause, for example:

- An incorrect calculation of mass, balance and drag.
- Release of a store type not intended by the crew.
- Release of a stores package that results in an out of balance condition.

Therefore, failure of this component could cause uncontrolled flight of the Exploiting Platform due to exceedance of the flight envelope or structural limits. This could lead to an uncontrolled crash. The result is likely to be loss of the Exploiting Platform and fatalities.

5.4.2.30.6.4 Security Considerations

The indicative security classification is SNEO.

This component contains knowledge of the Items (including stores and role-fit equipment) and their Location, from which combat effectiveness and possible mission purpose may be derived. This information is considered to be SNEO. The component will determine and verify the Inventory. Loss of integrity or availability of the Inventory may affect the ability of the Exploiting Platform to use its stores and sensors as intended or lead to a loss of confidence in the Exploiting Platform mass and balance.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to changes made to the Inventory.
- **Maintaining Audit Records** of changes made to the Inventory.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Proving **System Status and Monitoring** of the Inventory, with any unexpected changes to Inventory being a possible sign the Exploiting Platform has been compromised.

The component is expected to satisfy security enforcing functions by:

- **Verifying Integrity of Data** relating to the Exploiting Platform Inventory, when loaded and for any subsequent changes.

5.4.2.30.7 Services

5.4.2.30.7.1 Service Definitions

5.4.2.30.7.1.1 Inventory_Verification

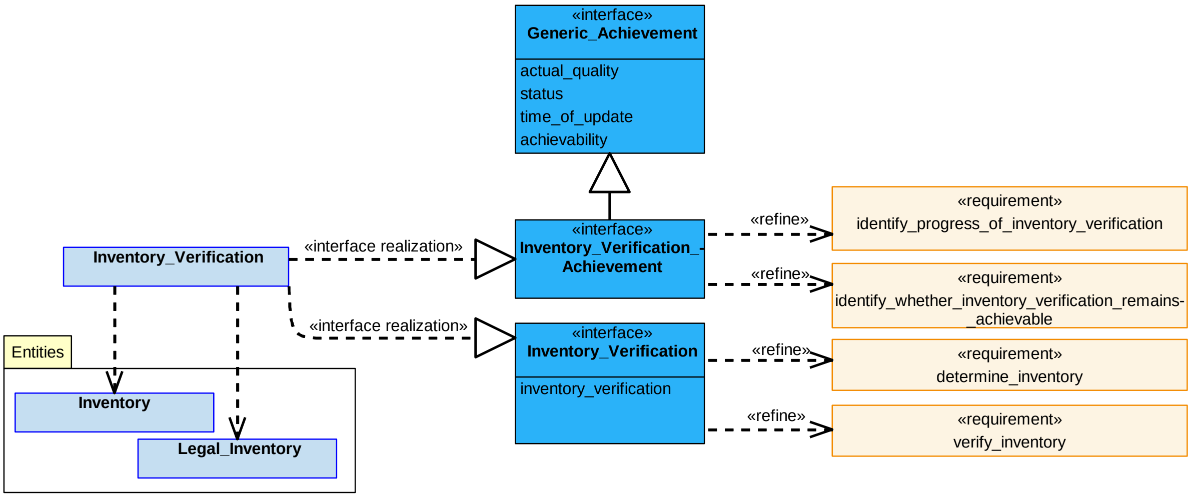


Figure 521: Inventory_Verification Service Definition

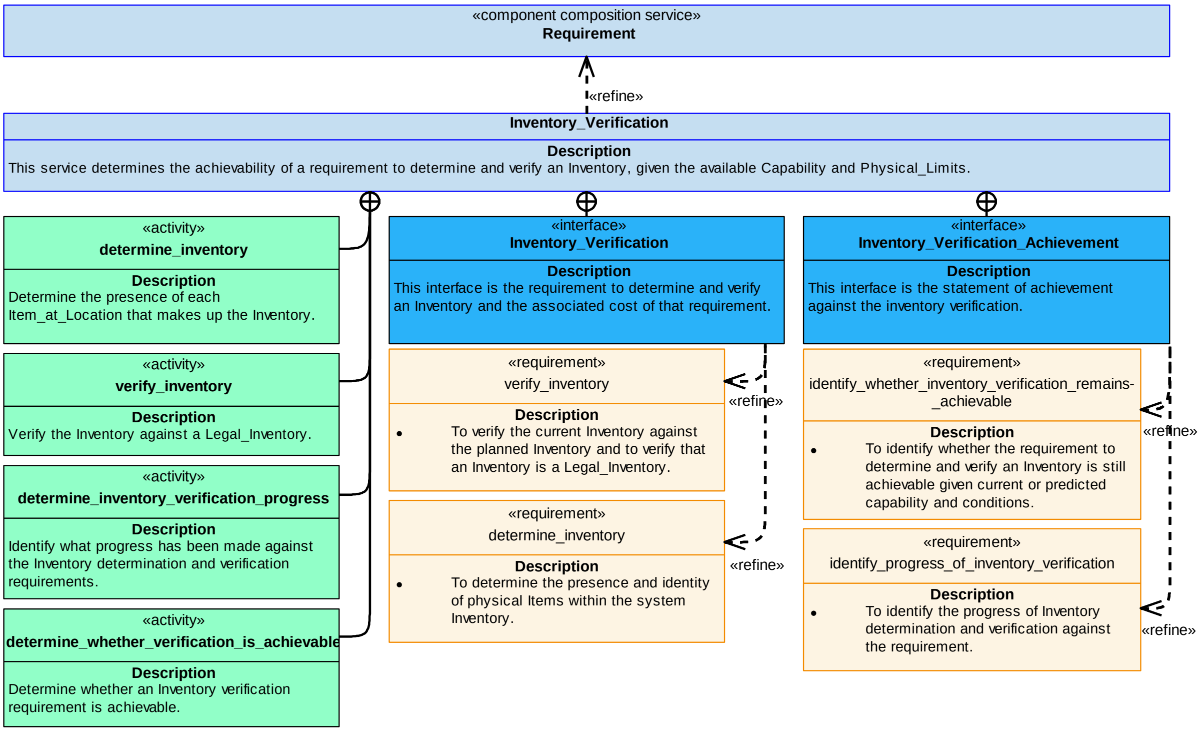

«component composition service»
Requirement
«refine»
Inventory_Verification
Description
This service determines the achievability of a requirement to determine and verify an Inventory, given the available Capability and Physical_Limits.
«activity»
determine_inventory
Description
Determine the presence of each Item_at_Location that makes up the Inventory.
«activity»
verify_inventory
Description
Verify the Inventory against a Legal_Inventory.
«activity»
determine_inventory_verification_progress
Description
Identify what progress has been made against the Inventory determination and verification requirements.
«activity»
determine_whether_verification_is_achievable
Description
Determine whether an Inventory verification requirement is achievable.
«interface»
Inventory_Verification
Description
This interface is the requirement to determine and verify an Inventory and the associated cost of that requirement.
«requirement»
verify_inventory
Description
To verify the current Inventory against the planned Inventory and to verify that an Inventory is a Legal_Inventory.
«refine»
«requirement»
determine_inventory
Description
To determine the presence and identity of physical Items within the system Inventory.
«refine»
«interface»
Inventory_Verification_Achievement
Description
This interface is the statement of achievement against the inventory verification.
«requirement»
identify_whether_inventory_verification_remains-_achievable
Description
To identify whether the requirement to determine and verify an Inventory is still achievable given current or predicted capability and conditions.
«refine»
«requirement»
identify_progress_of_inventory_verification
Description
To identify the progress of Inventory determination and verification against the requirement.
«refine»

Figure 522: Inventory_Verification Service Policy

Inventory_Verification

This service determines the achievability of a requirement to determine and verify an Inventory, given the available Capability and Physical_Limits.

Interfaces

Inventory_Verification

This interface is the requirement to determine and verify an Inventory and the associated cost of that requirement.

Attribute

inventory_verification	The definition of the requirement to determine and verify the Inventory. For example, this may be the requirement to verify the planned Inventory against the current Inventory.

Inventory_Verification_Achievement

This interface is the statement of achievement against the inventory verification.

Activities

determine_inventory

Determine the presence of each Item_at_Location that makes up the Inventory.

verify_inventory

Verify the Inventory against a Legal_Inventory.

determine_inventory_verification_progress

Identify what progress has been made against the Inventory determination and verification requirements.

determine_whether_verification_is_achievable

Determine whether an Inventory verification requirement is achievable.

5.4.2.30.7.1.2 Presence_Reporting_Enabler

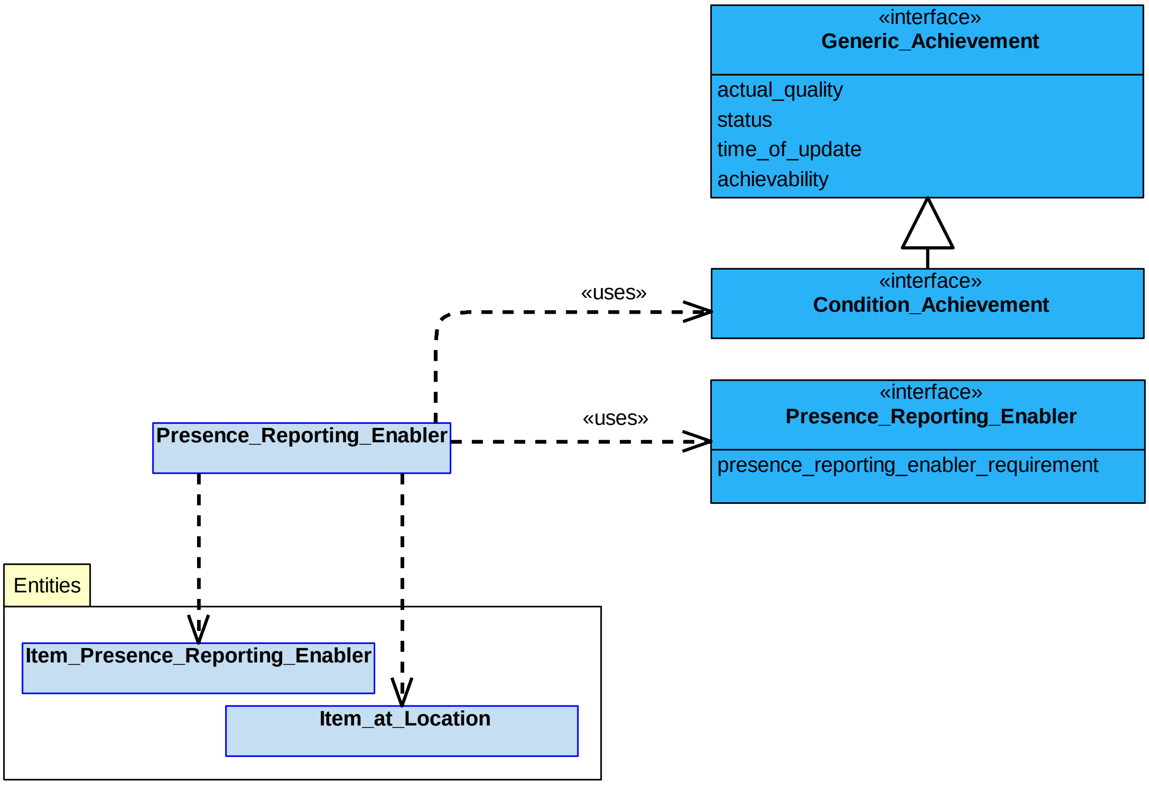


Figure 523: Presence_Reporting_Enabler Service Definition

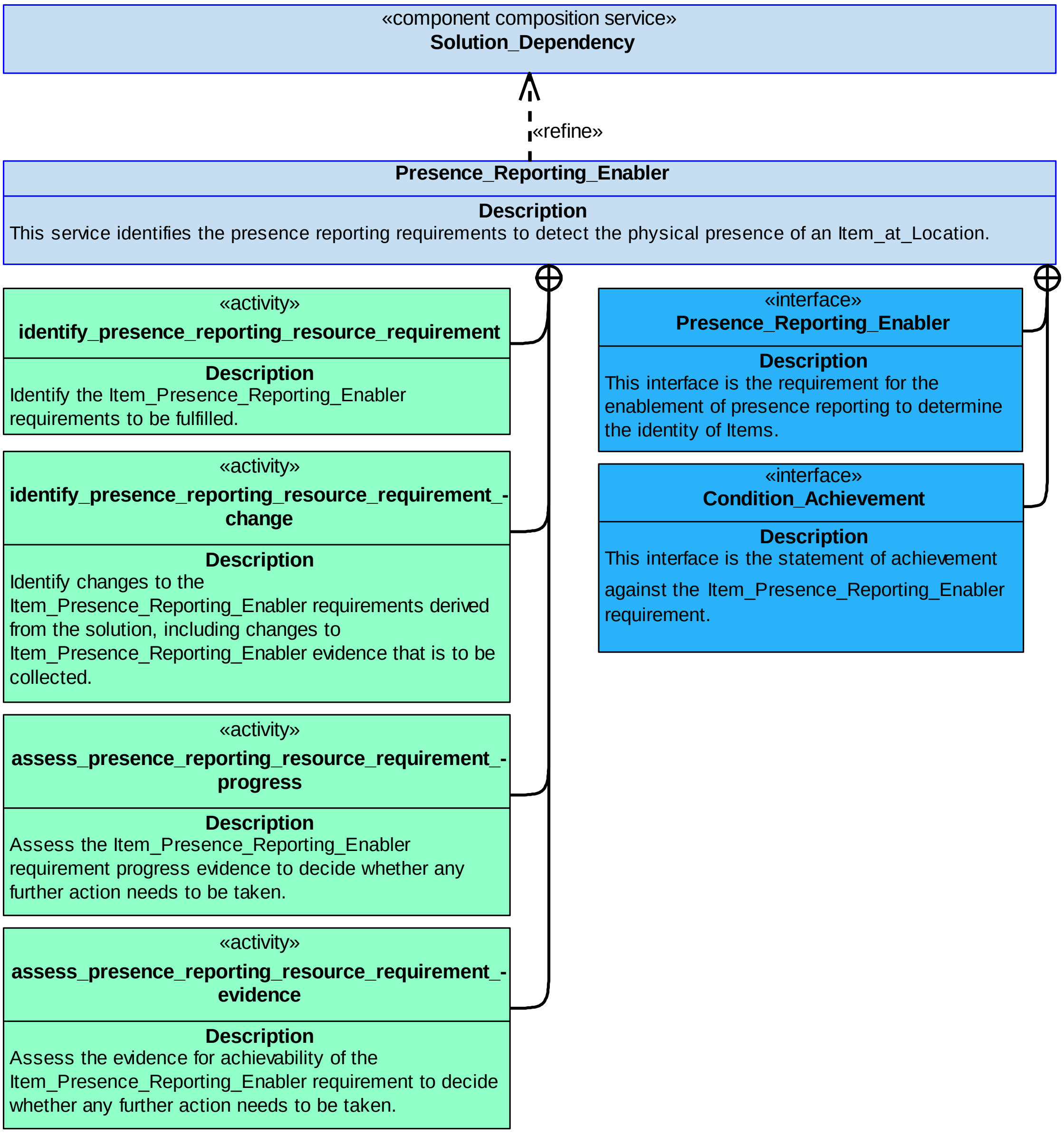

«component composition service»
Solution_Dependency
«refine»
Presence_Reporting_Enabler
Description
This service identifies the presence reporting requirements to detect the physical presence of an Item_at_Location.
«activity»
identify_presence_reporting_resource_requirement
Description
Identify the Item_Presence_Reporting_Enabler requirements to be fulfilled.
«activity»
identify_presence_reporting_resource_requirement_-change
Description
Identify changes to the Item_Presence_Reporting_Enabler requirements derived from the solution, including changes to Item_Presence_Reporting_Enabler evidence that is to be collected.
«activity»
assess_presence_reporting_resource_requirement_-progress
Description
Assess the Item_Presence_Reporting_Enabler requirement progress evidence to decide whether any further action needs to be taken.
«activity»
assess_presence_reporting_resource_requirement_-evidence
Description
Assess the evidence for achievability of the Item_Presence_Reporting_Enabler requirement to decide whether any further action needs to be taken.
«interface»
Presence_Reporting_Enabler
Description
This interface is the requirement for the enablement of presence reporting to determine the identity of Items.
«interface»
Condition_Achievement
Description
This interface is the statement of achievement against the Item_Presence_Reporting_Enabler requirement.

Figure 524: Presence_Reporting_Enabler Service Policy

Presence_Reporting_Enabler

This service identifies the presence reporting requirements to detect the physical presence of an Item_at_Location.

Interfaces

Presence_Reporting_Enabler

This interface is the requirement for the enablement of presence reporting to determine the identity of Items.

Attribute

presence_reporting_enabler_requirement	The requirement to enable the reporting of the presence and/or type of an Item_at_Location.

Condition_Achievement

This interface is the statement of achievement against the Item_Presence_Reporting_Enabler requirement.

Activities

identify_presence_reporting_resource_requirement

Identify the Item_Presence_Reporting_Enabler requirements to be fulfilled.

assess_presence_reporting_resource_requirement_evidence

Assess the evidence for achievability of the Item_Presence_Reporting_Enabler requirement to decide whether any further action needs to be taken.

identify_presence_reporting_resource_requirement_change

Identify changes to the Item_Presence_Reporting_Enabler requirements derived from the solution, including changes to Item_Presence_Reporting_Enabler evidence that is to be collected.

assess_presence_reporting_resource_requirement_progress

Assess the Item_Presence_Reporting_Enabler requirement progress evidence to decide whether any further action needs to be taken.

5.4.2.30.7.1.3 Inventory_Legality

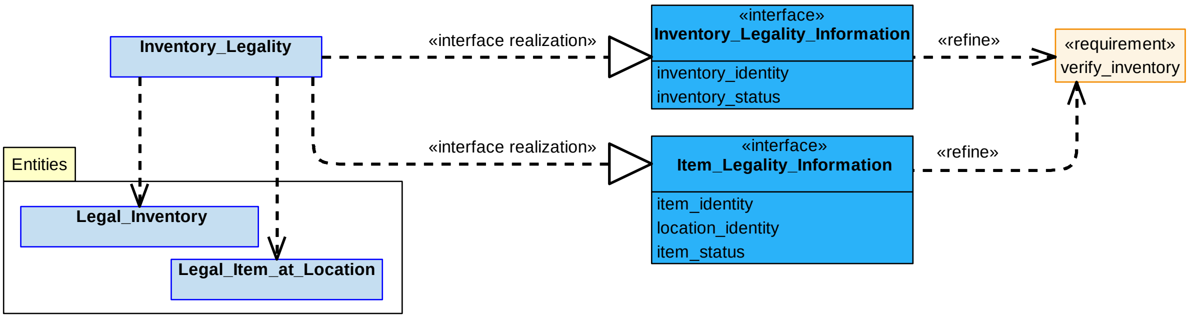


Figure 525: Inventory_Legality Service Definition

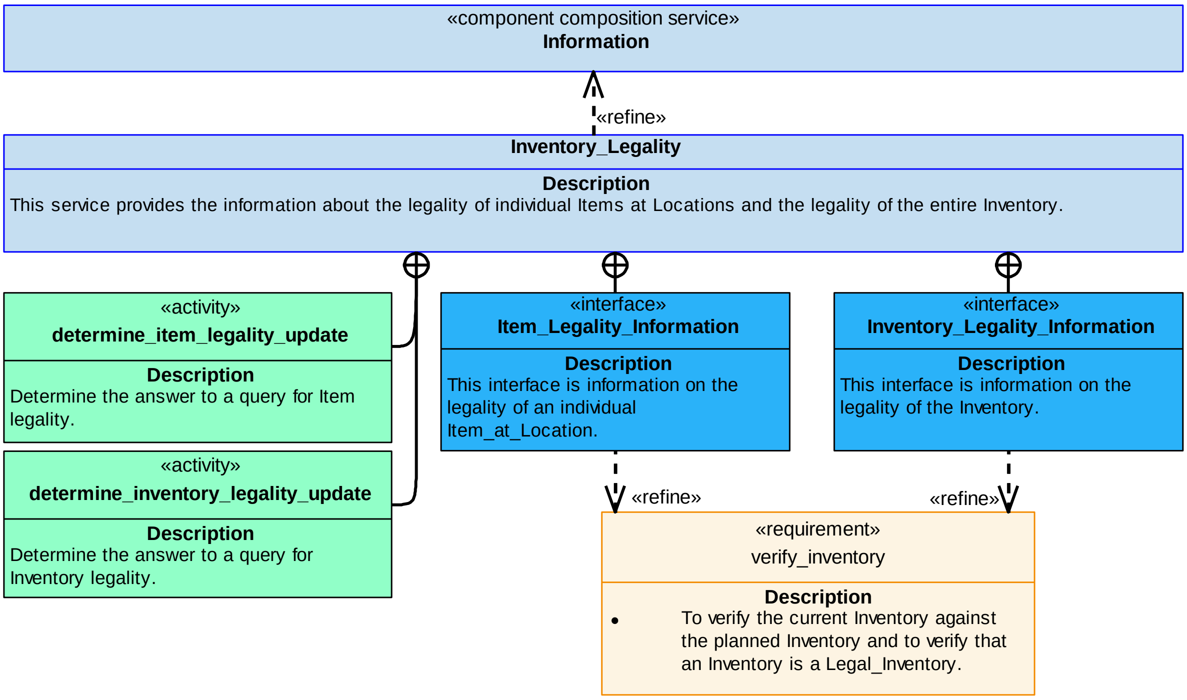

«component composition service»
Information
«refine»
Inventory_Legality
Description
This service provides the information about the legality of individual Items at Locations and the legality of the entire Inventory.
«activity»
determine_item_legality_update
Description
Determine the answer to a query for Item legality.
«activity»
determine_inventory_legality_update
Description
Determine the answer to a query for Inventory legality.
«interface»
Item_Legality_Information
Description
This interface is information on the legality of an individual Item_at_Location.
«interface»
Inventory_Legality_Information
Description
This interface is information on the legality of the Inventory.
«refine»
«refine»
«requirement»
verify_inventory
Description
To verify the current Inventory against the planned Inventory and to verify that an Inventory is a Legal_Inventory.

Figure 526: Inventory_Legality Service Policy

Inventory_Legality

This service provides the information about the legality of individual Items at Locations and the legality of the entire Inventory.

Interfaces

Inventory_Legality_Information

This interface is information on the legality of the Inventory.

Attributes

inventory_identity	The identity of the Inventory. The specified Inventory may be the current Inventory or may be a potential Inventory that might arise due to the addition, movement or release of an Item.
inventory_status	The legality status of the Inventory, e.g. Legal or Not Legal.

Item_Legality_Information

This interface is information on the legality of an individual Item_at_Location.

Attributes

item_identity	The identity of the Item.
location_identity	The identity of the Location.
item_status	The legality status of the Item, e.g. Legal or Not Legal.

Activities

determine_inventory_legality_update

Determine the answer to a query for Inventory legality.

determine_item_legality_update

Determine the answer to a query for Item legality.

5.4.2.30.7.1.4 Item_Location

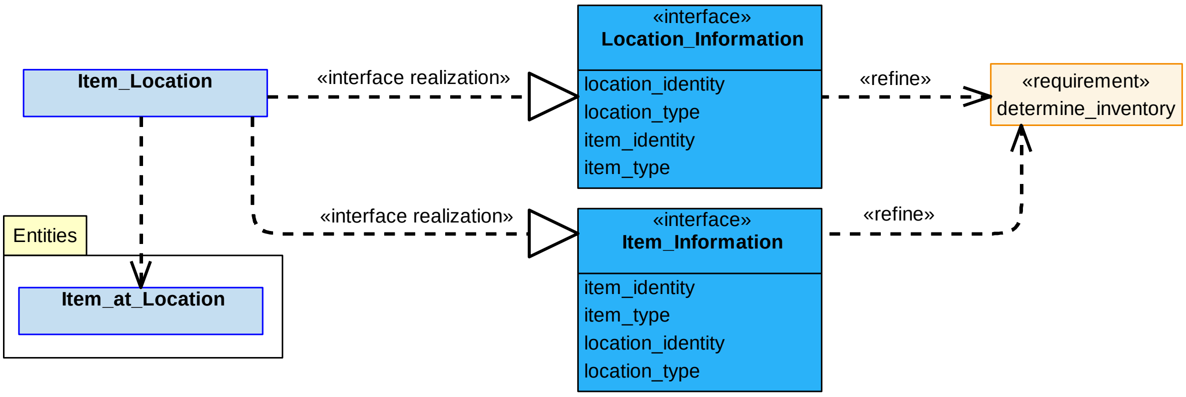


Figure 527: Item_Location Service Definition

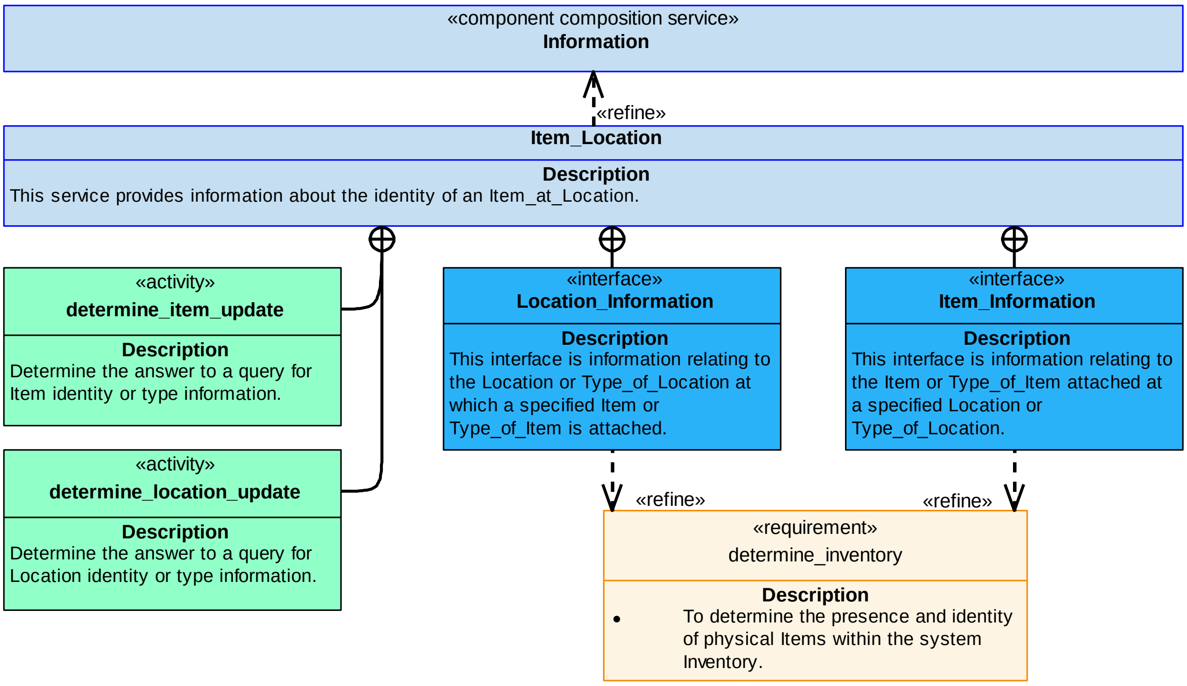

«component composition service»
Information
«refine»
Item_Location
Description
This service provides information about the identity of an Item_at_Location.
«activity»
determine_item_update
Description
Determine the answer to a query for Item identity or type information.
«activity»
determine_location_update
Description
Determine the answer to a query for Location identity or type information.
«interface»
Location_Information
Description
This interface is information relating to the Location or Type_of_Location at which a specified Item or Type_of_Item is attached.
«interface»
Item_Information
Description
This interface is information relating to the Item or Type_of_Item attached at a specified Location or Type_of_Location.
«refine»
«refine»
«requirement»
determine_inventory
Description
• To determine the presence and identity of physical Items within the system Inventory.

Figure 528: Item_Location Service Policy

Item_Location

This service provides information about the identity of an Item_at_Location.

Interfaces

Location_Information

This interface is information relating to the Location or Type_of_Location at which a specified Item or Type_of_Item is attached.

Attributes

location_identity	The identity of the Location at which the Item is located.
location_type	The Type_of_Location at which the Item is located. For example, an external pylon.
item_identity	The identity of the Item at the Location.
item_type	The Type_of_Item at the Location.

Item_Information

This interface is information relating to the Item or Type_of_Item attached at a specified Location or Type_of_Location.

Attributes

item_identity	The identity of the Item at the Location.
item_type	The Type_of_Item at the Location.
location_identity	The identity of the Location at which the Item is located.
location_type	The Type_of_Location at which the Item is located. For example, an external pylon.

Activities

determine_location_update

Determine the answer to a query for Location identity or type information.

determine_item_update

Determine the answer to a query for Item identity or type information.

5.4.2.30.7.1.5 Physical_Limit

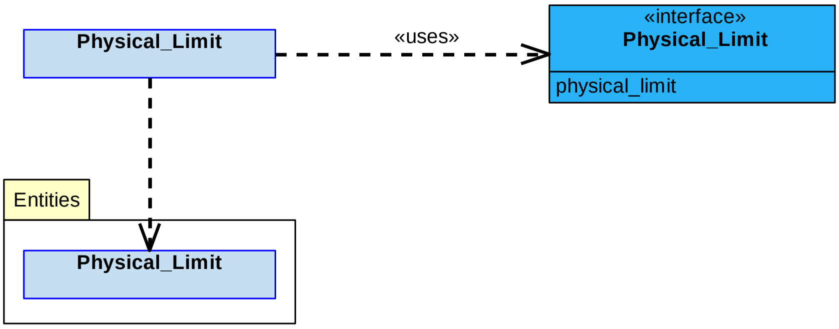


Figure 529: Physical_Limit Service Definition

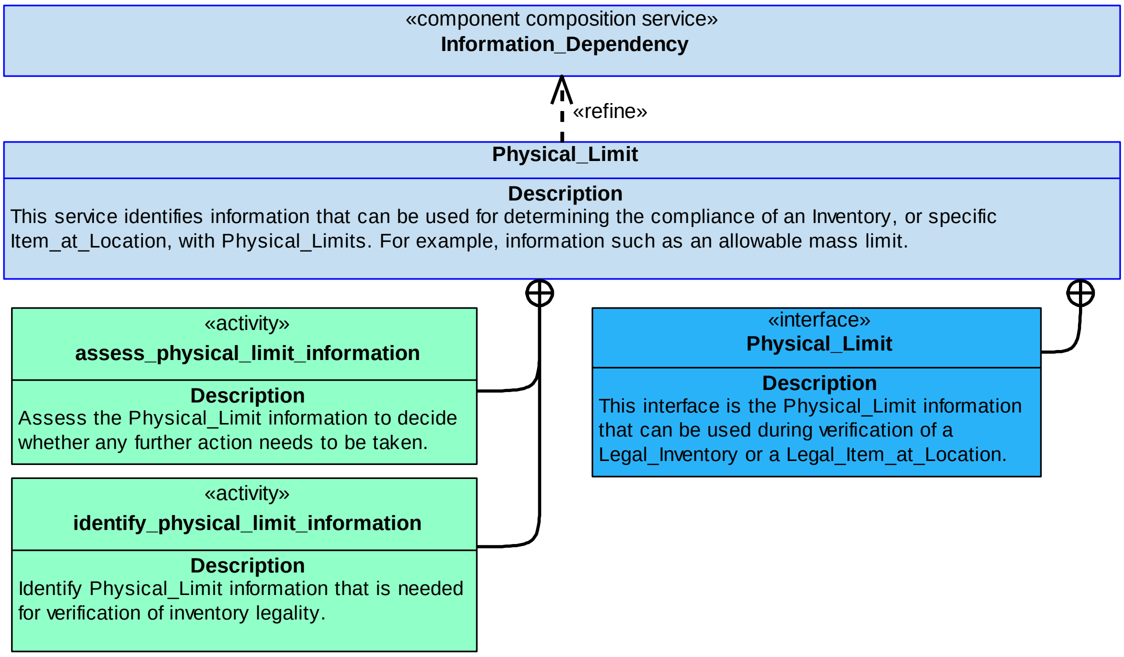

«component composition service»
Information_Dependency
«refine»
Physical_Limit
Description
This service identifies information that can be used for determining the compliance of an Inventory, or specific Item_at_Location, with Physical_Limits. For example, information such as an allowable mass limit.
«activity»
assess_physical_limit_information
Description
Assess the Physical_Limit information to decide whether any further action needs to be taken.
«activity»
identify_physical_limit_information
Description
Identify Physical_Limit information that is needed for verification of inventory legality.
«interface»
Physical_Limit
Description
This interface is the Physical_Limit information that can be used during verification of a Legal_Inventory or a Legal_Item_at_Location.

Figure 530: Physical_Limit Service Policy

Physical_Limit

This service identifies information that can be used for determining the compliance of an Inventory, or specific Item_at_Location, with Physical_Limits. For example, information such as an allowable mass limit.

<u>Interface</u>

Physical_Limit

This interface is the Physical_Limit information that can be used during verification of a Legal_Inventory or a Legal_Item_at_Location.

<u>Attribute</u>

physical_limit	The Physical_Limit information, e.g. the mass limit for a position associated with a Location.

<u>Activities</u>

assess_physical_limit_information

Assess the Physical_Limit information to decide whether any further action needs to be taken.

identify_physical_limit_information

Identify Physical_Limit information that is needed for verification of inventory legality.

5.4.2.30.7.1.6 Item_Presence_Evidence

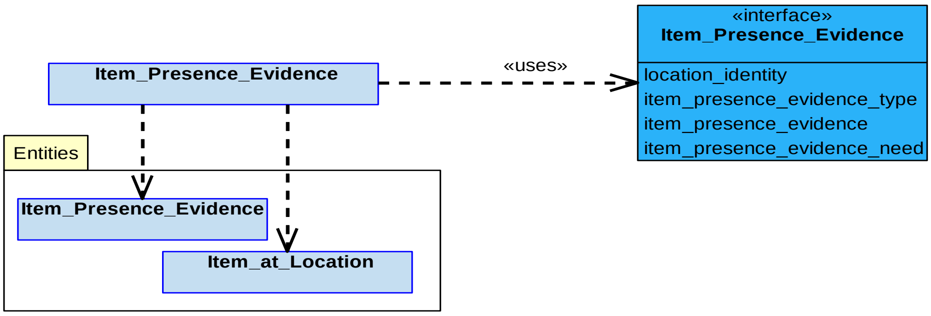


Figure 531: Item_Presence_Evidence Service Definition

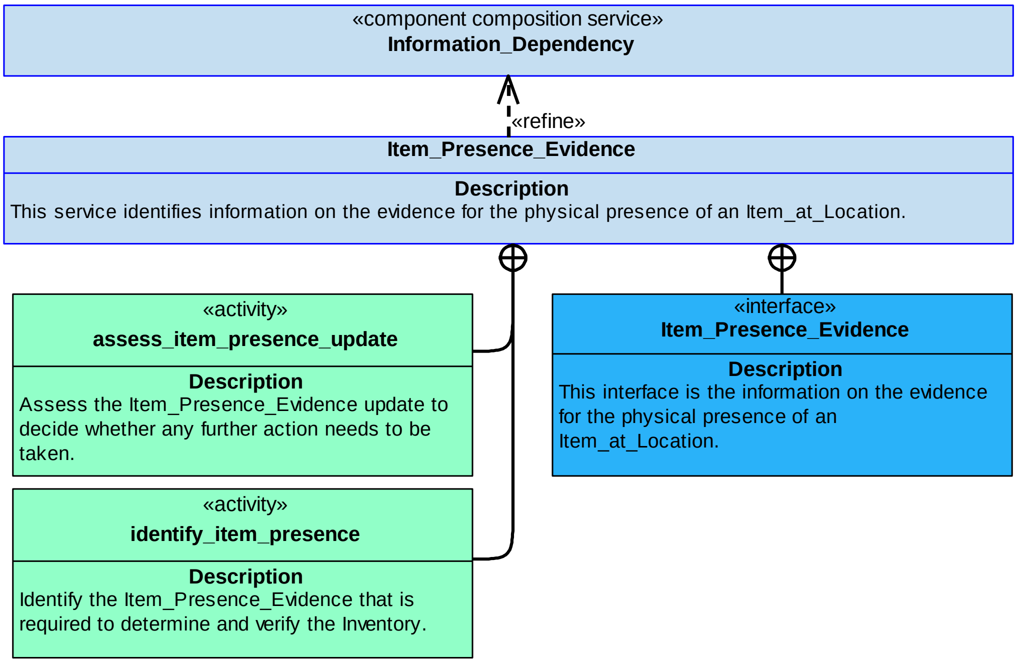

«component composition service»
Information_Dependency
«refine»
Item_Presence_Evidence
Description
This service identifies information on the evidence for the physical presence of an Item_at_Location.
«activity»
assess_item_presence_update
Description
Assess the Item_Presence_Evidence update to decide whether any further action needs to be taken.
«interface»
Item_Presence_Evidence
Description
This interface is the information on the evidence for the physical presence of an Item_at_Location.
«activity»
identify_item_presence
Description
Identify the Item_Presence_Evidence that is required to determine and verify the Inventory.

Figure 532: Item_Presence_Evidence Service Policy

Item_Presence_Evidence

This service identifies information on the evidence for the physical presence of an Item_at_Location.

Interface

Item_Presence_Evidence

This interface is the information on the evidence for the physical presence of an Item_at_Location.

Attributes

location_identity	The identity of a Location at which an Item may be located.
item_presence_evidence_type	The type of Item_Presence_Evidence which applies to a Location. For example, weight on pylon.
item_presence_evidence	Specific evidence for the presence of an Item_at_Location. For example, the specific value for the weight on pylon.

item_presence_evidence_need	The need for evidence on the physical presence of an Item_at_Location, e.g. a request for such evidence.

Activities

assess_item_presence_update

Assess the Item_Presence_Evidence update to decide whether any further action needs to be taken.

identify_item_presence

Identify the Item_Presence_Evidence that is required to determine and verify the Inventory.

5.4.2.30.7.1.7 Capability

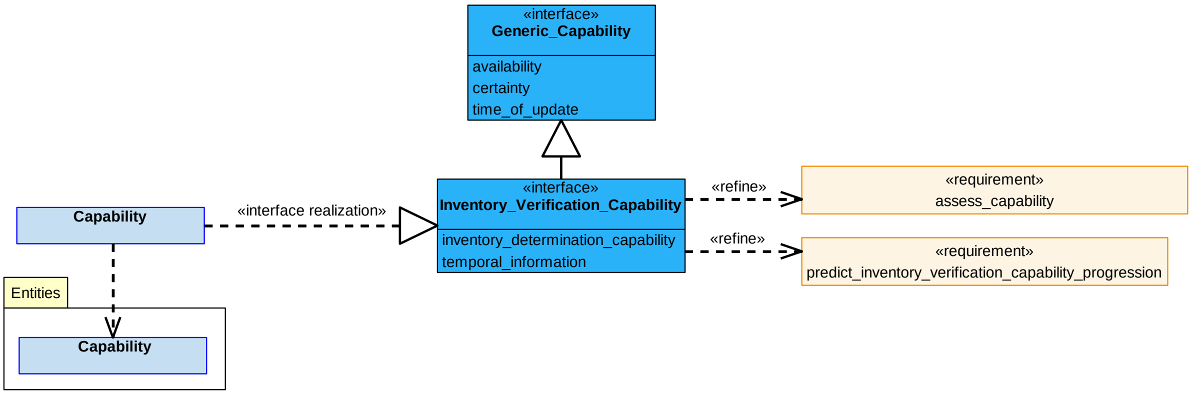


Figure 533: Capability Service Definition

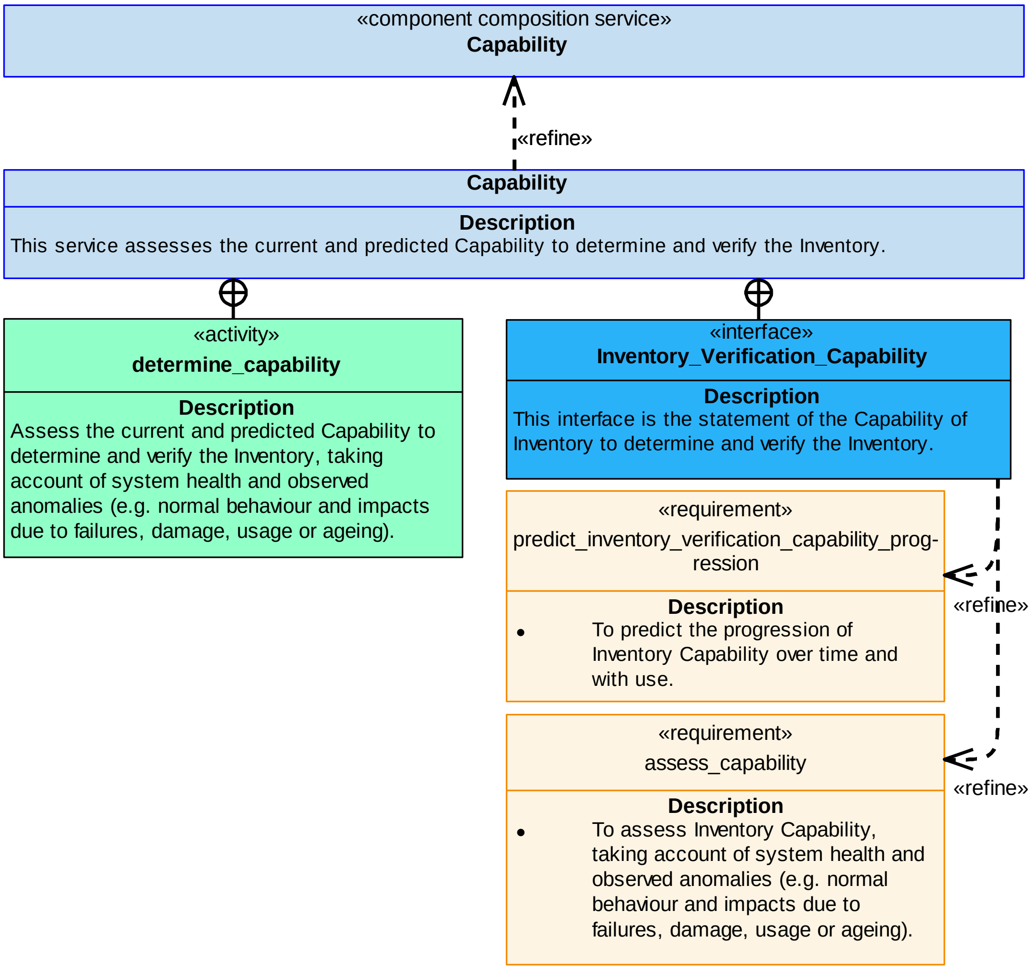

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to determine and verify the Inventory.
«activity»
determine_capability
Description
Assess the current and predicted Capability to determine and verify the Inventory, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Inventory_Verification_Capability
Description
This interface is the statement of the Capability of Inventory to determine and verify the Inventory.
«requirement»
predict_inventory_verification_capability_progression
Description
To predict the progression of Inventory Capability over time and with use.
«refine»
«requirement»
assess_capability
Description
To assess Inventory Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 534: Capability Service Policy

Capability

This service assesses the current and predicted Capability to determine and verify the Inventory.

<u>Interface</u>

Inventory_Verification_Capability

This interface is the statement of the Capability of Inventory to determine and verify the Inventory.

<u>Attributes</u>

inventory_determination_capability	The determination and verification of the Inventory that can be provided.
temporal_information	Information covering timing of the Capability, such as for how long the capability is likely to exist.

Activity

determine_capability

Assess the current and predicted Capability to determine and verify the Inventory, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.30.7.1.8 Capability_Evidence

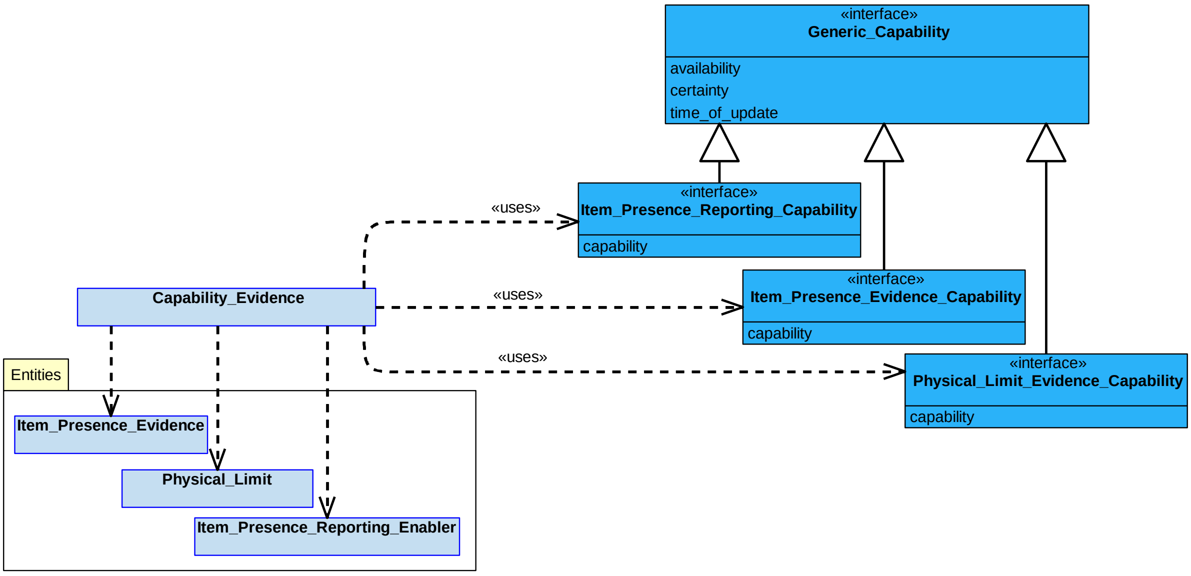


Figure 535: Capability_Evidence Service Definition

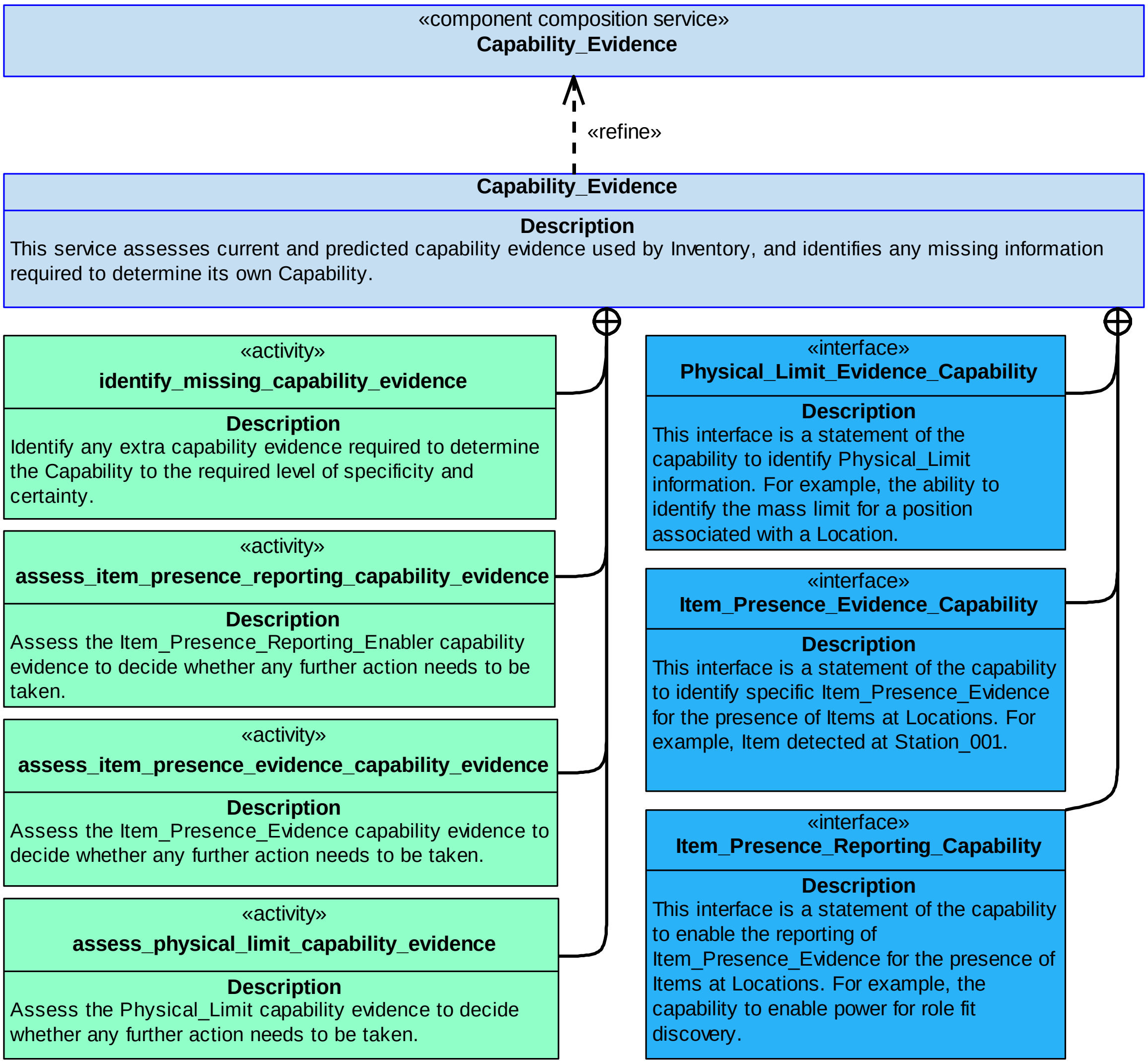

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service assesses current and predicted capability evidence used by Inventory, and identifies any missing information required to determine its own Capability.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«activity»
assess_item_presence_reporting_capability_evidence
Description
Assess the Item_Presence_Reporting_Enabler capability evidence to decide whether any further action needs to be taken.
«activity»
assess_item_presence_evidence_capability_evidence
Description
Assess the Item_Presence_Evidence capability evidence to decide whether any further action needs to be taken.
«activity»
assess_physical_limit_capability_evidence
Description
Assess the Physical_Limit capability evidence to decide whether any further action needs to be taken.
«interface»
Physical_Limit_Evidence_Capability
Description
This interface is a statement of the capability to identify Physical_Limit information. For example, the ability to identify the mass limit for a position associated with a Location.
«interface»
Item_Presence_Evidence_Capability
Description
This interface is a statement of the capability to identify specific Item_Presence_Evidence for the presence of Items at Locations. For example, Item detected at Station_001.
«interface»
Item_Presence_Reporting_Capability
Description
This interface is a statement of the capability to enable the reporting of Item_Presence_Evidence for the presence of Items at Locations. For example, the capability to enable power for role fit discovery.

Figure 536: Capability_Evidence Service Policy

Capability_Evidence

This service assesses current and predicted capability evidence used by Inventory, and identifies any missing information required to determine its own Capability.

Interfaces

Item_Presence_Reporting_Capability

This interface is a statement of the capability to enable the reporting of Item_Presence_Evidence for the presence of Items at Locations. For example, the capability to enable power for role fit discovery.

Attribute

capability	The capability to enable the reporting of Item_Presence_Evidence.

Item_Presence_Evidence_Capability

This interface is a statement of the capability to identify specific Item_Presence_Evidence for the presence of Items at Locations. For example, Item detected at Station_001.

Attribute

capability	The capability to identify Item_Presence_Evidence.

Physical_Limit_Evidence_Capability

This interface is a statement of the capability to identify Physical_Limit information. For example, the ability to identify the mass limit for a position associated with a Location.

Attribute

capability	The capability to identify evidence on Physical_Limits information.

Activities

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

assess_item_presence_reporting_capability_evidence

Assess the Item_Presence_Reporting_Enabler capability evidence to decide whether any further action needs to be taken.

assess_item_presence_evidence_capability_evidence

Assess the Item_Presence_Evidence capability evidence to decide whether any further action needs to be taken.

assess_physical_limit_capability_evidence

Assess the Physical_Limit capability evidence to decide whether any further action needs to be taken.

5.4.2.30.7.2 Service Dependencies

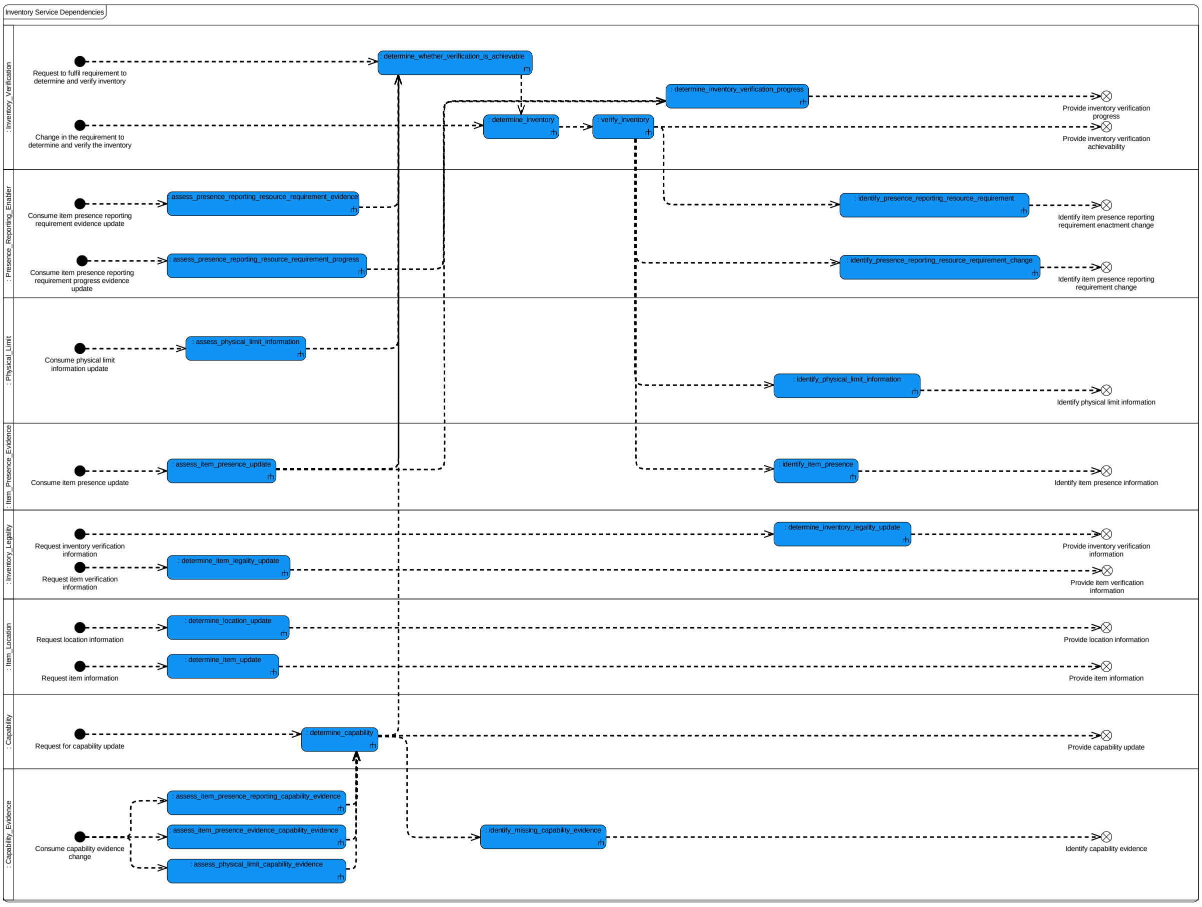

Inventory Service Dependencies
: Inventory_Verification
Request to fulfil requirement to determine and verify inventory
: determine_whether_verification_is_achievable
Change in the requirement to determine and verify the inventory
: determine_inventory
: verify_inventory
: determine_inventory_verification_progress
Provide inventory verification progress
Provide inventory verification achievability
: Presence_Reporting_Enabler
Consume item presence reporting requirement evidence update
: assess_presence_reporting_resource_requirement_evidence
Consume item presence reporting requirement progress evidence update
: assess_presence_reporting_resource_requirement_progress
: identify_presence_reporting_resource_requirement
Identify item presence reporting requirement enactment change
: identify_presence_reporting_resource_requirement_change
Identify item presence reporting requirement change
: Physical_Limit
Consume physical limit information update
: assess_physical_limit_information
: identify_physical_limit_information
Identify physical limit information
: Item_Presence_Evidence
Consume item presence update
: assess_item_presence_update
: identify_item_presence
Identify item presence information
: Inventory_Legality
Request inventory verification information
: determine_inventory_legality_update
Provide inventory verification information
Request item verification information
: determine_item_legality_update
Provide item verification information
: Item_Location
Request location information
: determine_location_update
Provide location information
Request item information
: determine_item_update
Provide item information
: Capability
Request for capability update
: determine_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_item_presence_reporting_capability_evidence
: assess_item_presence_evidence_capability_evidence
: assess_physical_limit_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 537: Inventory Service Dependencies

5.4.2.31 Jettison

5.4.2.31.1 Role

The role of Jettison is to coordinate the jettison of physical items from the platform.

5.4.2.31.2 Overview

Control Architecture

Jettison is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Jettison will identify the Jettison_Solution needed to meet the Requirements of a particular situation, including any Pre-conditions that need to be satisfied. The component will determine the Jettison_Package and coordinate the execution of its jettison at the appropriate time and in an appropriate area, requesting other components perform the necessary steps in jettisoning the package in the prescribed order.

Examples of Use

Jettison should be used where:

- Removable items such as stores, expendables or fuel need to be identified and coordinated for removal from the platform in order to reduce weight, improve safety or increase aerodynamic efficiency, etc.

5.4.2.31.3 Service Summary

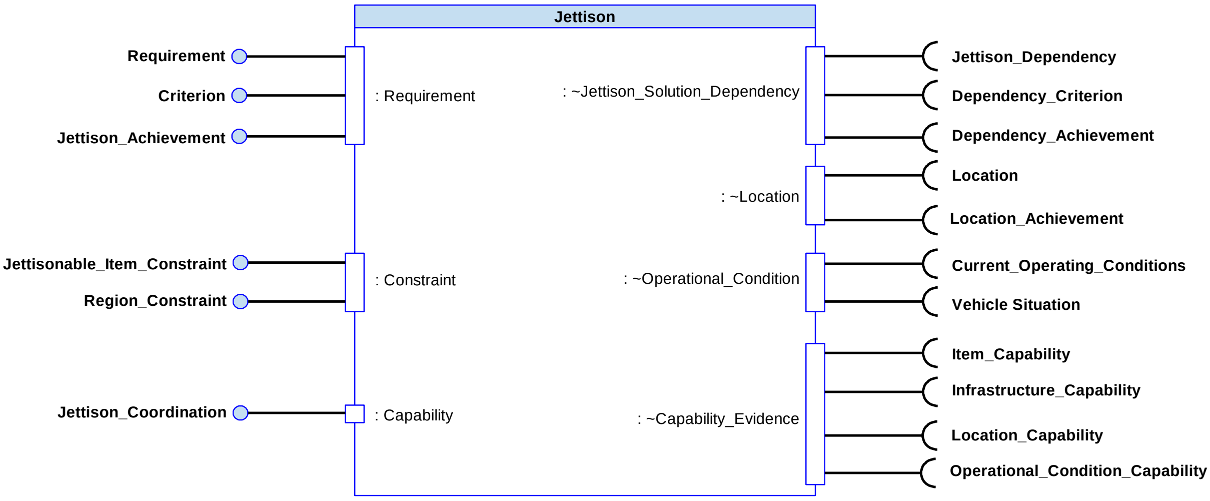


Figure 538: Jettison Service Summary

5.4.2.31.4 Responsibilities

capture_jettison_requirements

- To capture provided Requirements (e.g. mass to be removed) for jettison activities.

capture_measurement_criteria_for_jettison_actions

- To capture provided Measurement_Criterion/criteria for jettison activities.

capture_jettison_constraints

- To capture provided Constraints (e.g. stores that are not to be jettisoned) for jettison activities.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Jettison_Capability and Constraints.

determine_jettison_solution

- To determine a Jettison_Solution that meets the given Requirements and Constraints for jettison using available Jettison_Resources.

determine_predicted_quality_of_jettison_deliverables

- To determine the predicted quality of a proposed Jettison_Solution against given Measurement_Criterion/criteria.

identify_pre-conditions

- To identify Pre-conditions required to support the Jettison_Solution or a step of the jettison solution.

coordinate_jettison_dependencies

- To coordinate the execution of a Jettison_Solution.

identify_progress_of_jettison_solution

- To identify the progress of a Jettison_Solution against the Requirements.

determine_actual_quality_of_jettison_deliverables

- To determine the actual quality of the Jettison_Solution against the Measurement_Criterion/criteria.

capture_jettison_locations

- To capture locations in which jettison is allowable.

assess_jettison_capability

- To assess the Jettison_Capability to perform actions taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Jettison_Capability assessment.

predict_capability_progression

- To predict the progression of Jettison_Capability over time and with use.

5.4.2.31.5 Subject Matter Semantics

The subject matter of Jettison is physical items to be removed from the platform in order to maintain its safe and/or efficient operation.

Exclusions

The subject matter of Jettison does not include:

- The actual physical separation of items from the platform; it is only concerned with determining which items require jettison, the locations where jettison can be undertaken, and the coordination of the jettison.
- Determining additional items which need to be released at the same time (e.g. for aerodynamic or mass and balance reasons) and in what order to release items to make the jettison a valid release package.

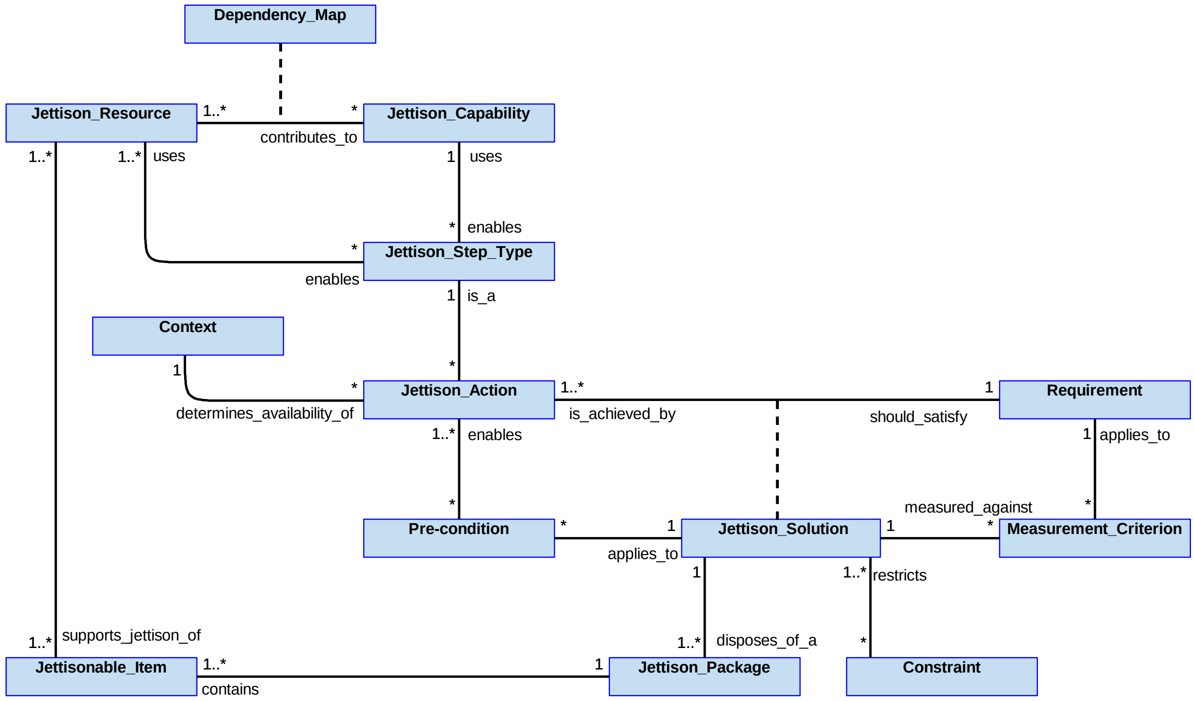


Figure 539: Jettison Semantics

5.4.2.31.5.1 Entities

Constraint

An externally imposed restriction, such as a restriction on the locations where the jettison may take place.

Context

Information that is required in order to determine a solution, e.g. the location of the vehicle and the operating conditions (weather, environment, etc.).

Dependency_Map

The range of jettison capabilities that the component is able to perform with its available Jettison_Resources.

Jettison_Action

An action that, when performed, contributes to the jettison of a physical item or fluid measure from the platform.

Jettison_Capability

The range of activities that can be carried out in order to perform a jettison.

Jettison_Package

The Jettisonable_Items selected to be jettisoned together as part of a Jettison_Solution.

Jettison_Resource

Something which can be instructed to carry out the activities required in order to perform a jettison, e.g. to release stores or remove fuel from the platform.

Jettison_Solution

A sequence of Jettison_Actions that are needed to meet the jettison Requirement.

Jettison_Step_Type

A type of action that can be carried out in order to contribute to performing a jettison, e.g. select the Jettison_Package or initiate jettison.

Jettisonable_Item

A physical object or fluid measure able to be jettisoned.

Measurement_Criterion

Something by which the quality or cost of the jettison will be measured, e.g. the mass of the package jettisoned.

Pre-condition

A condition that must be true before a jettison can take place, e.g. being in a safe envelope or an aperture being open.

Requirement

A requirement to achieve a goal for the removal of physical items from the platform, e.g. mass reduction or removal of unsafe items.

5.4.2.31.6 Design Rationale

5.4.2.31.6.1 Assumptions

- The jettison of items such as stores, expendables and fuel from the Exploiting Platform is a strategy that may be employed as, or as part of, a contingency action (e.g. in response to an emergency). The use of jettison as a contingency action includes the desire to reduce weight, dispose of unsafe items, or to improve the vehicle's aerodynamic efficiency.

- Rendering cryptographic key, algorithm or certificate material, or sensitive mission data held within stores, inaccessible may be a pre-condition or constraint before a physical jettison can take place.
- It is not expected that Jettison would reason about where a Jettison_Package will land; it would only be concerned with where the Jettison_Package is released from.
- A required Jettison_Package may form part of a larger release package to meet aerodynamic or mass and balance constraints.

5.4.2.31.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Jettison:

- Data Driving - Permitted jettison locations could be set via data as an alternative to identification by an authorised operator during the mission, and jettison procedures could also be set via data driving.

Extensions

- It is possible that extension components will be useful for the Jettison component in considering different types of jettisonable resources, e.g. for fuel or stores.

Exploitation Considerations

- It is not expected that Jettison would reason about where a Jettison_Package will land, but would instead use approved safe jettison regions defined by a third party.
- The Jettison_Solution may require the jettison of a certain quantity and type of stores (e.g. to reduce payload weight) or may require the jettison of a specific store (e.g. a safety jettison following malfunction) and may need to place requirements of either type on Jettison_Resources.

5.4.2.31.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

- If a failure occurs and jettison of fuel or stores is not performed (or the wrong item jettisoned) then this could result in a number of hazards, including (for an air vehicle):
    - Uncontrolled flight leading to an uncontrolled crash resulting in loss of the air vehicle and fatalities.
    - Uncontrolled fire leading to an uncontrolled crash resulting in loss of the air vehicle and fatalities.

There is only a need to perform a jettison if an initial failure has occurred, hence an indicative IDAL of DAL B is considered appropriate rather than DAL A.

This component will be involved in initiating actions by the Exploiting Platform to fulfil the jettison. However, interlocks and protection mechanisms that feed into other components will prevent them occurring when not required or incorrectly. For example:

- Interlocks can be used to prevent Stores Release, Release Effecting or external equipment jettisoning stores when not authorised (including not being in volumes of airspace where jettison is authorised).
- Stores Release will only jettison a package of stores that meets the appropriate Mass and Balance rules and may determine that other items need to be released first as part of the same package for safety reasons.

5.4.2.31.6.4 Security Considerations

The indicative security classification is SNEO

In order to plan a jettison, this component needs to know aspects of the stores configuration (jettisonability, weight, etc.) and status data and other relevant platform data from which it may be possible to deduce aspects of the combat effectiveness of the Exploiting Platform. These will have a security classification of SNEO and appropriate protective measures will be required to protect confidentiality. It is assumed that the integrity of a jettison request will be assured to prevent fraudulent jettison requests from reducing the capability of the platform (by removing weapons, fuel, etc.). Loss of the availability to perform timely jettison will adversely affect safety margins.

The component may be expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** to support non-repudiation of package selections and jettison commands.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **System Status and Monitoring** of selected jettison requests and packages, etc. Unexpected activity might be a sign of malicious intent.

The component may be expected to at least partially satisfy security enforcing functions by

- **Verifying Integrity of Data** for the jettison requests and packages (e.g. stores, mass and balance information).

5.4.2.31.7 Services

5.4.2.31.7.1 Service Definitions

5.4.2.31.7.1.1 Requirement

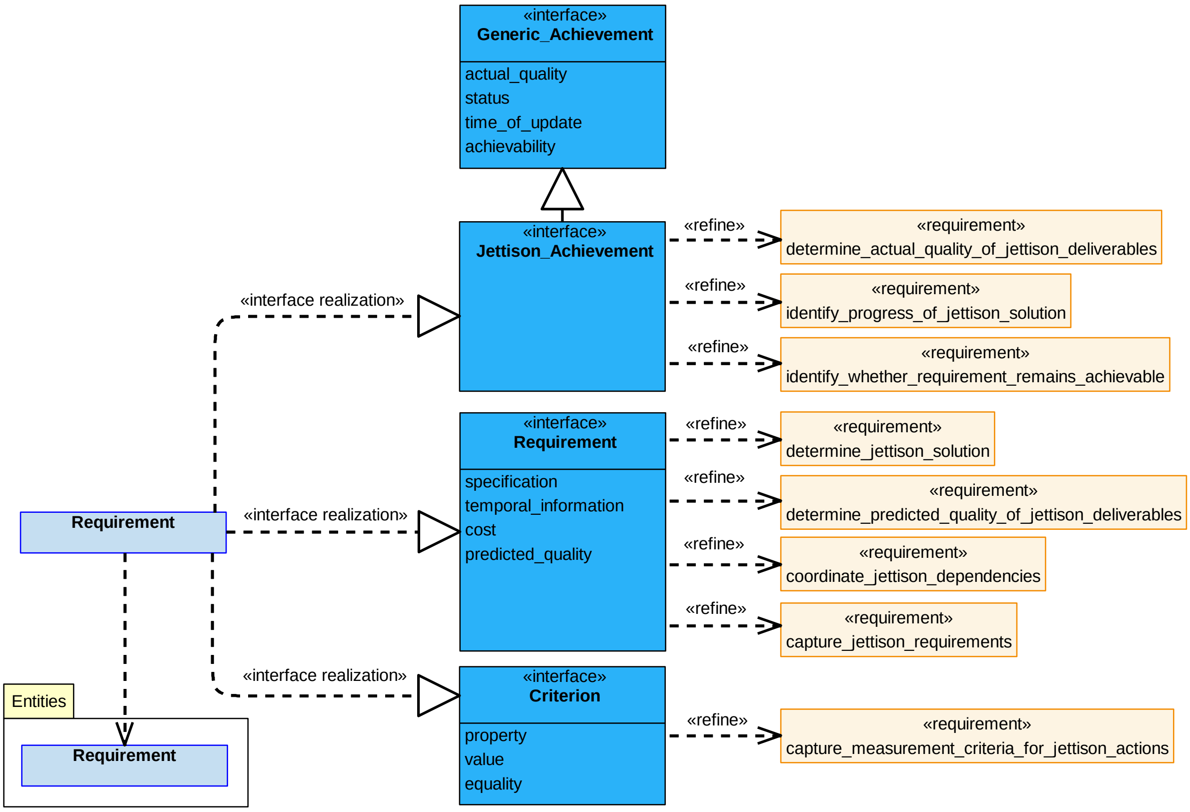


Figure 540: Requirement Service Definition

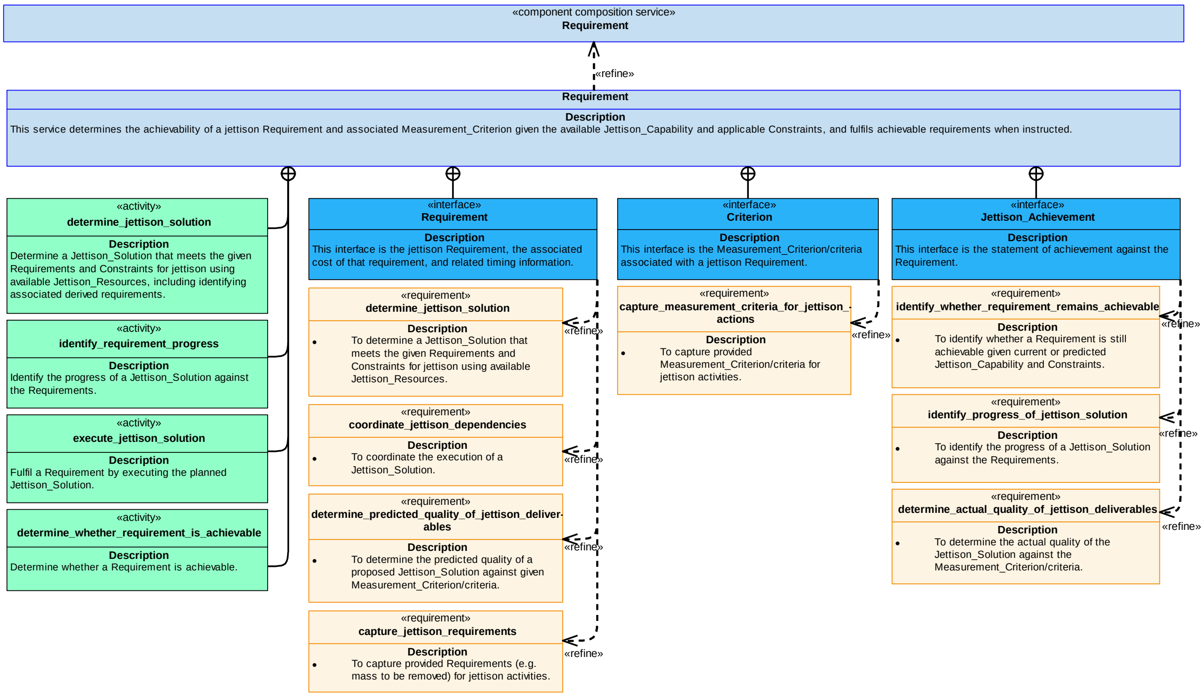

«component composition service»
Requirement
«refine»
Requirement
Description
This service determines the achievability of a jettison Requirement and associated Measurement_Criterion given the available Jettison_Capability and applicable Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_jettison_solution
Description
Determine a Jettison_Solution that meets the given Requirements and Constraints for jettison using available Jettison_Resources, including identifying associated derived requirements.
«activity»
identify_requirement_progress
Description
Identify the progress of a Jettison_Solution against the Requirements.
«activity»
execute_jettison_solution
Description
Fulfil a Requirement by executing the planned Jettison_Solution.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether a Requirement is achievable.
«interface»
Requirement
Description
This interface is the jettison Requirement, the associated cost of that requirement, and related timing information.
«requirement»
determine_jettison_solution
Description
To determine a Jettison_Solution that meets the given Requirements and Constraints for jettison using available Jettison_Resources.
«requirement»
coordinate_jettison_dependencies
Description
To coordinate the execution of a Jettison_Solution.
«requirement»
determine_predicted_quality_of_jettison_deliverables
Description
To determine the predicted quality of a proposed Jettison_Solution against given Measurement_Criterion/criteria.
«requirement»
capture_jettison_requirements
Description
To capture provided Requirements (e.g. mass to be removed) for jettison activities.
«interface»
Criterion
Description
This interface is the Measurement_Criterion/criteria associated with a jettison Requirement.
«requirement»
capture_measurement_criteria_for_jettison_actions
Description
To capture provided Measurement_Criterion/criteria for jettison activities.
«interface»
Jettison_Achievement
Description
This interface is the statement of achievement against the Requirement.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Requirement is still achievable given current or predicted Jettison_Capability and Constraints.
«requirement»
identify_progress_of_jettison_solution
Description
To identify the progress of a Jettison_Solution against the Requirements.
«requirement»
determine_actual_quality_of_jettison_deliverables
Description
To determine the actual quality of the Jettison_Solution against the Measurement_Criterion/criteria.
«refine»

Figure 541: Requirement Service Policy

Requirement

This service determines the achievability of a jettison Requirement and associated Measurement_Criterion given the available Jettison_Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Requirement

This interface is the jettison Requirement, the associated cost of that requirement, and related timing information.

Attributes

specification	The definition of the jettison Requirement, e.g. to remove mass from the Exploiting Platform.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the Jettison_Solution, e.g. resources used or time taken.
predicted_quality	How well the planned Jettison_Solution is predicted to satisfy the Requirement.

Criterion

This interface is the Measurement_Criterion/criteria associated with a jettison Requirement.

Attributes

property	The property to be measured, e.g. mass of packages.

value	The measured value of the property, e.g. 100kg.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Jettison_Achievement

This interface is the statement of achievement against the Requirement.

Activities

identify_requirement_progress

Identify the progress of a Jettison_Solution against the Requirements.

execute_jettison_solution

Fulfil a Requirement by executing the planned Jettison_Solution.

determine_jettison_solution

Determine a Jettison_Solution that meets the given Requirements and Constraints for jettison using available Jettison_Resources, including identifying associated derived requirements.

determine_whether_requirement_is_achievable

Determine whether a Requirement is achievable.

5.4.2.31.7.1.2 Jettison_Solution_Dependency

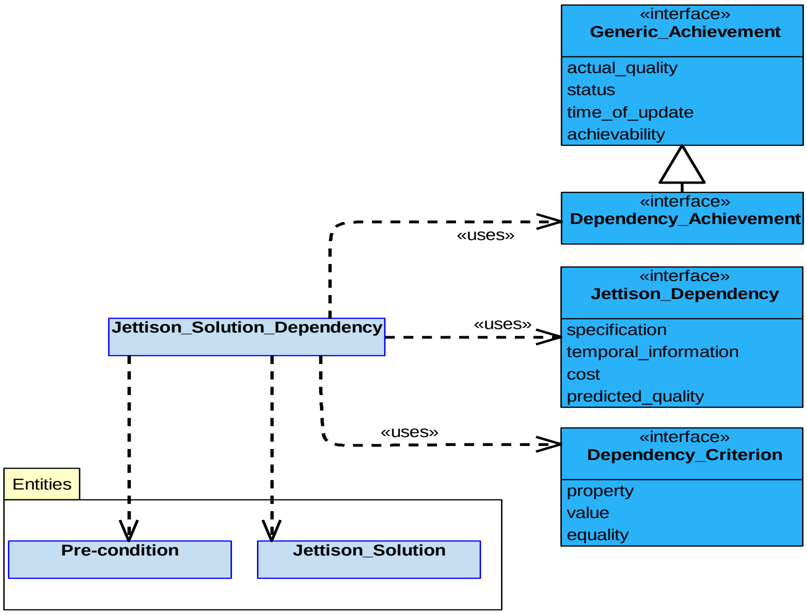


Figure 542: Jettison_Solution_Dependency Service Definition

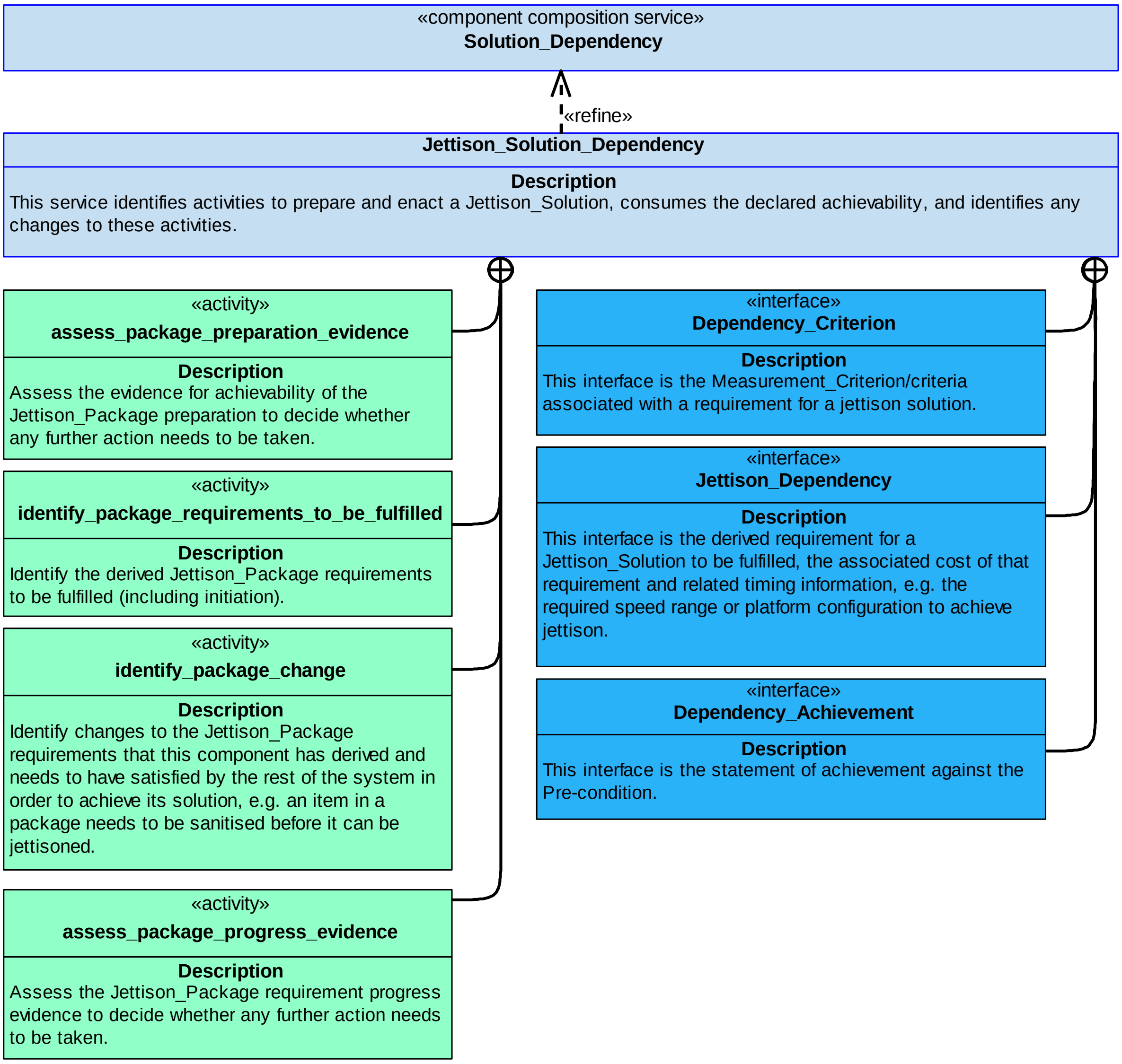

«component composition service»
Solution_Dependency
«refine»
Jettison_Solution_Dependency
Description
This service identifies activities to prepare and enact a Jettison_Solution, consumes the declared achievability, and identifies any changes to these activities.
«activity»
assess_package_preparation_evidence
Description
Assess the evidence for achievability of the Jettison_Package preparation to decide whether any further action needs to be taken.
«activity»
identify_package_requirements_to_be_fulfilled
Description
Identify the derived Jettison_Package requirements to be fulfilled (including initiation).
«activity»
identify_package_change
Description
Identify changes to the Jettison_Package requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution, e.g. an item in a package needs to be sanitised before it can be jettisoned.
«activity»
assess_package_progress_evidence
Description
Assess the Jettison_Package requirement progress evidence to decide whether any further action needs to be taken.
«interface»
Dependency_Criterion
Description
This interface is the Measurement_Criterion/criteria associated with a requirement for a jettison solution.
«interface»
Jettison_Dependency
Description
This interface is the derived requirement for a Jettison_Solution to be fulfilled, the associated cost of that requirement and related timing information, e.g. the required speed range or platform configuration to achieve jettison.
«interface»
Dependency_Achievement
Description
This interface is the statement of achievement against the Pre-condition.

Figure 543: Jettison_Solution_Dependency Service Policy

Jettison_Solution_Dependency

This service identifies activities to prepare and enact a Jettison_Solution, consumes the declared achievability, and identifies any changes to these activities.

Interfaces

Jettison_Dependency

This interface is the derived requirement for a Jettison_Solution to be fulfilled, the associated cost of that requirement and related timing information, e.g. the required speed range or platform configuration to achieve jettison.

Attributes

specification	The definition of the derived jettison solution requirement, e.g. prepare the package for jettison or arrange platform into the suitable configuration.

temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used or time taken.
predicted_quality	How well the planned solution is predicted to satisfy the derived requirement.

Dependency_Criterion

This interface is the Measurement_Criterion/criteria associated with a requirement for a jettison solution.

Attributes

property	The property to be measured, e.g. number of packages.
value	The measured value of the property, e.g. 2 packages.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Dependency_Achievement

This interface is the statement of achievement against the Pre-condition.

Activities

assess_package_preparation_evidence

Assess the evidence for achievability of the Jettison_Package preparation to decide whether any further action needs to be taken.

assess_package_progress_evidence

Assess the Jettison_Package requirement progress evidence to decide whether any further action needs to be taken.

identify_package_requirements_to_be_fulfilled

Identify the derived Jettison_Package requirements to be fulfilled (including initiation).

identify_package_change

Identify changes to the Jettison_Package requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution, e.g. an item in a package needs to be sanitised before it can be jettisoned.

5.4.2.31.7.1.3 Location

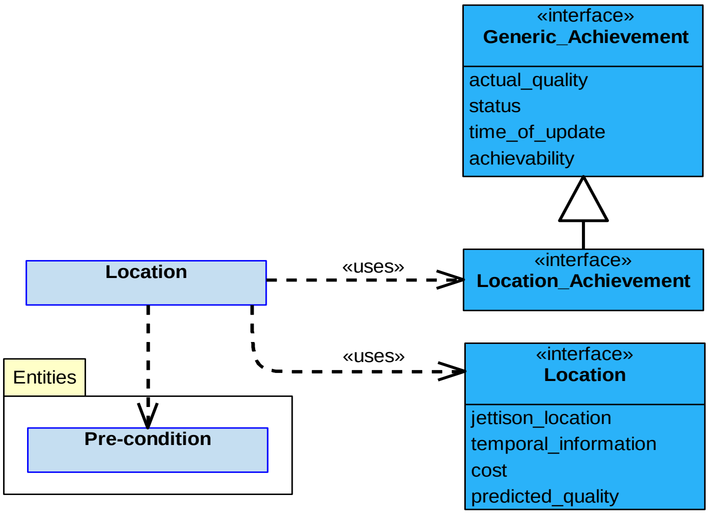
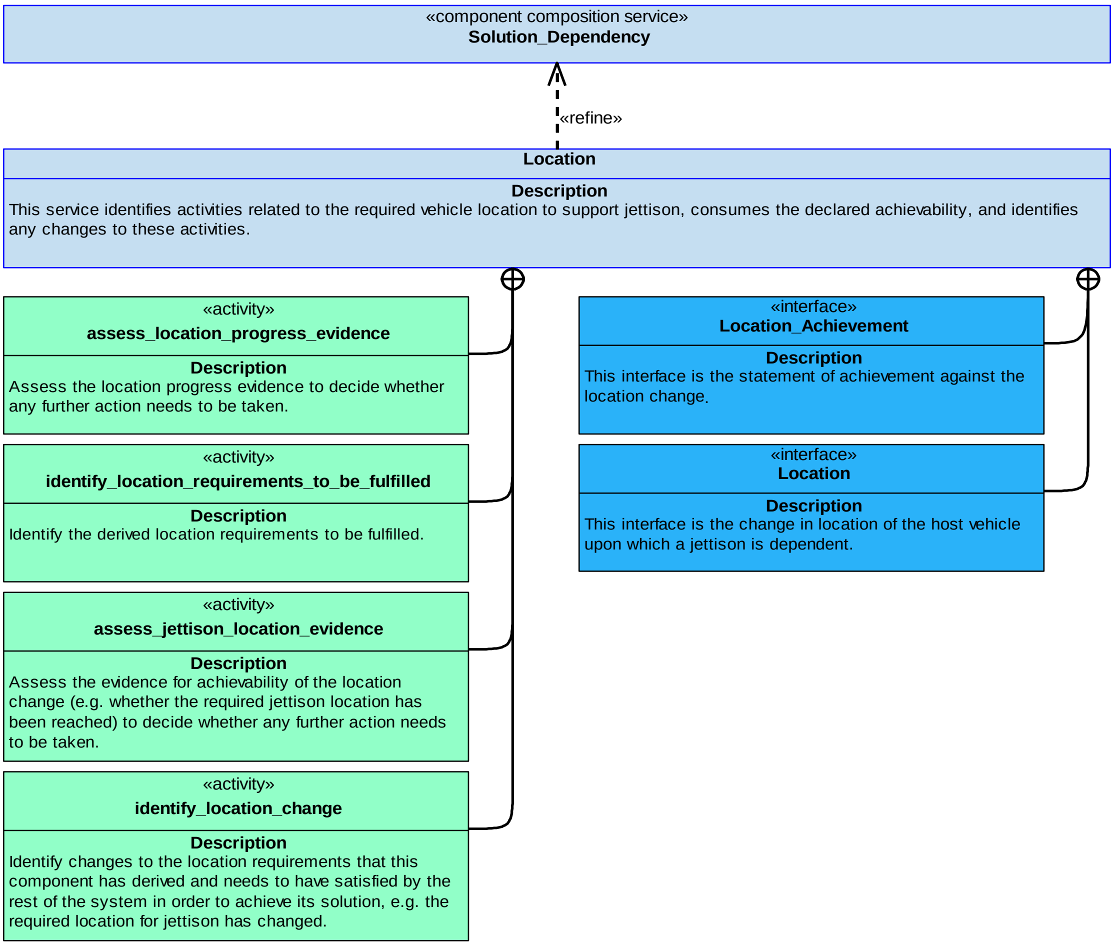


Figure 544: Location Service Definition



This service identifies activities related to the required vehicle location to support jettison, consumes the declared achievability, and identifies any changes to these activities.

Assess the location progress evidence to decide whether any further action needs to be taken.

Identify the derived location requirements to be fulfilled.

Assess the evidence for achievability of the location change (e.g. whether the required jettison location has been reached) to decide whether any further action needs to be taken.

Identify changes to the location requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution, e.g. the required location for jettison has changed.

This interface is the statement of achievement against the location change.

This interface is the change in location of the host vehicle upon which a jettison is dependent.

Figure 545: Location Service Policy

Location

This service identifies activities related to the required vehicle location to support jettison, consumes the declared achievability, and identifies any changes to these activities.

Interfaces

Location

This interface is the change in location of the host vehicle upon which a jettison is dependent.

Attributes

jettison_location	The location to be navigated to.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used, time taken.
predicted_quality	How well the planned location solution is predicted to satisfy the requirement.

Location_Achievement

This interface is the statement of achievement against the location change.

Activities

assess_jettison_location_evidence

Assess the evidence for achievability of the location change (e.g. whether the required jettison location has been reached) to decide whether any further action needs to be taken.

assess_location_progress_evidence

Assess the location progress evidence to decide whether any further action needs to be taken.

identify_location_change

Identify changes to the location requirements that this component has derived and needs to have satisfied by the rest of the system in order to achieve its solution, e.g. the required location for jettison has changed.

identify_location_requirements_to_be_fulfilled

Identify the derived location requirements to be fulfilled.

5.4.2.31.7.1.4 Operational_Condition

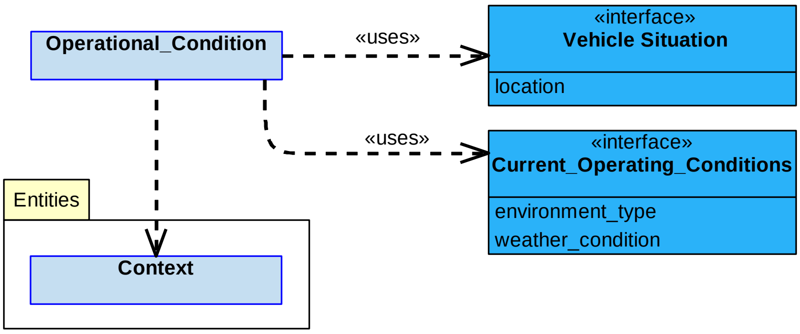


Figure 546: Operational_Condition Service Definition

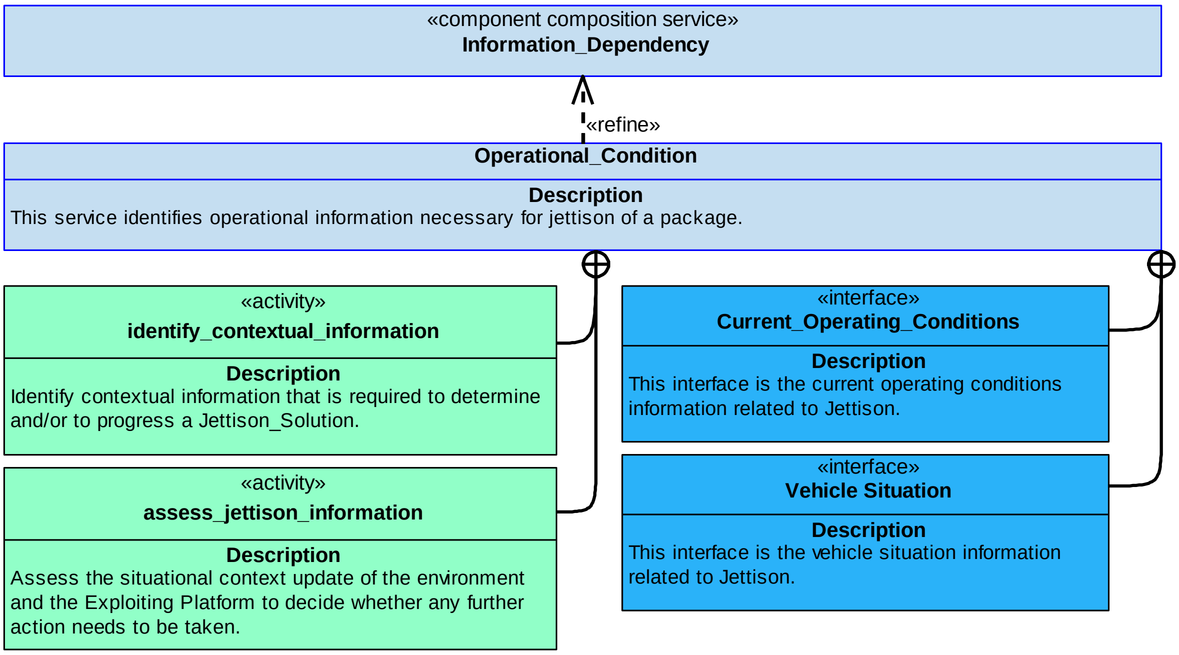

«component composition service»
Information_Dependency
«refine»
Operational_Condition
Description
This service identifies operational information necessary for jettison of a package.
«activity»
identify_contextual_information
Description
Identify contextual information that is required to determine and/or to progress a Jettison_Solution.
«activity»
assess_jettison_information
Description
Assess the situational context update of the environment and the Exploiting Platform to decide whether any further action needs to be taken.
«interface»
Current_Operating_Conditions
Description
This interface is the current operating conditions information related to Jettison.
«interface»
Vehicle Situation
Description
This interface is the vehicle situation information related to Jettison.

Figure 547: Operational_Condition Service Policy

Operational_Condition

This service identifies operational information necessary for jettison of a package.

Interfaces

Vehicle Situation

This interface is the vehicle situation information related to Jettison.

Attribute

location	The location of the vehicle.

Current_Operating_Conditions

This interface is the current operating conditions information related to Jettison.

Attributes

environment_type	Data regarding geographical information and environmental structures that the Exploiting Platform is currently operating in.
weather_condition	The state of a type of atmospheric condition at a given time and place.

Activities

assess_jettison_information

Assess the situational context update of the environment and the Exploiting Platform to decide whether any further action needs to be taken.

identify_contextual_information

Identify contextual information that is required to determine and/or to progress a Jettison_Solution.

5.4.2.31.7.1.5 Constraint

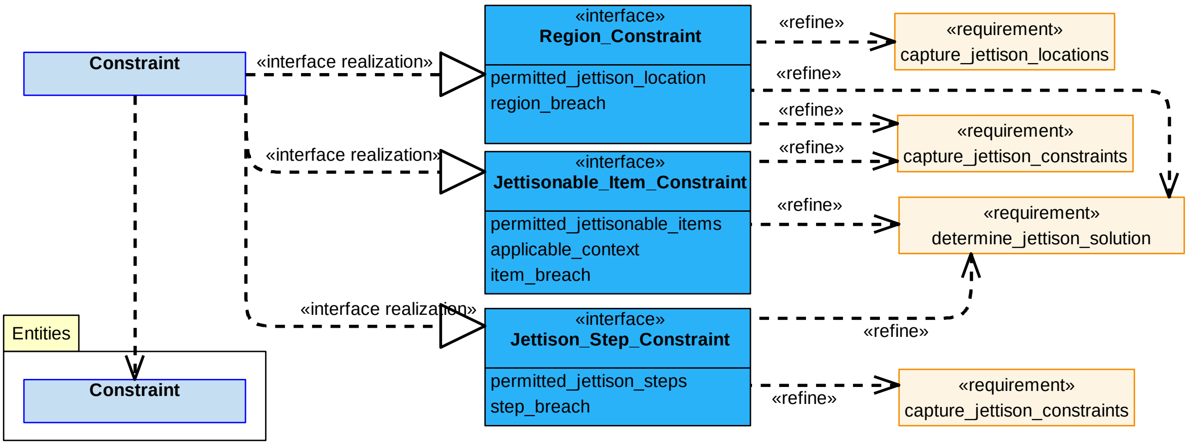
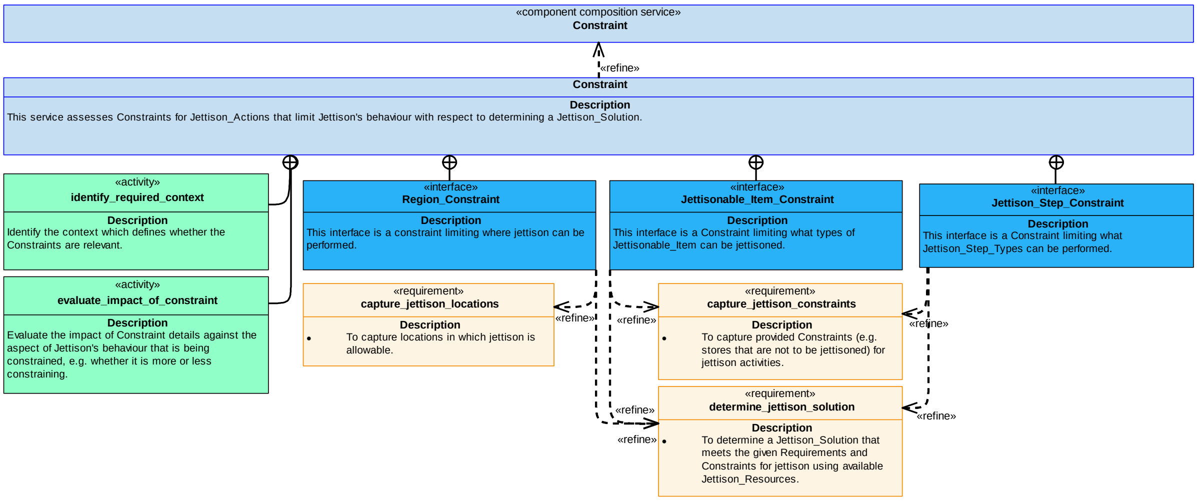


Figure 548: Constraint Service Definition



This service assesses Constraints for Jettison_Actions that limit Jettison's behaviour with respect to determining a Jettison_Solution.

Identify the context which defines whether the Constraints are relevant.

Evaluate the impact of Constraint details against the aspect of Jettison's behaviour that is being constrained, e.g. whether it is more or less constraining.

This interface is a constraint limiting where jettison can be performed.

This interface is a Constraint limiting what types of Jettisonable_Item can be jettisoned.

This interface is a Constraint limiting what Jettison_Step_Types can be performed.

- To capture locations in which jettison is allowable.

- To capture provided Constraints (e.g. stores that are not to be jettisoned) for jettison activities.

- To determine a Jettison_Solution that meets the given Requirements and Constraints for jettison using available Jettison_Resources.

Figure 549: Constraint Service Policy

Constraint

This service assesses Constraints for Jettison_Actions that limit Jettison's behaviour with respect to determining a Jettison_Solution.

Interfaces

Region_Constraint

This interface is a constraint limiting where jettison can be performed.

Attributes

permitted_jettison_location	The locations within which Jettison_Step_Types are permitted.
region_breach	A statement that the region constraint has been breached.

Jettisonable_Item_Constraint

This interface is a Constraint limiting what types of Jettisonable_Item can be jettisoned.

Attributes

permitted_jettisonable_items	The items or fluids that are permitted to be jettisoned.
applicable_context	The context in which the Constraint is applicable.
item_breach	A statement that the item Constraint has been breached.

Jettison_Step_Constraint

This interface is a Constraint limiting what Jettison_Step_Types can be performed.

Attributes

permitted_jettison_steps	The Jettison_Step_Types that are permitted.
step_breach	A statement that the jettison step constraint has been breached.

Activities

identify_required_context

Identify the context which defines whether the Constraints are relevant.

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of Jettison's behaviour that is being constrained, e.g. whether it is more or less constraining.

5.4.2.31.7.1.6 Capability

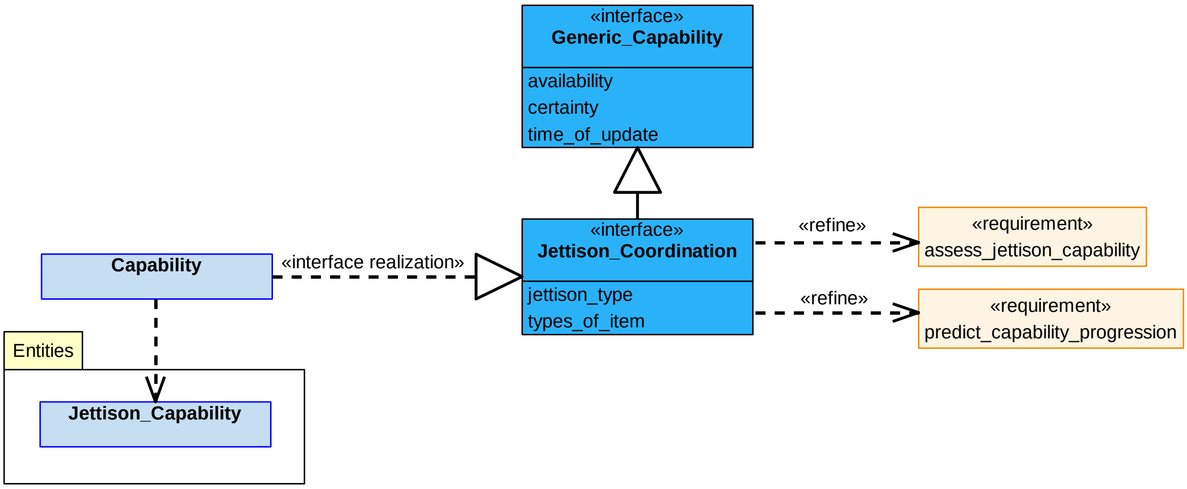
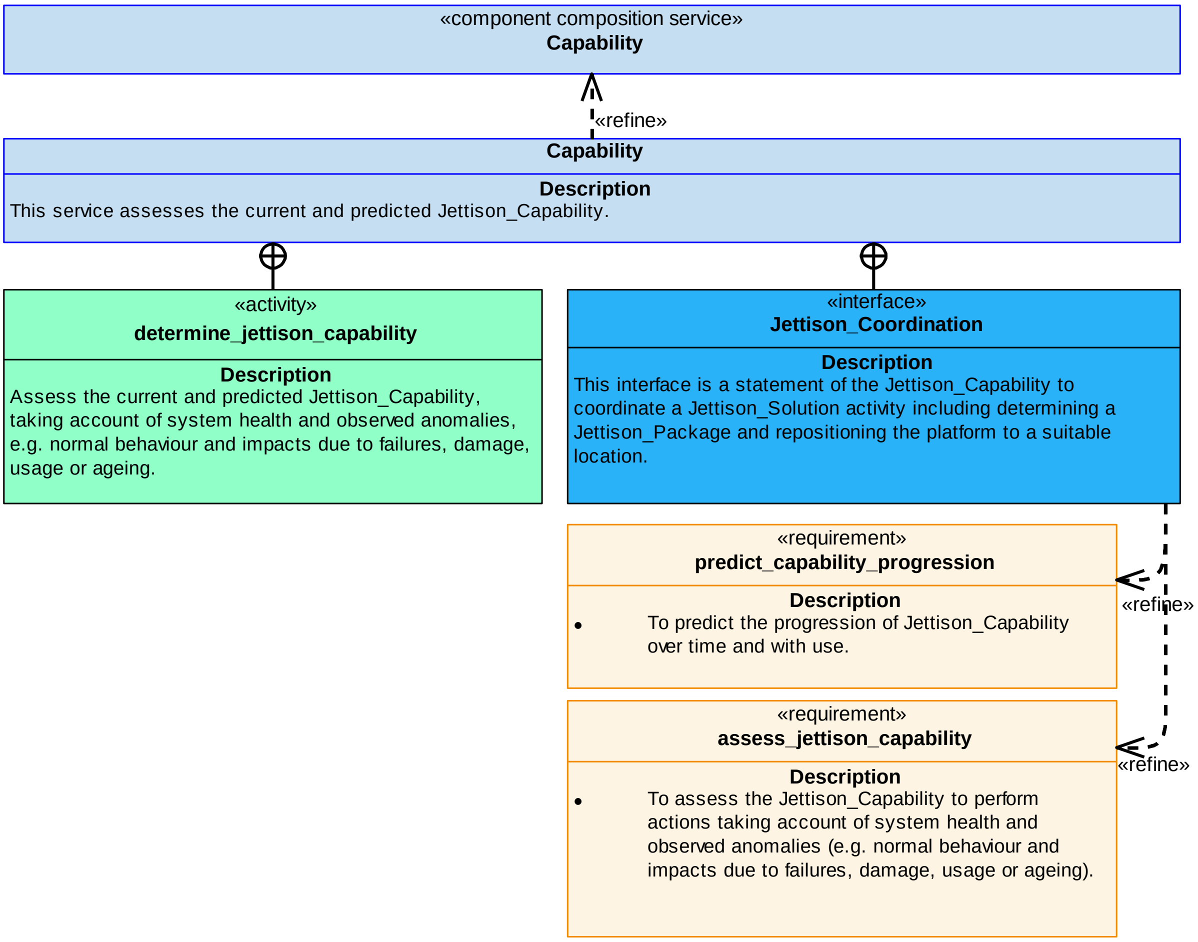


Figure 550: Capability Service Definition



This service assesses the current and predicted Jettison_Capability.

Assess the current and predicted Jettison_Capability, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.

This interface is a statement of the Jettison_Capability to coordinate a Jettison_Solution activity including determining a Jettison_Package and repositioning the platform to a suitable location.

- To predict the progression of Jettison_Capability over time and with use.

- To assess the Jettison_Capability to perform actions taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Figure 551: Capability Service Policy

Capability

This service assesses the current and predicted Jettison_Capability.

Interface

Jettison_Coordination

This interface is a statement of the Jettison_Capability to coordinate a Jettison_Solution activity including determining a Jettison_Package and repositioning the platform to a suitable location.

Attributes

jettison_type	The type of jettison that can be coordinated, e.g. mass reduction.
types_of_item	The types of Jettisonable_Item that can be within the Jettison_Package.

Activity

determine_jettison_capability

Assess the current and predicted Jettison_Capability, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.

5.4.2.31.7.1.7 Capability_Evidence

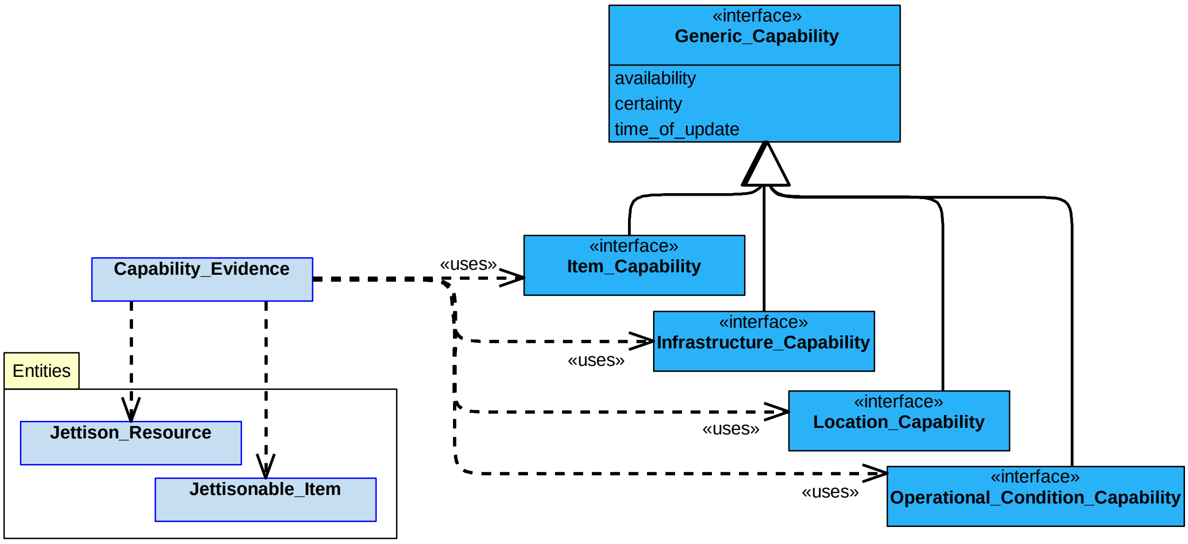
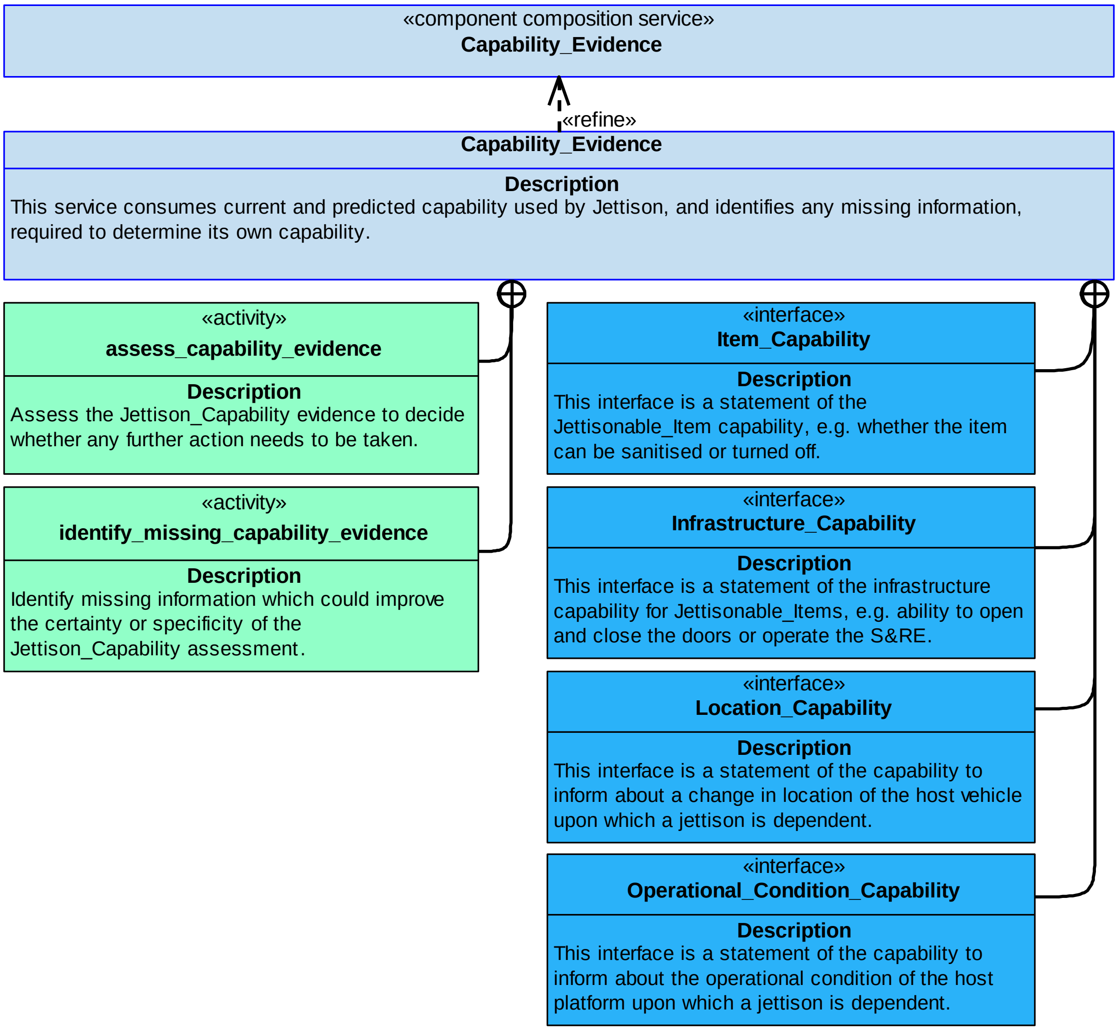


Figure 552: Capability_Evidence Service Definition



«component composition service»
Capability_Evidence

«refine»

Capability_Evidence

Description
This service consumes current and predicted capability used by Jettison, and identifies any missing information, required to determine its own capability.

«activity»
assess_capability_evidence

Description
Assess the Jettison_Capability evidence to decide whether any further action needs to be taken.

«activity»
identify_missing_capability_evidence

Description
Identify missing information which could improve the certainty or specificity of the Jettison_Capability assessment.

«interface»
Item_Capability

Description
This interface is a statement of the Jettisonable_Item capability, e.g. whether the item can be sanitised or turned off.

«interface»
Infrastructure_Capability

Description
This interface is a statement of the infrastructure capability for Jettisonable_Items, e.g. ability to open and close the doors or operate the S&RE.

«interface»
Location_Capability

Description
This interface is a statement of the capability to inform about a change in location of the host vehicle upon which a jettison is dependent.

«interface»
Operational_Condition_Capability

Description
This interface is a statement of the capability to inform about the operational condition of the host platform upon which a jettison is dependent.

Figure 553: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability used by Jettison, and identifies any missing information, required to determine its own capability.

Interfaces

Item_Capability

This interface is a statement of the Jettisonable_Item capability, e.g. whether the item can be sanitised or turned off.

Infrastructure_Capability

This interface is a statement of the infrastructure capability for Jettisonable_Items, e.g. ability to open and close the doors or operate the S&RE.

Operational_Condition_Capability

This interface is a statement of the capability to inform about the operational condition of the host platform upon which a jettison is dependent.

Location_Capability

This interface is a statement of the capability to inform about a change in location of the host vehicle upon which a jettison is dependent.

Activities

assess_capability_evidence

Assess the Jettison_Capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify missing information which could improve the certainty or specificity of the Jettison_Capability assessment.

5.4.2.31.7.2 Service Dependencies

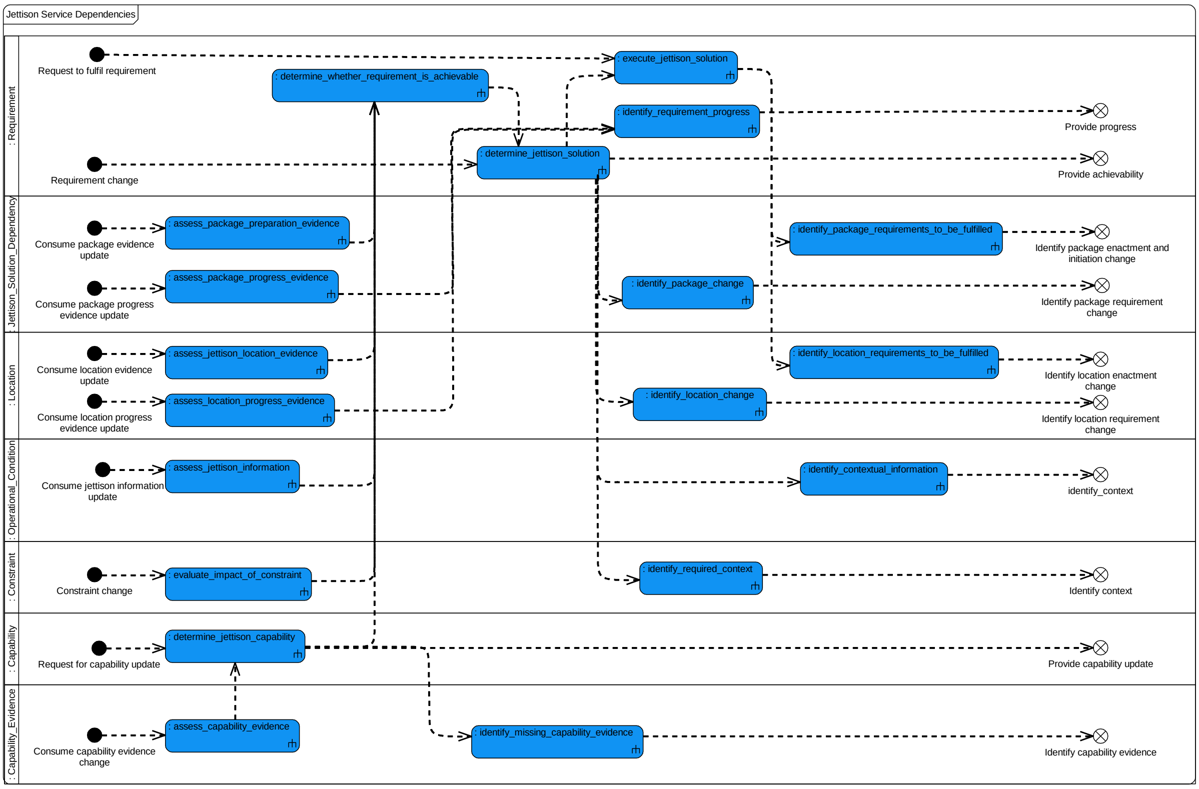

Jettison Service Dependencies
: Requirement
: Jettison_Solution_Dependency
: Location
: Operational_Condition
: Constraint
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_jettison_solution
: determine_whether_requirement_is_achievable
: identify_requirement_progress
Provide progress
: determine_jettison_solution
Requirement change
Provide achievability
Consume package evidence update
: assess_package_preparation_evidence
: identify_package_requirements_to_be_fulfilled
Identify package enactment and initiation change
Consume package progress evidence update
: assess_package_progress_evidence
: identify_package_change
Identify package requirement change
Consume location evidence update
: assess_jettison_location_evidence
: identify_location_requirements_to_be_fulfilled
Identify location enactment change
Consume location progress evidence update
: assess_location_progress_evidence
: identify_location_change
Identify location requirement change
Consume jettison information update
: assess_jettison_information
: identify_contextual_information
identify_context
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Request for capability update
: determine_jettison_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 554: Jettison Service Dependencies

5.4.2.32 Lights

5.4.2.32.1 Role

The role of Lights is to control the lighting on an Exploiting Platform.

5.4.2.32.2 Overview

Control Architecture

Lights is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Lights will identify the Lighting_Solution needed to satisfy the Requirements of a particular situation, taking into account any Pre-conditions and Constraints. The component will determine the best Light to achieve an Outcome which will then be evaluated against the original Lighting_Solution.

Examples of Use

Lights could be used where:

- There are requirements to have lights flashing at different frequencies in order to communicate.
- There are requirements to illuminate objects in variable lighting environments.

5.4.2.32.3 Service Summary

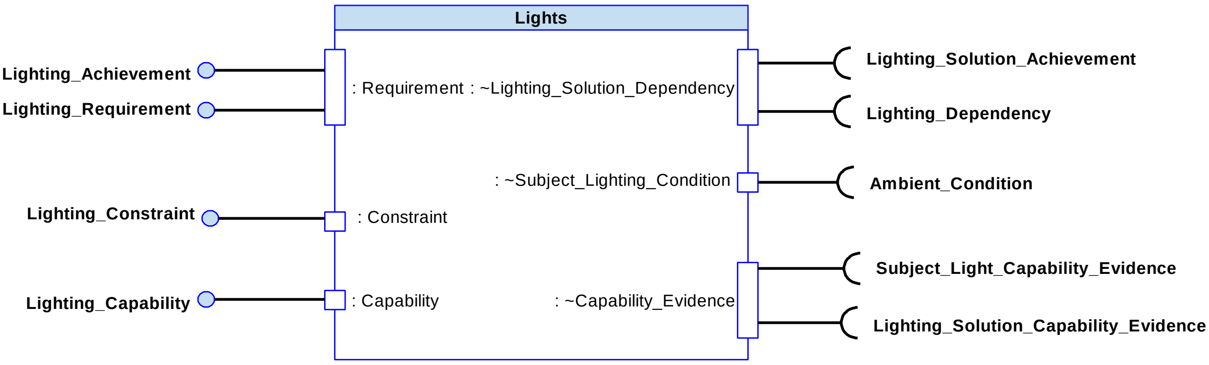

Lights
Lighting_Achievement
Lighting_Requirement
: Requirement : ~Lighting_Solution_Dependency
Lighting_Solution_Achievement
Lighting_Dependency
: ~Subject_Lighting_Condition
Ambient_Condition
Lighting_Constraint
: Constraint
Subject_Light_Capability_Evidence
Lighting_Capability
: Capability
: ~Capability_Evidence
Lighting_Solution_Capability_Evidence

Figure 555: Lights Service Summary

5.4.2.32.4 Responsibilities

capture_lighting_requirements

- To capture provided lighting Requirements.

capture_lighting_constraints

- To capture Constraints on any potential Lighting_Solution.

determine_lighting_solution

- To determine the available Lighting_Solution which best meets the Requirements and Constraints.

identify_lighting_solution_in_progress_remains_feasible

- To identify whether a Lighting_Solution in progress remains feasible given current resources.

implement_lighting_solution

- To control lighting in accordance with a planned Lighting_Solution.

assess_capability

- To assess the Capability to provide lighting taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the Lights Capability over time and with use.

5.4.2.32.5 Subject Matter Semantics

The subject matter of Lights is the resources that can be used to provide illumination.

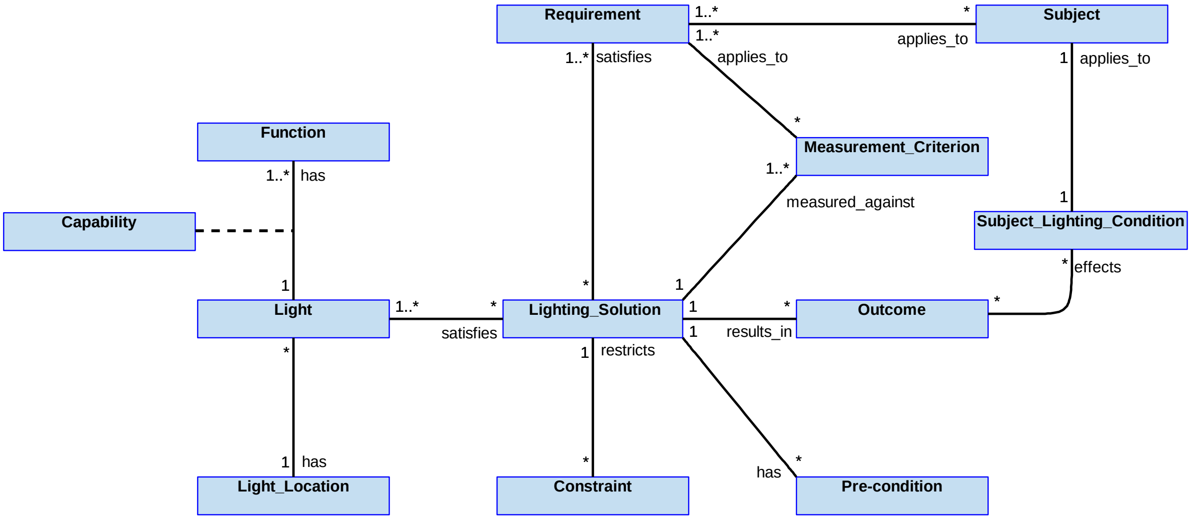


Figure 556: Lights Semantics

5.4.2.32.5.1 Entities

Capability

The capability of the component based upon the function of lights and their availability.

Constraint

An externally imposed restriction, e.g. a limit on the use or illumination level of a light.

Subject_Lighting_Condition

The condition of lighting on a subject, e.g. the level of ambient light or impact of lighting.

Function

The ability of a light to produce light types, colour of light, and angle of the beam, e.g. to produce visible red light with a 90 degree spread.

Light

An instance of a light source, e.g. a lamp or a source of chemical illumination such as a glow stick.

Light_Location

The location of a light on an Exploiting Platform, e.g. on the port wing.

Lighting_Solution

A solution to control the lights on an Exploiting Platform, e.g. turn on all the lights in cockpit or dim the landing lights. The solution will take into account light direction, its ability to move (e.g. on a turret) and intensity.

Measurement_Criterion

The criteria by which the quality or cost of lighting can be measured against.

Outcome

A change, or changes in Subject_Lighting_Condition which results from executing the planned solution.

Pre-condition

A condition that must be true, e.g. the Exploiting Platform is in a planned position.

Requirement

A requirement to achieve a light or lighting effect, e.g. provide a certain amount of lumens over a certain area.

Subject

The intended target for a light, e.g. a search area for a spotlight.

5.4.2.32.6 Design Rationale

5.4.2.32.6.1 Assumptions

- The lighting that is controlled includes searchlights and bay lighting/cockpit lighting as well as navigation lights, beacons and lights used for communication (such as landing lights).
- This component is not intended to cover small indicator lights (such as status indicators on role fit equipment or HMI instrument panels, including backlights).

- The majority of lighting plans for navigational lighting and beacons will be pre-defined in accordance with Environment Integration rules for signalling.
- External lighting can be controlled to support covert operations, through mission data and authorised operator command. The component will not have direct knowledge of EMCON rules.
- The component could hold pre-defined lighting configurations for an Exploiting Platforms external lighting and implement changes between them.

5.4.2.32.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Lights:

- Data Driving **-** To facilitate different pre-defined configurations between different operators.

5.4.2.32.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component may cause loss of external lighting, external lighting when not required or an incorrect combination of lights (where they are used for back-up communications). Whilst external lighting is used to mitigate aircraft collisions (particularly during ground operations) the failure of external lighting would not directly cause a catastrophic collision as other safety barriers are present (e.g. de-confliction by ATC or ACAS). Therefore, failure of this component is judged to be a ‘large reduction in safety margins’ (severity critical) which would require an indicative IDAL of DAL B.

5.4.2.32.6.4 Security Considerations

The indicative security classification is O.

This component is responsible for the control of lighting, including searchlights and navigation lights and beacons. Some lights and their usage will be subject to international requirements, although exemptions can apply. This component will have integrity and availability requirements appropriate to support use of the Exploiting Platform's lights around other vehicles, and whilst on a mission, unintended lighting activation/de-activation will affect the ability to act covertly.

The component may be expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to changes to lighting plans and protocols, etc.
- **Maintaining Audit Records** relating to use of lighting during a mission, including authorisation to turn off lighting required by regulations, etc.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is not expected to directly implement security enforcing functions.

5.4.2.32.7 Services

5.4.2.32.7.1 Service Definitions

5.4.2.32.7.1.1 Requirement

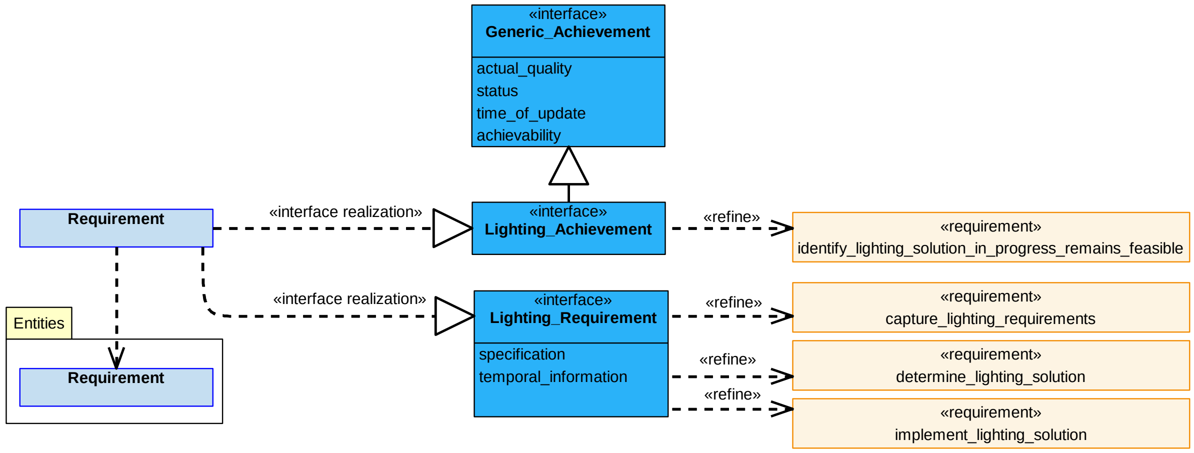
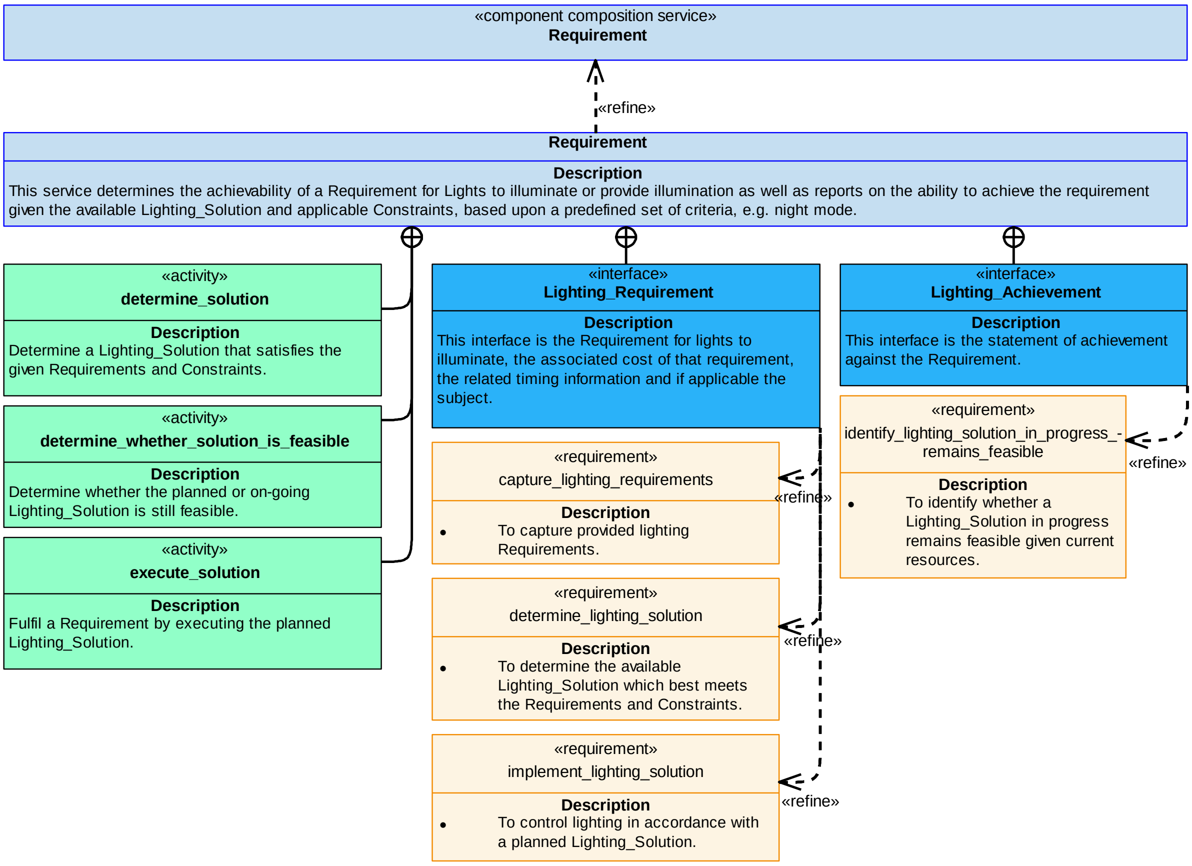


Figure 557: Requirement Service Definition



Requirement

Description
This service determines the achievability of a Requirement for Lights to illuminate or provide illumination as well as reports on the ability to achieve the requirement given the available Lighting_Solution and applicable Constraints, based upon a predefined set of criteria, e.g. night mode.

Description
Determine a Lighting_Solution that satisfies the given Requirements and Constraints.

Description
Determine whether the planned or on-going Lighting_Solution is still feasible.

Description
Fulfil a Requirement by executing the planned Lighting_Solution.

Description
This interface is the Requirement for lights to illuminate, the associated cost of that requirement, the related timing information and if applicable the subject.

Description
This interface is the statement of achievement against the Requirement.

Description
- To capture provided lighting Requirements.

Description
- To determine the available Lighting_Solution which best meets the Requirements and Constraints.

Description
- To control lighting in accordance with a planned Lighting_Solution.

Description
- To identify whether a Lighting_Solution in progress remains feasible given current resources.

Figure 558: Requirement Service Policy

Requirement

This service determines the achievability of a Requirement for Lights to illuminate or provide illumination as well as reports on the ability to achieve the requirement given the available Lighting_Solution and applicable Constraints, based upon a predefined set of criteria, e.g. night mode.

Interfaces

Lighting_Requirement

This interface is the Requirement for lights to illuminate, the associated cost of that requirement, the related timing information and if applicable the subject.

Attributes

specification	The request to provide or cease providing illumination.
temporal_information	Information covering timing, such as start and end times.

Lighting_Achievement

This interface is the statement of achievement against the Requirement.

Activities

determine_solution

Determine a Lighting_Solution that satisfies the given Requirements and Constraints.

execute_solution

Fulfil a Requirement by executing the planned Lighting_Solution.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Lighting_Solution is still feasible.

5.4.2.32.7.1.2 Lighting_Solution_Dependency

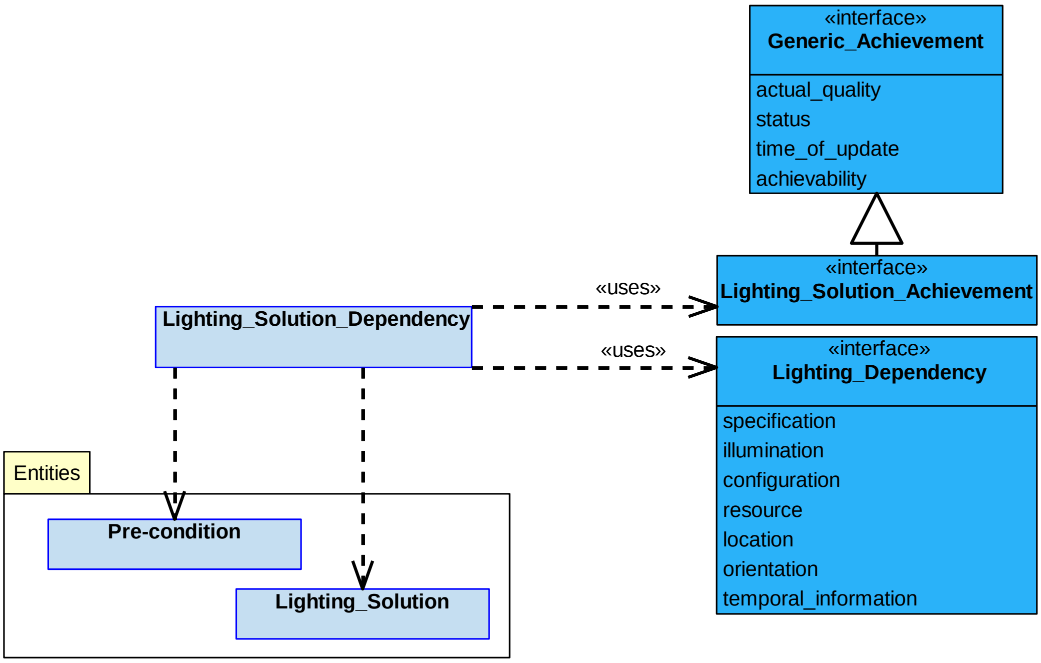


Figure 559: Lighting_Solution_Dependency Service Definition

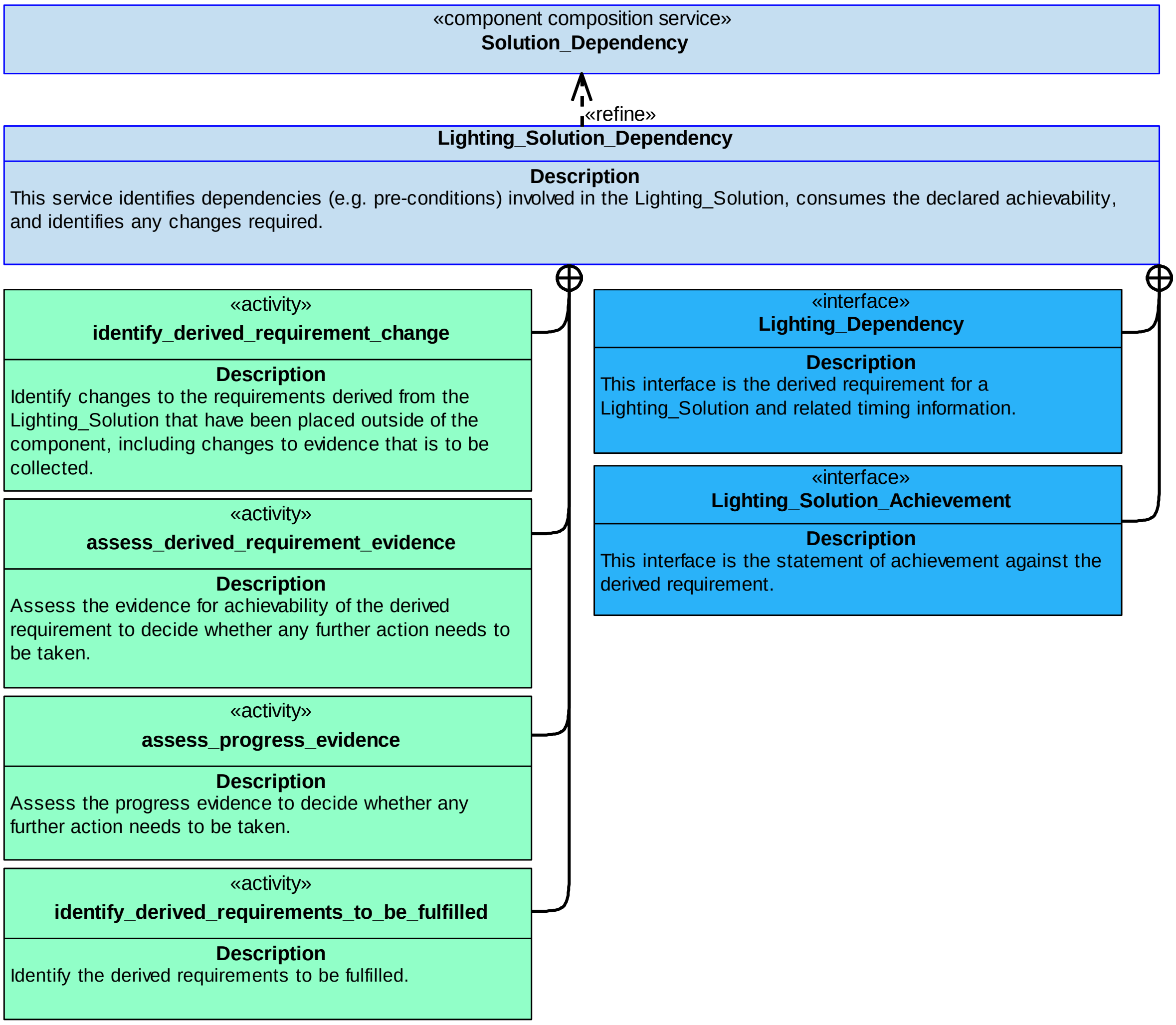

«component composition service»
Solution_Dependency
«refine»
Lighting_Solution_Dependency
Description
This service identifies dependencies (e.g. pre-conditions) involved in the Lighting_Solution, consumes the declared achievability, and identifies any changes required.
«activity»
identify_derived_requirement_change
Description
Identify changes to the requirements derived from the Lighting_Solution that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
assess_derived_requirement_evidence
Description
Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
identify_derived_requirements_to_be_fulfilled
Description
Identify the derived requirements to be fulfilled.
«interface»
Lighting_Dependency
Description
This interface is the derived requirement for a Lighting_Solution and related timing information.
«interface»
Lighting_Solution_Achievement
Description
This interface is the statement of achievement against the derived requirement.

Figure 560: Lighting_Solution_Dependency Service Policy

Lighting_Solution_Dependency

This service identifies dependencies (e.g. pre-conditions) involved in the Lighting_Solution, consumes the declared achievability, and identifies any changes required.

Interfaces

Lighting_Dependency

This interface is the derived requirement for a Lighting_Solution and related timing information.

Attributes

specification	The specification of the derived requirement e.g. for the Exploiting Platform to be in a particular orientation.
illumination	The illumination of one or more Lights to achieve a Lighting_Solution.
configuration	A particular configuration of the Exploiting Platform that is necessary to achieve the Lighting_Solution.
resource	An instance of a resource that is required to achieve a Lighting_Solution (e.g. power).

location	A particular position of one or more Lights that is necessary to achieve the Lighting_Solution.
orientation	A particular orientation of one or more Lights that is necessary to achieve the Lighting_Solution.
temporal_information	Information covering timing, such as start and end times.

Lighting_Solution_Achievement

This interface is the statement of achievement against the derived requirement.

Activities

assess_derived_requirement_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

identify_derived_requirement_change

Identify changes to the requirements derived from the Lighting_Solution that have been placed outside of the component, including changes to evidence that is to be collected.

identify_derived_requirements_to_be_fulfilled

Identify the derived requirements to be fulfilled.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

5.4.2.32.7.1.3 Subject_Lighting_Condition

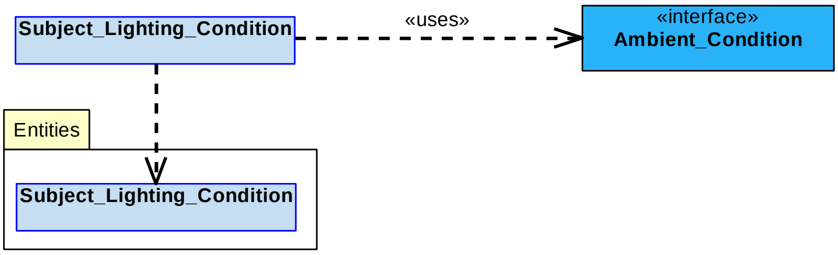


Figure 561: Subject_Lighting_Condition Service Definition

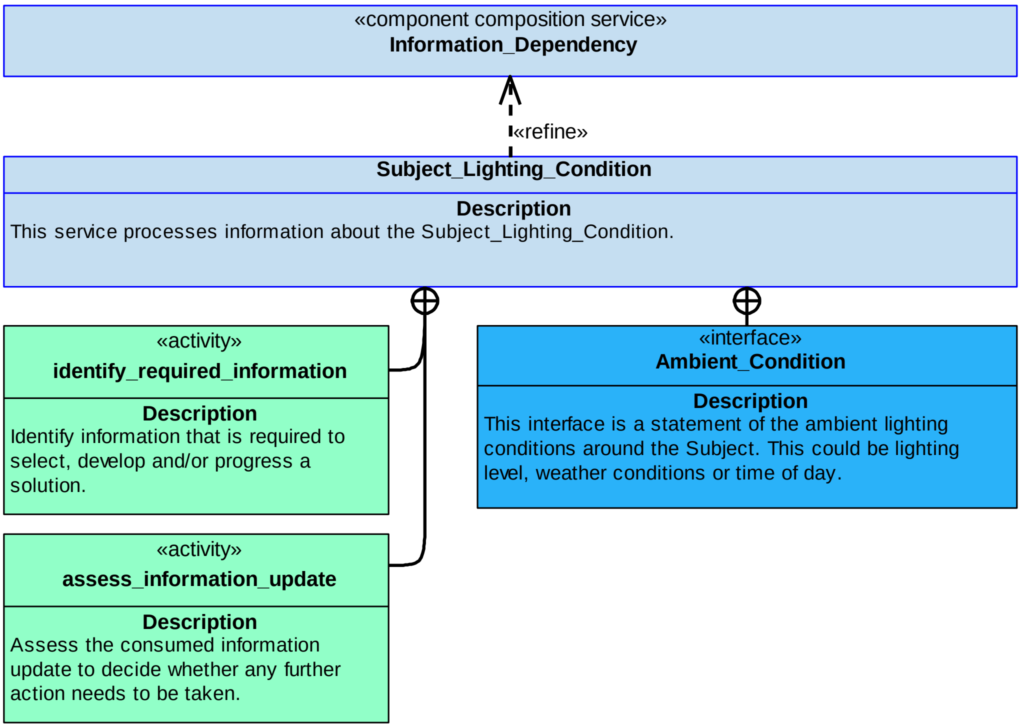

«component composition service»
Information_Dependency
«refine»
Subject_Lighting_Condition
Description
This service processes information about the Subject_Lighting_Condition.
«activity»
identify_required_information
Description
Identify information that is required to select, develop and/or progress a solution.
«activity»
assess_information_update
Description
Assess the consumed information update to decide whether any further action needs to be taken.
«interface»
Ambient_Condition
Description
This interface is a statement of the ambient lighting conditions around the Subject. This could be lighting level, weather conditions or time of day.

Figure 562: Subject_Lighting_Condition Service Policy

Subject_Lighting_Condition

This service processes information about the Subject_Lighting_Condition.

<u>Interface</u>

Ambient_Condition

This interface is a statement of the ambient lighting conditions around the Subject. This could be lighting level, weather conditions or time of day.

<u>Activities</u>

assess_information_update

Assess the consumed information update to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to select, develop and/or progress a solution.

5.4.2.32.7.1.4 Constraint

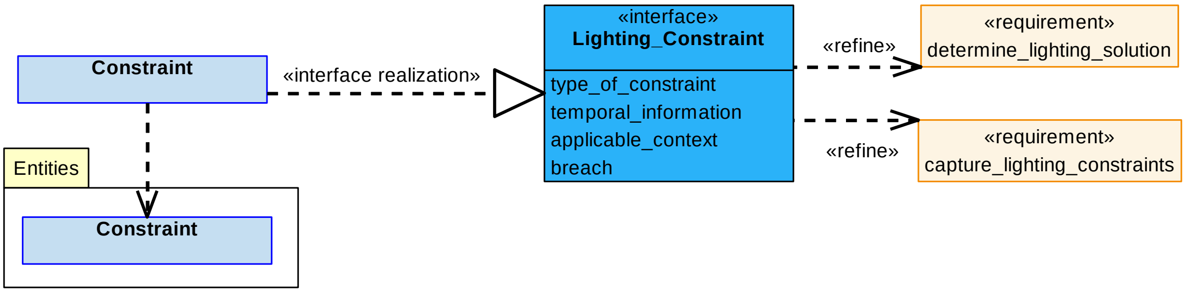


Figure 563: Constraint Service Definition

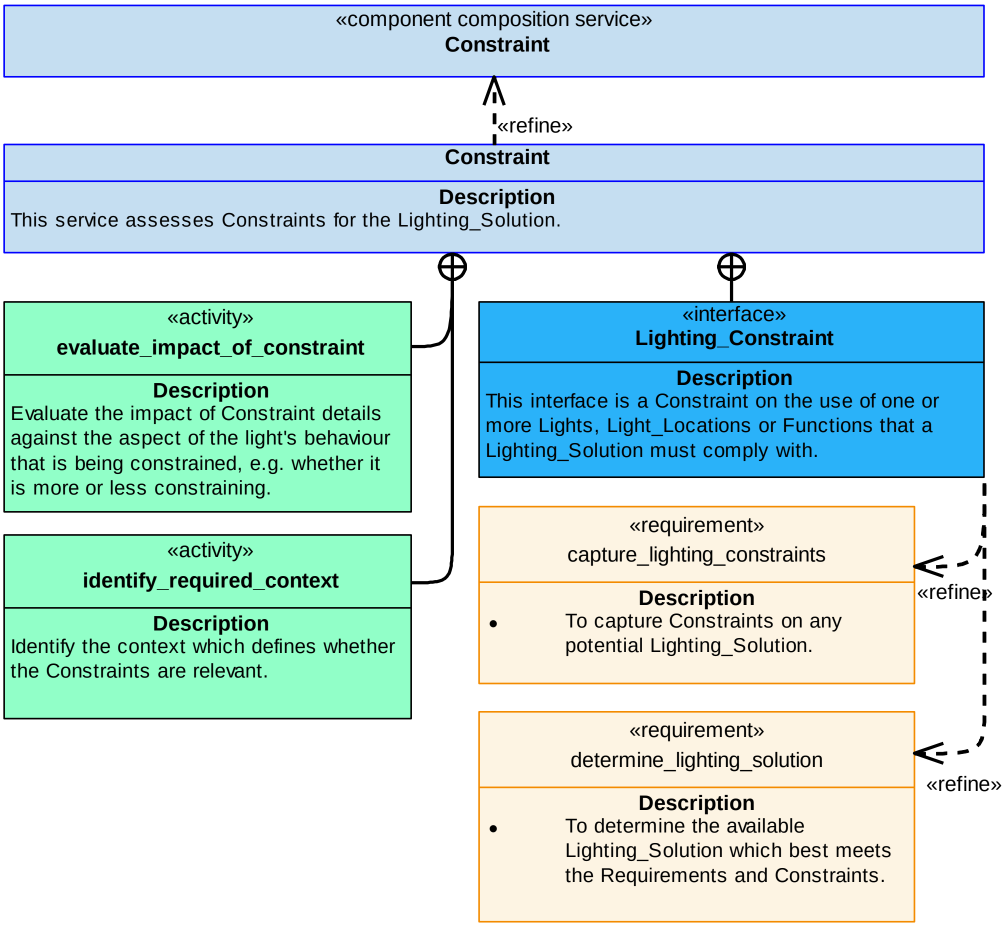

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints for the Lighting_Solution.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of the light's behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Lighting_Constraint
Description
This interface is a Constraint on the use of one or more Lights, Light_Locations or Functions that a Lighting_Solution must comply with.
«requirement»
capture_lighting_constraints
Description
• To capture Constraints on any potential Lighting_Solution.
«refine»
«requirement»
determine_lighting_solution
Description
• To determine the available Lighting_Solution which best meets the Requirements and Constraints.
«refine»

Figure 564: Constraint Service Policy

Constraint

This service assesses Constraints for the Lighting_Solution.

Interface

Lighting_Constraint

This interface is a Constraint on the use of one or more Lights, Light_Locations or Functions that a Lighting_Solution must comply with.

Attributes

type_of_constraint	The nature of the constraint.
temporal_information	Information relating to the times or durations when the constraint applies.
applicable_context	The context in which the constraint is applicable.
breach	A statement that the constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of the light's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.32.7.1.5 Capability

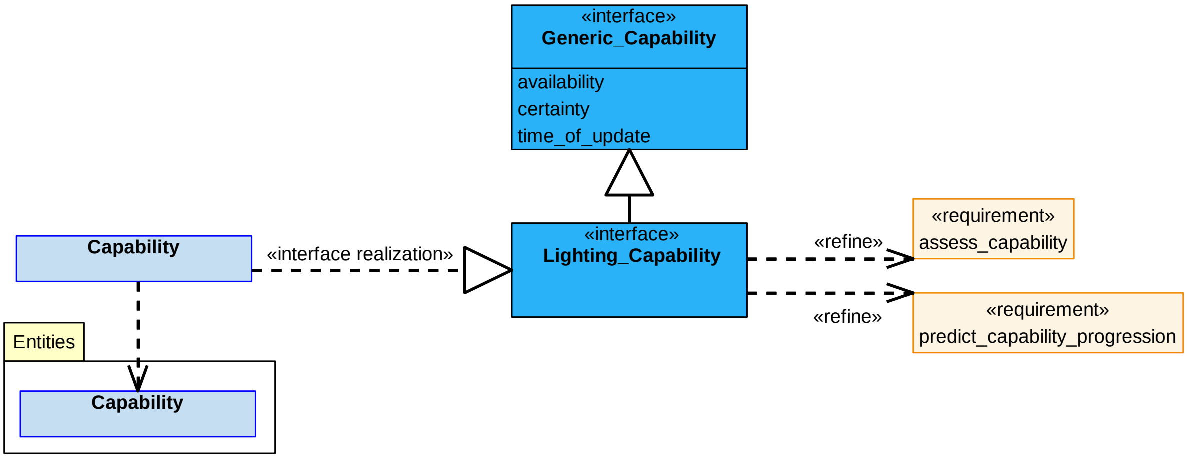


Figure 565: Capability Service Definition

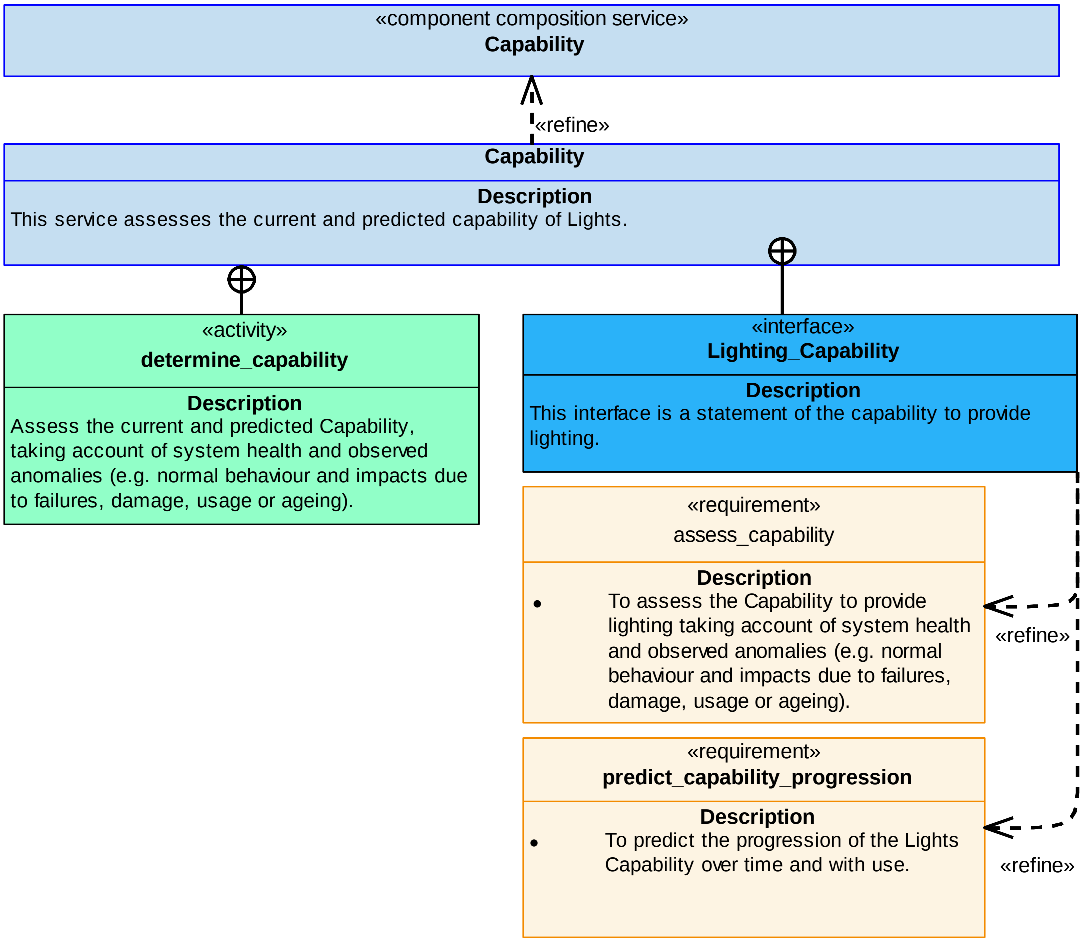

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability of Lights.
«activity»
determine_capability
Description
Assess the current and predicted Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Lighting_Capability
Description
This interface is a statement of the capability to provide lighting.
«requirement»
assess_capability
Description
To assess the Capability to provide lighting taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the Lights Capability over time and with use.
«refine»

Figure 566: Capability Service Policy

Capability

This service assesses the current and predicted capability of Lights.

Interface

Lighting_Capability

This interface is a statement of the capability to provide lighting.

Activity

determine_capability

Assess the current and predicted Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.32.7.1.6 Capability_Evidence

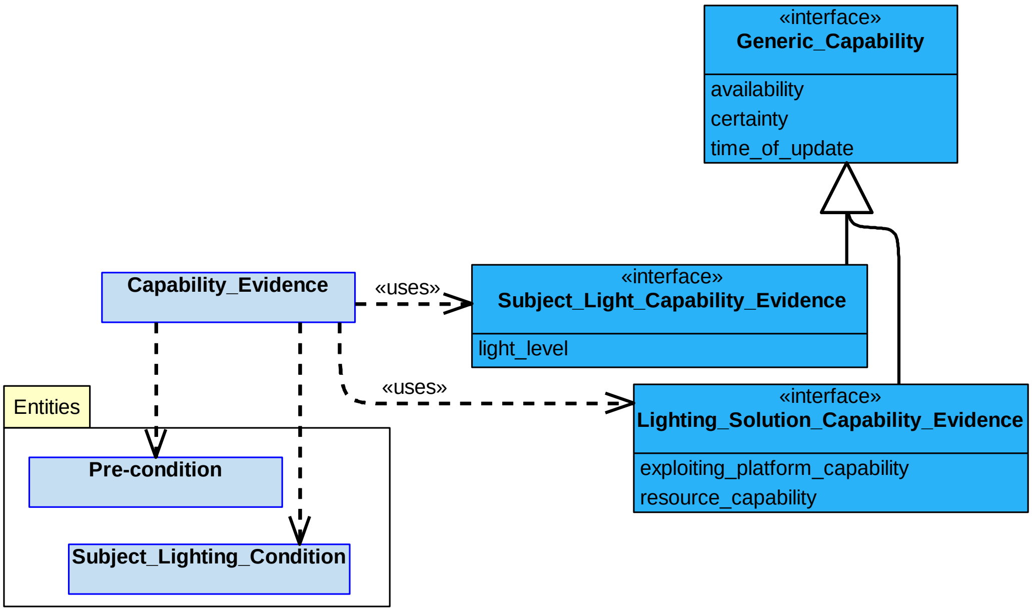


Figure 567: Capability Evidence Service Definition

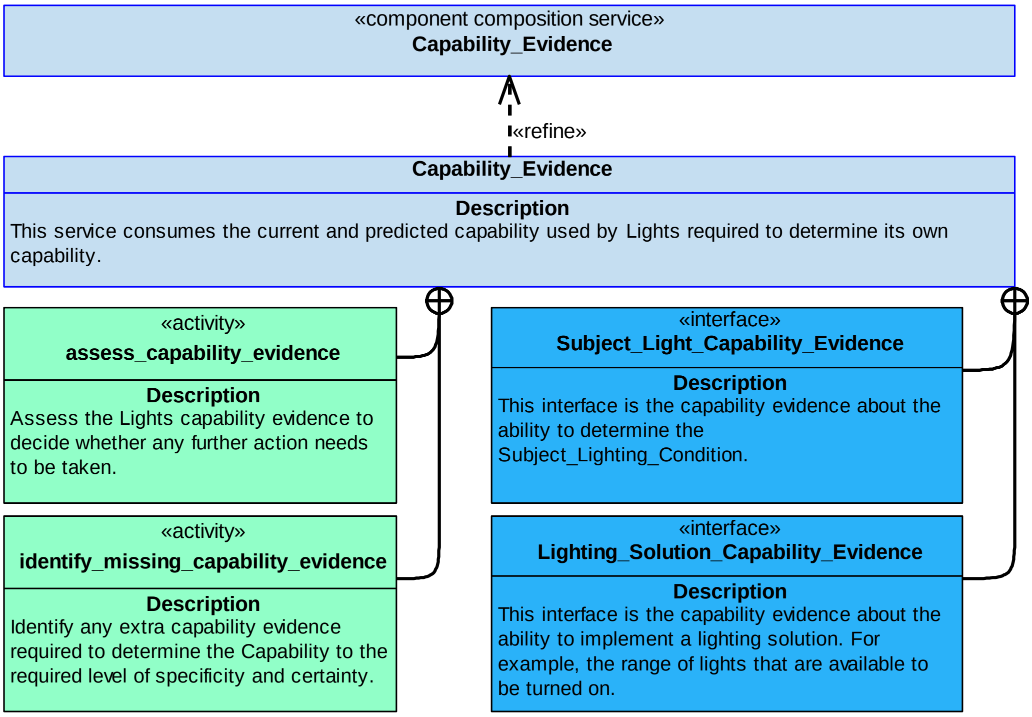

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted capability used by Lights required to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the Lights capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Subject_Light_Capability_Evidence
Description
This interface is the capability evidence about the ability to determine the Subject_Lighting_Condition.
«interface»
Lighting_Solution_Capability_Evidence
Description
This interface is the capability evidence about the ability to implement a lighting solution. For example, the range of lights that are available to be turned on.

Figure 568: Capability Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capability used by Lights required to determine its own capability.

Interfaces

Subject_Light_Capability_Evidence

This interface is the capability evidence about the ability to determine the Subject_Lighting_Condition.

Attribute

light_level	An indication of the capability to receive information about amount of ambient light in the environment.

Lighting_Solution_Capability_Evidence

This interface is the capability evidence about the ability to implement a lighting solution. For example, the range of lights that are available to be turned on.

Attributes

exploiting_platform_capability	An indication of the capability of the Exploiting Platform to contribute to a lighting solution. For example, whether the Exploiting Platform can orient itself or not.
resource_capability	An indication of the state of a resource and whether it can contribute to a lighting solution. For example, whether or not power can be provided.

Activities

assess_capability_evidence

Assess the Lights capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.32.7.2 Service Dependencies

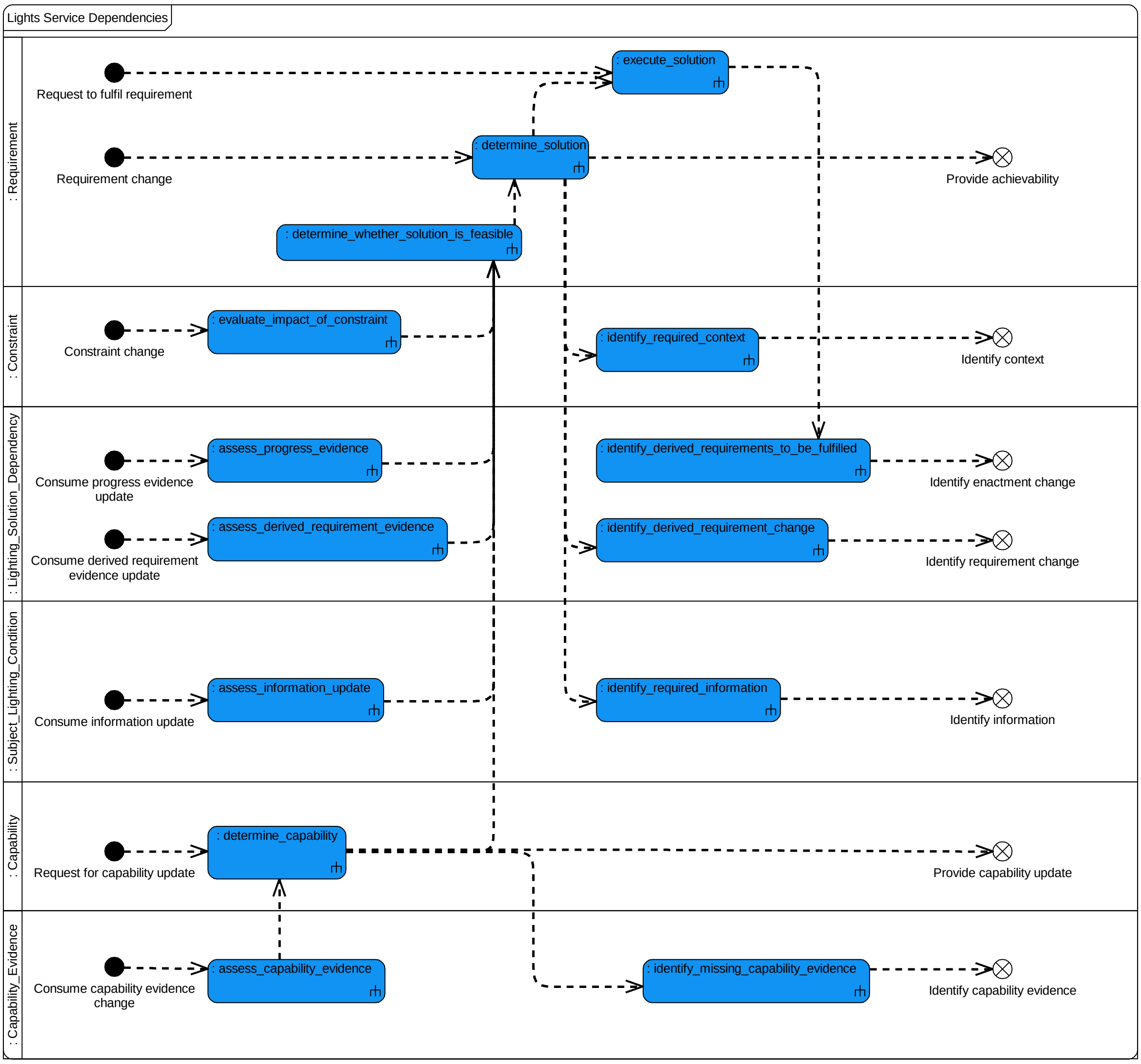

Lights Service Dependencies
: Requirement
: Constraint
: Lighting_Solution_Dependency
: Subject_Lighting_Condition
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_solution
Requirement change
: determine_solution
Provide achievability
: determine_whether_solution_is_feasible
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume progress evidence update
: assess_progress_evidence
: identify_derived_requirements_to_be_fulfilled
Identify enactment change
Consume derived requirement evidence update
: assess_derived_requirement_evidence
: identify_derived_requirement_change
Identify requirement change
Consume information update
: assess_information_update
: identify_required_information
Identify information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 569: Lights Service Dependencies

5.4.2.33 Location and Orientation

5.4.2.33.1 Role

The role of Location and Orientation is to determine the location and spatial orientation including any derivatives of a platform.

5.4.2.33.2 Overview

Control Architecture

Location and Orientation is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

A Requirement is placed on Location and Orientation to provide an estimate of the Location and/or Orientation (or Derivatives of these) of a Platform. Source_Parameters from one or more Sources are used to determine the Location, Orientation or Derivative, defined against a particular Reference_Frame.

Examples of Use

Location and Orientation will be used where:

- The determination of Location and/or Orientation (or Derivatives of these) is required in order to safely and efficiently control some aspect of the Platform's operation.

5.4.2.33.3 Service Summary

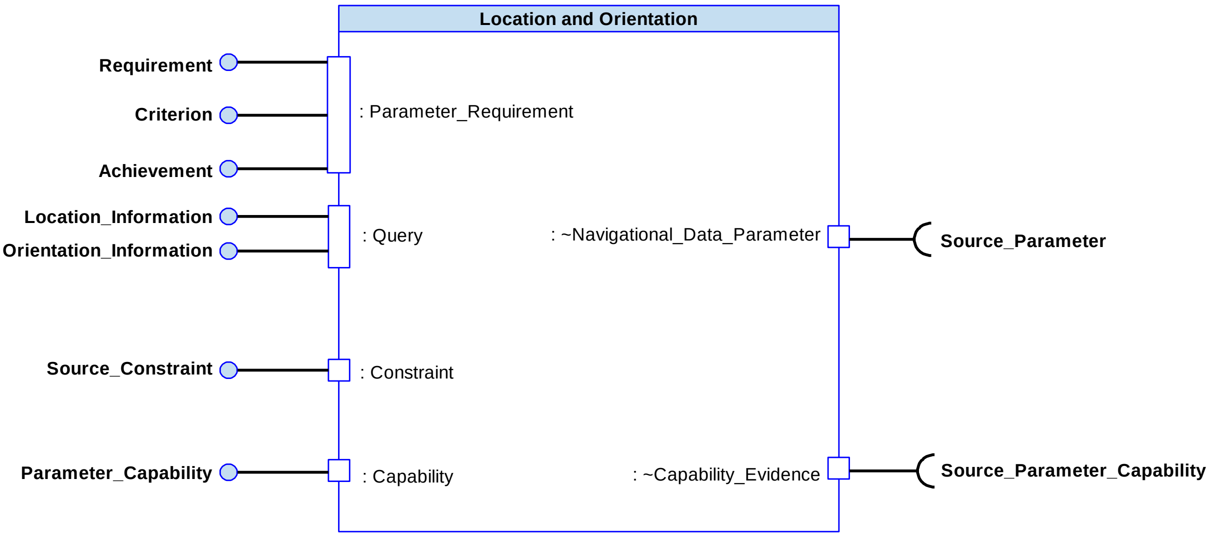


Figure 570: Location and Orientation Service Summary

5.4.2.33.4 Responsibilities

capture_parameter_requirements

- To capture the Requirements for determining Location, Orientation and Derivatives.

capture_constraints

- To capture Constraints on the use of Sources.

determine_if_requirement_is_achievable

- To determine if a Requirement is achievable given current Capability and Constraints.

determine_location

- To determine the Location of a Platform relative to a Reference_Frame.

determine_orientation

- To determine the Orientation of a Platform relative to a Reference_Frame.

determine_derivatives

- To determine the derivatives of Location and/or Orientation relative to the Reference_Frame.

determine_parameter_quality

- To determine the accuracy and precision of a Location, Orientation or Derivative parameter.

capture_reference_frame

- To capture given Reference_Frames. This includes absolute and relative Reference_Frames.

capture_source_parameters

- To capture given Source_Parameters from available Sources.

assess_parameter_capability

- To identify the system's Capability to determine the Location and/or Orientation (or a Derivative of these parameters) of a Platform, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the Location and Orientation Capability over time and with use.

5.4.2.33.5 Subject Matter Semantics

The subject matter of Location and Orientation is the Location and Orientation of a Platform and their Derivatives (e.g. angular acceleration).

Exclusions

The subject matter of Location and Orientation does not include:

- The determination of Location or Orientation for platforms that the Exploiting Platform is not responsible for. For example, vehicles detected by tactical sensors or other friendly air vehicles with their own capabilities to determine their own Location and Orientation.

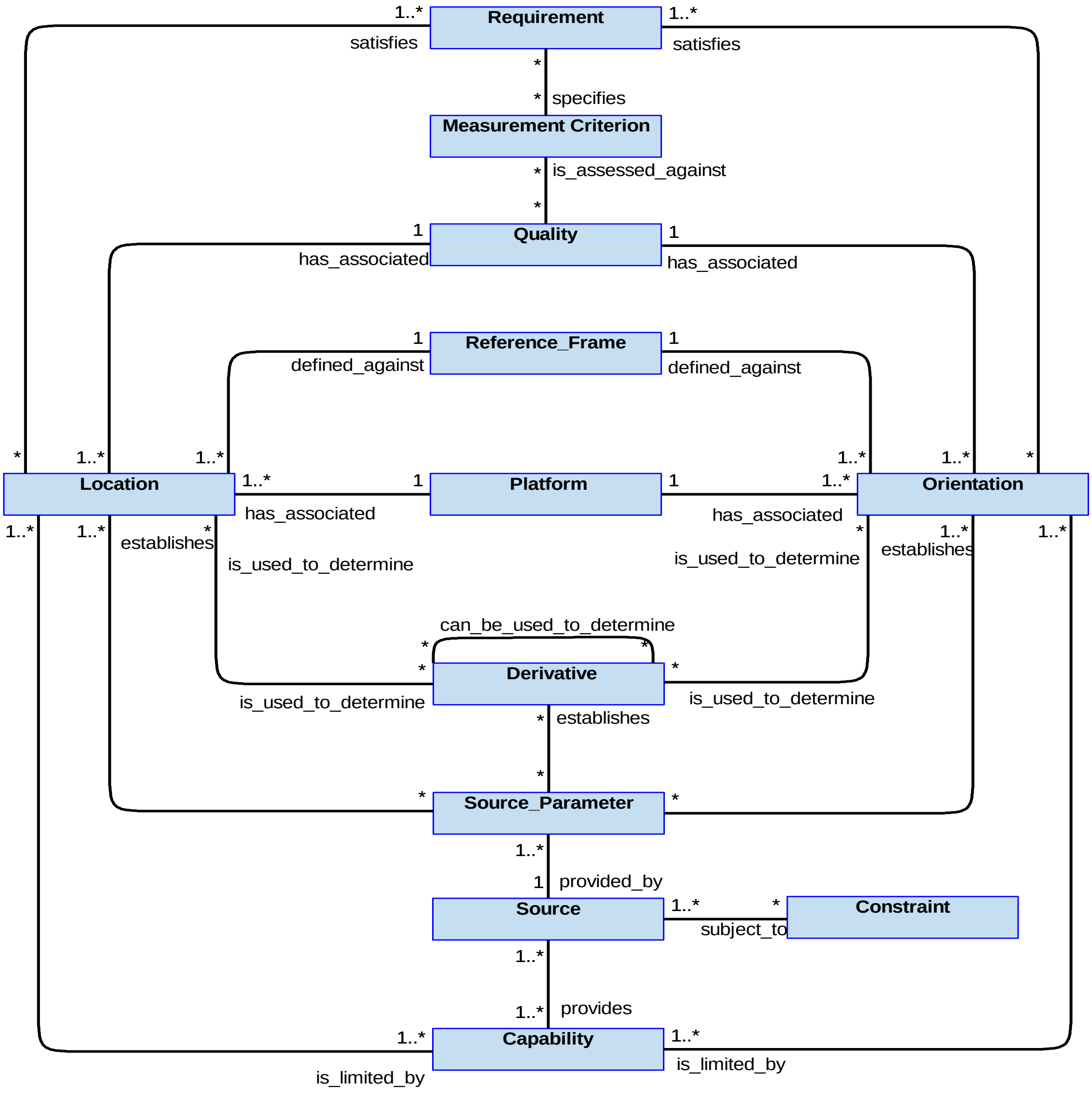


Figure 571: Location and Orientation Semantics

5.4.2.33.5.1 Entities

Quality

The accuracy and precision. This may take into account drift rate of an inertial solution parameter, quality of GNSS data, etc.

Capability

The capability to determine Location, Orientation and their Derivatives, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Constraint

An externally imposed restriction, e.g. it may be necessary to exclude a GNSS as a provider of Source_Parameters due to suspected spoofing or to limit specific data from a Source.

Location

An estimate of the location of a Platform defined against a Reference_Frame. This could be expressed as latitude, longitude, altitude, etc.

Orientation

An estimate of the orientation of a Platform defined against a Reference_Frame. This could include attitude (pitch, roll and yaw).

Platform

The asset for which the positional information is to be determined. The asset could either be the Exploiting Platform, including a part of it (e.g. the air vehicle or the ground station), or another air vehicle whose capabilities are handled by the Exploiting Platform (e.g. a less capable UAV).

Source_Parameter

Any form of navigational data, including positions, velocities, accelerations, etc. A Source_Parameter may have an associated accuracy, precision and validity provided by the Source.

Reference_Frame

An abstract coordinate system defining a set of physical reference points which uniquely:

- locate a coordinate system (the 'origin').
- orient the coordinate system.
- identify standardised measurements used to identify the relative location of physical reference points.

This could be absolute (e.g. a fixed coordinate system) or relative (e.g. relative to a given object).

Source

A Source of parameters. For example, a GNSS unit or an accelerometer.

Measurement Criterion

A measure against which achievement of the Requirement can be assessed.

Requirement

A specification for the provision of a solution to determine Location, Orientation or a Derivative of one of these parameters.

Derivative

A derivative of Location or Orientation relative to a Reference_Frame, e.g. the angular velocity, angular acceleration, linear velocity and linear acceleration.

5.4.2.33.6 Design Rationale

5.4.2.33.6.1 Assumptions

None.

5.4.2.33.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Location and Orientation:

- Data Driving - The parameters maintained by the Location and Orientation component for use throughout the system can be designed to be data-driven at build (i.e. development) time. This would allow the component to be configurable and flexible to cater for any set of required information and allow reusability between multiple Exploiting Platforms.

Extensions

- Extension components could be deployed for some aspects of Location and Orientation, such as to deal with Dead Reckoning, for example.

Other Factors that were Taken into Account

- Whilst the subject matter of Vehicle External Environment and Location and Orientation are closely related, the information determined by each component and the Sources used to create them do not directly overlap.
- Safety concerns mean that measurements supplied to Vehicle External Environment must be treated differently as they cover concepts such as the vehicle's attitude to the local airflow, regardless of the platform's overall orientation, and hence impact flight control safety integrity. The Location and Orientation capability of some Exploiting Platforms may not use environment information in navigational reasoning (e.g. if determining Location using navigational beacons).
- The separation of concerns between these two components aids in configurability and resilience against obsolescence requirements.

Exploitation Considerations

- There may be many ways to determine the Location or Orientation using different combinations of Source_Parameters. An Exploiting Programme will need to define how Location and Orientation will be determined, for example, by applying weightings to Source_Parameters. The rules for combining these Source_Parameters will be contained within Location and Orientation.
- Location and Orientation determines the Location or Orientation using Source_Parameters from multiple Sources, according to provided Requirements. It is, however, not within the scope of Location and Orientation to place requirements on to the Sources to initiate data collection, or to manage the Sources.

- Determining that a particular Source or Source_Parameter has been subjected to cyber attack (e.g. GNSS jamming or spoofing) is not the responsibility of this component (see the Cyber Defence component). However, for exploitations with multiple Sources (e.g. inertial and GNSS) this component may reduce the weighting or exclude a Source, based on a comparison with other Sources or the accuracy of Source_Parameters. For example, GNSS may be excluded if the GNSS derived location were to deviate significantly from an inertial derived location.
- An Exploiting Programme will need to decide whether or how to use an associated Source_Parameter, should a deviation from the required or expected accuracy be detected, or if it is detected that the data is invalid. This aids the reliability of the component.

5.4.2.33.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- In the case of an air vehicle, failure of this component in determining orientation may cause uncontrolled flight of the air vehicle due to exceedance of the flight envelope/structural limits/loss of stability. This could lead to loss of structural integrity of the air vehicle and/or an uncontrolled crash.
- Failure of this component in determining location would cause uncontrolled flight of the air vehicle. Whilst the air vehicle would be within the aerodynamic limits of the air vehicle, the path of the air vehicle would not be controlled.
- The result is likely to be loss of the air vehicle and fatalities.

5.4.2.33.6.4 Security Considerations

The indicative security classification is SNEO.

This component is responsible for the determination of the location and spatial orientation of the Platform, using both internal and external Sources. Whilst the source data and location/orientation in civil airspace or public areas will generally be O, during military operations the details will be SNEO with associated requirements for confidentiality. This component relies upon the integrity of its sources, and should therefore only use those considered trustworthy. GNSS spoofing and denial is a particular concern as this could impact the accuracy of this component and the behaviour of the platform. Loss of availability can have significant safety and operational consequences and should be protected accordingly.

The component is expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** used for determining location and orientation as being allowable sources.
- **Logging of Security Data** of attempted access to non-whitelisted sources, interruptions in operation, changes to positional data, etc.
- **Maintaining Audit Records** of where the Exploiting Platform is throughout the mission.

- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is considered to at least partially satisfy security enforcing function through its involvement in:

- **Verifying Integrity of Data** received; where Quality is below that expected, as may be the case with spoofed data, it will coordinate with other components to attempt to improve it.

5.4.2.33.7 Services

5.4.2.33.7.1 Service Definitions

5.4.2.33.7.1.1 Parameter_Requirement

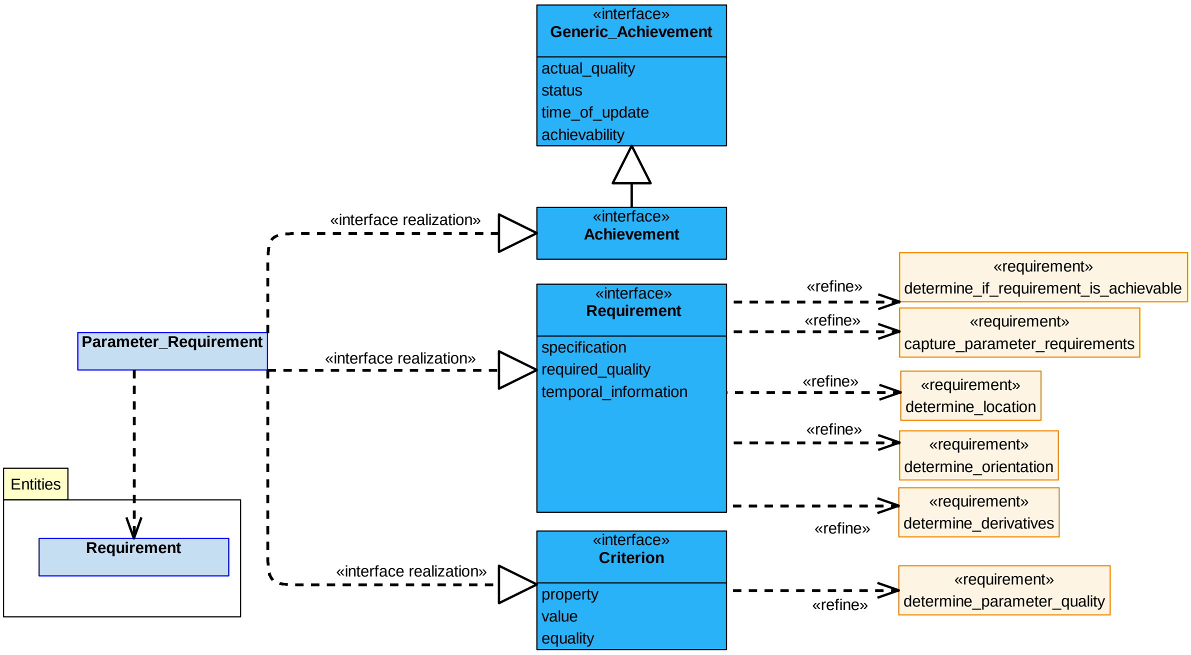


Figure 572: Parameter_Requirement Service Definition

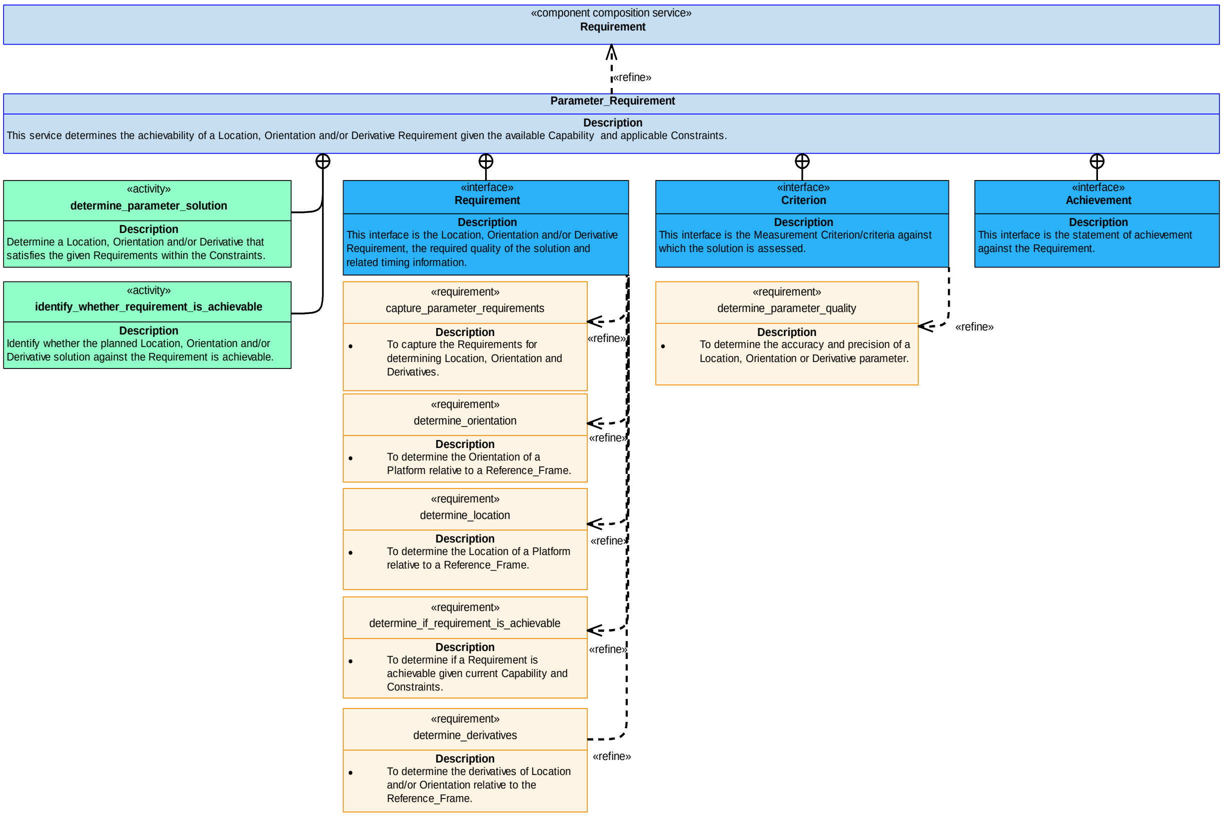

«component composition service»
Requirement
«refine»
Parameter_Requirement
Description
This service determines the achievability of a Location, Orientation and/or Derivative Requirement given the available Capability and applicable Constraints.
«activity»
determine_parameter_solution
Description
Determine a Location, Orientation and/or Derivative that satisfies the given Requirements within the Constraints.
«activity»
identify_whether_requirement_is_achievable
Description
Identify whether the planned Location, Orientation and/or Derivative solution against the Requirement is achievable.
«interface»
Requirement
Description
This interface is the Location, Orientation and/or Derivative Requirement, the required quality of the solution and related timing information.
«interface»
Criterion
Description
This interface is the Measurement Criterion/criteria against which the solution is assessed.
«interface»
Achievement
Description
This interface is the statement of achievement against the Requirement.
«requirement»
capture_parameter_requirements
Description
To capture the Requirements for determining Location, Orientation and Derivatives.
«refine»
«requirement»
determine_orientation
Description
To determine the Orientation of a Platform relative to a Reference_Frame.
«refine»
«requirement»
determine_location
Description
To determine the Location of a Platform relative to a Reference_Frame.
«refine»
«requirement»
determine_if_requirement_is_achievable
Description
To determine if a Requirement is achievable given current Capability and Constraints.
«refine»
«requirement»
determine_derivatives
Description
To determine the derivatives of Location and/or Orientation relative to the Reference_Frame.
«refine»
«requirement»
determine_parameter_quality
Description
To determine the accuracy and precision of a Location, Orientation or Derivative parameter.
«refine»

Figure 573: Parameter_Requirement Service Policy

Parameter_Requirement

This service determines the achievability of a Location, Orientation and/or Derivative Requirement given the available Capability and applicable Constraints.

Interfaces

Requirement

This interface is the Location, Orientation and/or Derivative Requirement, the required quality of the solution and related timing information.

Attributes

specification	The specification of a Location and Orientation solution, e.g. the Sources and the rules for their combination in determining a Location and/or Orientation solution, including weighting or precedence of Source_Parameters.
required_quality	The required quality of the Location and Orientation solution, e.g. the required accuracy, regularity, or precision.
temporal_information	Information covering timing, such as start and end times.

Criterion

This interface is the Measurement Criterion/criteria against which the solution is assessed.

Attributes

property	The criterion property to be measured, e.g. a cost or quality factor such as time taken.
value	The value to be related to the measured property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Achievement

This interface is the statement of achievement against the Requirement.

Activities

determine_parameter_solution

Determine a Location, Orientation and/or Derivative that satisfies the given Requirements within the Constraints.

identify_whether_requirement_is_achievable

Identify whether the planned Location, Orientation and/or Derivative solution against the Requirement is achievable.

5.4.2.33.7.1.2 Query

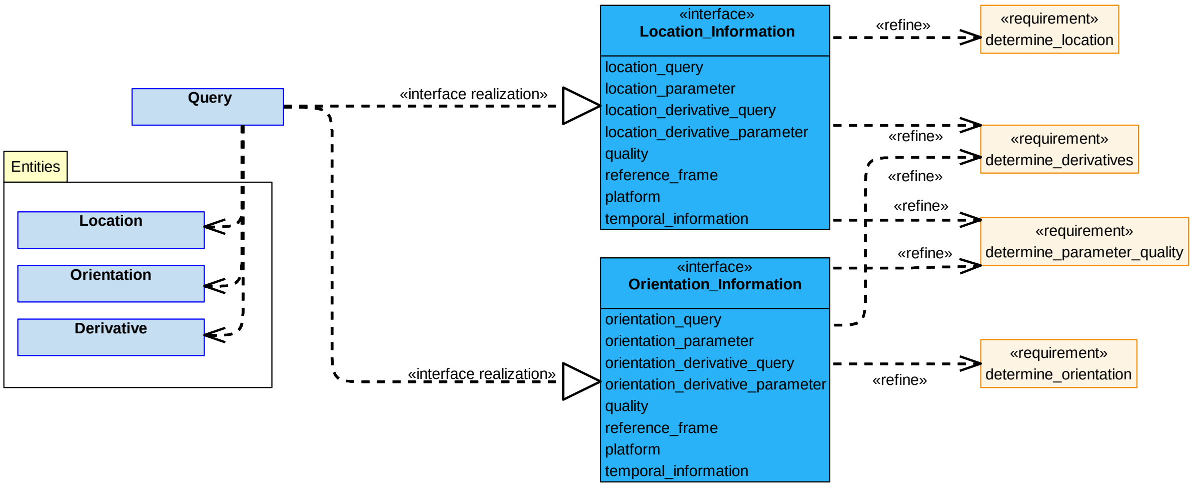


Figure 574: Query Service Definition

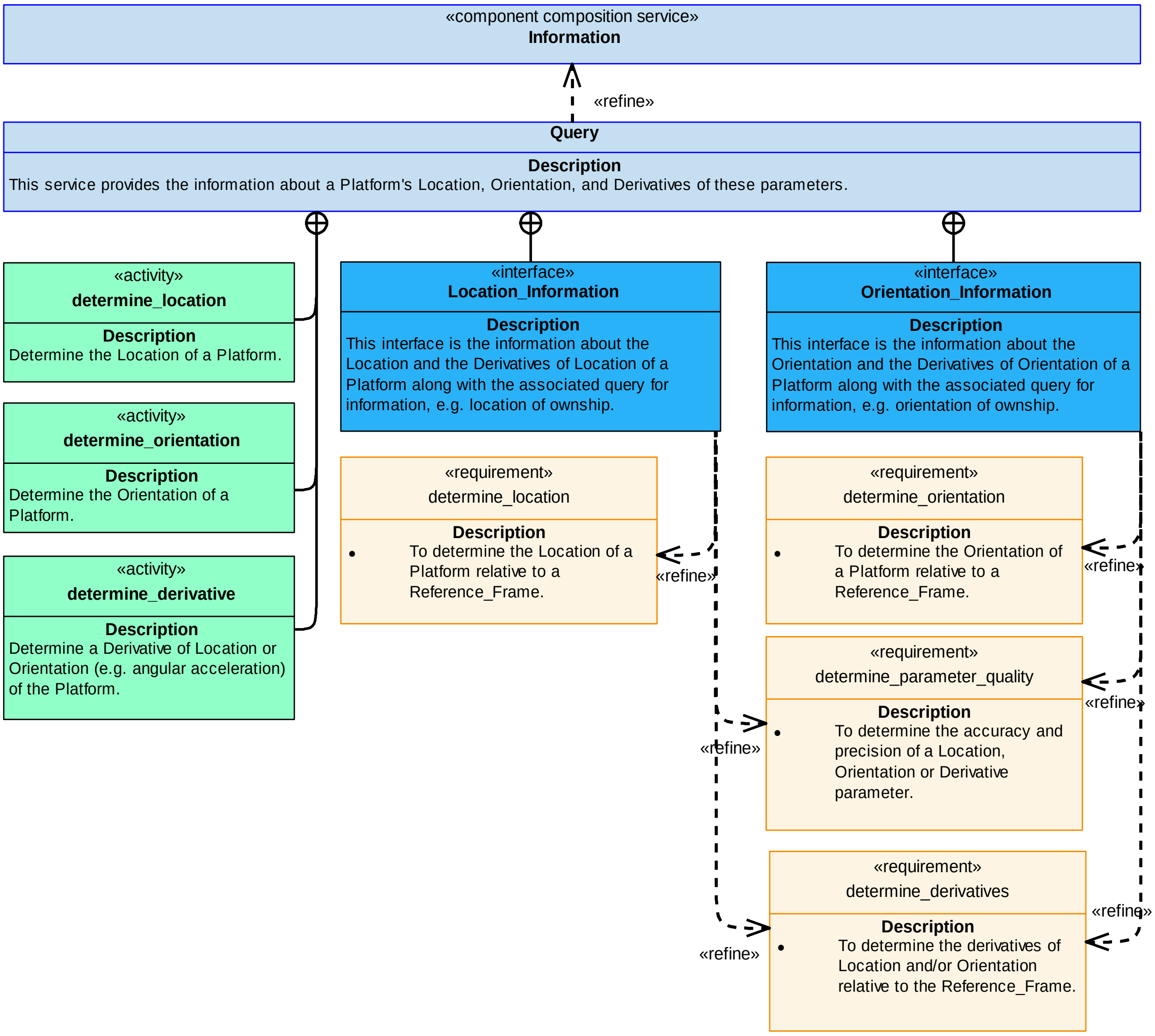

«component composition service»
Information
«refine»
Query
Description
This service provides the information about a Platform's Location, Orientation, and Derivatives of these parameters.
«activity»
determine_location
Description
Determine the Location of a Platform.
«activity»
determine_orientation
Description
Determine the Orientation of a Platform.
«activity»
determine_derivative
Description
Determine a Derivative of Location or Orientation (e.g. angular acceleration) of the Platform.
«interface»
Location_Information
Description
This interface is the information about the Location and the Derivatives of Location of a Platform along with the associated query for information, e.g. location of ownship.
«interface»
Orientation_Information
Description
This interface is the information about the Orientation and the Derivatives of Orientation of a Platform along with the associated query for information, e.g. orientation of ownship.
«requirement»
determine_location
Description
To determine the Location of a Platform relative to a Reference_Frame.
«requirement»
determine_orientation
Description
To determine the Orientation of a Platform relative to a Reference_Frame.
«requirement»
determine_parameter_quality
Description
To determine the accuracy and precision of a Location, Orientation or Derivative parameter.
«requirement»
determine_derivatives
Description
To determine the derivatives of Location and/or Orientation relative to the Reference_Frame.
«refine»
«refine»
«refine»
«refine»
«refine»
«refine»

Figure 575: Query Service Policy

Query

This service provides the information about a Platform's Location, Orientation, and Derivatives of these parameters.

<u>Interfaces</u>

Location_Information

This interface is the information about the Location and the Derivatives of Location of a Platform along with the associated query for information, e.g. location of ownship.

<u>Attributes</u>

location_query	The definition of the query for information about the Location of a Platform.

location_parameter	The Location of the Platform defined against a given Reference_Frame, e.g. latitude, longitude, and altitude expressed as geodetic coordinates.
location_derivative_query	The definition of the query for information about a Derivative of Location of a Platform.
location_derivative_parameter	A Derivative of Location of the Platform relative to a Reference_Frame, e.g. including but not limited to, linear velocity, linear acceleration, and linear jerk.
quality	The Quality of the Location or Location Derivative information.
reference_frame	The Reference_Frame to be used for the query.
platform	The Platform for which the Location or Location Derivative information is being provided, e.g. ownship.
temporal_information	Timing information, such as when the Location information was provided.

Orientation_Information

This interface is the information about the Orientation and the Derivatives of Orientation of a Platform along with the associated query for information, e.g. orientation of ownship.

Attributes

orientation_query	The definition of the query for information about the Orientation of a Platform.
orientation_parameter	The Orientation of the Platform relative to a Reference_Frame.
orientation_derivative_query	The definition of the query for information about a Derivative of Orientation of a Platform.
orientation_derivative_parameter	A Derivative of Orientation of the Platform relative to a Reference_Frame, e.g. including but not limited to, angular velocity, angular acceleration, and angular jerk.
quality	The Quality of the Orientation or Orientation Derivative information.
reference_frame	The Reference_Frame to be used for the query.
platform	The Platform for which the Orientation or Orientation Derivative information is being provided, e.g. ownship.
temporal_information	Timing information, such as when the Orientation information was provided.

Activities

determine_location

Determine the Location of a Platform.

determine_orientation

Determine the Orientation of a Platform.

determine_derivative

Determine a Derivative of Location or Orientation (e.g. angular acceleration) of the Platform.

5.4.2.33.7.1.3 Navigational_Data_Parameter

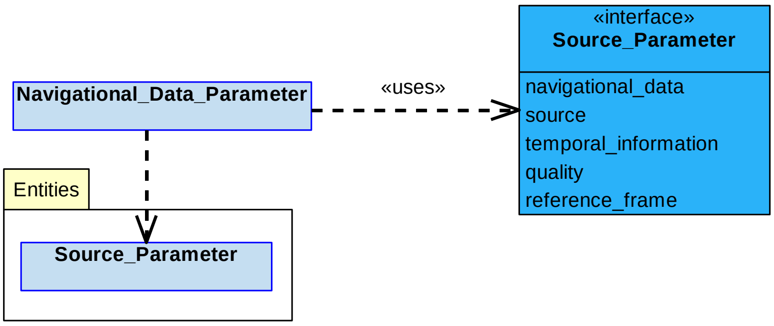


Figure 576: Navigational_Data_Parameter Service Definition

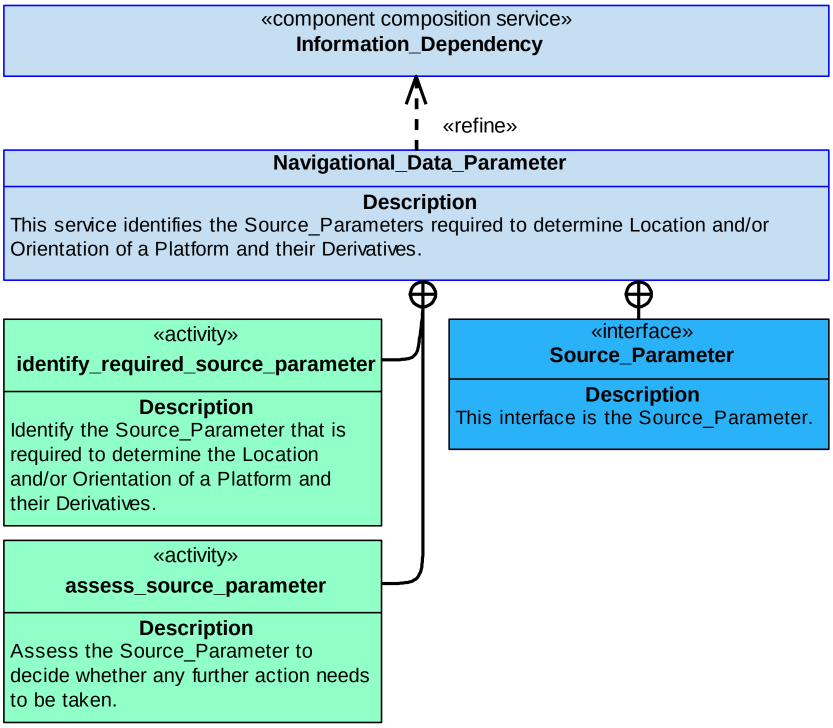

«component composition service»
Information_Dependency
«refine»
Navigational_Data_Parameter
Description
This service identifies the Source_Parameters required to determine Location and/or Orientation of a Platform and their Derivatives.
«activity»
identify_required_source_parameter
Description
Identify the Source_Parameter that is required to determine the Location and/or Orientation of a Platform and their Derivatives.
«interface»
Source_Parameter
Description
This interface is the Source_Parameter.
«activity»
assess_source_parameter
Description
Assess the Source_Parameter to decide whether any further action needs to be taken.

Figure 577: Navigational_Data_Parameter Service Policy

Navigational_Data_Parameter

This service identifies the Source_Parameters required to determine Location and/or Orientation of a Platform and their Derivatives.

Interface

Source_Parameter

This interface is the Source_Parameter.

Attributes

navigational_data	The navigational data, e.g. positions, velocities, accelerations.
source	The Source, e.g. a GNSS unit.

temporal_information	Timing information, such as when the Source_Parameter was taken.
quality	The quality of the Source_Parameter.
reference_frame	The abstract coordinate system that defines a set of physical reference points.

Activities

assess_source_parameter

Assess the Source_Parameter to decide whether any further action needs to be taken.

identify_required_source_parameter

Identify the Source_Parameter that is required to determine the Location and/or Orientation of a Platform and their Derivatives.

5.4.2.33.7.1.4 Constraint

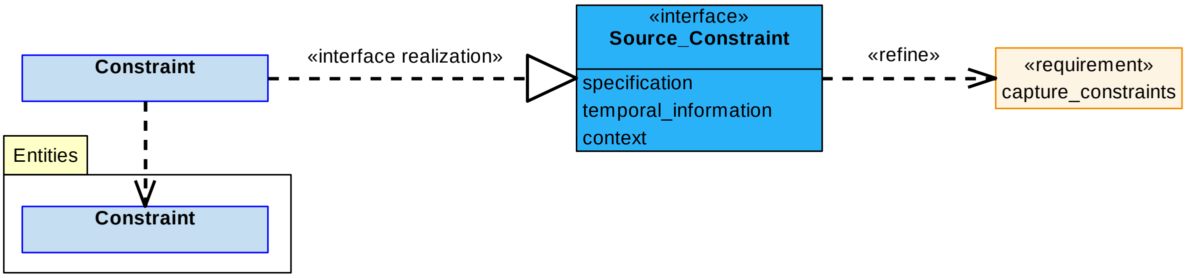


Figure 578: Constraint Service Definition

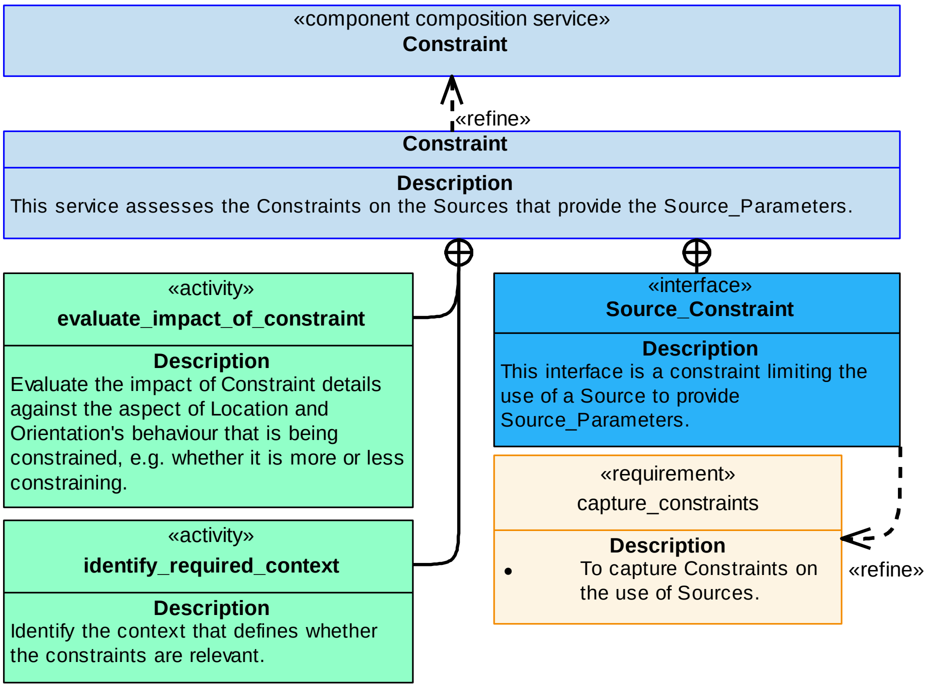

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Constraints on the Sources that provide the Source_Parameters.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of Location and Orientation's behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context that defines whether the constraints are relevant.
«interface»
Source_Constraint
Description
This interface is a constraint limiting the use of a Source to provide Source_Parameters.
«requirement»
capture_constraints
Description
• To capture Constraints on the use of Sources.
«refine»

Figure 579: Constraint Service Policy

Constraint

This service assesses the Constraints on the Sources that provide the Source_Parameters.

Interface

Source_Constraint

This interface is a constraint limiting the use of a Source to provide Source_Parameters.

Attributes

specification	Specification of the Constraint, such as a restriction on the type of information that can be obtained from a Source.
temporal_information	Information covering timing of a Constraint, such as start time and duration, or end time.
context	The context in which the Constraint is applicable, e.g. in polar regions due to the accuracy of the data Sources in these regions.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of Location and Orientation's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context that defines whether the constraints are relevant.

5.4.2.33.7.1.5 Capability

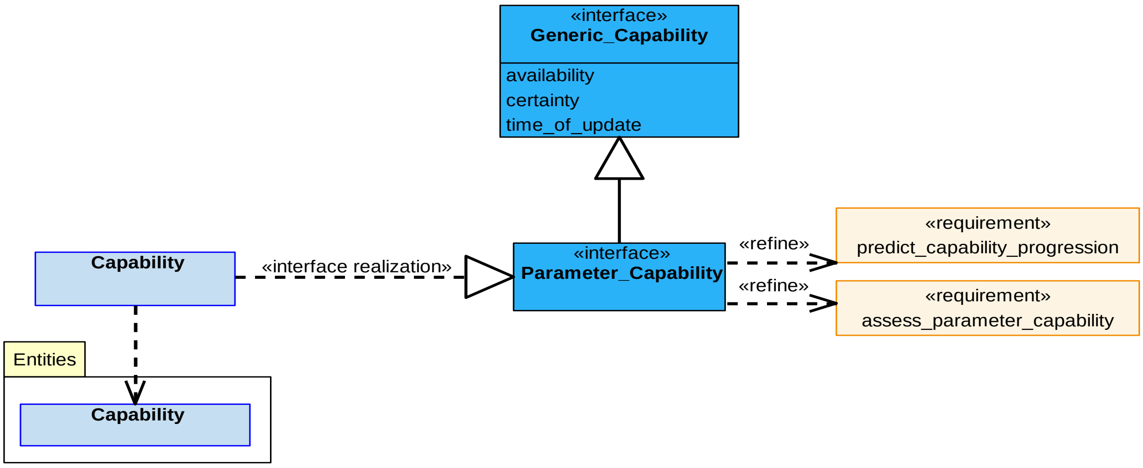


Figure 580: Capability Service Definition

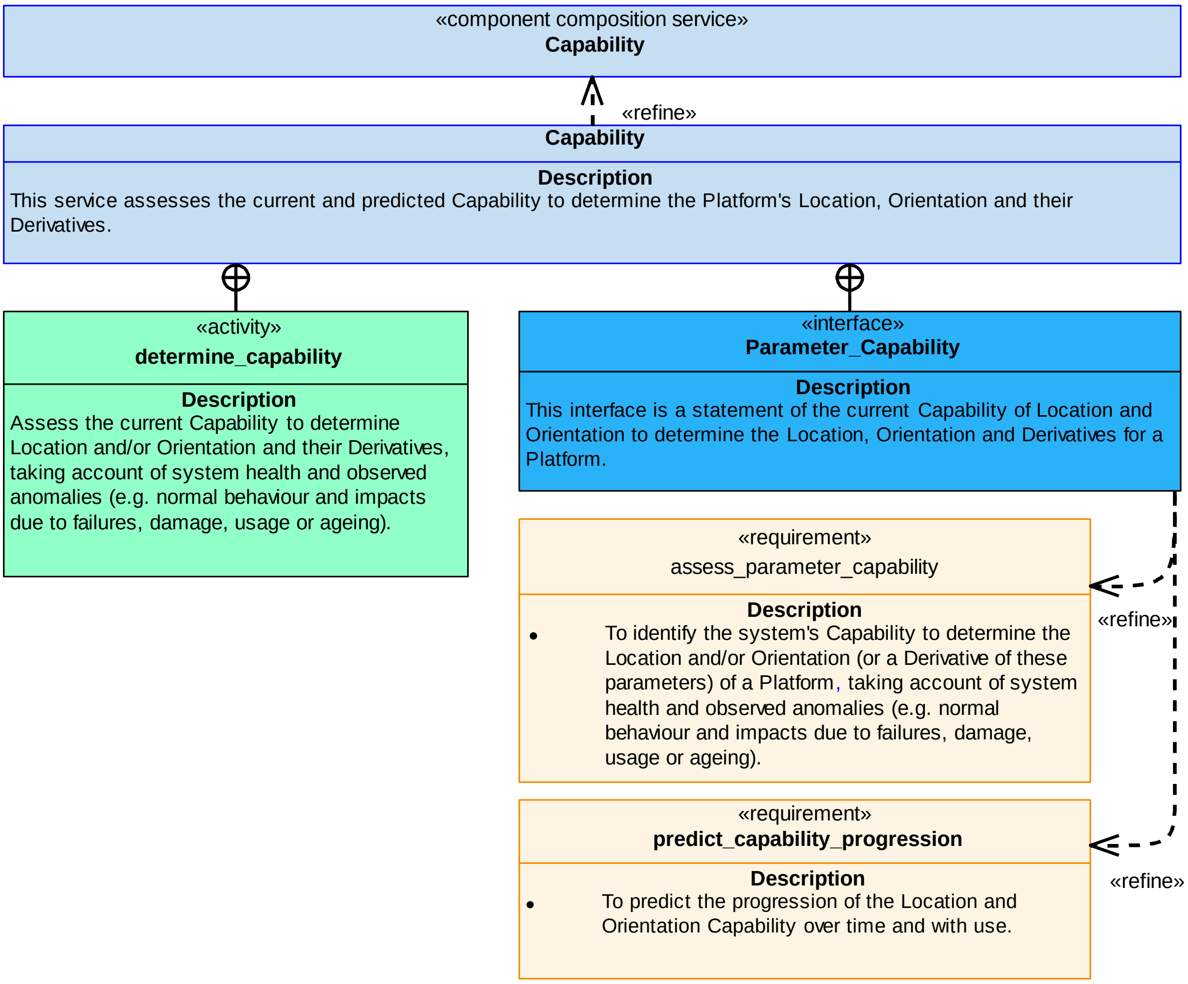

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to determine the Platform's Location, Orientation and their Derivatives.
«activity»
determine_capability
Description
Assess the current Capability to determine Location and/or Orientation and their Derivatives, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Parameter_Capability
Description
This interface is a statement of the current Capability of Location and Orientation to determine the Location, Orientation and Derivatives for a Platform.
«requirement»
assess_parameter_capability
Description
• To identify the system's Capability to determine the Location and/or Orientation (or a Derivative of these parameters) of a Platform, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
• To predict the progression of the Location and Orientation Capability over time and with use.
«refine»

Figure 581: Capability Service Policy

Capability

This service assesses the current and predicted Capability to determine the Platform's Location, Orientation and their Derivatives.

Interface

Parameter_Capability

This interface is a statement of the current Capability of Location and Orientation to determine the Location, Orientation and Derivatives for a Platform.

Activity

determine_capability

Assess the current Capability to determine Location and/or Orientation and their Derivatives, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.33.7.1.6 Capability_Evidence

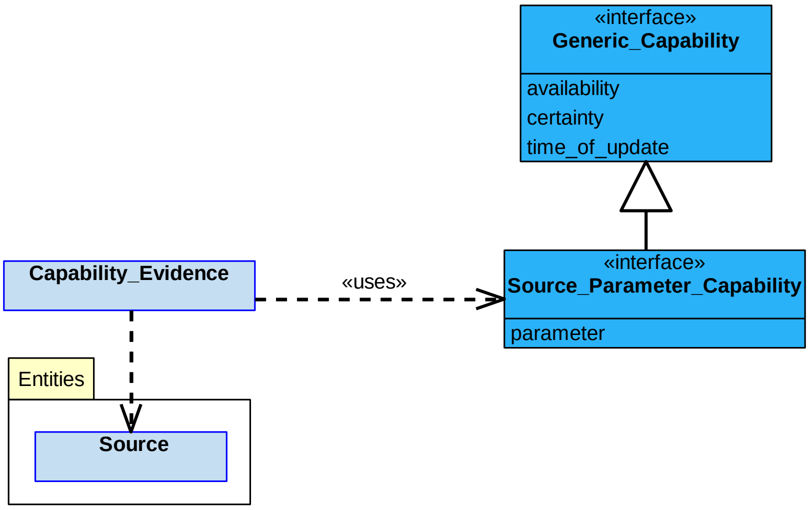


Figure 582: Capability_Evidence Service Definition

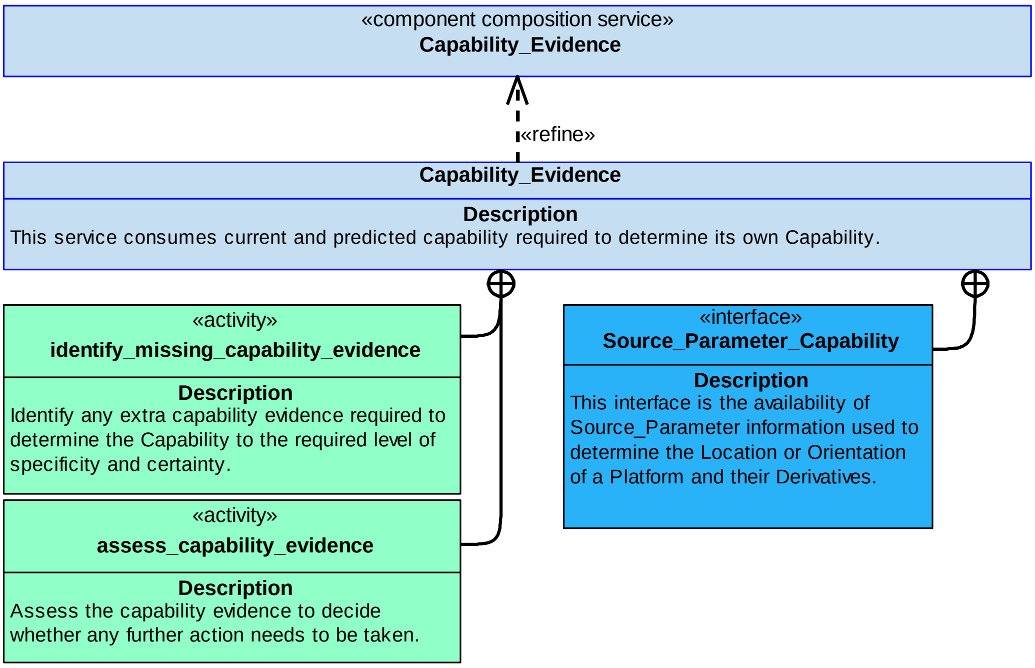

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability required to determine its own Capability.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«interface»
Source_Parameter_Capability
Description
This interface is the availability of Source_Parameter information used to determine the Location or Orientation of a Platform and their Derivatives.

Figure 583: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability required to determine its own Capability.

Interface

Source_Parameter_Capability

This interface is the availability of Source_Parameter information used to determine the Location or Orientation of a Platform and their Derivatives.

Attribute

parameter	Specification of the parameter to which the capability statement applies.

Activities

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

5.4.2.33.7.2 Service Dependencies

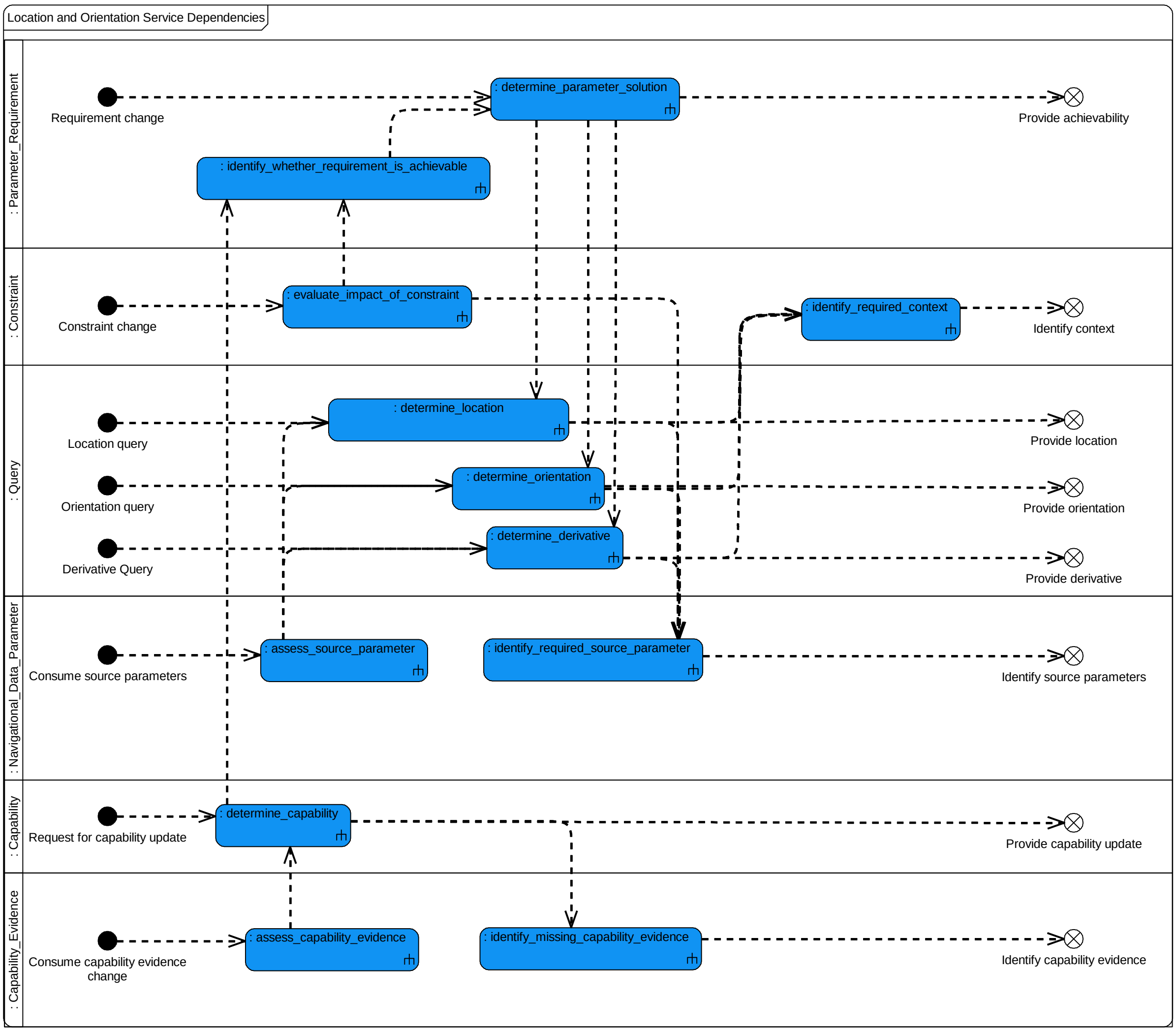

Location and Orientation Service Dependencies
: Parameter_Requirement
: Constraint
: Query
: Navigational_Data_Parameter
: Capability
: Capability_Evidence
Requirement change
: determine_parameter_solution
Provide achievability
: identify_whether_requirement_is_achievable
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Location query
: determine_location
Provide location
Orientation query
: determine_orientation
Provide orientation
Derivative Query
: determine_derivative
Provide derivative
Consume source parameters
: assess_source_parameter
: identify_required_source_parameter
Identify source parameters
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 584: Location and Orientation Service Dependencies

5.4.2.34 Mass and Balance

5.4.2.34.1 Role

The role of Mass and Balance is to determine the impact of contributing elements on the total mass, moments of inertia and centre of mass of configurations.

5.4.2.34.2 Overview

Control Architecture

Mass and Balance is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

The Mass and Balance component gathers data about the current and evolving Total_Mass, Centre_of_Mass and Moment_of_Inertia of the Contributing_Elements of an Exploiting Platform and determines possible changes to keep the Configuration within the Mass_and_Balance_Limits.

Examples of Use

Mass and Balance will be used where:

- The calculation of Total_Mass, balance and Moment_of_Inertia are required in order to safely and efficiently control some aspect of the Exploiting Platform’s operation.

5.4.2.34.3 Service Summary

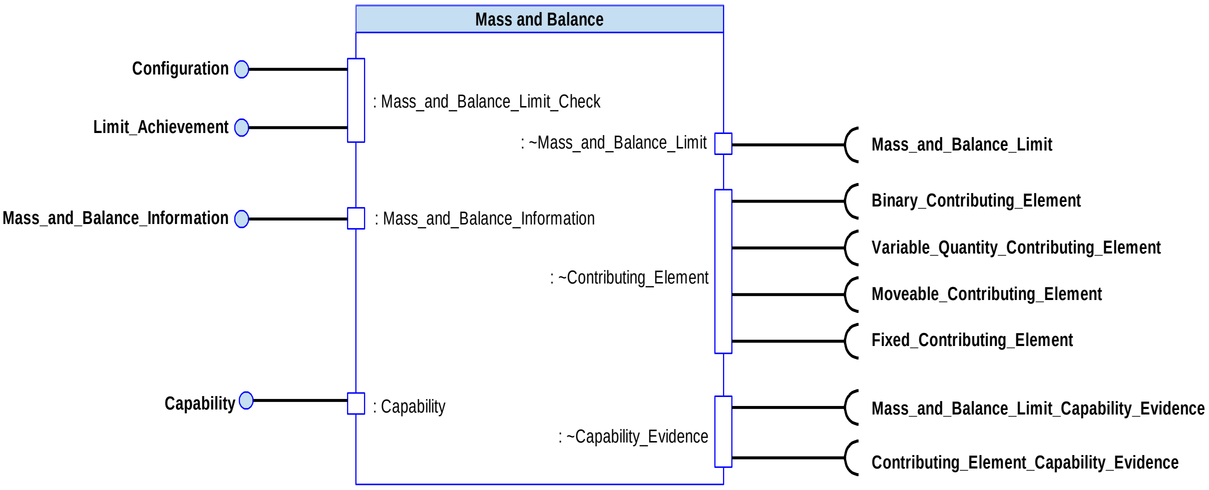

Mass and Balance
Configuration
Limit_Achievement
Mass_and_Balance_Information
Capability
: Mass_and_Balance_Limit_Check
: ~Mass_and_Balance_Limit
: Mass_and_Balance_Information
: ~Contributing_Element
: Capability
: ~Capability_Evidence
Mass_and_Balance_Limit
Binary_Contributing_Element
Variable_Quantity_Contributing_Element
Moveable_Contributing_Element
Fixed_Contributing_Element
Mass_and_Balance_Limit_Capability_Evidence
Contributing_Element_Capability_Evidence

Figure 585: Mass and Balance Service Summary

5.4.2.34.4 Responsibilities

capture_inertia_and_balance_requirements

- To capture requirements for maintaining the Moment_of_Inertia and Centre_of_Mass for a given Configuration.

capture_mass_and_balance_limit

- To capture the Mass_and_Balance_Limits.

determine_changes_conform_to_limits

- To determine whether Configuration changes (e.g. a planned stores release package) conform to Mass_and_Balance_Limits.

determine_total_mass

- To determine the Total_Mass of Contributing_Elements within Configurations.

determine_centre_of_mass

- To determine the Centre_of_Mass of Contributing_Elements within Configurations.

determine_moment_of_inertia

- To determine the Moment_of_Inertia of Contributing_Elements within Configurations.

determine_mass_changes

- To determine Total_Mass, Moment_of_Inertia and Centre_of_Mass changes which would result from variations in the Contributing_Element(s) within Configurations.

determine_optimum_configuration_balance

- To determine the optimum Configuration of Contributing_Elements taking into account Total_Mass, Moment_of_Inertia and Centre_of_Mass in order to maintain balance of the Exploiting Platform.

suggest_mass_reconfiguration

- To determine Contributing_Element(s) which could be varied in order to achieve the desired Total_Mass, Moment_of_Inertia or Centre_of_Mass of a Configuration.

capture_current_mass_and_balance_configs

- To capture current Configurations of Contributing_Elements.

capture_planned_mass_and_balance_configs

- To capture planned Configurations of Contributing_Elements.

assess_mass_and_balance_capability

- To predict the progression of the capability to determine whether potential Configuration changes will conform to mass and balance limits and to determine Configuration changes required to maintain mass and balance limits or optimise the Configuration, over time and with use.

predict_capability_progression

- To predict the progression of available Capability over time and with use.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the capability assessment.

5.4.2.34.5 Subject Matter Semantics

The subject matter of Mass and Balance is the mass, centre of mass and the moments of inertia of the Exploiting Platform.

Exclusions

The subject matter of Mass and Balance does not include:

- The command of actions to prevent or correct an imbalance or exceedance of limits.

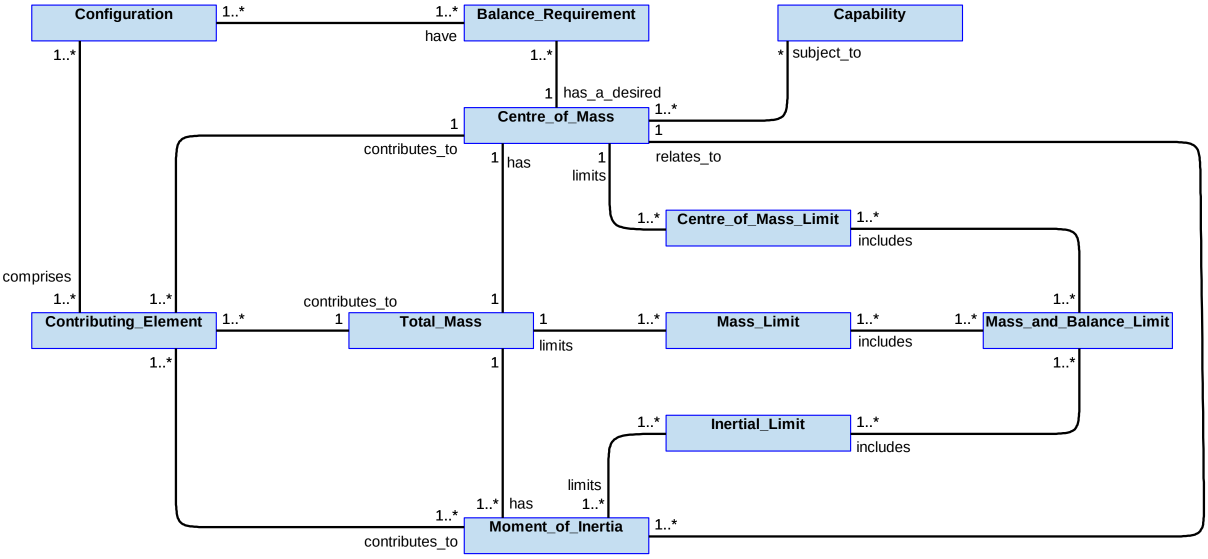


Figure 586: Mass and Balance Semantics

5.4.2.34.5.1 Entities

Balance_Requirement

A requirement for values of Centre_of_Mass placed upon a Configuration of Contributing_Elements.

Capability

The capability to determine Centre_of_Mass and their Derivatives, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Centre_of_Mass

The centre of mass arising from the contributions of Contributing_Elements in a Configuration.

Centre_of_Mass_Limit

A limit for Centre_of_Mass.

Configuration

A group of Contributing_Elements to be reasoned about by Mass and Balance.

Contributing_Element

A physical element with a mass and position.

Inertial_Limit

A limit for Moment_of_Inertia.

Mass_and_Balance_Limit

A set of individual limits for Total_Mass, Moment_of_Inertia and Centre_of_Mass.

Mass_Limit

A limit for Total_Mass.

Moment_of_Inertia

A property of a Configuration (or Contributing_Element thereof) that determines the torque needed for a desired angular acceleration about a rotational axis.

Total_Mass

The combined mass of all Contributing_Elements in a Configuration.

5.4.2.34.6 Design Rationale

5.4.2.34.6.1 Assumptions

- The component will have knowledge of Contributing_Elements and how they affect Total_Mass, Moment_of_Inertia and Centre_of_Mass.

5.4.2.34.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Mass and Balance:

- Data Driving - data driving of the Contributing_Elements, their Configuration and applicable Mass_and_Balance_Limits to enable them to be configured for a particular Exploiting Platform.

Extensions

- It may be appropriate to use extension components to cater for differing calculations that may apply to the various types of vehicle or platform, or the behaviour of some Contributing_Elements, that Mass and Balance could be used for.

Other Factors that were Taken into Account

- When considering the mass of the Exploiting Platform, all relevant variations of mass should be considered, including active and passive gravitational mass, inertial mass, etc.

Exploitation Considerations

- It will be possible for Contributing_Elements to vary, e.g. mass of fuel. It is also possible for an element to cease to contribute, e.g. a released store is no longer a Contributing_Element.

5.4.2.34.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- In the case of an air vehicle, incorrect calculation of mass or centre of gravity, or incorrect determination of store balance rules could cause the air vehicle to be flown outside safe limits and result in uncontrolled flight / loss of structural integrity. This could lead to an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

5.4.2.34.6.4 Security Considerations

The indicative security classification is O-S.

This component is responsible for maintaining the mass, moment of inertia and centre of mass for the vehicle, the details of which are considered to be O-S, however care should be taken where it is possible to deduce performance or fuel loading, etc. through data aggregation, e.g. within any data recorded for audit purposes. As such, appropriate protection is required to be in place to protect the confidentiality of this information. This is one of a series of components that will assist in identifying if form and fit integrity has been interfered with.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of mass, inertia and balance during a mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Reporting **System Status and Monitoring** for unexpected shifts in mass, moment of inertia or balance, and registering configuration feedback following mass transfers (e.g. fuel balancing).
- Providing **Warnings and Notifications** of an imbalance. An unexpected mass or shift in balance may indicate that the form or fit of the Exploiting Platform has been compromised.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.34.7 Services

5.4.2.34.7.1 Service Definitions

5.4.2.34.7.1.1 Mass_and_Balance_Limit_Check

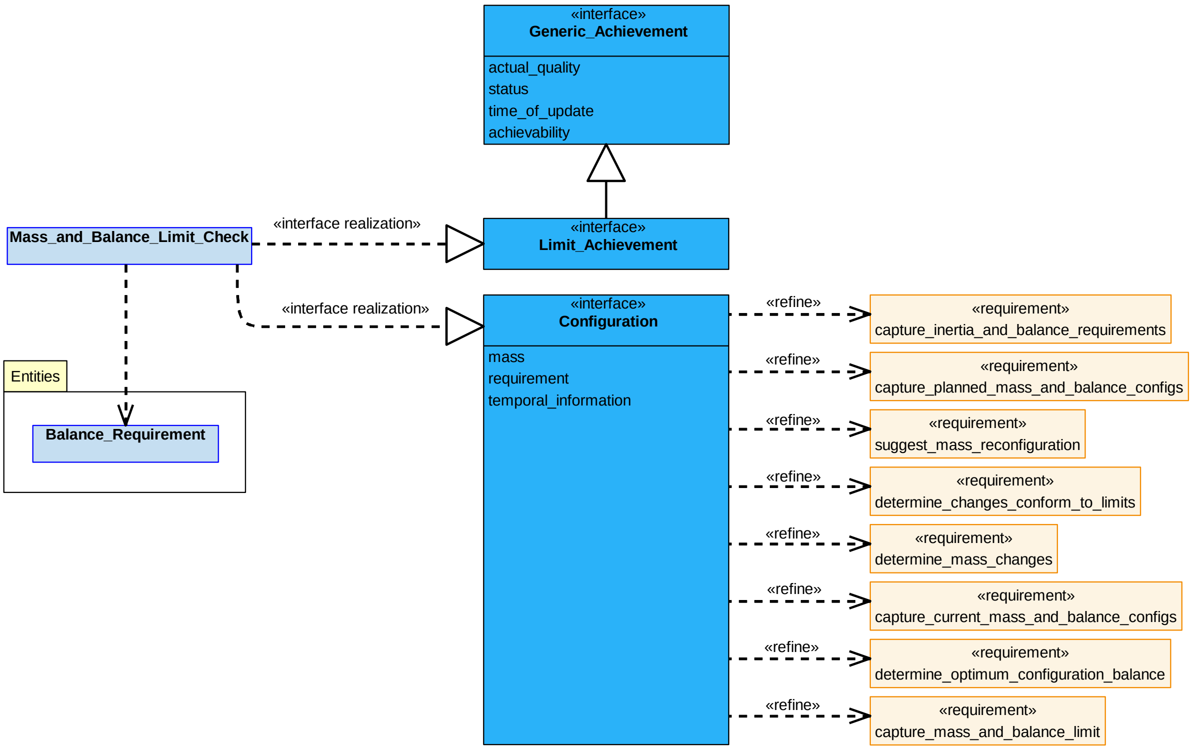
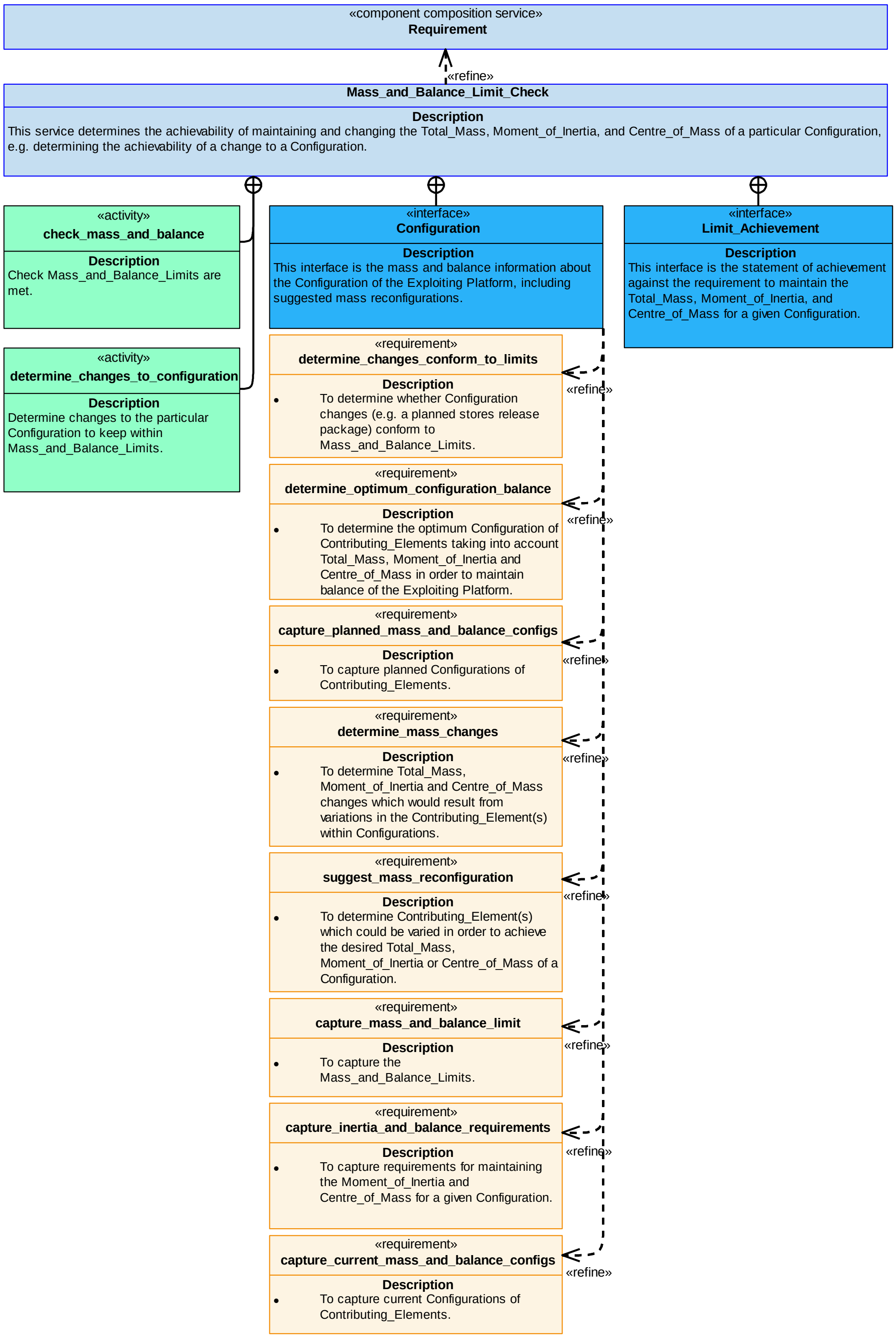


Figure 587: Mass_and_Balance_Limit_Check Service Definition



This service determines the achievability of maintaining and changing the Total_Mass, Moment_of_Inertia, and Centre_of_Mass of a particular Configuration, e.g. determining the achievability of a change to a Configuration.

Check Mass_and_Balance_Limits are met.

Determine changes to the particular Configuration to keep within Mass_and_Balance_Limits.

This interface is the mass and balance information about the Configuration of the Exploiting Platform, including suggested mass reconfigurations.

This interface is the statement of achievement against the requirement to maintain the Total_Mass, Moment_of_Inertia, and Centre_of_Mass for a given Configuration.

- To determine whether Configuration changes (e.g. a planned stores release package) conform to Mass_and_Balance_Limits.

- To determine the optimum Configuration of Contributing_Elements taking into account Total_Mass, Moment_of_Inertia and Centre_of_Mass in order to maintain balance of the Exploiting Platform.

- To capture planned Configurations of Contributing_Elements.

- To determine Total_Mass, Moment_of_Inertia and Centre_of_Mass changes which would result from variations in the Contributing_Element(s) within Configurations.

- To determine Contributing_Element(s) which could be varied in order to achieve the desired Total_Mass, Moment_of_Inertia or Centre_of_Mass of a Configuration.

- To capture the Mass_and_Balance_Limits.

- To capture requirements for maintaining the Moment_of_Inertia and Centre_of_Mass for a given Configuration.

- To capture current Configurations of Contributing_Elements.

Figure 588: Mass_and_Balance_Limit_Check Service Policy

Mass_and_Balance_Limit_Check

This service determines the achievability of maintaining and changing the Total_Mass, Moment_of_Inertia, and Centre_of_Mass of a particular Configuration, e.g. determining the achievability of a change to a Configuration.

Interfaces

Configuration

This interface is the mass and balance information about the Configuration of the Exploiting Platform, including suggested mass reconfigurations.

Attributes

mass	The mass and balance information about a Configuration, e.g. each Contributing_Element's mass contribution.
requirement	The definition of the requirement to manage mass and balance, e.g. a reconfiguration.
temporal_information	Information covering timing, such as start and end times.

Limit_Achievement

This interface is the statement of achievement against the requirement to maintain the Total_Mass, Moment_of_Inertia, and Centre_of_Mass for a given Configuration.

Activities

check_mass_and_balance

Check Mass_and_Balance_Limits are met.

determine_changes_to_configuration

Determine changes to the particular Configuration to keep within Mass_and_Balance_Limits.

5.4.2.34.7.1.2 Mass_and_Balance_Limit

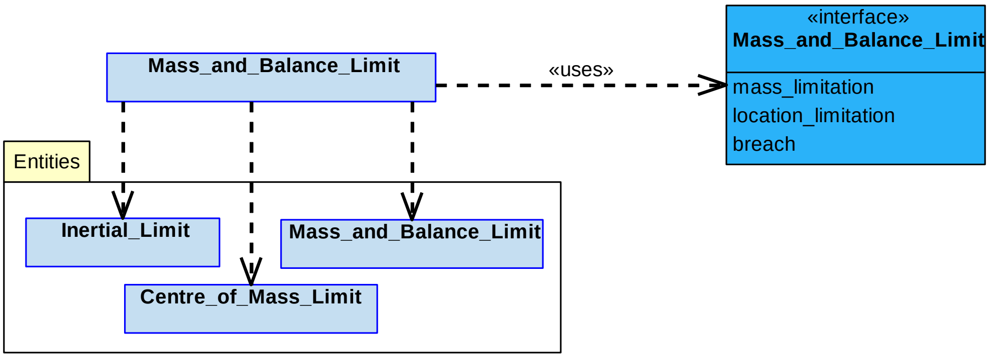


Figure 589: Mass_and_Balance_Limit Service Definition

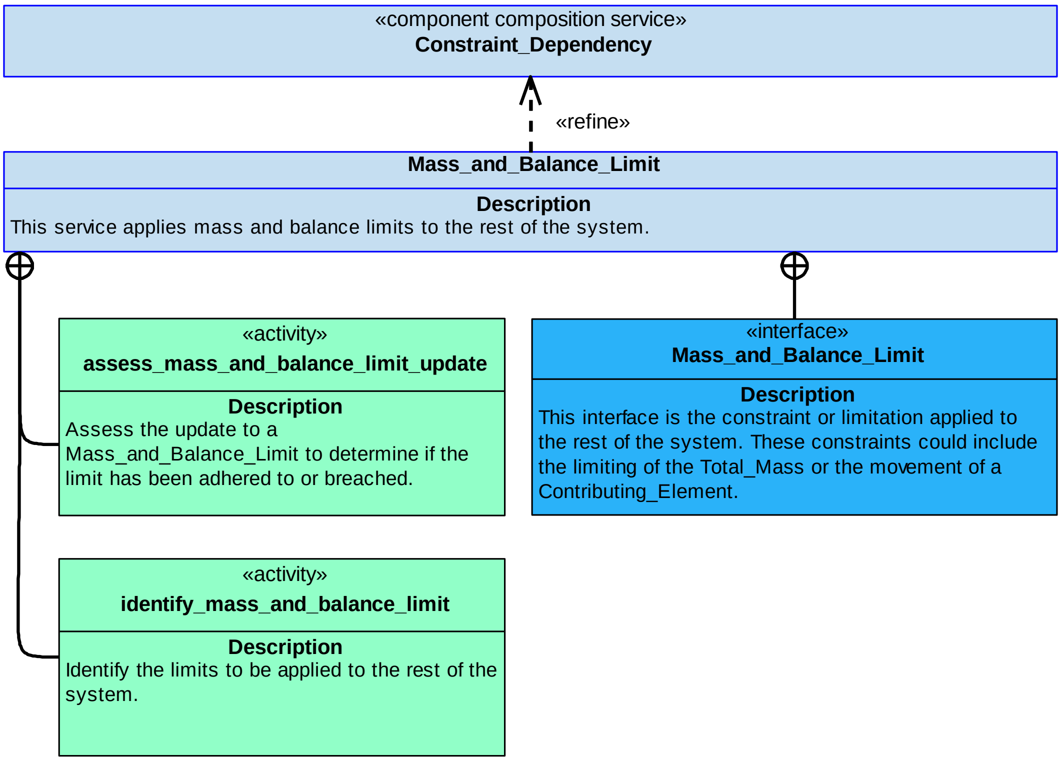

«component composition service»
Constraint_Dependency
«refine»
Mass_and_Balance_Limit
Description
This service applies mass and balance limits to the rest of the system.
«activity»
assess_mass_and_balance_limit_update
Description
Assess the update to a Mass_and_Balance_Limit to determine if the limit has been adhered to or breached.
«interface»
Mass_and_Balance_Limit
Description
This interface is the constraint or limitation applied to the rest of the system. These constraints could include the limiting of the Total_Mass or the movement of a Contributing_Element.
«activity»
identify_mass_and_balance_limit
Description
Identify the limits to be applied to the rest of the system.

Figure 590: Mass_and_Balance_Limit Service Policy

Mass_and_Balance_Limit

This service applies mass and balance limits to the rest of the system.

Interface

Mass_and_Balance_Limit

This interface is the constraint or limitation applied to the rest of the system. These constraints could include the limiting of the Total_Mass or the movement of a Contributing_Element.

Attributes

mass_limitation	A limit on the mass of a Contributing_Element in a particular location.
location_limitation	A limit on the location of a Contributing_Element.
breach	A statement that a Contributing_Element is breaching a limitation, or is likely to breach a limitation if enforced.

Activities

assess_mass_and_balance_limit_update

Assess the update to a Mass_and_Balance_Limit to determine if the limit has been adhered to or breached.

identify_mass_and_balance_limit

Identify the limits to be applied to the rest of the system.

5.4.2.34.7.1.3 Mass_and_Balance_Information

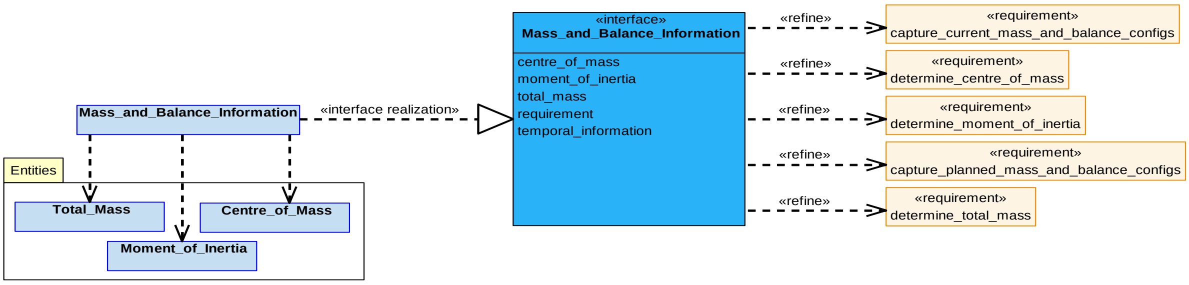


Figure 591: Mass_and_Balance_Information Service Definition

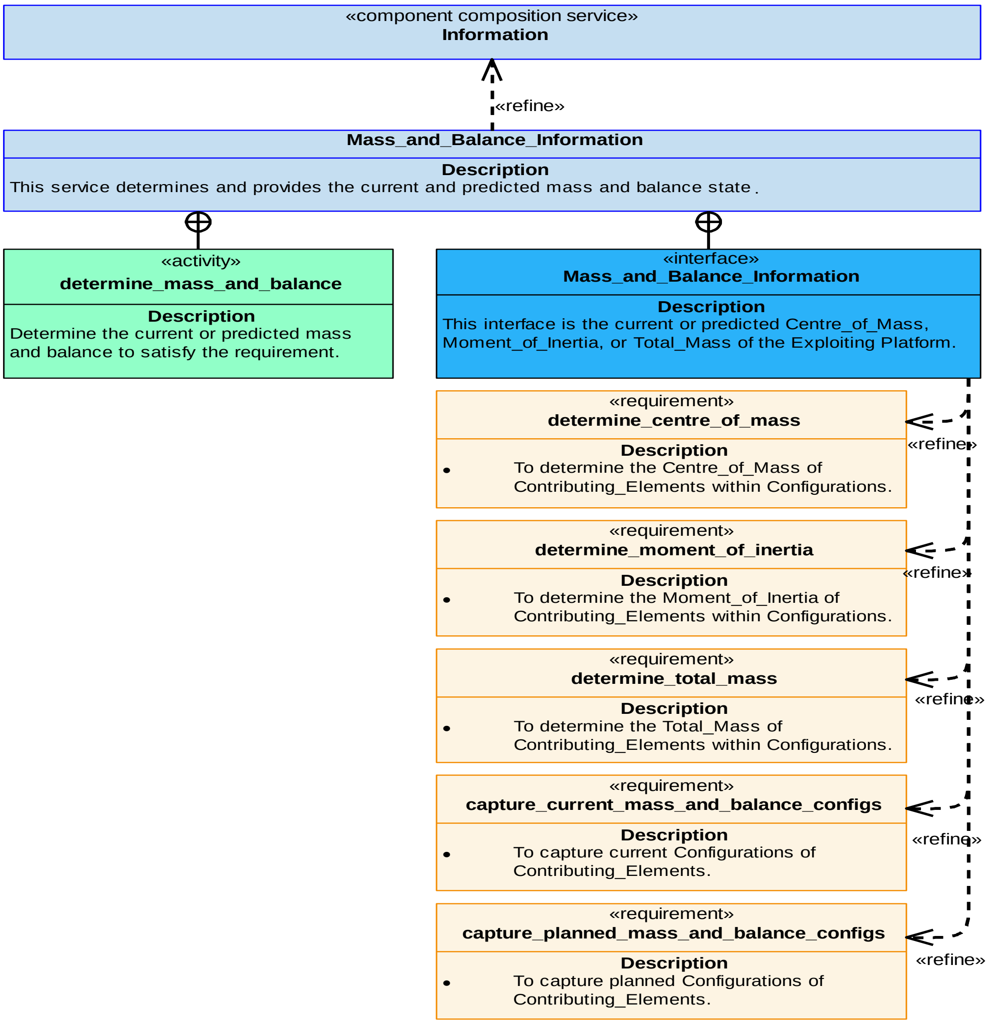

«component composition service»
Information
«refine»
Mass_and_Balance_Information
Description
This service determines and provides the current and predicted mass and balance state .
«activity»
determine_mass_and_balance
Description
Determine the current or predicted mass and balance to satisfy the requirement.
«interface»
Mass_and_Balance_Information
Description
This interface is the current or predicted Centre_of_Mass, Moment_of_Inertia, or Total_Mass of the Exploiting Platform.
«requirement»
determine_centre_of_mass
Description
To determine the Centre_of_Mass of Contributing_Elements within Configurations.
«requirement»
determine_moment_of_inertia
Description
To determine the Moment_of_Inertia of Contributing_Elements within Configurations.
«requirement»
determine_total_mass
Description
To determine the Total_Mass of Contributing_Elements within Configurations.
«requirement»
capture_current_mass_and_balance_configs
Description
To capture current Configurations of Contributing_Elements.
«requirement»
capture_planned_mass_and_balance_configs
Description
To capture planned Configurations of Contributing_Elements.
«refine»

Figure 592: Mass_and_Balance_Information Service Policy

Mass_and_Balance_Information

This service determines and provides the current and predicted mass and balance state.

Interface

Mass_and_Balance_Information

This interface is the current or predicted Centre_of_Mass, Moment_of_Inertia, or Total_Mass of the Exploiting Platform.

Attributes

centre_of_mass	The information about Centre_of_Mass.
moment_of_inertia	The information about Moment_of_Inertia.
total_mass	The information about Total_Mass.
requirement	The definition of what information is required.
temporal_information	Information covering timing, such as the time a prediction was made, and the time of an event for which a prediction was made.

Activity

determine_mass_and_balance

Determine the current or predicted mass and balance to satisfy the requirement.

5.4.2.34.7.1.4 Contributing_Element

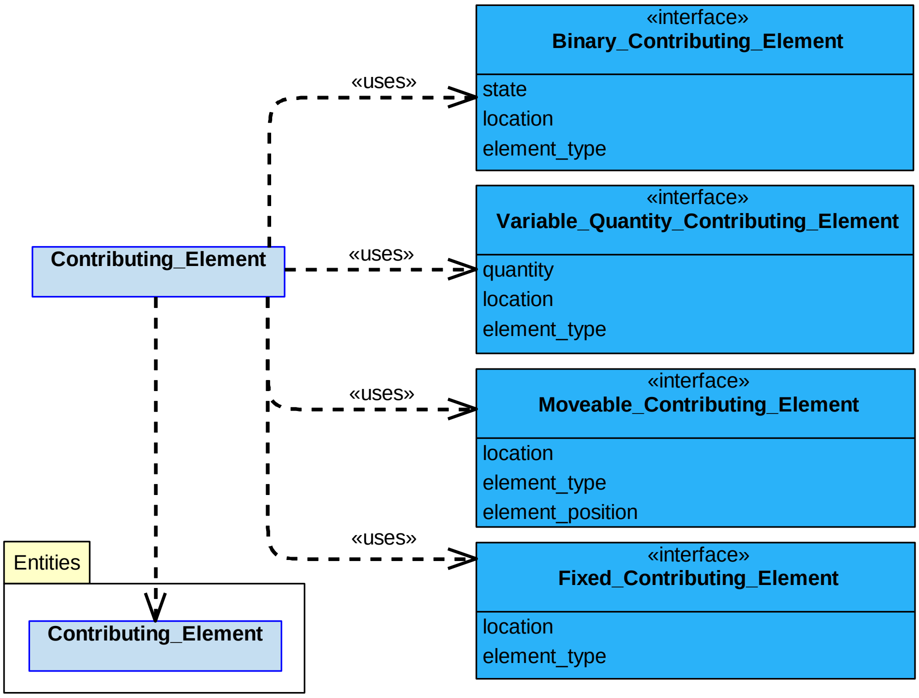


Figure 593: Contributing_Element Service Definition

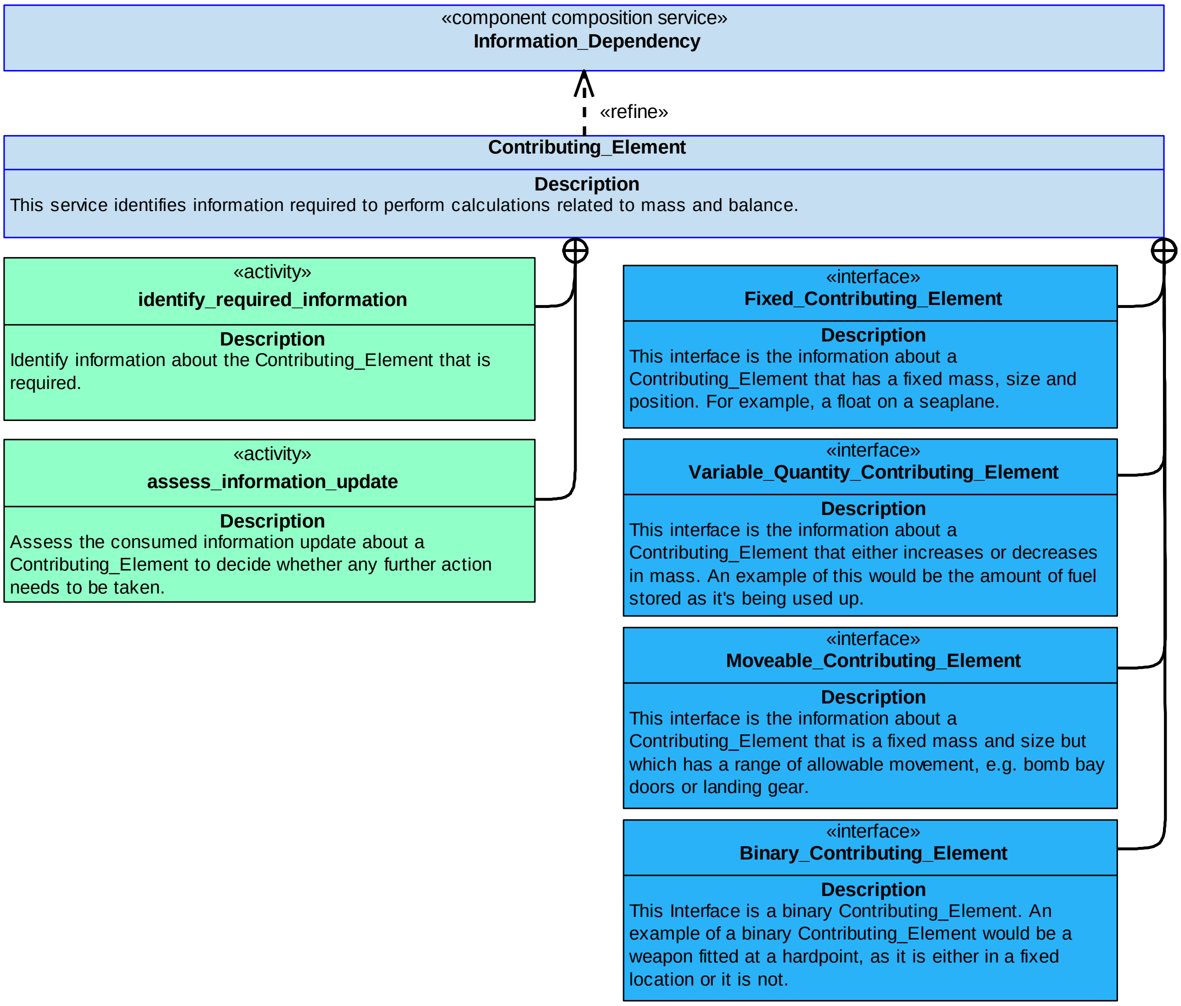

«component composition service»
Information_Dependency
«refine»
Contributing_Element
Description
This service identifies information required to perform calculations related to mass and balance.
«activity»
identify_required_information
Description
Identify information about the Contributing_Element that is required.
«activity»
assess_information_update
Description
Assess the consumed information update about a Contributing_Element to decide whether any further action needs to be taken.
«interface»
Fixed_Contributing_Element
Description
This interface is the information about a Contributing_Element that has a fixed mass, size and position. For example, a float on a seaplane.
«interface»
Variable_Quantity_Contributing_Element
Description
This interface is the information about a Contributing_Element that either increases or decreases in mass. An example of this would be the amount of fuel stored as it's being used up.
«interface»
Moveable_Contributing_Element
Description
This interface is the information about a Contributing_Element that is a fixed mass and size but which has a range of allowable movement, e.g. bomb bay doors or landing gear.
«interface»
Binary_Contributing_Element
Description
This Interface is a binary Contributing_Element. An example of a binary Contributing_Element would be a weapon fitted at a hardpoint, as it is either in a fixed location or it is not.

Figure 594: Contributing_Element Service Policy

Contributing_Element

This service identifies information required to perform calculations related to mass and balance.

Interfaces

Binary_Contributing_Element

This Interface is a binary Contributing_Element. An example of a binary Contributing_Element would be a weapon fitted at a hardpoint, as it is either in a fixed location or it is not.

Attributes

state	Presence or absence of an element.
location	The location of a Contributing_Element relative to an Exploiting Platform.
element_type	The type of element, e.g. a specific type of a weapon.

Variable_Quantity_Contributing_Element

This interface is the information about a Contributing_Element that either increases or decreases in mass. An example of this would be the amount of fuel stored as it's being used up.

Attributes

quantity	The amount(s) of elements remaining. For example, the amount of fuel left.
location	The positional information of a Contributing_Element.
element_type	The type of element, e.g. fuel.

Moveable_Contributing_Element

This interface is the information about a Contributing_Element that is a fixed mass and size but which has a range of allowable movement, e.g. bomb bay doors or landing gear.

Attributes

location	The location of a Contributing_Element on an Exploiting Platform, e.g. the left hand side (for landing gear), or rear (for bomb bay door).
element_type	The type of element.
element_position	The position of the element within a range of allowable movement. For example, landing gear may be fully lowered, retracted, or any position between.

Fixed_Contributing_Element

This interface is the information about a Contributing_Element that has a fixed mass, size and position. For example, a float on a seaplane.

Attributes

location	The location of a Contributing_Element on an Exploiting Platform.
element_type	The type of element.

Activities

assess_information_update

Assess the consumed information update about a Contributing_Element to decide whether any further action needs to be taken.

identify_required_information

Identify information about the Contributing_Element that is required.

5.4.2.34.7.1.5 Capability

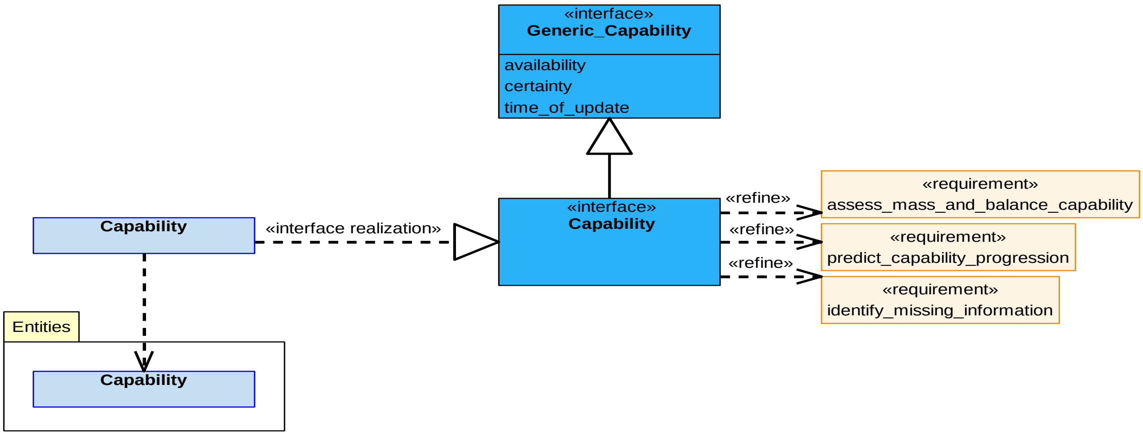
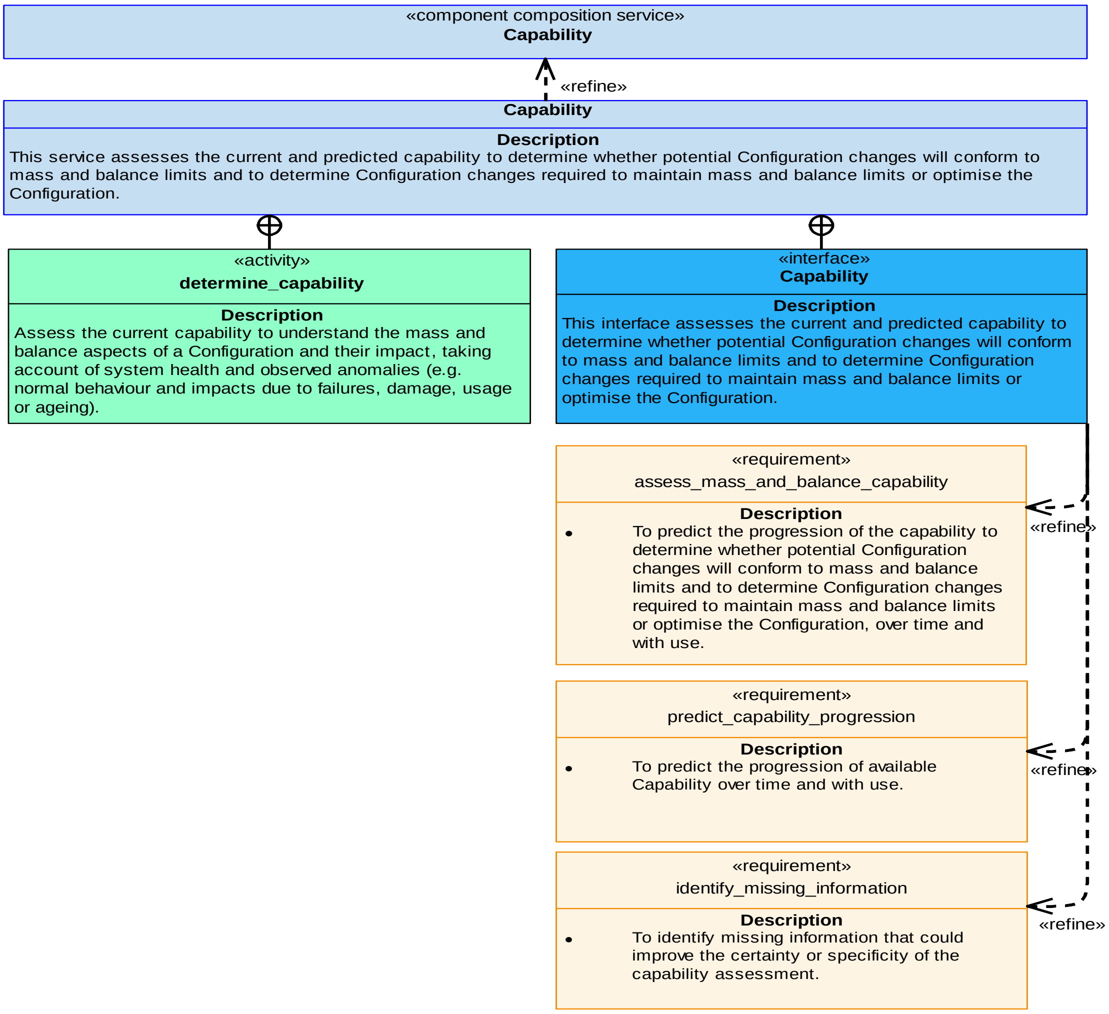


Figure 595: Capability Service Definition



This service assesses the current and predicted capability to determine whether potential Configuration changes will conform to mass and balance limits and to determine Configuration changes required to maintain mass and balance limits or optimise the Configuration.

Assess the current capability to understand the mass and balance aspects of a Configuration and their impact, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

This interface assesses the current and predicted capability to determine whether potential Configuration changes will conform to mass and balance limits and to determine Configuration changes required to maintain mass and balance limits or optimise the Configuration.

- To predict the progression of the capability to determine whether potential Configuration changes will conform to mass and balance limits and to determine Configuration changes required to maintain mass and balance limits or optimise the Configuration, over time and with use.

- To predict the progression of available Capability over time and with use.

- To identify missing information that could improve the certainty or specificity of the capability assessment.

Figure 596: Capability Service Policy

Capability

This service assesses the current and predicted capability to determine whether potential Configuration changes will conform to mass and balance limits and to determine Configuration changes required to maintain mass and balance limits or optimise the Configuration.

Interface

Capability

This interface assesses the current and predicted capability to determine whether potential Configuration changes will conform to mass and balance limits and to determine Configuration changes required to maintain mass and balance limits or optimise the Configuration.

Activity

determine_capability

Assess the current capability to understand the mass and balance aspects of a Configuration and their impact, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.34.7.1.6 Capability_Evidence

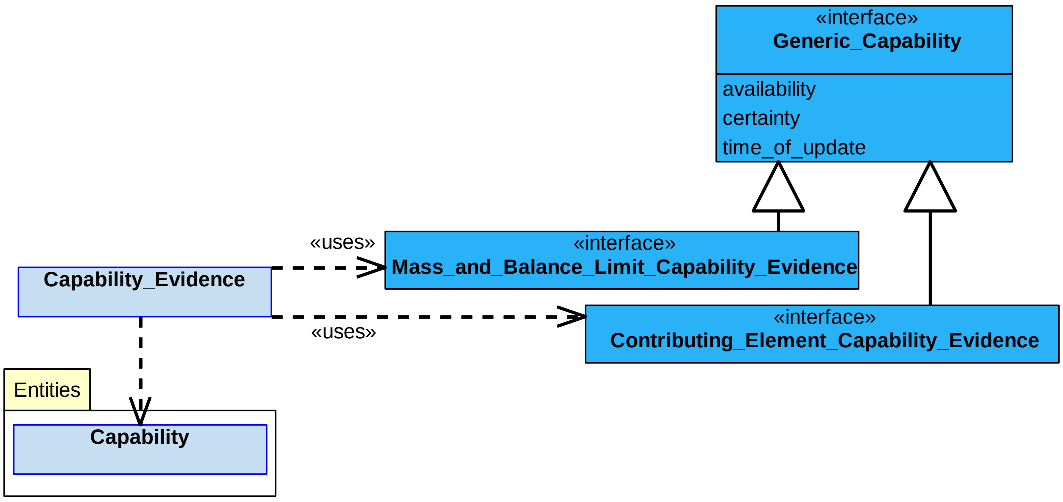


Figure 597: Capability_Evidence Service Definition

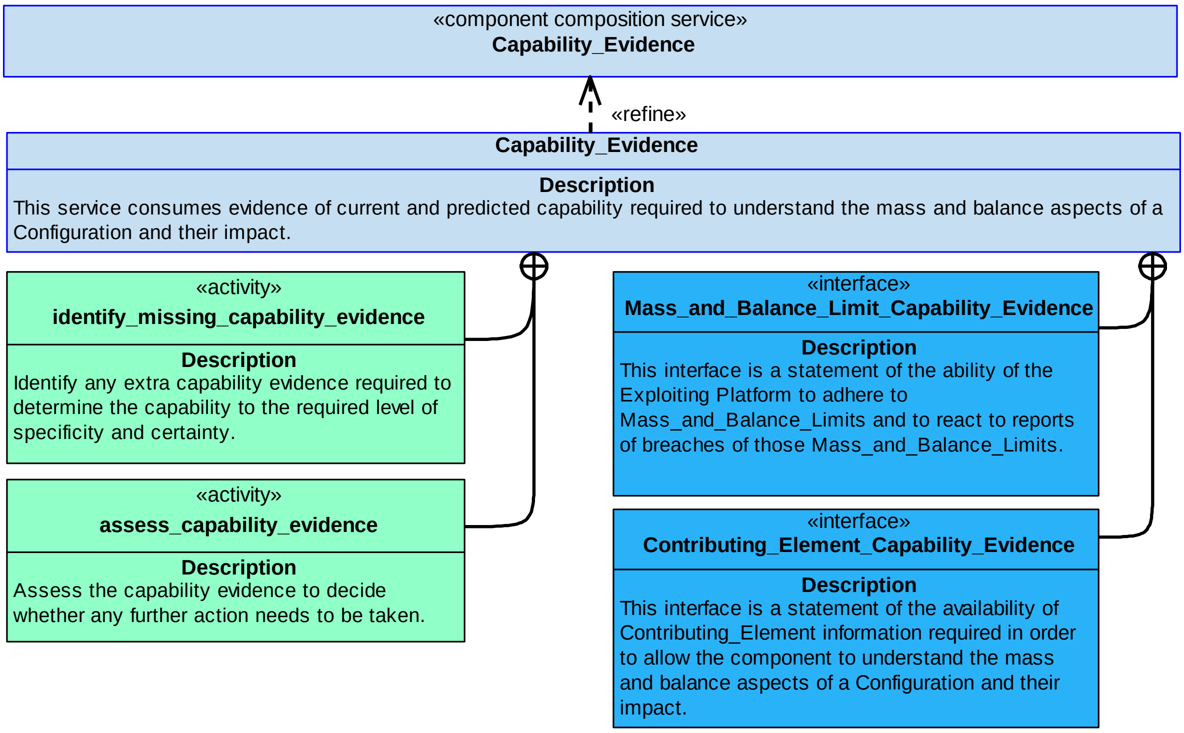

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes evidence of current and predicted capability required to understand the mass and balance aspects of a Configuration and their impact.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«interface»
Mass_and_Balance_Limit_Capability_Evidence
Description
This interface is a statement of the ability of the Exploiting Platform to adhere to Mass_and_Balance_Limits and to react to reports of breaches of those Mass_and_Balance_Limits.
«interface»
Contributing_Element_Capability_Evidence
Description
This interface is a statement of the availability of Contributing_Element information required in order to allow the component to understand the mass and balance aspects of a Configuration and their impact.

Figure 598: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence of current and predicted capability required to understand the mass and balance aspects of a Configuration and their impact.

Interfaces

Mass_and_Balance_Limit_Capability_Evidence

This interface is a statement of the ability of the Exploiting Platform to adhere to Mass_and_Balance_Limits and to react to reports of breaches of those Mass_and_Balance_Limits.

Contributing_Element_Capability_Evidence

This interface is a statement of the availability of Contributing_Element information required in order to allow the component to understand the mass and balance aspects of a Configuration and their impact.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.34.7.2 Service Dependencies

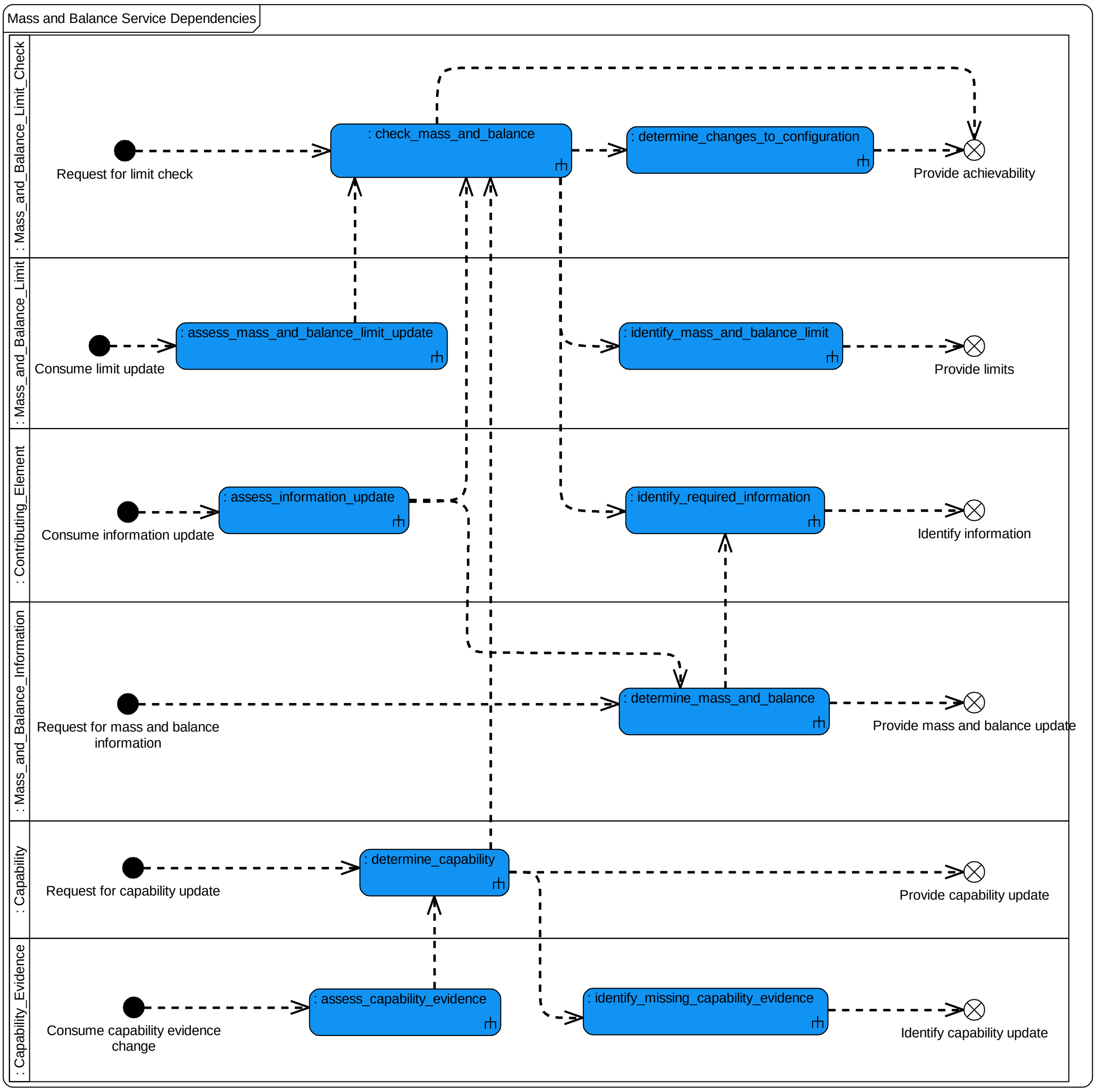

Mass and Balance Service Dependencies
: Mass_and_Balance_Limit_Check
: Mass_and_Balance_Limit
: Contributing_Element
: Mass_and_Balance_Information
: Capability
: Capability_Evidence
Request for limit check
: check_mass_and_balance
: determine_changes_to_configuration
Provide achievability
Consume limit update
: assess_mass_and_balance_limit_update
: identify_mass_and_balance_limit
Provide limits
Consume information update
: assess_information_update
: identify_required_information
Identify information
Request for mass and balance information
: determine_mass_and_balance
Provide mass and balance update
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability update

Figure 599: Mass and Balance Service Dependencies

5.4.2.35 Mechanical Positioning

5.4.2.35.1 Role

The role of Mechanical Positioning is to control the position of an element of a physical structure (e.g. a door, flight control surface, or landing gear) in relation to that structure.

5.4.2.35.2 Overview

Control Architecture

Mechanical Positioning is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When there is a Requirement for movement of a Physical_Element on the Exploiting Platform, Mechanical Positioning will determine the required position of its associated Effector(s) (e.g. the position of a hydraulic actuator or rotation of a servomotor) to attain the position required by the Physical_Element.

Examples of Use

Mechanical Positioning will be used for all mechanically operated Physical_Elements, such as:

- Doors
- Flight control surfaces

5.4.2.35.3 Service Summary

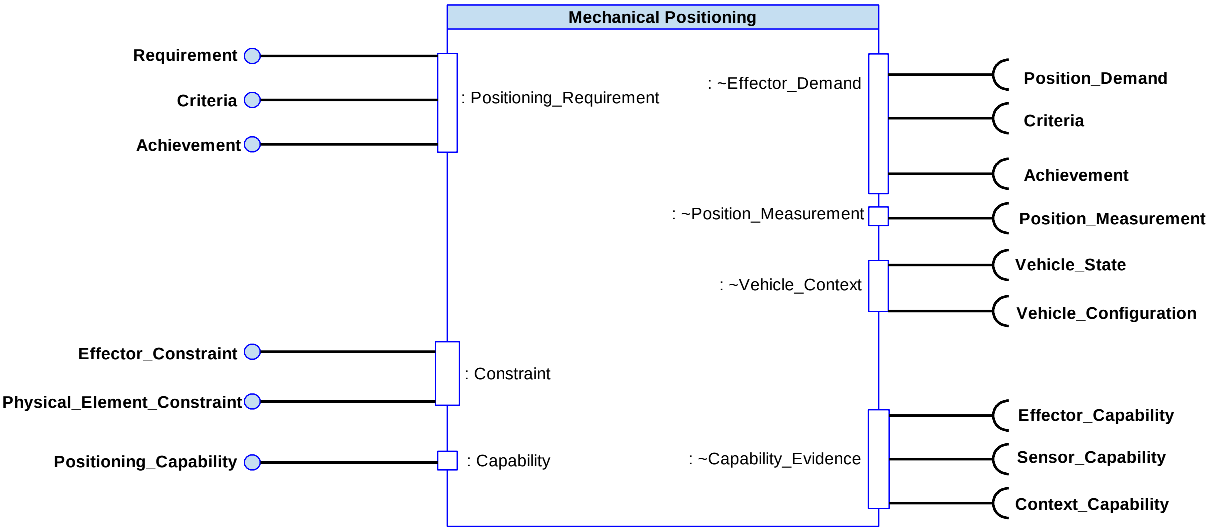


Figure 600: Mechanical Positioning Service Summary

5.4.2.35.4 Responsibilities

capture_positioning_requirements

- To capture provided positioning Requirements for a Physical_Element.

capture_measurement_criteria

- To capture given Measurement_Criterion.

capture_positioning_constraints

- To capture provided Physical_Element_Constraints for a Physical_Element.

capture_effector_constraints

- To capture provided Effector_Constraints for an Effector.

determine_required_position_solution

- To determine the Effector positions to achieve the required position of a Physical_Element.

identify_positioning_solution_in_progress_remains_feasible

- To identify whether a Positioning_Solution in progress remains feasible against particular Requirements and Measurement_Criterion/criteria given current resources.

control_position

- To control the position of a Physical_Element by placing positioning dependencies on Effectors.

identify_positioning_progress

- To identify the progress of a Positioning_Solution against a Requirement.

assess_movement_capability

- To assess the Movement_Capability of a Physical_Element taking account of the capability of the Effector(s), system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_capability_information

- To identify missing information which could improve the certainty or specificity of Movement_Capability determination.

predict_movement_capability_progression

- To predict the progression of the Movement_Capability of a Physical_Element over time and with use.

5.4.2.35.5 Subject Matter Semantics

The subject matter of Mechanical Positioning is moveable Physical_Elements and their positions (including orientations).

Exclusions

The subject matter of Mechanical Positioning does not include:

- The capability delivered by a Physical_Element it is controlling.
- How positional feedback is provided.
- The direct interaction with effectors or an understanding of how a required Effector position is achieved.

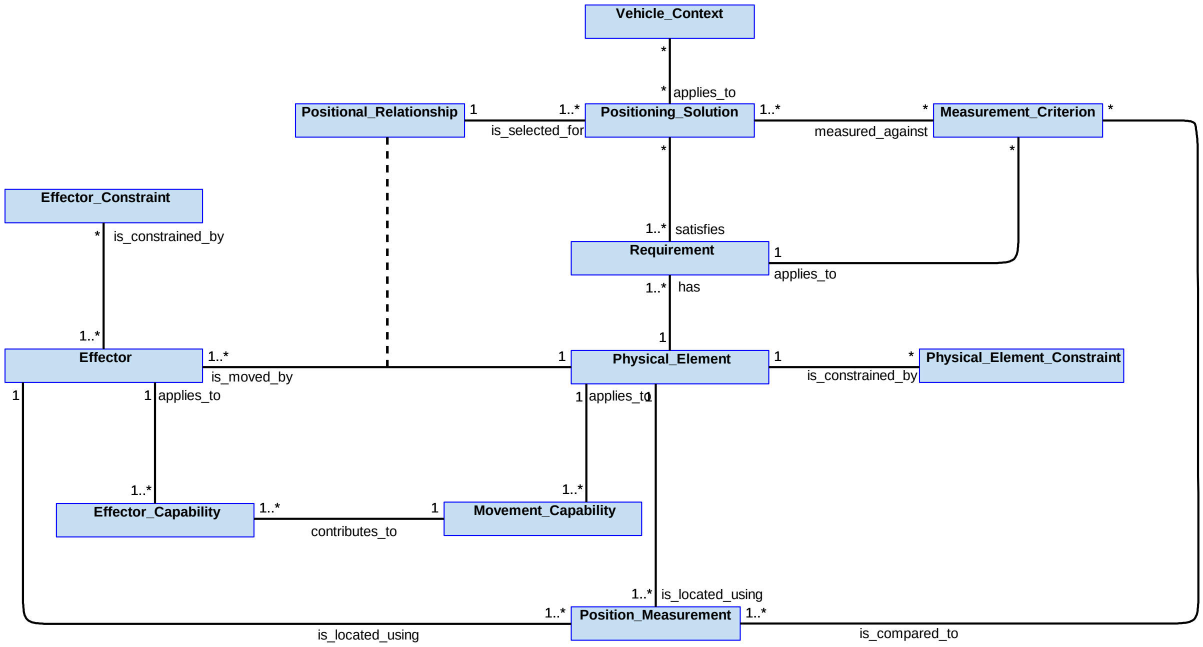


Figure 601: Mechanical Positioning Semantics

5.4.2.35.5.1 Entities

Effector

Something used to enact a desired change in the position (including orientation) of an associated Physical_Element, e.g. a control surface actuator or a servo motor.

Effector_Constraint

A limitation on the way that an Effector may be used to meet a Requirement, e.g. range or rate of movement.

Physical_Element

An element on the Exploiting Platform that needs to be positioned. This will primarily be a structural entity (e.g. a door, control surface, or undercarriage) but may include other positionable items of role fit equipment.

Physical_Element_Constraint

A limitation on the way that a Physical_Element may be moved or positioned.

Positional_Relationship

The relationship between the position (including orientation) or movement of an Effector and that of its associated Physical_Element, e.g. that 20mm of actuator travel equates to 5 degrees of control surface deflection.

Requirement

The requirement to move a Physical_Element to a defined position (including orientation).

Movement_Capability

The ability of the component to implement a positioning Requirement for a Physical_Element.

Position_Measurement

The determined values about the position of an Effector or a Physical_Element.

Effector_Capability

Capability of an Effector to position a Physical_Element.

Measurement_Criterion

A criterion used to determine the quality of a Positioning_Solution against a Requirement.

Positioning_Solution

A selected set of Effector actions that can be used to achieve a required Physical_Element position.

Vehicle_Context

Information about the context in which the Positioning_Solution is being carried out, e.g. airspeed and orientation of the platform.

5.4.2.35.6 Design Rationale

5.4.2.35.6.1 Assumptions

- Several Effectors may be used to control a single Physical_Element (e.g. a number of actuators attached to a control surface); however, each Effector will only control a single Physical_Element. Therefore, separate instances of the Mechanical Positioning component could be used to independently control different Physical_Element.
- The determination of the safety related limits for an Effector's position or use would have been agreed outside this component.
- This component understands the way that any part of a Physical_Element can be moved but does not have the performance details of an individual Effector.

5.4.2.35.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Mechanical Positioning:

- Resource Management - Each instantiation of Mechanical Positioning will represent one or more Physical_Element resource and its associated Effector(s) and hence will allow for resource management.
- Data Driving - The design of an Exploiting Platform will define all Effectors that are needed to control Physical_Elements. Therefore, the full extent of the data within the component could be data-driven using build time data to provide configurability.

Extensions

- It is not expected that extension components will be needed.

Exploitation Considerations

- There are numerous methods for controlling Effectors with varying levels of complexity, e.g. closed loop position control, endstop sensors or specific positions only. Defining this information during the development process means the component has reusability.
- The abstraction of the purpose of an Effector (to move control surfaces, open doors, etc.) provides the component with reusability across multiple Exploiting Programmes and resilience against obsolescence to Exploiting Platforms with different needs.
- Mechanical Positioning is responsible for determining the required effector position to achieve a given physical element position, whilst the Effectors component would actually control the Effectors themselves.

5.4.2.35.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- The uses of this component include determining the required position of Effectors that position Physical_Elements including control surfaces, undercarriage, airbrakes, thrust reversers, weapon bay doors and aperture doors. In the case of an air vehicle, the most severe cases of failure for this component would cause uncontrolled flight due to exceedance of the flight envelope or structural limits. This could lead to an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.
- Particular instances of the component may be developed to lower DALs where the consequences of failure are less severe. For example, where a navigation light is concealed behind a door within the air vehicle structure.

5.4.2.35.6.4 Security Considerations

The indicative security classification is O-S.

This component controls Physical_Elements on an Exploiting Platform including control surfaces, undercarriage and doors, etc. As such, the security classification is considered unlikely to be above O-S. The integrity and availability of this component can have an impact on the combat effectiveness of the Exploiting Platform, e.g. unauthorised opening of apertures can adversely affect the signature of the platform, and inability to open them may prevent the delivery of weapons to the target. Integrity and availability will need to be protected according to the elements being controlled by the component.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to positioning of Physical_Elements during the mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** including feedback on requested and actual element position (in association with sensors), with deviation from the expected position being a possible sign of cyber activity.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.35.7 Services

5.4.2.35.7.1 Service Definitions

5.4.2.35.7.1.1 Positioning_Requirement

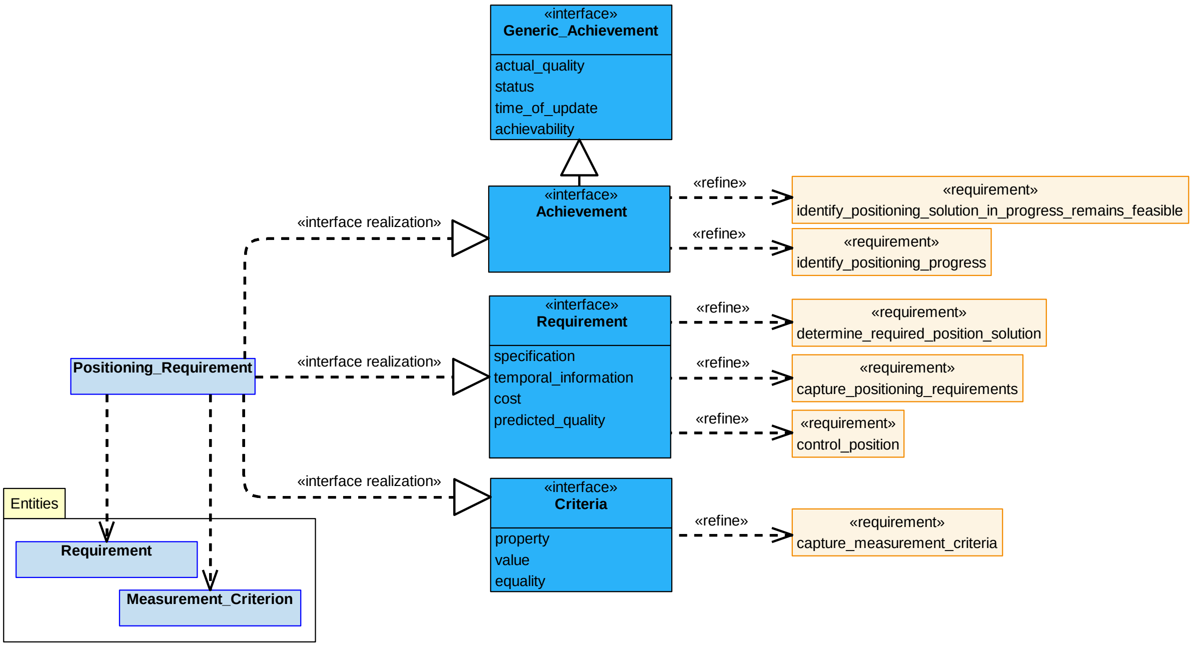


Figure 602: Positioning_Requirement Service Definition

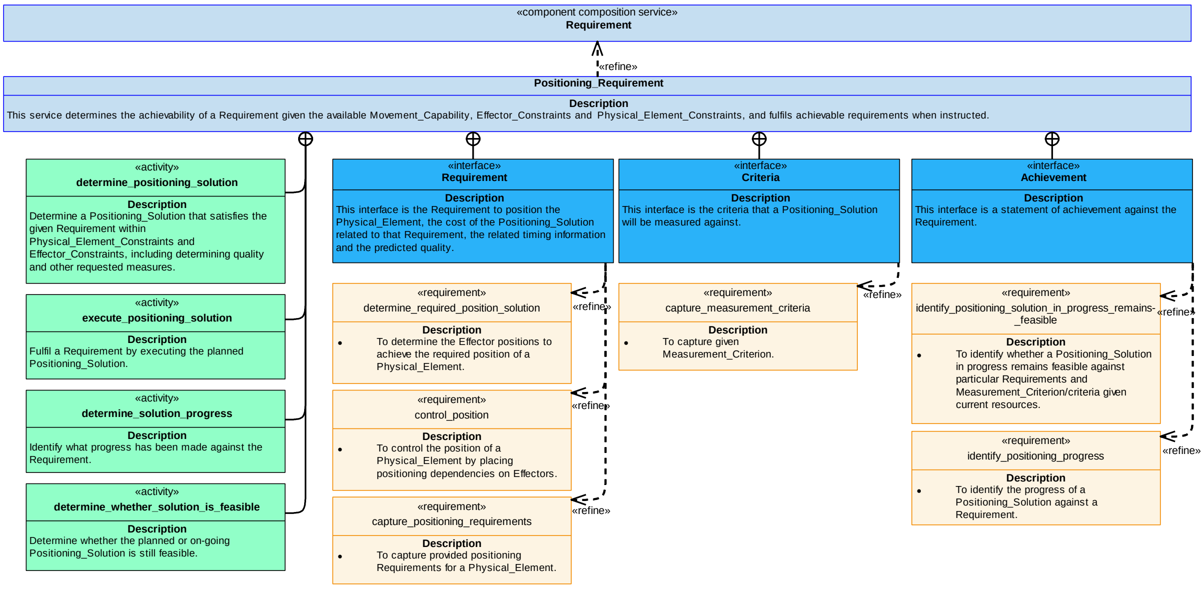

«component composition service»
Requirement
«refine»
Positioning_Requirement
Description
This service determines the achievability of a Requirement given the available Movement_Capability, Effector_Constraints and Physical_Element_Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_positioning_solution
Description
Determine a Positioning_Solution that satisfies the given Requirement within Physical_Element_Constraints and Effector_Constraints, including determining quality and other requested measures.
«activity»
execute_positioning_solution
Description
Fulfil a Requirement by executing the planned Positioning_Solution.
«activity»
determine_solution_progress
Description
Identify what progress has been made against the Requirement.
«activity»
determine_whether_solution_is_feasible
Description
Determine whether the planned or on-going Positioning_Solution is still feasible.
«interface»
Requirement
Description
This interface is the Requirement to position the Physical_Element, the cost of the Positioning_Solution related to that Requirement, the related timing information and the predicted quality.
«interface»
Criteria
Description
This interface is the criteria that a Positioning_Solution will be measured against.
«interface»
Achievement
Description
This interface is a statement of achievement against the Requirement.
«requirement»
determine_required_position_solution
Description
To determine the Effector positions to achieve the required position of a Physical_Element.
«refine»
«requirement»
control_position
Description
To control the position of a Physical_Element by placing positioning dependencies on Effectors.
«refine»
«requirement»
capture_positioning_requirements
Description
To capture provided positioning Requirements for a Physical_Element.
«refine»
«requirement»
capture_measurement_criteria
Description
To capture given Measurement_Criterion.
«refine»
«requirement»
identify_positioning_solution_in_progress_remains_feasible
Description
To identify whether a Positioning_Solution in progress remains feasible against particular Requirements and Measurement_Criterion/criteria given current resources.
«refine»
«requirement»
identify_positioning_progress
Description
To identify the progress of a Positioning_Solution against a Requirement.
«refine»

Figure 603: Positioning_Requirement Service Policy

Positioning_Requirement

This service determines the achievability of a Requirement given the available Movement_Capability, Effector_Constraints and Physical_Element_Constraints, and fulfils achievable requirements when instructed.

Interfaces

Requirement

This interface is the Requirement to position the Physical_Element, the cost of the Positioning_Solution related to that Requirement, the related timing information and the predicted quality.

Attributes

specification	The definition of the Requirement, e.g. to move a Physical_Element to a new position and the required rate of travel.
temporal_information	Information covering timing, such as the positioning start and end times.
cost	The cost of positioning the Physical_Element, e.g. resources expended.
predicted_quality	How well the Positioning_Solution is predicted to meet the Requirement.

Criteria

This interface is the criteria that a Positioning_Solution will be measured against.

Attributes

property	The property to be measured, e.g. the tolerance in the final position of the Physical_Element.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than or equal to.

Achievement

This interface is a statement of achievement against the Requirement.

Activities

determine_positioning_solution

Determine a Positioning_Solution that satisfies the given Requirement within Physical_Element_Constraints and Effector_Constraints, including determining quality and other requested measures.

execute_positioning_solution

Fulfil a Requirement by executing the planned Positioning_Solution.

determine_solution_progress

Identify what progress has been made against the Requirement.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Positioning_Solution is still feasible.

5.4.2.35.7.1.2 Effector_Demand

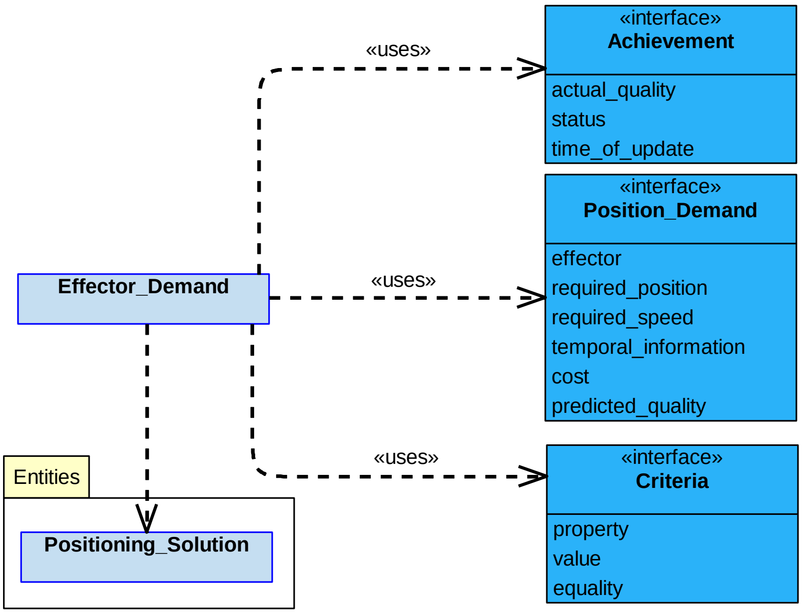


Figure 604: Effector_Demand Service Definition

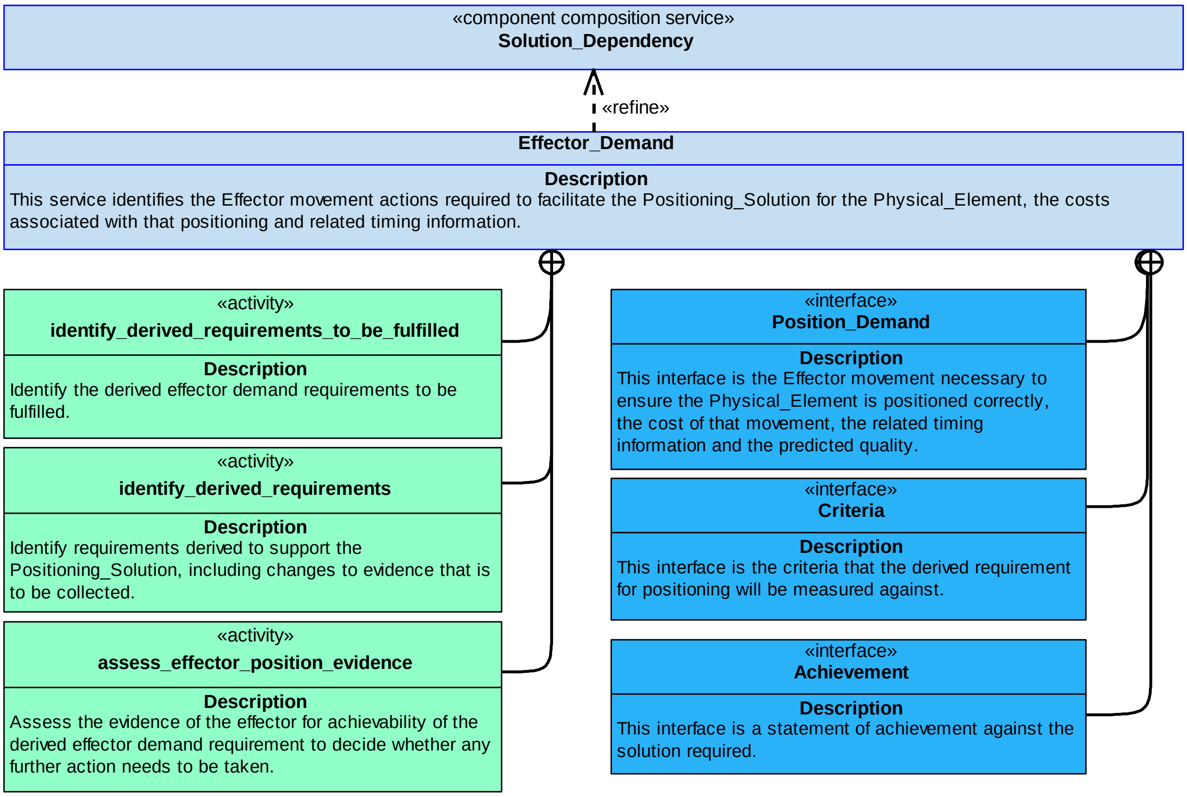

«component composition service»
Solution_Dependency
«refine»
Effector_Demand
Description
This service identifies the Effector movement actions required to facilitate the Positioning_Solution for the Physical_Element, the costs associated with that positioning and related timing information.
«activity»
identify_derived_requirements_to_be_fulfilled
Description
Identify the derived effector demand requirements to be fulfilled.
«activity»
identify_derived_requirements
Description
Identify requirements derived to support the Positioning_Solution, including changes to evidence that is to be collected.
«activity»
assess_effector_position_evidence
Description
Assess the evidence of the effector for achievability of the derived effector demand requirement to decide whether any further action needs to be taken.
«interface»
Position_Demand
Description
This interface is the Effector movement necessary to ensure the Physical_Element is positioned correctly, the cost of that movement, the related timing information and the predicted quality.
«interface»
Criteria
Description
This interface is the criteria that the derived requirement for positioning will be measured against.
«interface»
Achievement
Description
This interface is a statement of achievement against the solution required.

Figure 605: Effector_Demand Service Policy

Effector_Demand

This service identifies the Effector movement actions required to facilitate the Positioning_Solution for the Physical_Element, the costs associated with that positioning and related timing information.

Interfaces

Achievement

This interface is a statement of achievement against the solution required.

Attributes

actual_quality	How well the effector has satisfied the solution.
status	A high-level representation of achievement of the solution (e.g. not started, in progress, or complete).
time_of_update	The time at which an achievement update occurred.

Position_Demand

This interface is the Effector movement necessary to ensure the Physical_Element is positioned correctly, the cost of that movement, the related timing information and the predicted quality.

Attributes

effector	The specific Effector being moved.
required_position	The position the Effector needs to be moved into (e.g. 10mm extended or fully retracted).

required_speed	The speed the Effector needs to be moved at (e.g. extension at 10mm per second).
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used.
predicted_quality	How well the planned movement is predicted to satisfy the requirement.

Criteria

This interface is the criteria that the derived requirement for positioning will be measured against.

Attributes

property	The property to be measured, e.g. the inferred position of the Effector.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than or equal to.

Activities

identify_derived_requirements_to_be_fulfilled

Identify the derived effector demand requirements to be fulfilled.

identify_derived_requirements

Identify requirements derived to support the Positioning_Solution, including changes to evidence that is to be collected.

assess_effector_position_evidence

Assess the evidence of the effector for achievability of the derived effector demand requirement to decide whether any further action needs to be taken.

5.4.2.35.7.1.3 Position_Measurement

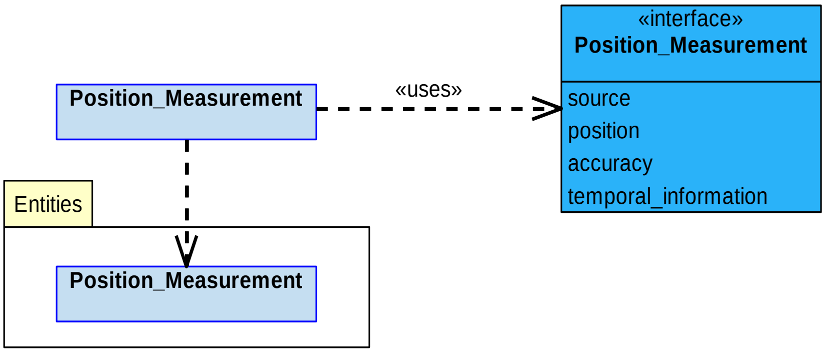


Figure 606: Position_Measurement Service Definition

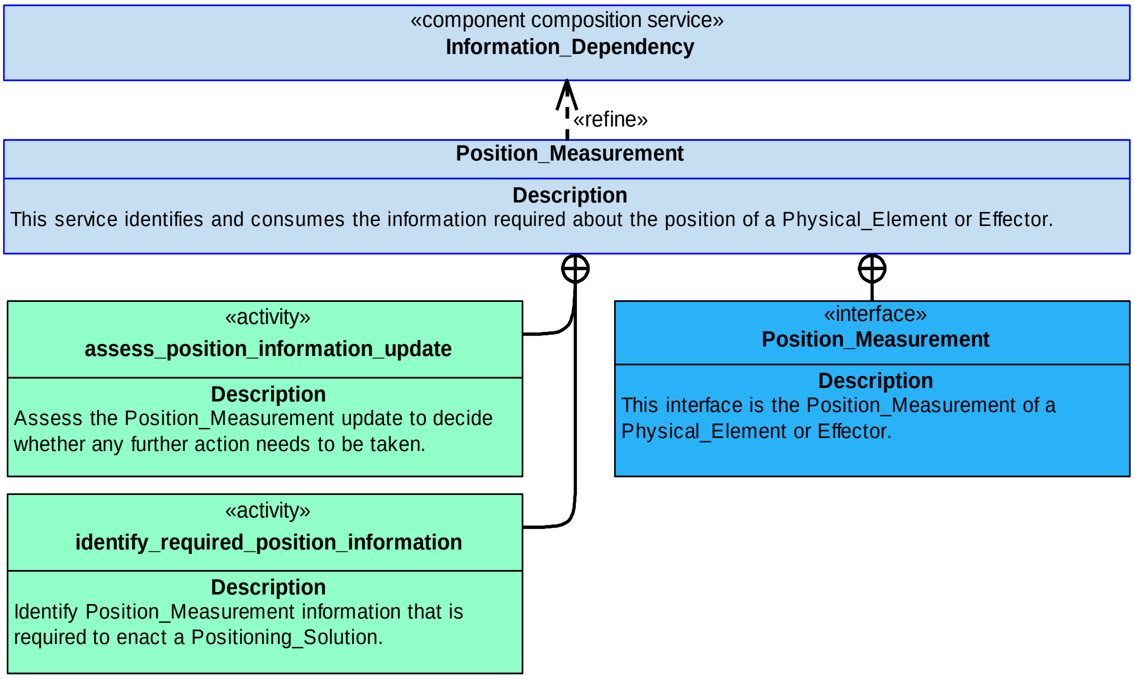

«component composition service»
Information_Dependency
«refine»
Position_Measurement
Description
This service identifies and consumes the information required about the position of a Physical_Element or Effector.
«activity»
assess_position_information_update
Description
Assess the Position_Measurement update to decide whether any further action needs to be taken.
«activity»
identify_required_position_information
Description
Identify Position_Measurement information that is required to enact a Positioning_Solution.
«interface»
Position_Measurement
Description
This interface is the Position_Measurement of a Physical_Element or Effector.

Figure 607: Position_Measurement Service Policy

Position_Measurement

This service identifies and consumes the information required about the position of a Physical_Element or Effector.

Interface

Position_Measurement

This interface is the Position_Measurement of a Physical_Element or Effector.

Attributes

source	The source of the Effector/Physical_Element position information.
position	The position of the Effector/Physical_Element (e.g. 10 degrees open or 5mm extended).
accuracy	The level of accuracy in the reported information.
temporal_information	Information covering the timing of the information being reported.

Activities

assess_position_information_update

Assess the Position_Measurement update to decide whether any further action needs to be taken.

identify_required_position_information

Identify Position_Measurement information that is required to enact a Positioning_Solution.

5.4.2.35.7.1.4 Vehicle_Context

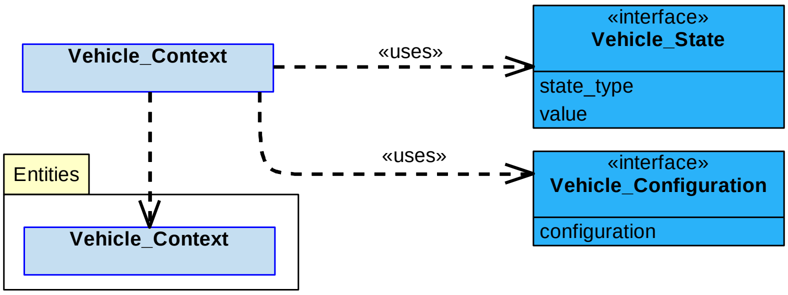


Figure 608: Vehicle_Context Service Definition

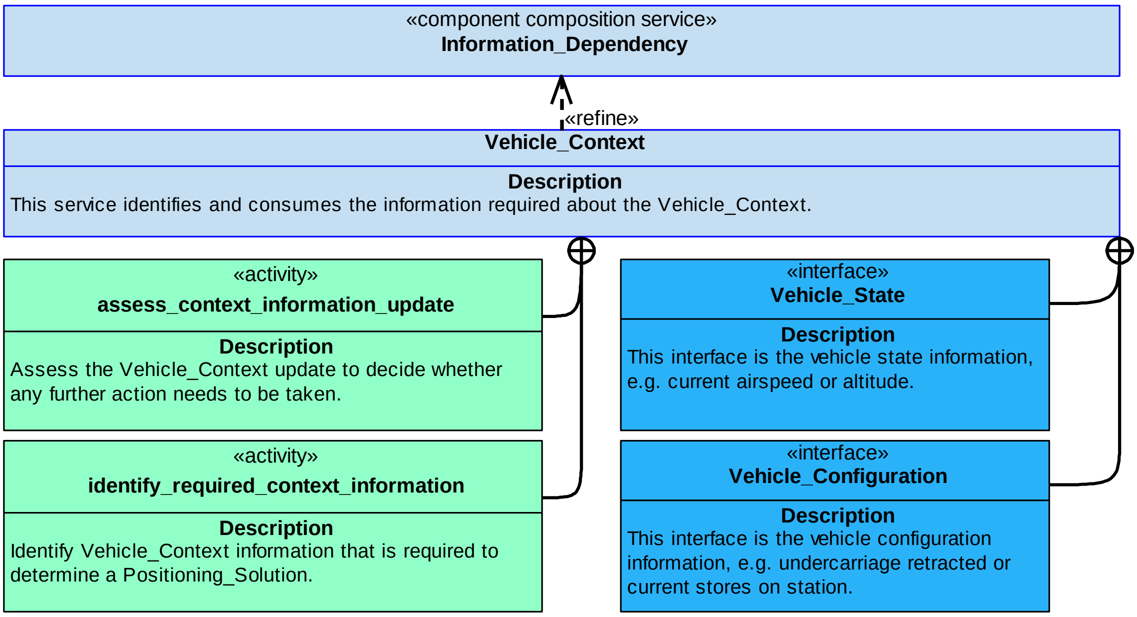

«component composition service»
Information_Dependency
«refine»
Vehicle_Context
Description
This service identifies and consumes the information required about the Vehicle_Context.
«activity»
assess_context_information_update
Description
Assess the Vehicle_Context update to decide whether any further action needs to be taken.
«activity»
identify_required_context_information
Description
Identify Vehicle_Context information that is required to determine a Positioning_Solution.
«interface»
Vehicle_State
Description
This interface is the vehicle state information, e.g. current airspeed or altitude.
«interface»
Vehicle_Configuration
Description
This interface is the vehicle configuration information, e.g. undercarriage retracted or current stores on station.

Figure 609: Vehicle_Context Service Policy

Vehicle_Context

This service identifies and consumes the information required about the Vehicle_Context.

Interfaces

Vehicle_State

This interface is the vehicle state information, e.g. current airspeed or altitude.

Attributes

state_type	The type of information relating to the vehicle state, such as an aircraft's altitude, airspeed, pitch or roll.
value	The value of the state property.

Vehicle_Configuration

This interface is the vehicle configuration information, e.g. undercarriage retracted or current stores on station.

Attribute

configuration	Information relating to the configuration of the vehicle.

Activities

assess_context_information_update

Assess the Vehicle_Context update to decide whether any further action needs to be taken.

identify_required_context_information

Identify Vehicle_Context information that is required to determine a Positioning_Solution.

5.4.2.35.7.1.5 Constraint

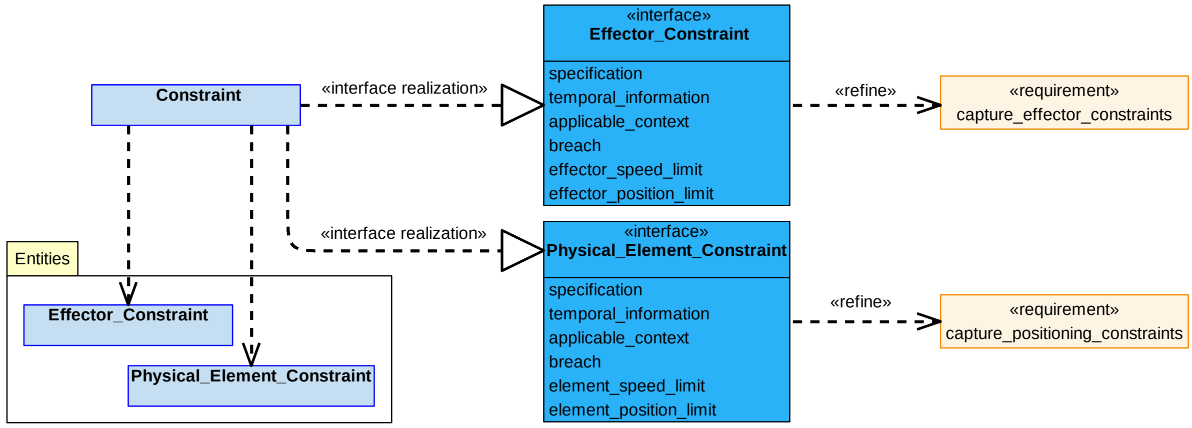
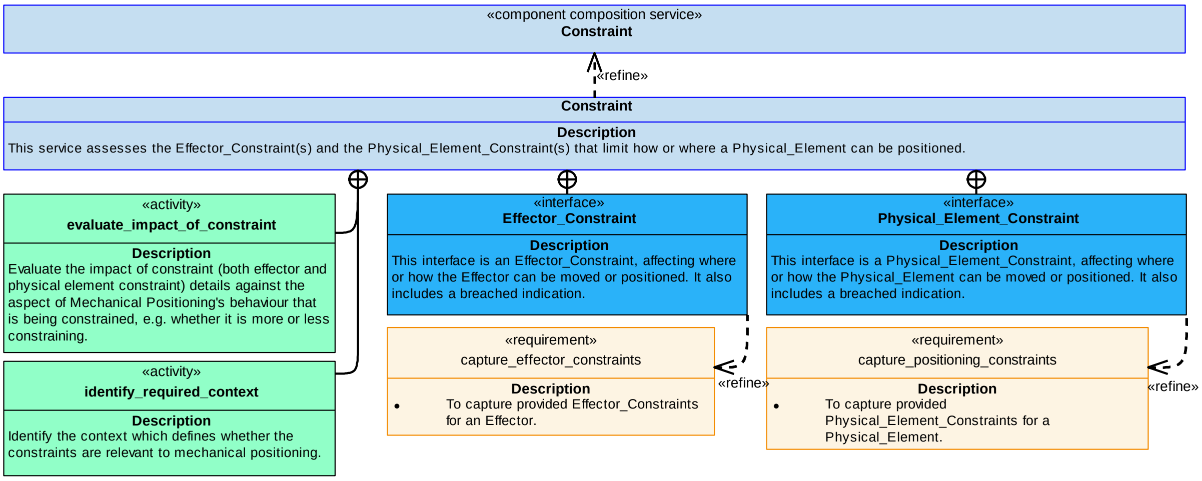


Figure 610: Constraint Service Definition



This service assesses the Effector_Constraint(s) and the Physical_Element_Constraint(s) that limit how or where a Physical_Element can be positioned.

Evaluate the impact of constraint (both effector and physical element constraint) details against the aspect of Mechanical Positioning's behaviour that is being constrained, e.g. whether it is more or less constraining.

Identify the context which defines whether the constraints are relevant to mechanical positioning.

This interface is an Effector_Constraint, affecting where or how the Effector can be moved or positioned. It also includes a breached indication.

This interface is a Physical_Element_Constraint, affecting where or how the Physical_Element can be moved or positioned. It also includes a breached indication.

- To capture provided Effector_Constraints for an Effector.

- To capture provided Physical_Element_Constraints for a Physical_Element.

Figure 611: Constraint Service Policy

Constraint

This service assesses the Effector_Constraint(s) and the Physical_Element_Constraint(s) that limit how or where a Physical_Element can be positioned.

Interfaces

Effector_Constraint

This interface is an Effector_Constraint, affecting where or how the Effector can be moved or positioned. It also includes a breached indication.

Attributes

specification	Information about the Effector.
temporal_information	Timing information on when the limit is applicable, e.g. start and end times.
applicable_context	The context within which the limit is applicable.
breach	A statement that the limit has been breached.
effector_speed_limit	A limit placed on the speed an Effector can be moved at.
effector_position_limit	A limit placed on the positions an Effector can be moved into.

Physical_Element_Constraint

This interface is a Physical_Element_Constraint, affecting where or how the Physical_Element can be moved or positioned. It also includes a breached indication.

Attributes

specification	Information about the Physical_Element.
temporal_information	Timing information on when the limit is applicable, e.g. start and end times.
applicable_context	The context within which the limit is applicable.
breach	A statement that the limit has been breached.
element_speed_limit	A limit placed on the speed a Physical_Element can be moved at.
element_position_limit	A limit placed on the positions a Physical_Element can be moved into.

Activities

evaluate_impact_of_constraint

Evaluate the impact of constraint (both effector and physical element constraint) details against the aspect of Mechanical Positioning's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the constraints are relevant to mechanical positioning.

5.4.2.35.7.1.6 Capability

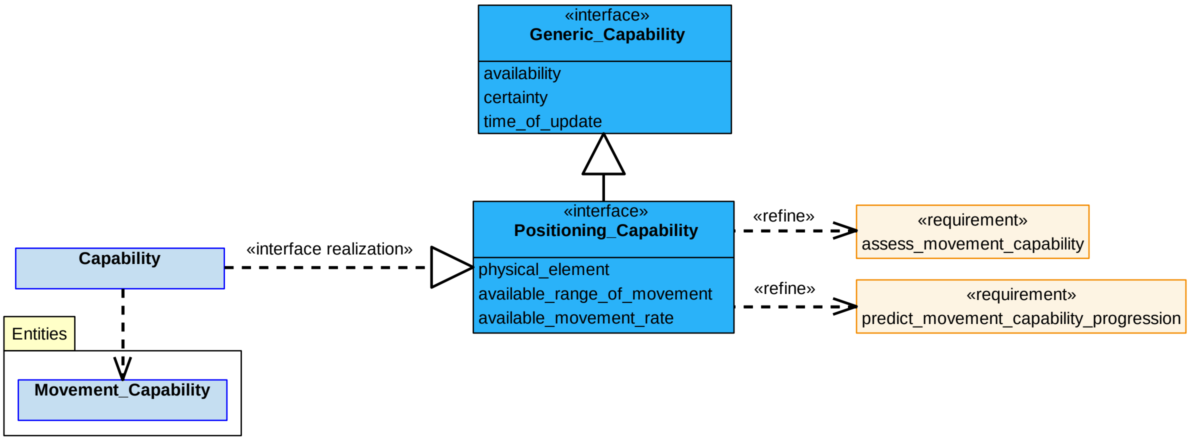
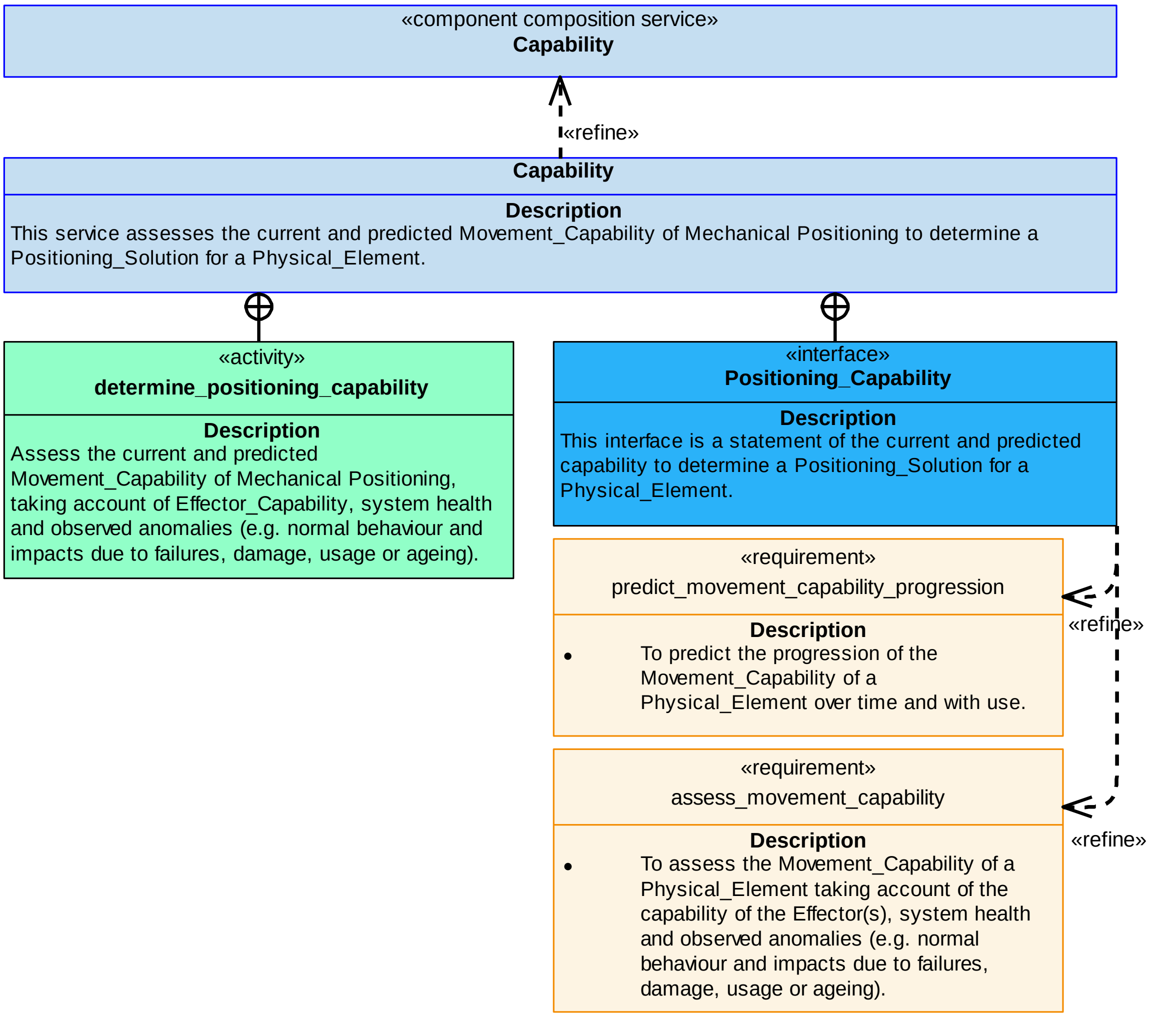


Figure 612: Capability Service Definition


«component composition service»
Capability
«refine»
Capability
«activity»
determine_positioning_capability
«interface»
Positioning_Capability
«requirement»
predict_movement_capability_progression
«refine»
«requirement»
assess_movement_capability
«refine»

Description

This service assesses the current and predicted Movement_Capability of Mechanical Positioning to determine a Positioning_Solution for a Physical_Element.

Description

Assess the current and predicted Movement_Capability of Mechanical Positioning, taking account of Effector_Capability, system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Description

This interface is a statement of the current and predicted capability to determine a Positioning_Solution for a Physical_Element.

Description

- To predict the progression of the Movement_Capability of a Physical_Element over time and with use.

Description

- To assess the Movement_Capability of a Physical_Element taking account of the capability of the Effector(s), system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Figure 613: Capability Service Policy

Capability

This service assesses the current and predicted Movement_Capability of Mechanical Positioning to determine a Positioning_Solution for a Physical_Element.

Interface

Positioning_Capability

This interface is a statement of the current and predicted capability to determine a Positioning_Solution for a Physical_Element.

Attributes

physical_element	This is the definition of the Physical_Element.
available_range_of_movement	The range of available positions into which the Physical_Element can be moved.
available_movement_rate	The available movement rates at which the Physical_Element can be moved.

Activity

determine_positioning_capability

Assess the current and predicted Movement_Capability of Mechanical Positioning, taking account of Effector_Capability, system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.35.7.1.7 Capability_Evidence

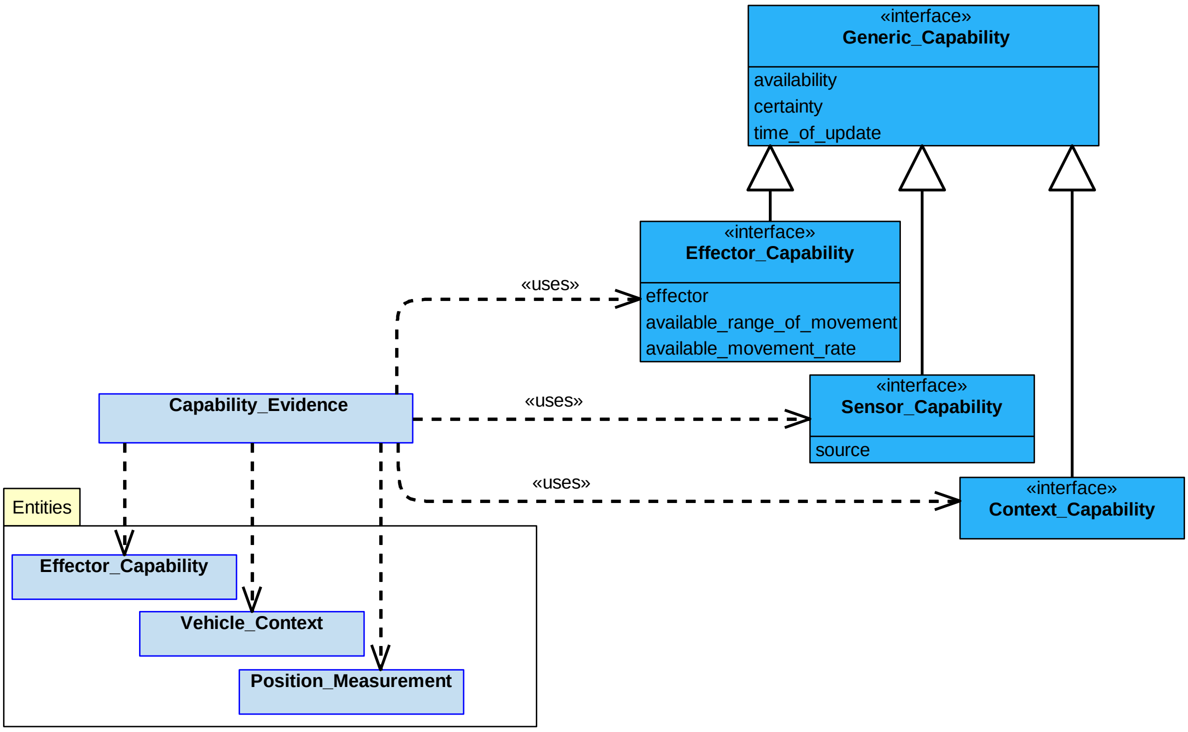


Figure 614: Capability_Evidence Service Definition

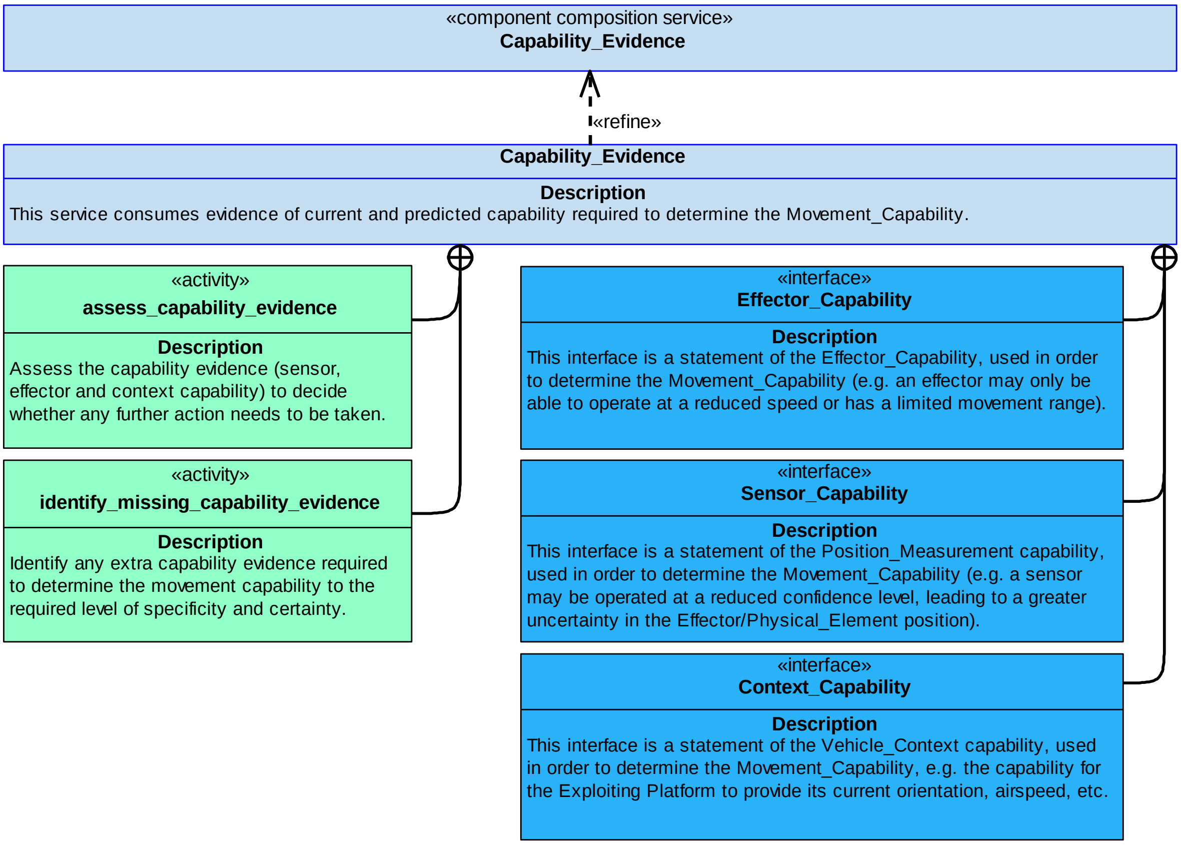

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes evidence of current and predicted capability required to determine the Movement_Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence (sensor, effector and context capability) to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the movement capability to the required level of specificity and certainty.
«interface»
Effector_Capability
Description
This interface is a statement of the Effector_Capability, used in order to determine the Movement_Capability (e.g. an effector may only be able to operate at a reduced speed or has a limited movement range).
«interface»
Sensor_Capability
Description
This interface is a statement of the Position_Measurement capability, used in order to determine the Movement_Capability (e.g. a sensor may be operated at a reduced confidence level, leading to a greater uncertainty in the Effector/Physical_Element position).
«interface»
Context_Capability
Description
This interface is a statement of the Vehicle_Context capability, used in order to determine the Movement_Capability, e.g. the capability for the Exploiting Platform to provide its current orientation, airspeed, etc.

Figure 615: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence of current and predicted capability required to determine the Movement_Capability.

Interfaces

Effector_Capability

This interface is a statement of the Effector_Capability, used in order to determine the Movement_Capability (e.g. an effector may only be able to operate at a reduced speed or has a limited movement range).

Attributes

effector	This is the definition of the Effector.
available_range_of_movement	The range of available positions into which the Effector can be moved.
available_movement_rate	The available movement rates at which the Effector can be moved.

Sensor_Capability

This interface is a statement of the Position_Measurement capability, used in order to determine the Movement_Capability (e.g. a sensor may be operated at a reduced confidence level, leading to a greater uncertainty in the Effector/Physical_Element position).

Attribute

source	The identification of the sensor, which will identify the Effector/Physical_Element whose position is defined by the Position_Measurement.

Context_Capability

This interface is a statement of the Vehicle_Context capability, used in order to determine the Movement_Capability, e.g. the capability for the Exploiting Platform to provide its current orientation, airspeed, etc.

Activities

assess_capability_evidence

Assess the capability evidence (sensor, effector and context capability) to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the movement capability to the required level of specificity and certainty.

5.4.2.35.7.2 Service Dependencies

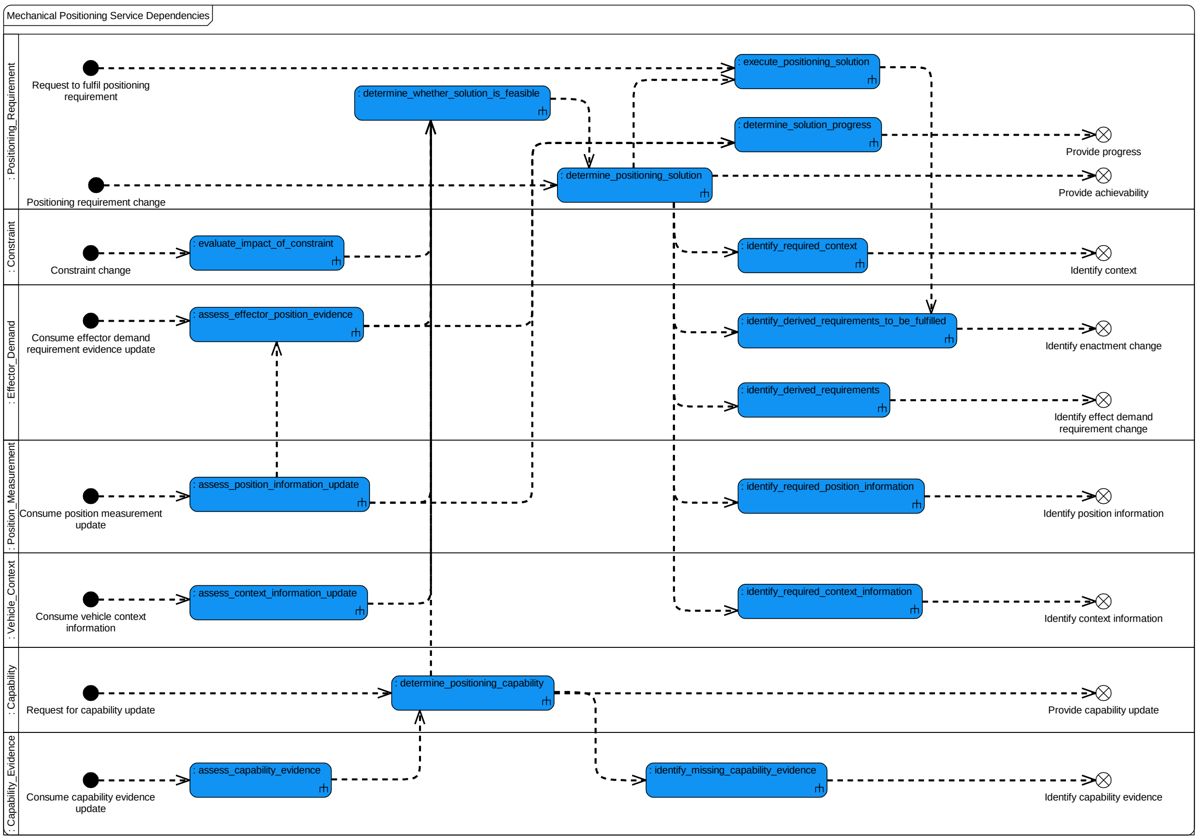

Mechanical Positioning Service Dependencies
: Positioning_Requirement
: Constraint
: Effector_Demand
: Position_Measurement
: Vehicle_Context
: Capability
: Capability_Evidence
Request to fulfil positioning requirement
Positioning requirement change
Constraint change
Consume effector demand requirement evidence update
Consume position measurement update
Consume vehicle context information
Request for capability update
Consume capability evidence update
: execute_positioning_solution
: determine_whether_solution_is_feasible
: determine_solution_progress
: determine_positioning_solution
: evaluate_impact_of_constraint
: identify_required_context
: assess_effector_position_evidence
: identify_derived_requirements_to_be_fulfilled
: identify_derived_requirements
: assess_position_information_update
: identify_required_position_information
: assess_context_information_update
: identify_required_context_information
: determine_positioning_capability
: assess_capability_evidence
: identify_missing_capability_evidence
Provide progress
Provide achievability
Identify context
Identify enactment change
Identify effect demand requirement change
Identify position information
Identify context information
Provide capability update
Identify capability evidence

Figure 616: Mechanical Positioning Service Dependencies

5.4.2.36 Navigation Sensing

5.4.2.36.1 Role

The role of Navigation Sensing is to coordinate the activities and resources required to provide a navigation solution of the required quality.

5.4.2.36.2 Overview

Control Architecture

Navigation Sensing is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a Requirement is received, Navigation Sensing will determine a Navigation_Solution_Scheme defining the activities and resources needed to deliver a solution of the required quality. This includes the identification of resource availability and configuration requirements, methods for processing sources of navigation data and other dependencies (e.g. a minimum altitude requirement to achieve use of navigation beacons). The Navigation_Solution_Scheme is then enacted, which will involve coordinated control of dependent activities and Navigation_Resources. Navigation Sensing subsequently monitors the ongoing achievement against the requirement and adjusts the solution as necessary in response to changes in achieved quality, constraints, capability and environmental conditions.

Examples of Use

Navigation Sensing should be used where:

- The coordination of multiple resources (e.g. GNSS, Inertial Navigation System, ILS, TACAN or VOR) and activities (e.g. the processing of navigation data) is needed to deliver the required navigations solutions.

5.4.2.36.3 Service Summary

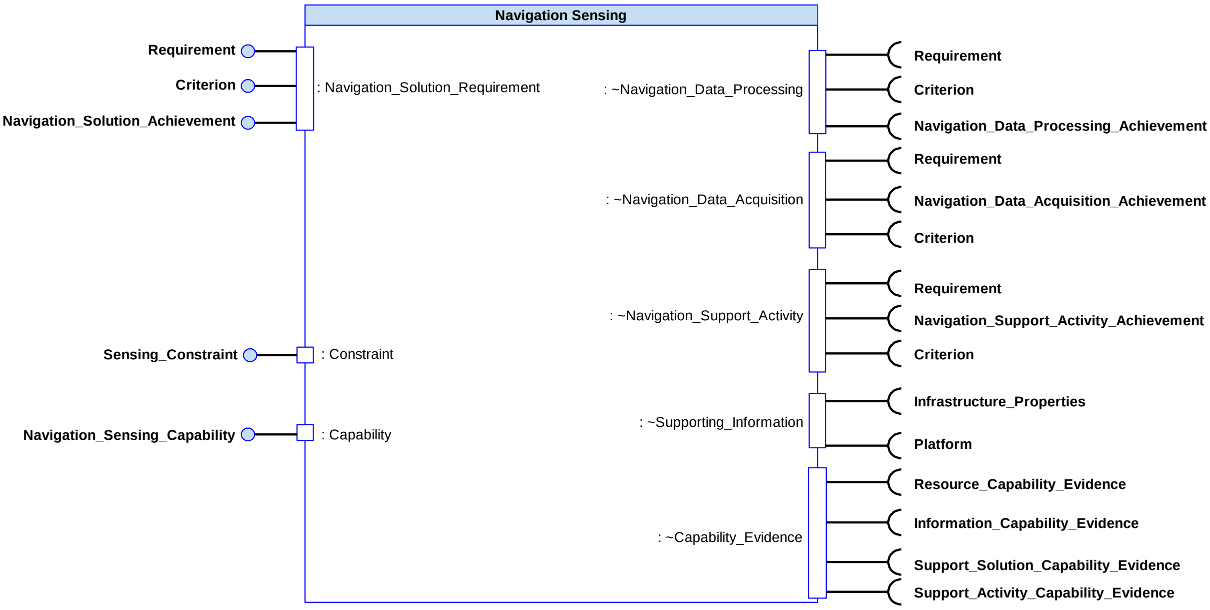

Navigation Sensing
Requirement
Criterion
Navigation_Solution_Achievement
: Navigation_Solution_Requirement
Sensing_Constraint
: Constraint
Navigation_Sensing_Capability
: Capability
: ~Navigation_Data_Processing
Requirement
Criterion
Navigation_Data_Processing_Achievement
: ~Navigation_Data_Acquisition
Requirement
Navigation_Data_Acquisition_Achievement
Criterion
: ~Navigation_Support_Activity
Requirement
Navigation_Support_Activity_Achievement
Criterion
: ~Supporting_Information
Infrastructure_Properties
Platform
: ~Capability_Evidence
Resource_Capability_Evidence
Information_Capability_Evidence
Support_Solution_Capability_Evidence
Support_Activity_Capability_Evidence

Figure 617: Navigation Sensing Service Summary

5.4.2.36.4 Responsibilities

capture_navigation_requirements

- To capture the Requirements for navigation, e.g. provision of the most accurate or most robust positional information.

capture_solution_measurement_criteria

- To capture given Measurement_Criterion/criteria (e.g. stability or timing) against which a Navigation_Solution_Scheme will be assessed.

capture_navigation_solution_constraints

- To capture the Constraints which must be observed when determining a Navigation_Solution_Scheme (e.g. resource restrictions, transmission restrictions, or exclusions).

identify_whether_requirement_is_achievable

- To identify whether a Requirement is achievable given the current or predicted Capability and Constraints.

determine_navigation_solution

- To determine a Navigation_Solution_Scheme using the available Capability, which meets the Requirements and satisfies the Constraints.

determine_predicted_quality_of_navigation_solution_scheme

- To determine the predicted quality of a Navigation_Solution_Scheme against the Measurement_Criterion/criteria.

determine_solution_actions

- To determine the activities required to support a Navigation_Solution_Scheme (e.g. the configuration of Navigation_Resources, for the provision or processing of navigation data, and determination of required support activities).

determine_solution_dependencies

- To identify dependencies to support a Navigation_Solution_Scheme or a step of the Navigation_Solution_Scheme.

coordinate_navigation_solution

- To coordinate the actions required to implement a Navigation_Solution_Scheme, e.g. by commanding the instruction of a Navigation_Resource for the provision or processing of navigation data.

identify_progress_of_navigation_solution

- To identify the progress of a Navigation_Solution_Scheme and achievement against the Requirement.

determine_actual_quality_of_navigation_solution_scheme

- To determine the actual quality of the delivered Navigation_Solution_Scheme, measured against the Requirements and Measurement_Criterion/criteria.

determine_solution_cost

- To determine the cost of a Navigation_Solution_Scheme against given Measurement_Criterion/criteria.

assess_capability

- To assess the Capability of the component taking account of capability of Navigation_Resources, system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of Capability determination.

predict_capability_progression

- To predict the progression of Capability over time and with use.

5.4.2.36.5 Subject Matter Semantics

The subject matter of Navigation Sensing is the resources that provide navigation data and the methods used to process that data, in order to meet the navigation needs of the platform.

Exclusions

The subject matter of Navigation Sensing does not include:

- The processing of navigation data provided by a Navigation_Resource.

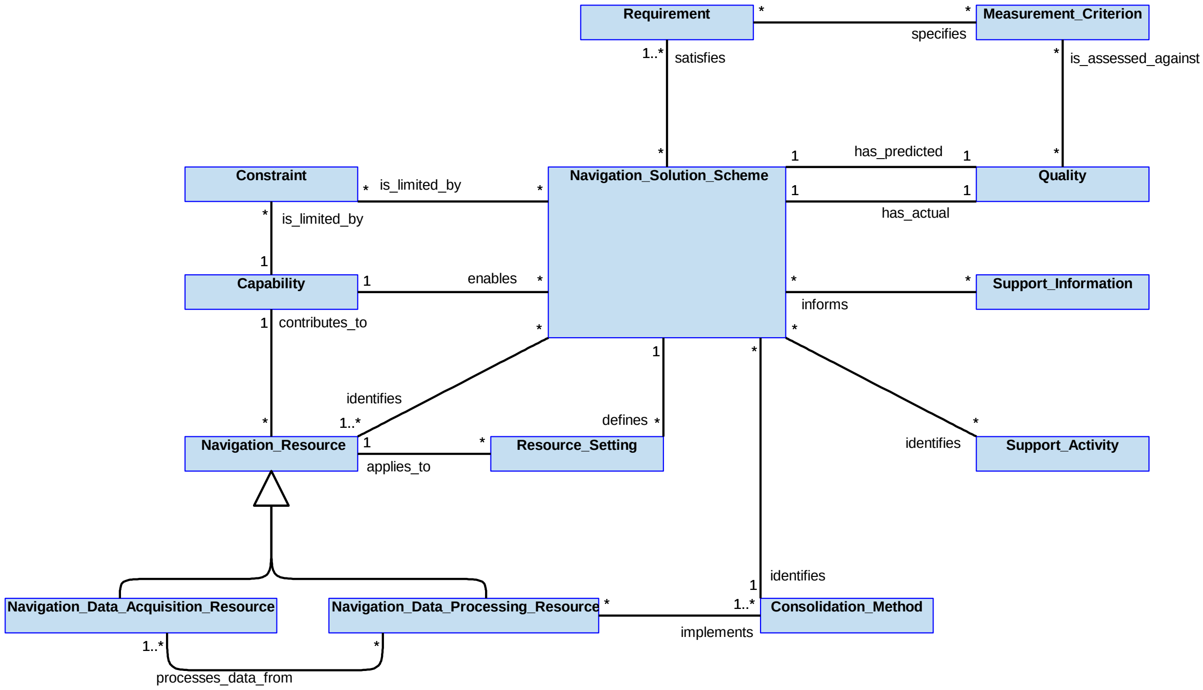


Figure 618: Navigation Sensing Semantics

5.4.2.36.5.1 Entities

Capability

The range of navigation solutions that the component is able to provide with the available resources and navigation data processing methods.

Constraint

An externally imposed restriction, e.g. a restriction on the use of the radar altimeter within the current airspace.

Measurement_Criterion

A measure against which achievement of the Requirement can be assessed, e.g. positional accuracy within a stated tolerance.

Requirement

A specification for the provision of a navigation solution, e.g. provision of the most accurate or most robust positional information.

Resource_Setting

A parameter, mode or variable that may be configured on a source, or processor, of navigation data.

Navigation_Solution_Scheme

A scheme comprising a chosen set of navigation data and the methodology for processing this data, that is expected to provide a navigation solution of the necessary quality (e.g. accuracy, availability, or stability) to meet the Requirement.

Navigation_Resource

A source of navigation data or navigation data processing capability, e.g. equipment like GNSS or DME, or the processing capability to consolidate multiple sources of navigation data.

Support_Information

A piece of information related to the platform, e.g. location, orientation, the platform operating context such as environmental information (e.g. visibility), or military/civil operating rules.

Consolidation_Method

A method for consolidating and cross checking the outputs from navigation information sources, including the identification of the relevant information sources and how they should be used, e.g. the identification of rules for weighting their use and the need for any filtering of results.

Support_Activity

A dependency that must be satisfied in order to deliver the navigation solution. Examples include:

- An inertial navigation system must be aligned or an almanac loaded.
- The Exploiting Platform must remain within beacon coverage areas.
- The Exploiting Platform must achieve immediate beacon connectivity.

Quality

A measure of the effectiveness or adequacy of a navigation solution that is expected or achieved (e.g. the accuracy of position, orientation, velocity and acceleration).

Navigation_Data_Acquisition_Resource

A source of navigation data, e.g. equipment like GNSS or DME.

Navigation_Data_Processing_Resource

A navigation data processing capability, e.g. to consolidate multiple sources of navigation data.

5.4.2.36.6 Design Rationale

5.4.2.36.6.1 Assumptions

- The types of Navigation_Resource will be updated extremely rarely.
- Supported Navigation_Resources will not change during operation - although *available* or *useable* resources will change.
- Methods of determining a Navigation_Solution_Scheme will be updated rarely (e.g. would be common to variants and updated across them).
- A Requirement placed for the Navigation_Solution_Scheme may be to conform to a policy. Some policies would be global (e.g. a policy for RNP0.1) and update rarely. Others may be mission related and updated frequently.
- Navigation Sensing will not have the ability to inhibit the equipment required to provide the minimum vehicle control. For example, inhibiting / excluding an Inertial Navigation System

where this is the only source of orientation and primary source of position is expected to be prevented by the Exploiting Platform.

- The system behaves in accordance with the actual navigation quality achieved as determined by other components.
- Other components are the sources of the information relating to the actual Navigation_Solution_Scheme performance and it is not sourced from the Navigation Sensing component.

5.4.2.36.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Navigation Sensing:

- Data Driving - The component may be data-driven to allow the component to be reusable between multiple Exploiting Programmes and maintainable as behaviours change and resources are replaced:
    - The types of Navigation_Resource (using build time data) and types of data source (during operation) and their compatibility with each other.
    - The methods by which Navigation Sensing derives Navigation_Solution_Schemes (using build time data).
    - Common policies for groupings of quality requirements (during operation).

Extensions

- Extension components may be utilised to cater for different types of Navigation_Resource.

Exploitation Considerations

- Navigation Sensing will need to represent a model of the theoretical quality for a Navigation_Solution_Scheme, based on the capabilities of available Navigation_Resources.
- The responses of the integrated navigation information against the theoretical Navigation_Solution_Scheme will be monitored. Should the component detect a large deviation from the required or expected behaviour, resources may be reconfigured or excluded or their unexpected performance be reported.
- An individual Navigation_Resource may not be useable (due to health or constraints placed on the component), and so would not be considered in the determination of a Navigation_Solution_Scheme.

5.4.2.36.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component may result in the loss of precision navigation data (e.g. of sufficient accuracy to land on a runway). This may cause an air vehicle to perform a controlled trajectory termination or forced landing in order to minimise third party fatalities or the crew needing to eject (if the visibility is low enough the runway cannot be seen).

5.4.2.36.6.4 Security Considerations

The indicative security classification is O.

This component selects and configures the Navigation_Resources to provide the Navigation_Solution_Scheme, contributing to determining the location, orientation, velocity or acceleration of the platform. The navigation solution will generally be considered O, however the use of some resources or policies may lead to higher confidentiality requirements. This component does not handle the sensor output, therefore it should not be possible to derive vehicle location from data within this component. This component can restrict the availability of navigation resource data to the system, preventing the use of resources suspected of being spoofed, but also potentially limiting the precision of the solution when done without need.

The component is expected to at least partially satisfy Security Related Functions by:

- **Identifying Data Sources** used for determining navigation data as being allowable sources.
- **Logging of Security Data** for changes in configuration and policies, and access to resources, etc.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring**, with unexpected loss of quality being a possible indicator of a cyber attack.

The component is considered unlikely to implement security enforcing functions.

5.4.2.36.7 Services

5.4.2.36.7.1 Service Definitions

5.4.2.36.7.1.1 Navigation_Solution_Requirement

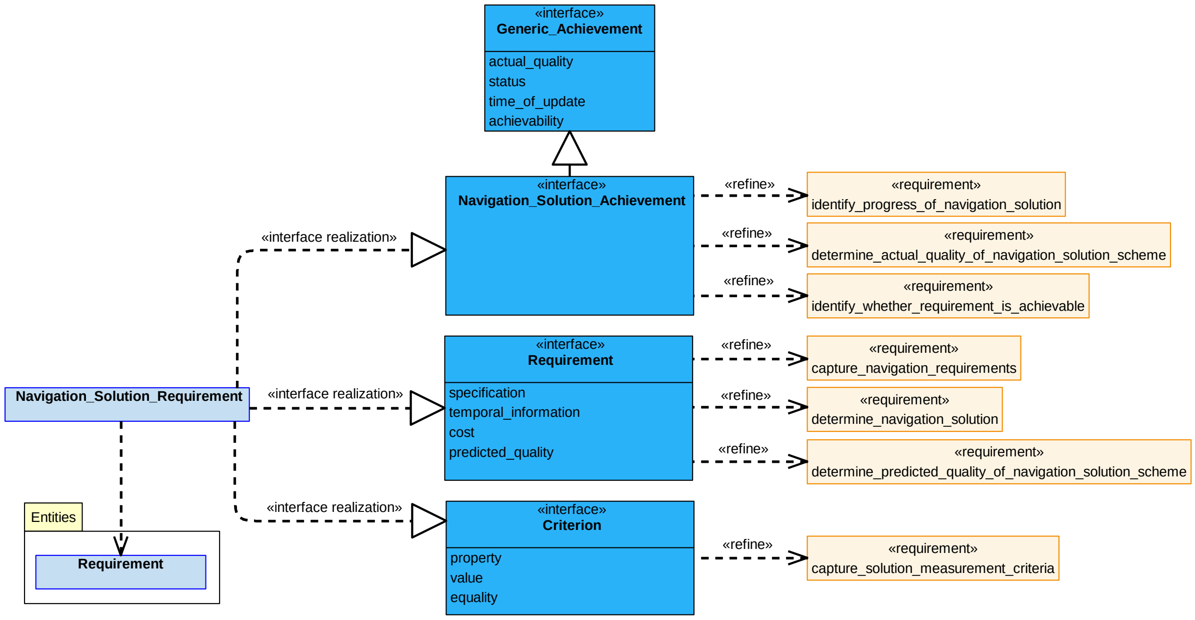
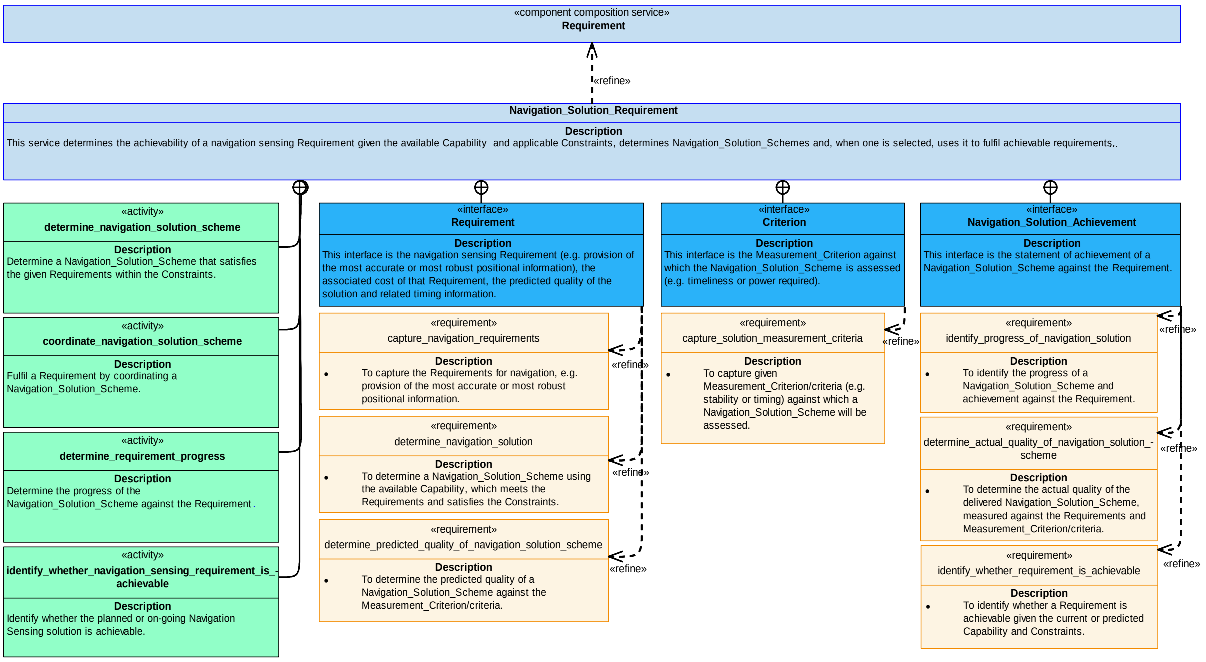


Figure 619: Navigation_Solution_Requirement Service Definition


«component composition service»
Requirement
«refine»
Navigation_Solution_Requirement
Description
«activity»
determine_navigation_solution_scheme
Description
«activity»
coordinate_navigation_solution_scheme
Description
«activity»
determine_requirement_progress
Description
«activity»
identify_whether_navigation_sensing_requirement_is_-achievable
Description
«interface»
Requirement
Description
«requirement»
capture_navigation_requirements
Description
«refine»
«requirement»
determine_navigation_solution
Description
«refine»
«requirement»
determine_predicted_quality_of_navigation_solution_scheme
Description
«refine»
«interface»
Criterion
Description
«requirement»
capture_solution_measurement_criteria
Description
«refine»
«interface»
Navigation_Solution_Achievement
Description
«requirement»
identify_progress_of_navigation_solution
Description
«refine»
«requirement»
determine_actual_quality_of_navigation_solution_-scheme
Description
«refine»
«requirement»
identify_whether_requirement_is_achievable
Description
«refine»

This service determines the achievability of a navigation sensing Requirement given the available Capability and applicable Constraints, determines Navigation_Solution_Schemes and, when one is selected, uses it to fulfil achievable requirements,.

Determine a Navigation_Solution_Scheme that satisfies the given Requirements within the Constraints.

Fulfil a Requirement by coordinating a Navigation_Solution_Scheme.

Determine the progress of the Navigation_Solution_Scheme against the Requirement.

Identify whether the planned or on-going Navigation Sensing solution is achievable.

This interface is the navigation sensing Requirement (e.g. provision of the most accurate or most robust positional information), the associated cost of that Requirement, the predicted quality of the solution and related timing information.

- To capture the Requirements for navigation, e.g. provision of the most accurate or most robust positional information.

- To determine a Navigation_Solution_Scheme using the available Capability, which meets the Requirements and satisfies the Constraints.

- To determine the predicted quality of a Navigation_Solution_Scheme against the Measurement_Criterion/criteria.

This interface is the Measurement_Criterion against which the Navigation_Solution_Scheme is assessed (e.g. timeliness or power required).

- To capture given Measurement_Criterion/criteria (e.g. stability or timing) against which a Navigation_Solution_Scheme will be assessed.

This interface is the statement of achievement of a Navigation_Solution_Scheme against the Requirement.

- To identify the progress of a Navigation_Solution_Scheme and achievement against the Requirement.

- To determine the actual quality of the delivered Navigation_Solution_Scheme, measured against the Requirements and Measurement_Criterion/criteria.

- To identify whether a Requirement is achievable given the current or predicted Capability and Constraints.

Figure 620: Navigation_Solution_Requirement Service Policy

Navigation_Solution_Requirement

This service determines the achievability of a navigation sensing Requirement given the available Capability and applicable Constraints, determines Navigation_Solution_Schemes and, when one is selected, uses it to fulfil achievable requirements.

Interfaces

Navigation_Solution_Achievement

This interface is the statement of achievement of a Navigation_Solution_Scheme against the Requirement.

Criterion

This interface is the Measurement_Criterion against which the Navigation_Solution_Scheme is assessed (e.g. timeliness or power required).

Attributes

property	The property to be measured, e.g. accuracy, stability, or timing.
value	The value related to the property to be measured.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Requirement

This interface is the navigation sensing Requirement (e.g. provision of the most accurate or most robust positional information), the associated cost of that Requirement, the predicted quality of the solution and related timing information.

Attributes

specification	The definition of the navigation sensing Requirement, e.g. provision of the most accurate or most robust positional information.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the Navigation_Solution_Scheme, for example: resources used or time taken.
predicted_quality	A measure of how well the planned Navigation_Solution_Scheme is predicted to satisfy the Requirement, taking in to account the navigation sensing capability and constraints.

Activities

determine_navigation_solution_scheme

Determine a Navigation_Solution_Scheme that satisfies the given Requirements within the Constraints.

coordinate_navigation_solution_scheme

Fulfil a Requirement by coordinating a Navigation_Solution_Scheme.

determine_requirement_progress

Determine the progress of the Navigation_Solution_Scheme against the Requirement.

identify_whether_navigation_sensing_requirement_is_achievable

Identify whether the planned or on-going Navigation Sensing solution is achievable.

5.4.2.36.7.1.2 Navigation_Data_Processing

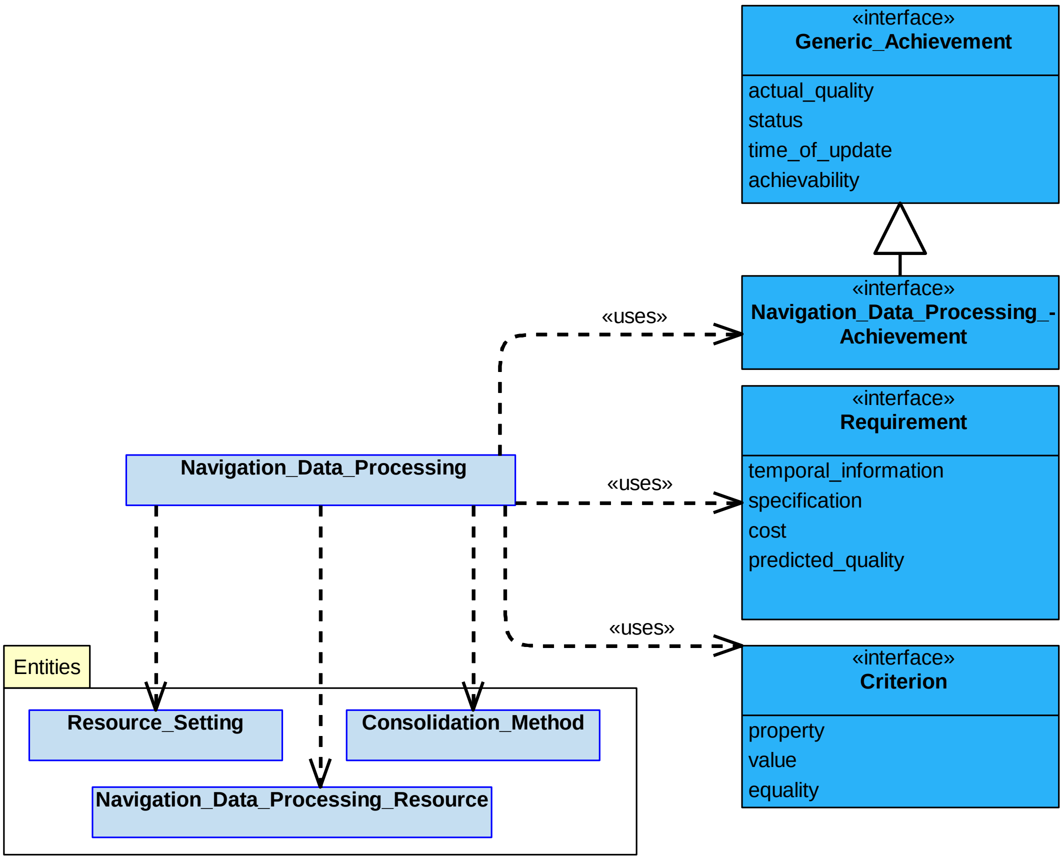


Figure 621: Navigation_Data_Processing Service Definition

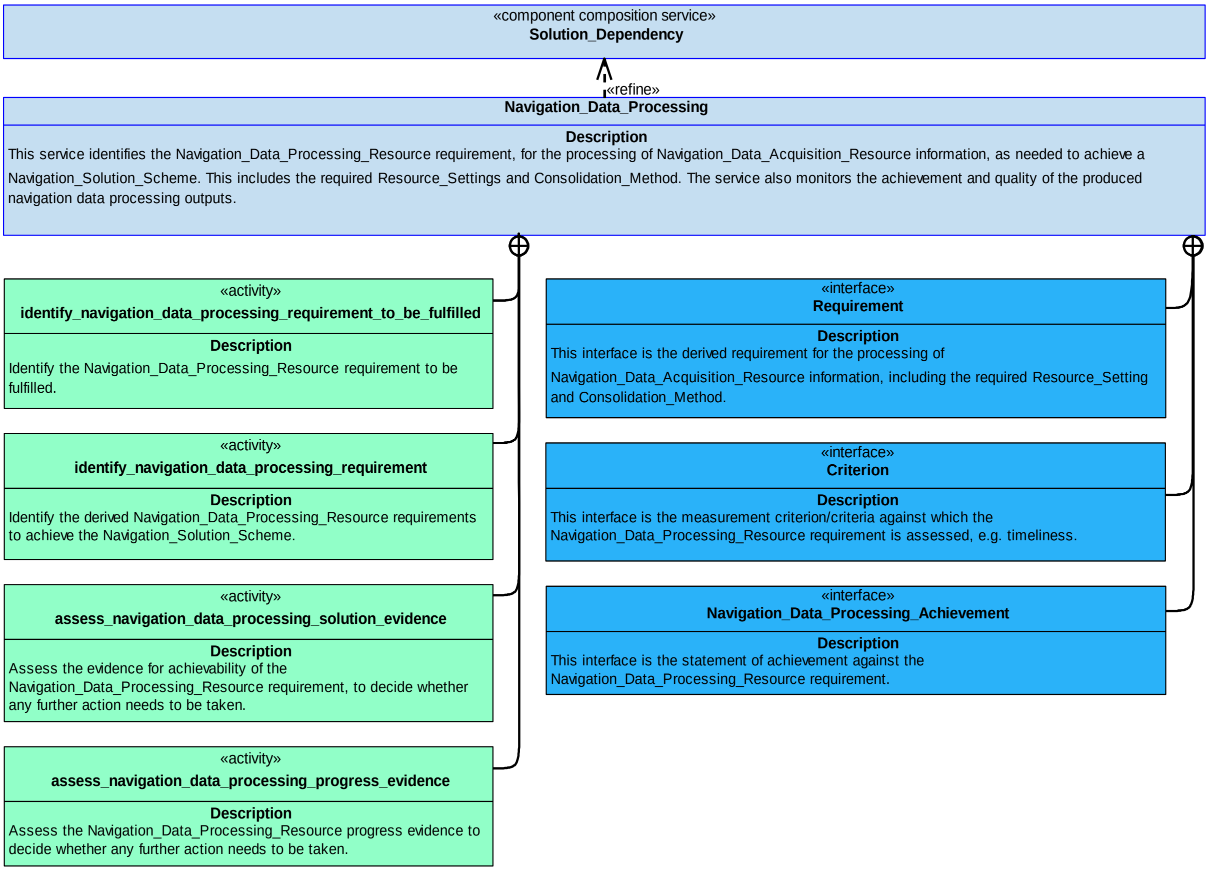

«component composition service»
Solution_Dependency
«refine»
Navigation_Data_Processing
Description
This service identifies the Navigation_Data_Processing_Resource requirement, for the processing of Navigation_Data_Acquisition_Resource information, as needed to achieve a Navigation_Solution_Scheme. This includes the required Resource_Settings and Consolidation_Method. The service also monitors the achievement and quality of the produced navigation data processing outputs.
«activity»
identify_navigation_data_processing_requirement_to_be_fulfilled
Description
Identify the Navigation_Data_Processing_Resource requirement to be fulfilled.
«activity»
identify_navigation_data_processing_requirement
Description
Identify the derived Navigation_Data_Processing_Resource requirements to achieve the Navigation_Solution_Scheme.
«activity»
assess_navigation_data_processing_solution_evidence
Description
Assess the evidence for achievability of the Navigation_Data_Processing_Resource requirement, to decide whether any further action needs to be taken.
«activity»
assess_navigation_data_processing_progress_evidence
Description
Assess the Navigation_Data_Processing_Resource progress evidence to decide whether any further action needs to be taken.
«interface»
Requirement
Description
This interface is the derived requirement for the processing of Navigation_Data_Acquisition_Resource information, including the required Resource_Setting and Consolidation_Method.
«interface»
Criterion
Description
This interface is the measurement criterion/criteria against which the Navigation_Data_Processing_Resource requirement is assessed, e.g. timeliness.
«interface»
Navigation_Data_Processing_Achievement
Description
This interface is the statement of achievement against the Navigation_Data_Processing_Resource requirement.

Figure 622: Navigation_Data_Processing Service Policy

Navigation_Data_Processing

This service identifies the Navigation_Data_Processing_Resource requirement, for the processing of Navigation_Data_Acquisition_Resource information, as needed to achieve a Navigation_Solution_Scheme. This includes the required Resource_Settings and Consolidation_Method. The service also monitors the achievement and quality of the produced navigation data processing outputs.

Interfaces

Navigation_Data_Processing_Achievement

This interface is the statement of achievement against the Navigation_Data_Processing_Resource requirement.

Criterion

This interface is the measurement criterion/criteria against which the Navigation_Data_Processing_Resource requirement is assessed, e.g. timeliness.

Attributes

property	The property to be measured, e.g. error margin.
value	The value to be related to the measured property.

equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Requirement

This interface is the derived requirement for the processing of Navigation_Data_Acquisition_Resource information, including the required Resource_Setting and Consolidation_Method.

Attributes

temporal_information	Information covering timing, such as start and end times.
specification	The definition of the Navigation_Data_Processing_Resource requirement, including specification of the sources of navigation data, how they are to be used (e.g. whether it is to be incorporated into the solution or used for checks to ensure that the solution remains within the expected bounds) and the Consolidation_Method to be applied to those sources.
cost	The cost of executing a Navigation_Data_Processing_Resource solution, e.g. resources used, or time taken.
predicted_quality	A measure of how well the Navigation_Data_Processing_Resource solution is expected to satisfy the Navigation_Data_Processing_Resource requirement, given the Navigation_Data_Processing_Resource capability and constraints, and the specified Navigation_Data_Acquisition_Resources and Consolidation_Method.

Activities

identify_navigation_data_processing_requirement_to_be_fulfilled

Identify the Navigation_Data_Processing_Resource requirement to be fulfilled.

identify_navigation_data_processing_requirement

Identify the derived Navigation_Data_Processing_Resource requirements to achieve the Navigation_Solution_Scheme.

assess_navigation_data_processing_solution_evidence

Assess the evidence for achievability of the Navigation_Data_Processing_Resource requirement, to decide whether any further action needs to be taken.

assess_navigation_data_processing_progress_evidence

Assess the Navigation_Data_Processing_Resource progress evidence to decide whether any further action needs to be taken.

5.4.2.36.7.1.3 Navigation_Data_Acquisition

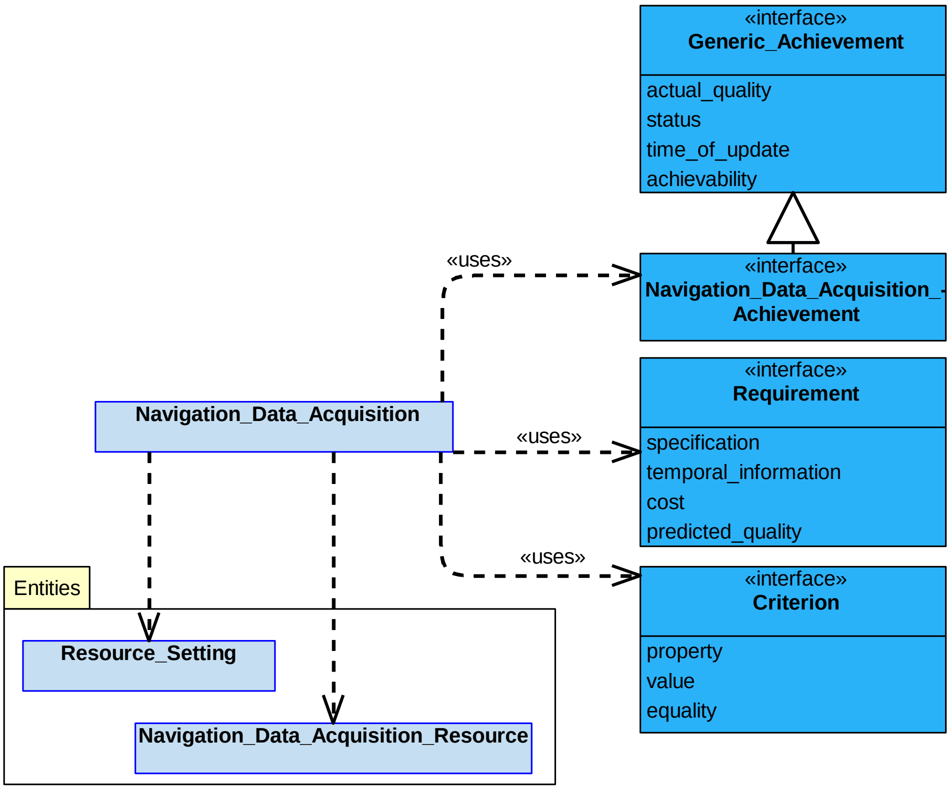


Figure 623: Navigation_Data_Acquisition Service Definition

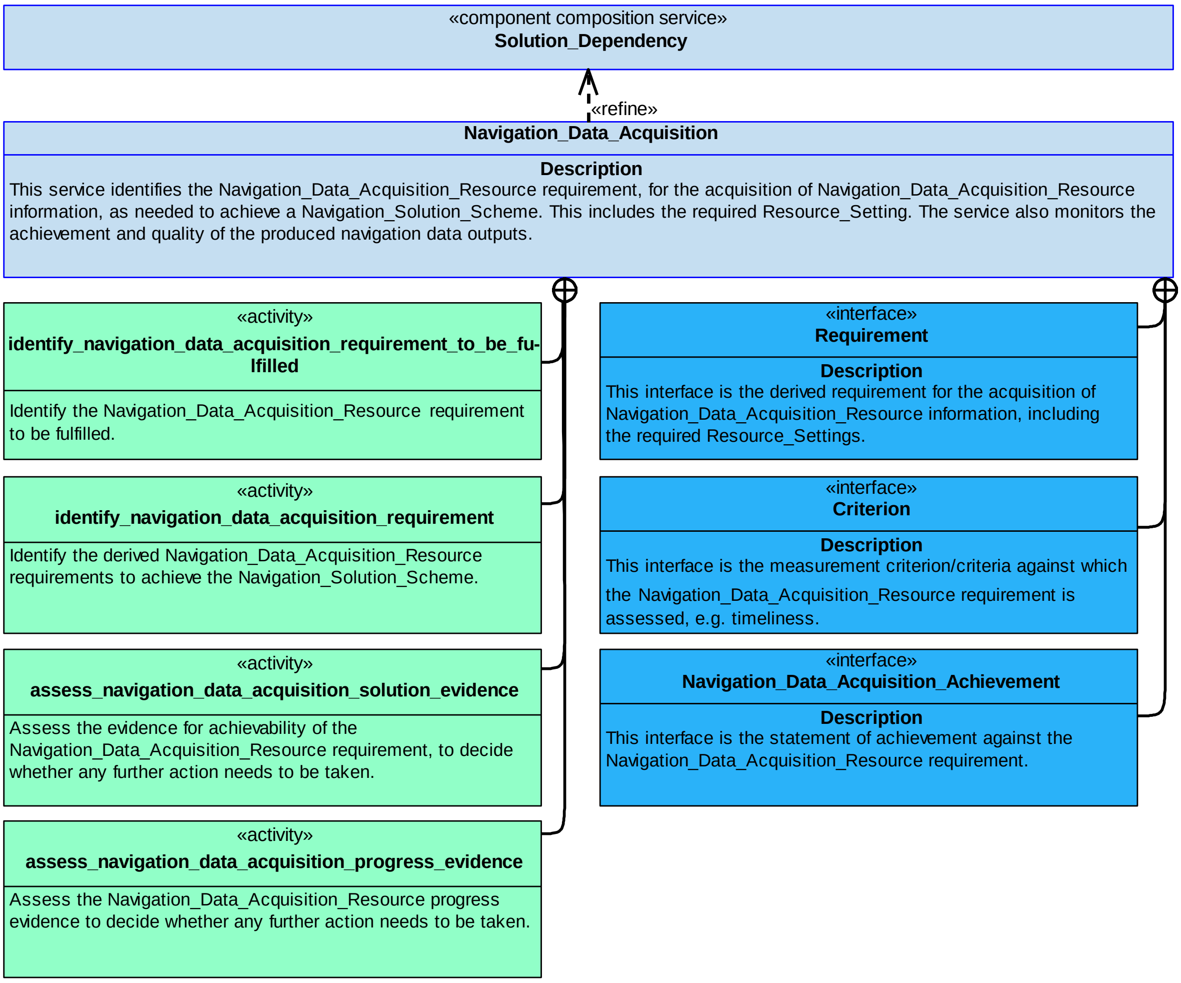

«component composition service»
Solution_Dependency
«refine»
Navigation_Data_Acquisition
Description
This service identifies the Navigation_Data_Acquisition_Resource requirement, for the acquisition of Navigation_Data_Acquisition_Resource information, as needed to achieve a Navigation_Solution_Scheme. This includes the required Resource_Setting. The service also monitors the achievement and quality of the produced navigation data outputs.
«activity»
identify_navigation_data_acquisition_requirement_to_be_fulfilled
Identify the Navigation_Data_Acquisition_Resource requirement to be fulfilled.
«activity»
identify_navigation_data_acquisition_requirement
Identify the derived Navigation_Data_Acquisition_Resource requirements to achieve the Navigation_Solution_Scheme.
«activity»
assess_navigation_data_acquisition_solution_evidence
Assess the evidence for achievability of the Navigation_Data_Acquisition_Resource requirement, to decide whether any further action needs to be taken.
«activity»
assess_navigation_data_acquisition_progress_evidence
Assess the Navigation_Data_Acquisition_Resource progress evidence to decide whether any further action needs to be taken.
«interface»
Requirement
Description
This interface is the derived requirement for the acquisition of Navigation_Data_Acquisition_Resource information, including the required Resource_Settings.
«interface»
Criterion
Description
This interface is the measurement criterion/criteria against which the Navigation_Data_Acquisition_Resource requirement is assessed, e.g. timeliness.
«interface»
Navigation_Data_Acquisition_Achievement
Description
This interface is the statement of achievement against the Navigation_Data_Acquisition_Resource requirement.

Figure 624: Navigation_Data_Acquisition Service Policy

Navigation_Data_Acquisition

This service identifies the Navigation_Data_Acquisition_Resource requirement, for the acquisition of Navigation_Data_Acquisition_Resource information, as needed to achieve a Navigation_Solution_Scheme. This includes the required Resource_Setting. The service also monitors the achievement and quality of the produced navigation data outputs.

Interfaces

Navigation_Data_Acquisition_Achievement

This interface is the statement of achievement against the Navigation_Data_Acquisition_Resource requirement.

Requirement

This interface is the derived requirement for the acquisition of Navigation_Data_Acquisition_Resource information, including the required Resource_Settings.

Attributes

specification	The definition of the Navigation_Data_Acquisition_Resource requirement, including specification of the Resource_Settings for the Navigation_Data_Acquisition_Resource, e.g. the mode of operation, operating channel, and frequency of data capture.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the setting, for example resources used, or time taken.
predicted_quality	The measures(s) of how well the Navigation_Data_Acquisition_Resource solution is expected to satisfy the Navigation_Data_Acquisition_Resource requirement, given the Navigation_Data_Acquisition_Resource capability and constraints, and specified Resource_Setting.

Criterion

This interface is the measurement criterion/criteria against which the Navigation_Data_Acquisition_Resource requirement is assessed, e.g. timeliness.

Attributes

property	The property to be measured, e.g. availability, stability.
value	The value to be related to the measured property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

identify_navigation_data_acquisition_requirement_to_be_fulfilled

Identify the Navigation_Data_Acquisition_Resource requirement to be fulfilled.

identify_navigation_data_acquisition_requirement

Identify the derived Navigation_Data_Acquisition_Resource requirements to achieve the Navigation_Solution_Scheme.

assess_navigation_data_acquisition_solution_evidence

Assess the evidence for achievability of the Navigation_Data_Acquisition_Resource requirement, to decide whether any further action needs to be taken.

assess_navigation_data_acquisition_progress_evidence

Assess the Navigation_Data_Acquisition_Resource progress evidence to decide whether any further action needs to be taken.

5.4.2.36.7.1.4 Navigation_Support_Activity

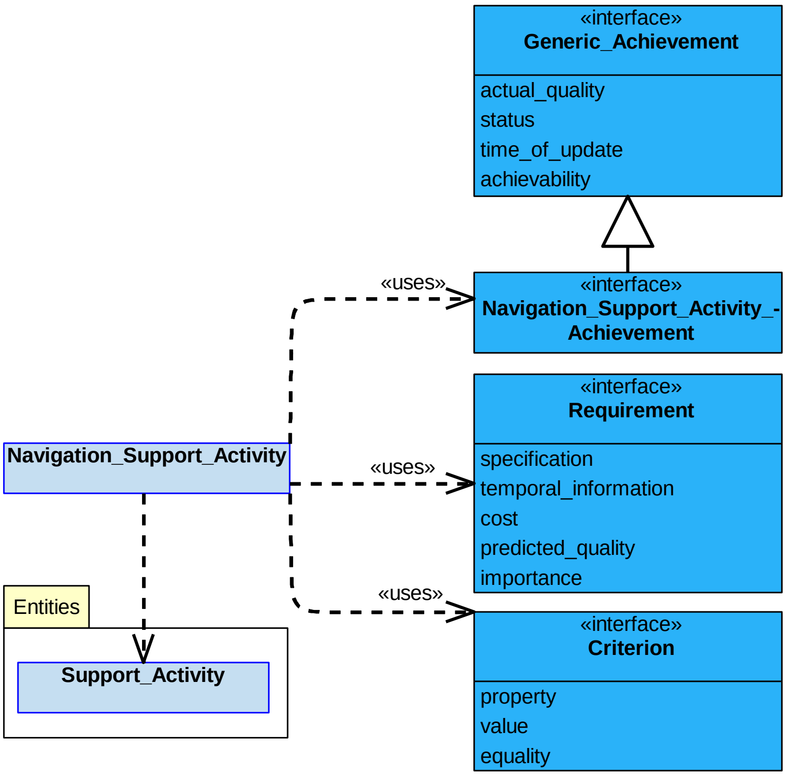


Figure 625: Navigation_Support_Activity Service Definition

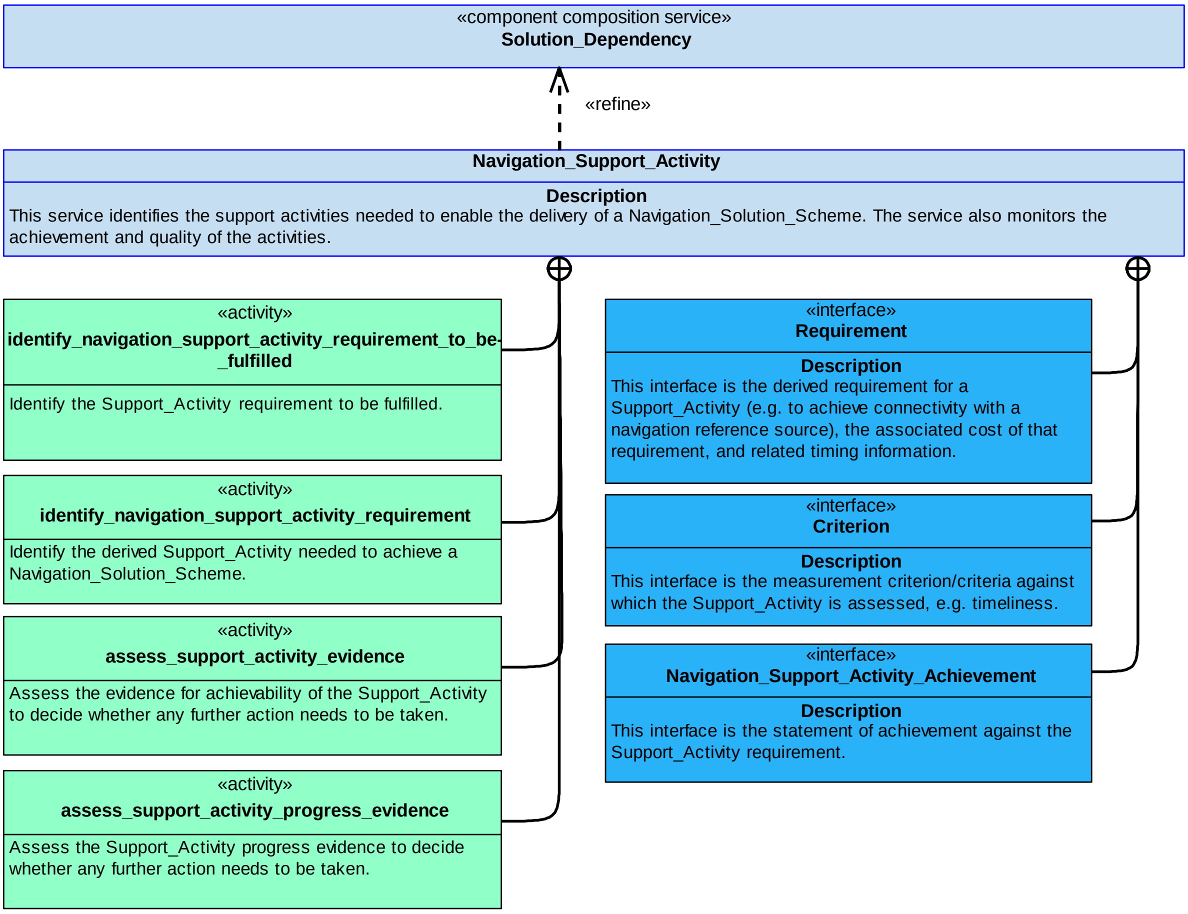

«component composition service»
Solution_Dependency
«refine»
Navigation_Support_Activity
Description
This service identifies the support activities needed to enable the delivery of a Navigation_Solution_Scheme. The service also monitors the achievement and quality of the activities.
«activity»
identify_navigation_support_activity_requirement_to_be_fulfilled
Identify the Support_Activity requirement to be fulfilled.
«activity»
identify_navigation_support_activity_requirement
Identify the derived Support_Activity needed to achieve a Navigation_Solution_Scheme.
«activity»
assess_support_activity_evidence
Assess the evidence for achievability of the Support_Activity to decide whether any further action needs to be taken.
«activity»
assess_support_activity_progress_evidence
Assess the Support_Activity progress evidence to decide whether any further action needs to be taken.
«interface»
Requirement
Description
This interface is the derived requirement for a Support_Activity (e.g. to achieve connectivity with a navigation reference source), the associated cost of that requirement, and related timing information.
«interface»
Criterion
Description
This interface is the measurement criterion/criteria against which the Support_Activity is assessed, e.g. timeliness.
«interface»
Navigation_Support_Activity_Achievement
Description
This interface is the statement of achievement against the Support_Activity requirement.

Figure 626: Navigation_Support_Activity Service Policy

Navigation_Support_Activity

This service identifies the support activities needed to enable the delivery of a Navigation_Solution_Scheme. The service also monitors the achievement and quality of the activities.

Interfaces

Navigation_Support_Activity_Achievement

This interface is the statement of achievement against the Support_Activity requirement.

Requirement

This interface is the derived requirement for a Support_Activity (e.g. to achieve connectivity with a navigation reference source), the associated cost of that requirement, and related timing information.

Attributes

specification	The definition of the Support_Activity requirement, for example, i) a dependency that must be satisfied (e.g. an inertial navigation system must be aligned), ii) a constraint that needs to be observed (e.g. remain within beacons coverage areas) or iii) a specific activity, such as the need to achieve immediate connectivity with a navigation reference source.
temporal_information	Information covering timing, such as start and end times.

cost	The cost of executing the activity, e.g. resources used or time taken.
predicted_quality	The measure(s) of how well the proposed Support_Activity solution is expected to satisfy the Support_Activity requirement.
importance	A measure of the criticality or urgency of the Support_Activity to enable the Navigation_Solution_Scheme.

Criterion

This interface is the measurement criterion/criteria against which the Support_Activity is assessed, e.g. timeliness.

Attributes

property	The property to be measured, e.g. percentage complete.
value	The value to be related to the measured property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

identify_navigation_support_activity_requirement

Identify the derived Support_Activity needed to achieve a Navigation_Solution_Scheme.

assess_support_activity_evidence

Assess the evidence for achievability of the Support_Activity to decide whether any further action needs to be taken.

assess_support_activity_progress_evidence

Assess the Support_Activity progress evidence to decide whether any further action needs to be taken.

identify_navigation_support_activity_requirement_to_be_fulfilled

Identify the Support_Activity requirement to be fulfilled.

5.4.2.36.7.1.5 Supporting_Information

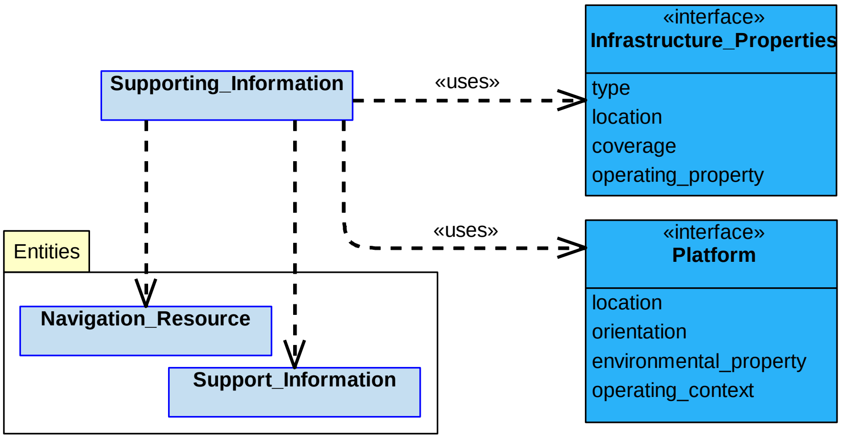


Figure 627: Supporting_Information Service Definition

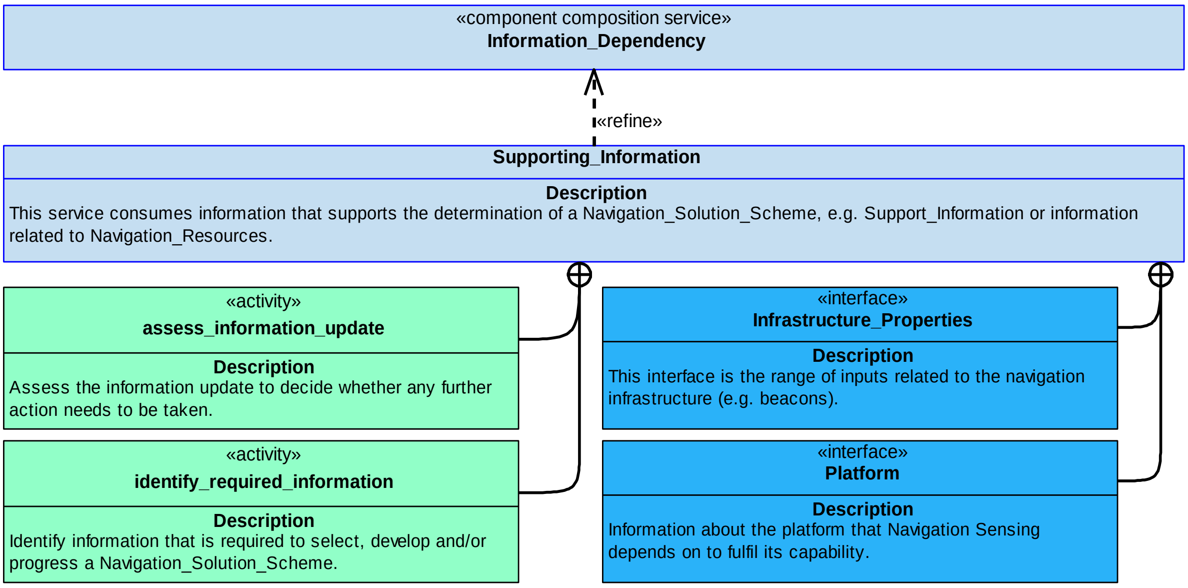

«component composition service»
Information_Dependency
«refine»
Supporting_Information
Description
This service consumes information that supports the determination of a Navigation_Solution_Scheme, e.g. Support_Information or information related to Navigation_Resources.
«activity»
assess_information_update
Description
Assess the information update to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify information that is required to select, develop and/or progress a Navigation_Solution_Scheme.
«interface»
Infrastructure_Properties
Description
This interface is the range of inputs related to the navigation infrastructure (e.g. beacons).
«interface»
Platform
Description
Information about the platform that Navigation Sensing depends on to fulfil its capability.

Figure 628: Supporting_Information Service Policy

Supporting_Information

This service consumes information that supports the determination of a Navigation_Solution_Scheme, e.g. Support_Information or information related to Navigation_Resources.

Interfaces

Infrastructure_Properties

This interface is the range of inputs related to the navigation infrastructure (e.g. beacons).

Attributes

type	The type of category of the infrastructure.
location	The location of the infrastructure element.
coverage	The range of operation or the volume of space within which the infrastructure element can be used.
operating_property	An operational property of an infrastructure element, e.g. frequency or mode.

Platform

Information about the platform that Navigation Sensing depends on to fulfil its capability.

Attributes

location	The position of the platform.
orientation	The attitude of the platform.
environmental_property	Information relating to the environment affecting the platform, e.g. visibility.
operating_context	Information relating to the operating context of the platform, e.g. military/civil operating rules.

Activities

assess_information_update

Assess the information update to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to select, develop and/or progress a Navigation_Solution_Scheme.

5.4.2.36.7.1.6 Constraint

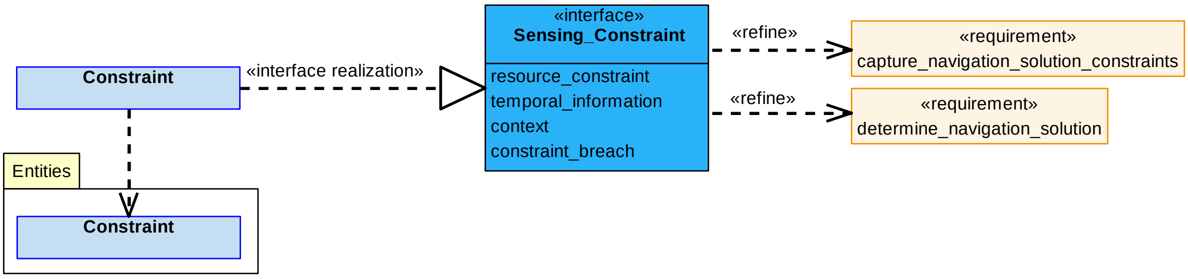


Figure 629: Constraint Service Definition

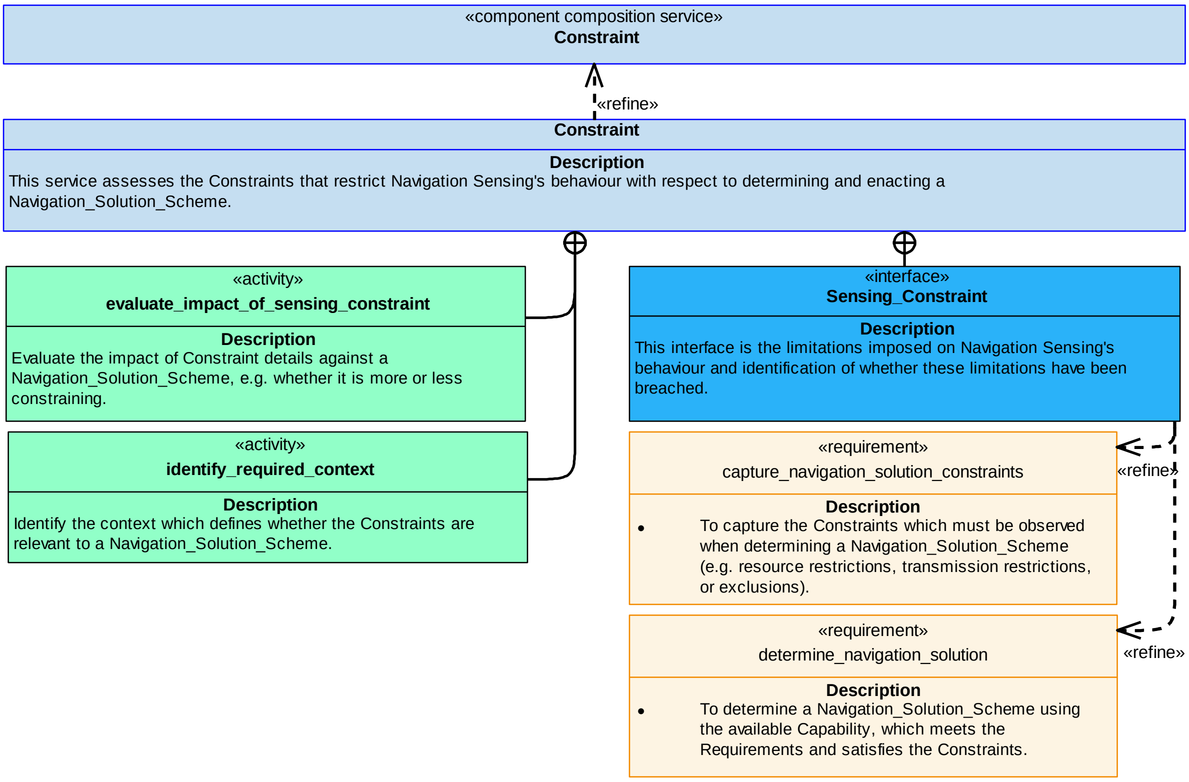

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Constraints that restrict Navigation Sensing's behaviour with respect to determining and enacting a Navigation_Solution_Scheme.
«activity»
evaluate_impact_of_sensing_constraint
Description
Evaluate the impact of Constraint details against a Navigation_Solution_Scheme, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant to a Navigation_Solution_Scheme.
«interface»
Sensing_Constraint
Description
This interface is the limitations imposed on Navigation Sensing's behaviour and identification of whether these limitations have been breached.
«requirement»
capture_navigation_solution_constraints
Description
To capture the Constraints which must be observed when determining a Navigation_Solution_Scheme (e.g. resource restrictions, transmission restrictions, or exclusions).
«refine»
«requirement»
determine_navigation_solution
Description
To determine a Navigation_Solution_Scheme using the available Capability, which meets the Requirements and satisfies the Constraints.
«refine»

Figure 630: Constraint Service Policy

Constraint

This service assesses the Constraints that restrict Navigation Sensing's behaviour with respect to determining and enacting a Navigation_Solution_Scheme.

Interface

Sensing_Constraint

This interface is the limitations imposed on Navigation Sensing's behaviour and identification of whether these limitations have been breached.

Attributes

resource_constraint	Resource Constraints applied to the Navigation_Solution_Scheme, e.g. limiting use of a Navigation_Resource (e.g. to restrict EM transmissions) or limiting the use of the information provided by the Navigation_Resource (e.g. due to spoofing).
temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
context	The context in which the Constraint is applicable.
constraint_breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_sensing_constraint

Evaluate the impact of Constraint details against a Navigation_Solution_Scheme, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant to a Navigation_Solution_Scheme.

5.4.2.36.7.1.7 Capability

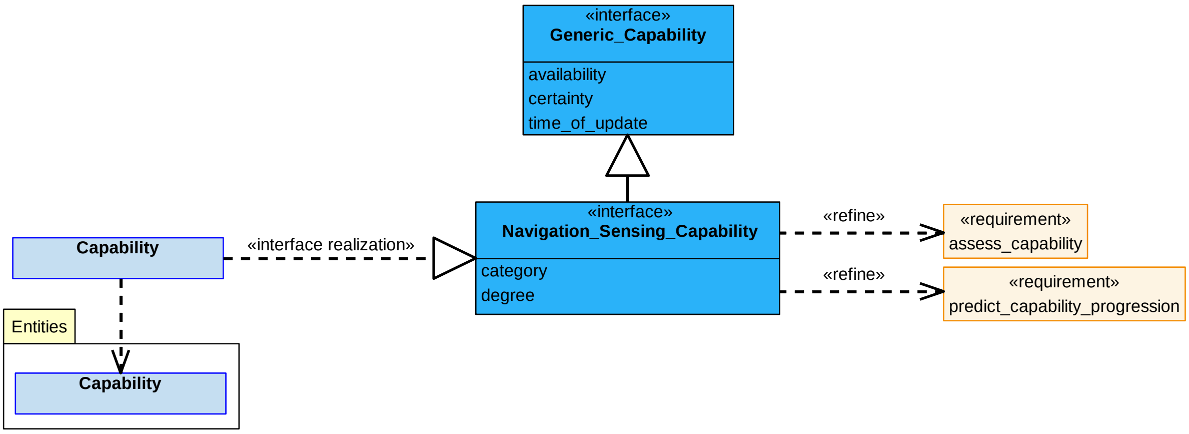


Figure 631: Capability Service Definition

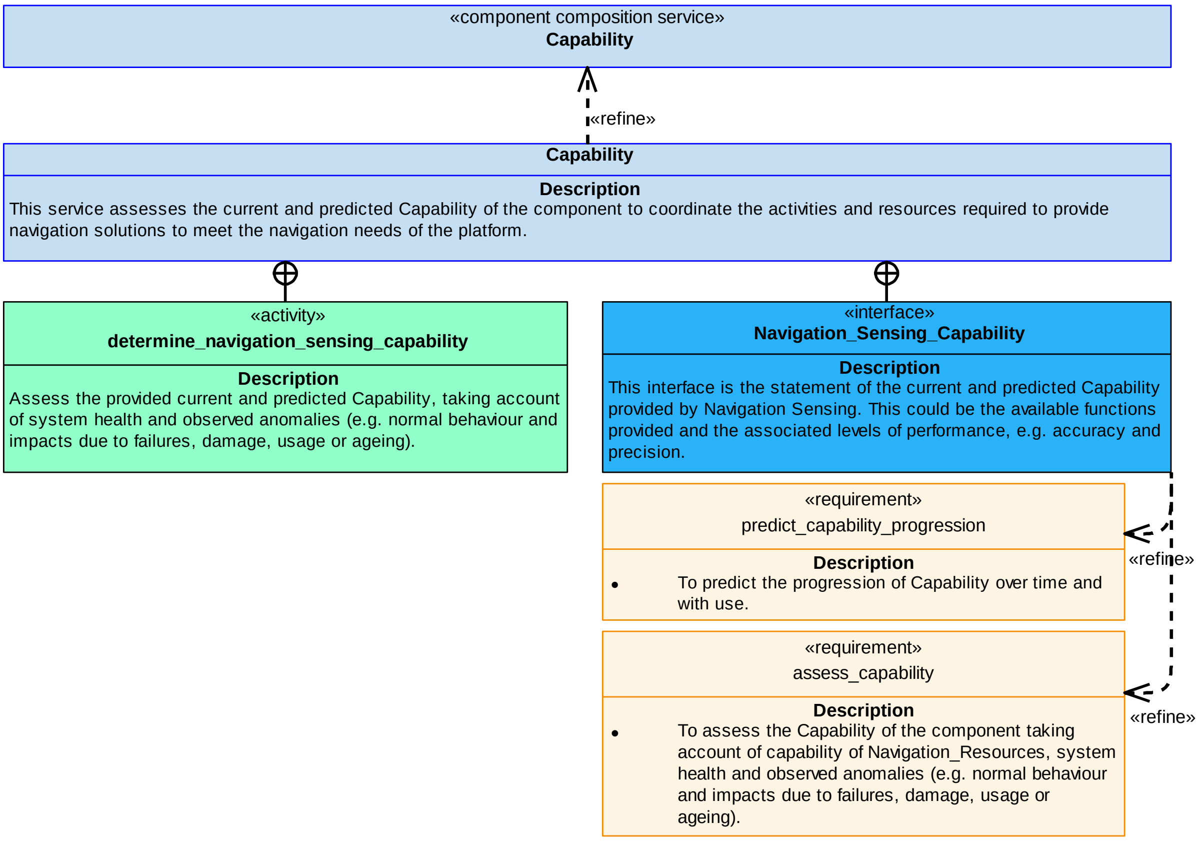

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability of the component to coordinate the activities and resources required to provide navigation solutions to meet the navigation needs of the platform.
«activity»
determine_navigation_sensing_capability
Description
Assess the provided current and predicted Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Navigation_Sensing_Capability
Description
This interface is the statement of the current and predicted Capability provided by Navigation Sensing. This could be the available functions provided and the associated levels of performance, e.g. accuracy and precision.
«requirement»
predict_capability_progression
Description
To predict the progression of Capability over time and with use.
«refine»
«requirement»
assess_capability
Description
To assess the Capability of the component taking account of capability of Navigation_Resources, system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 632: Capability Service Policy

Capability

This service assesses the current and predicted Capability of the component to coordinate the activities and resources required to provide navigation solutions to meet the navigation needs of the platform.

Interface

Navigation_Sensing_Capability

This interface is the statement of the current and predicted Capability provided by Navigation Sensing. This could be the available functions provided and the associated levels of performance, e.g. accuracy and precision.

Attributes

category	The type or category of Capability that is being provided.
degree	The level of performance or effectiveness that can be achieved for the associated Capability.

Activity

determine_navigation_sensing_capability

Assess the provided current and predicted Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.36.7.1.8 Capability_Evidence

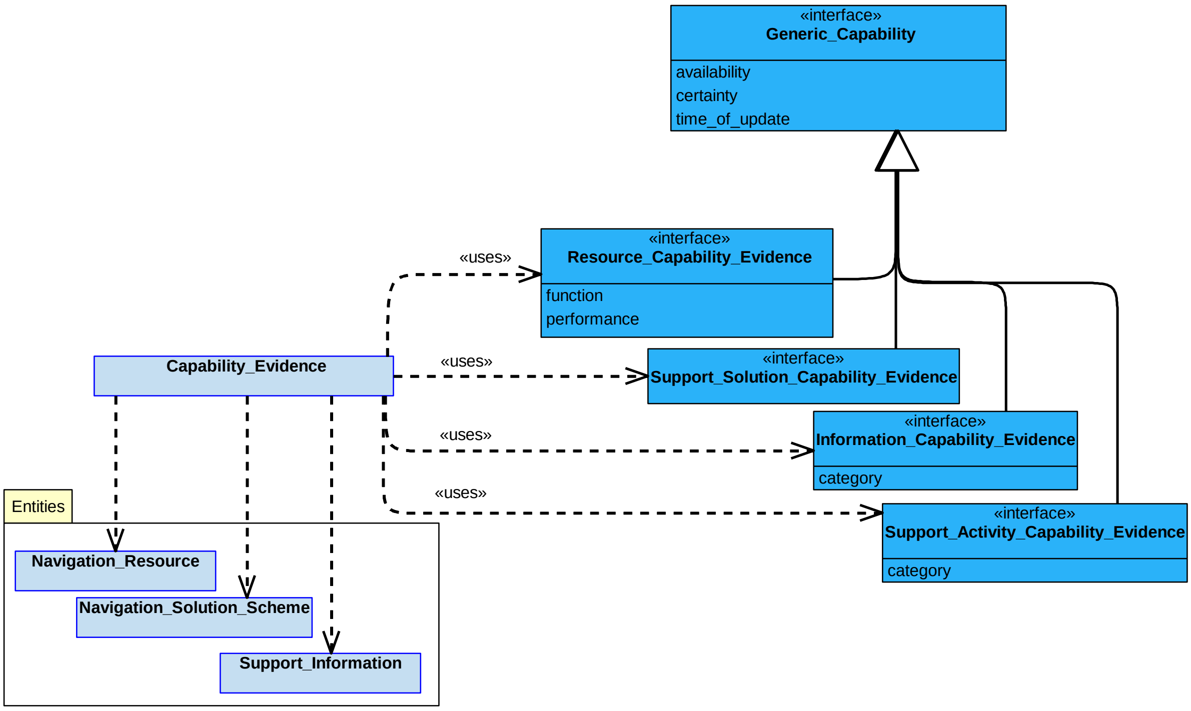
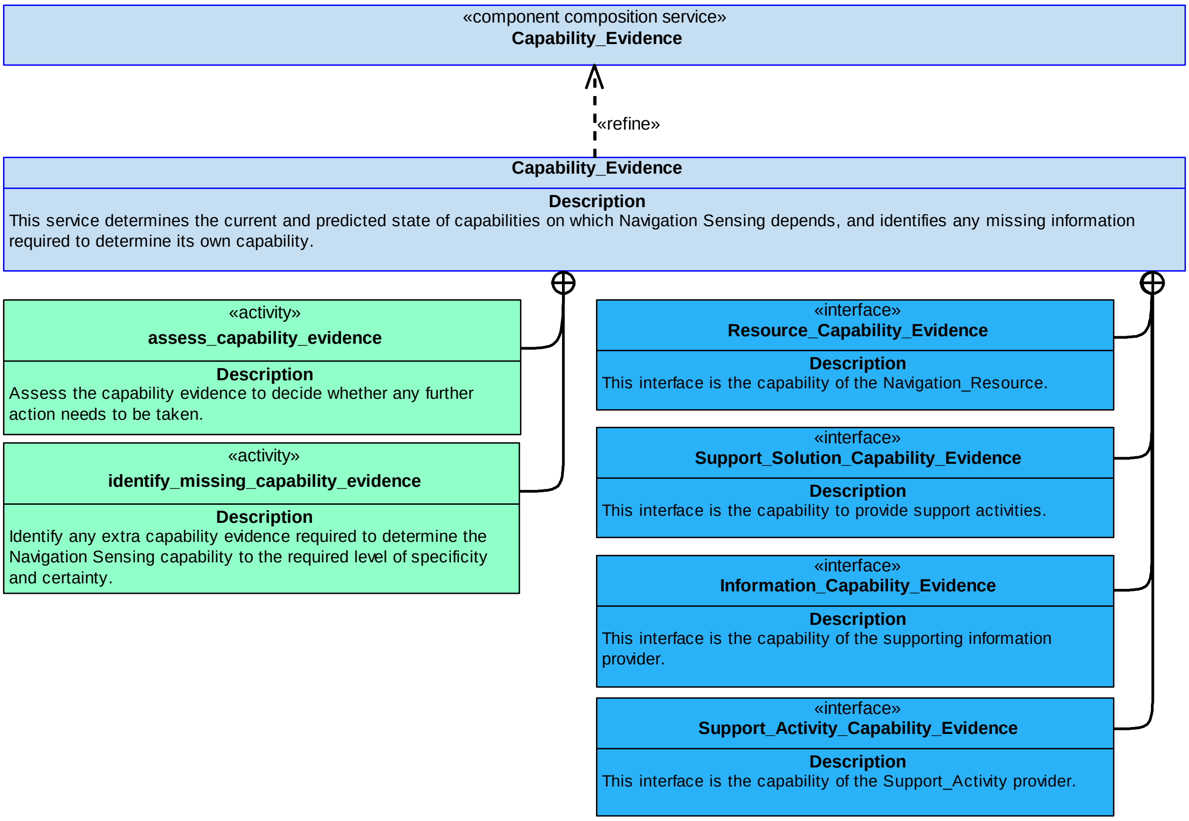


Figure 633: Capability_Evidence Service Definition


«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
«activity»
assess_capability_evidence
«activity»
identify_missing_capability_evidence
«interface»
Resource_Capability_Evidence
«interface»
Support_Solution_Capability_Evidence
«interface»
Information_Capability_Evidence
«interface»
Support_Activity_Capability_Evidence

Description
This service determines the current and predicted state of capabilities on which Navigation Sensing depends, and identifies any missing information required to determine its own capability.

Description
Assess the capability evidence to decide whether any further action needs to be taken.

Description
Identify any extra capability evidence required to determine the Navigation Sensing capability to the required level of specificity and certainty.

Description
This interface is the capability of the Navigation_Resource.

Description
This interface is the capability to provide support activities.

Description
This interface is the capability of the supporting information provider.

Description
This interface is the capability of the Support_Activity provider.

Figure 634: Capability_Evidence Service Policy

Capability_Evidence

This service determines the current and predicted state of capabilities on which Navigation Sensing depends, and identifies any missing information required to determine its own capability.

Interfaces

Resource_Capability_Evidence

This interface is the capability of the Navigation_Resource.

Attributes

function	The specific function or technique about which Capability can be defined, including the control options for its use, e.g. the ability to measure range, measure bearing, detect navigation signals (e.g. GNSS), provide position and provide pressure altitude.
performance	The level or degree of capability available for the associated function, taking into account the type, location and fit of the sensor equipment on the vehicle, e.g. field of regard, sampling rate and minimum detectable signal.

Support_Solution_Capability_Evidence

This interface is the capability to provide support activities.

Information_Capability_Evidence

This interface is the capability of the supporting information provider.

Attribute

category	The category of information that capability evidence is being provided for, e.g. platform or other infrastructure elements.

Support_Activity_Capability_Evidence

This interface is the capability of the Support_Activity provider.

Attribute

category	The category of Support_Activity that capability evidence is being provided for, e.g. capability reporting by a navigation system.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Navigation Sensing capability to the required level of specificity and certainty.

5.4.2.36.7.2 Service Dependencies

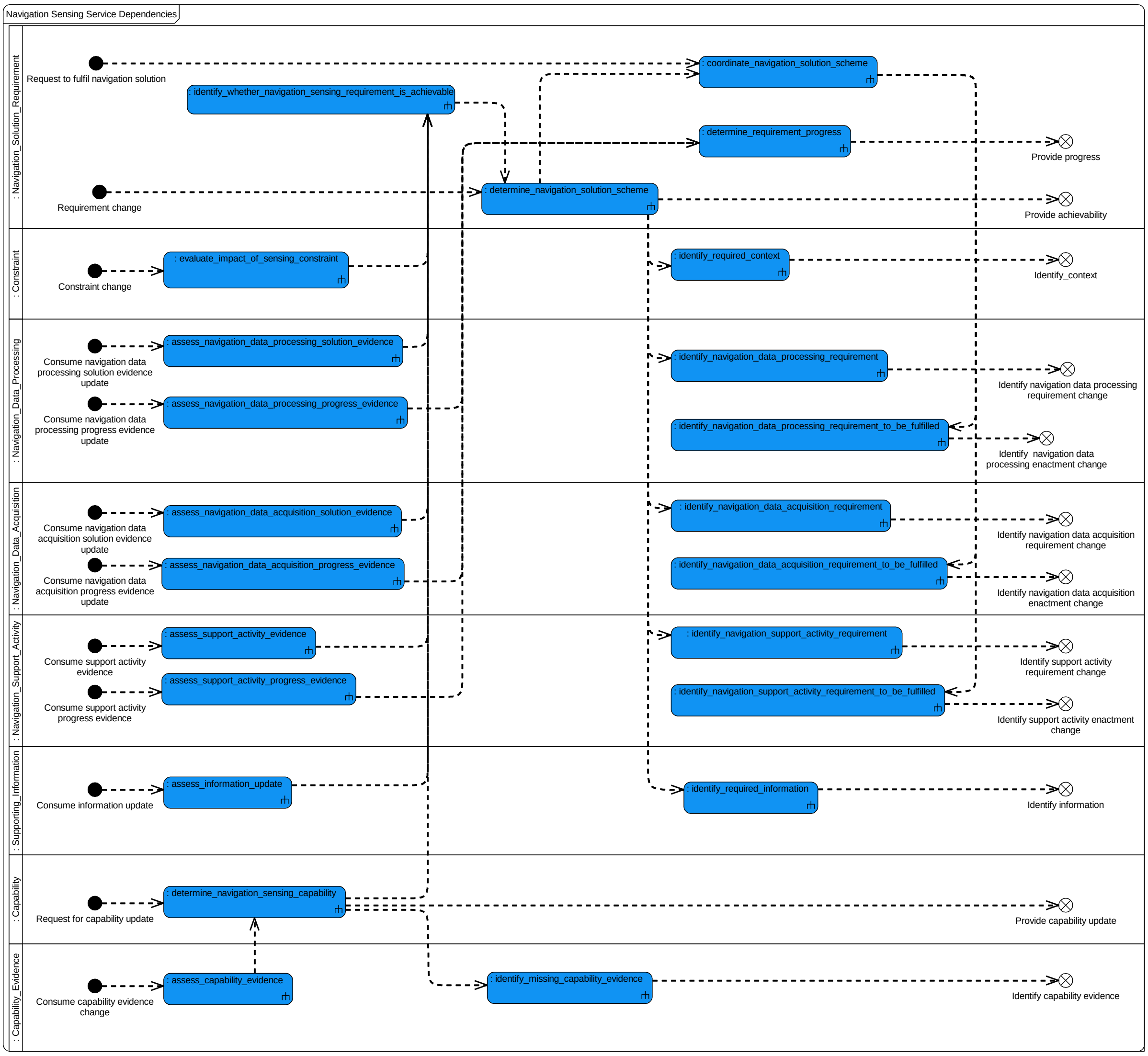

Navigation Sensing Service Dependencies
: Navigation_Solution_Requirement
: Constraint
: Navigation_Data_Processing
: Navigation_Data_Acquisition
: Navigation_Support_Activity
: Supporting_Information
: Capability
: Capability_Evidence
Request to fulfil navigation solution
: identify_whether_navigation_sensing_requirement_is_achievable
: coordinate_navigation_solution_scheme
: determine_requirement_progress
Provide progress
Requirement change
: determine_navigation_solution_scheme
Provide achievability
Constraint change
: evaluate_impact_of_sensing_constraint
: identify_required_context
Identify_context
Consume navigation data processing solution evidence update
: assess_navigation_data_processing_solution_evidence
: identify_navigation_data_processing_requirement
Identify navigation data processing requirement change
Consume navigation data processing progress evidence update
: assess_navigation_data_processing_progress_evidence
: identify_navigation_data_processing_requirement_to_be_fulfilled
Identify navigation data processing enactment change
Consume navigation data acquisition solution evidence update
: assess_navigation_data_acquisition_solution_evidence
: identify_navigation_data_acquisition_requirement
Identify navigation data acquisition requirement change
Consume navigation data acquisition progress evidence update
: assess_navigation_data_acquisition_progress_evidence
: identify_navigation_data_acquisition_requirement_to_be_fulfilled
Identify navigation data acquisition enactment change
Consume support activity evidence
: assess_support_activity_evidence
: identify_navigation_support_activity_requirement
Identify support activity requirement change
Consume support activity progress evidence
: assess_support_activity_progress_evidence
: identify_navigation_support_activity_requirement_to_be_fulfilled
Identify support activity enactment change
Consume information update
: assess_information_update
: identify_required_information
Identify information
Request for capability update
: determine_navigation_sensing_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 635: Navigation Sensing Service Dependencies

5.4.2.37 Network Routes

5.4.2.37.1 Role

The role of Network Routes is to deliver data over a network.

5.4.2.37.2 Overview

Control Architecture

Network Routes is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Network Routes coordinates the routing of data between Nodes, determining the next 'hop' in the transmission route across a network, for a specific Data_Unit.

Network Routes also maintains the network node configuration, measures the congestion and available bandwidth at points in the network and other aspects of Transmission_Quality against given measurement criteria, and uses this information to manage the network's effective use.

Examples of Use

This component may be used where:

- There is a need to determine the next onward ‘hop’ to be taken through a network for a specific Data_Unit.
- There is a need to monitor traffic flow through Nodes (e.g. to determine if a denial of service attack is taking place or to determine network performance).
- There is a need to apply Constraints on the use of hops, to meet the requirements of the system (e.g. in order to reduce congestion elsewhere).

5.4.2.37.3 Service Summary

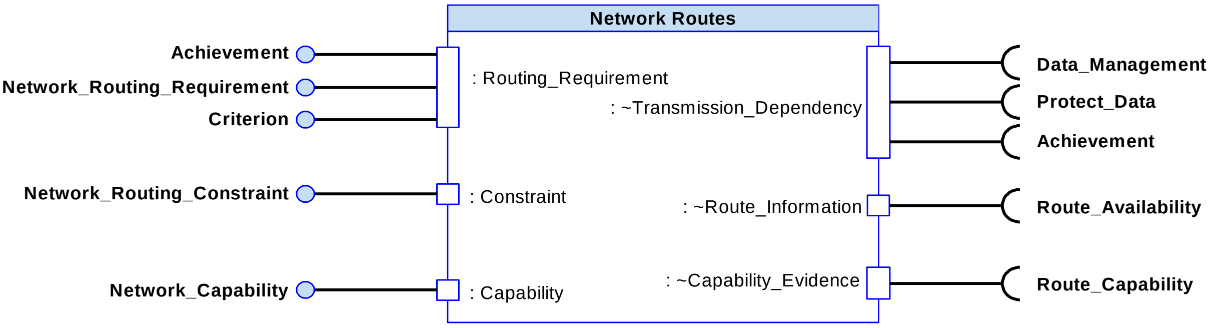


Figure 636: Network Routes Service Summary

5.4.2.37.4 Responsibilities

capture_requirements_for_network_routes

- To capture provided Transmission_Requirements.

capture_measurement_criteria_for_network_routes

- To capture provided Measurement_Criterion/criteria.

capture_constraints_for_network_routes

- To capture provided Constraints for the use of network routes.

manage_network_route_congestion

- To determine and manage the level of congestion within network Nodes.

determine_next_hop

- To select a Next_Hop from the Transmission_Routes.

determine_solution_feasibility

- To determine if a planned or on-going Next_Hop remains feasible given current capability and Constraints.

maintain_network_configuration

- To maintain the configuration of the network.

convey_data

- To convey data over a Next_Hop.

determine_network_route_quality

- To determine the Transmission_Quality - including but not limited to bandwidth utilisation, delay over a route and packet error rate.

assess_network_routes_capability

- To assess the capability provided by the network resources taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the capability assessment.

predict_network_routes_capability_progression

- To predict the progression of the capability of network routes over time and with use.

5.4.2.37.5 Subject Matter Semantics

The subject matter of Network Routes is the possible and active logical hops between Nodes within a multi-node network.

Exclusions

The subject matter of Network Routes does not include:

- Analysis of traffic information.

- Physical transmission management including encoding and decoding into electro-magnetic signals.
- Properties of physical platforms such as location and observability between platforms.
- Understanding the types of data to be passed (or its intended use).
- The wider network beyond the next hop to a Node.

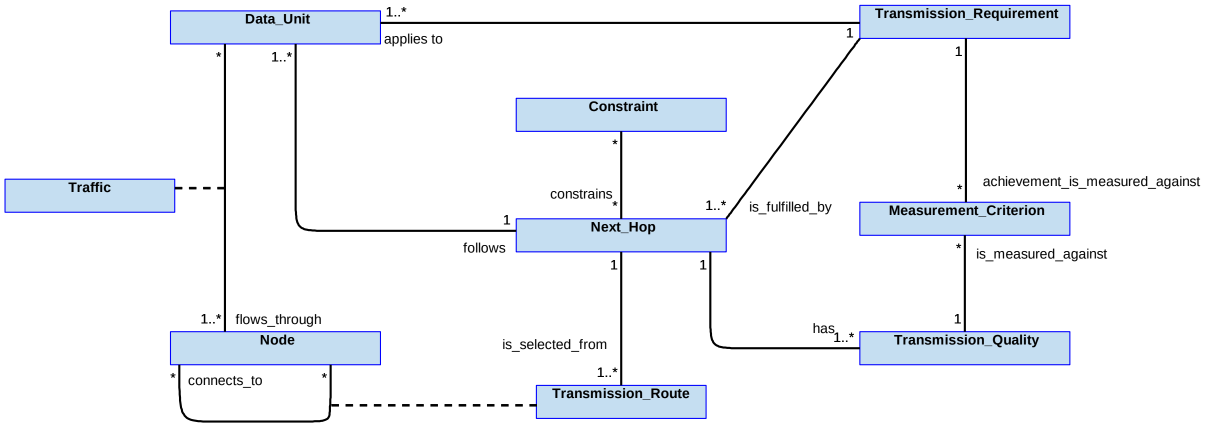


Figure 637: Network Routes Semantics

5.4.2.37.5.1 Entities

Transmission_Route

A possible transmission route that can be provided.

Constraint

A constraint which may impact and/or restrict the decisions made by this component, for example a usage limitation on a hop, or a disabled route.

Data_Unit

A specific and distinct element of data within an information flow.

Measurement_Criterion

A measurement criterion (e.g. bandwidth) against which a transmission route will be tested.

Node

A logical location within a network (e.g. a particular server) connected to other Nodes in the network.

Transmission_Quality

The measurable aspects of quality, such as congestion.

Transmission_Requirement

A requirement to deliver data in a network.

Traffic

The coordinated transmission of information flow comprising of Data_Units.

Next_Hop

The selected hop to the next node to achieve the end-to-end route across the network.

5.4.2.37.6 Design Rationale

5.4.2.37.6.1 Assumptions

- Network Routes generates traffic flow information; it does not analyse it, traffic flow analysis will be done by another component, for example Networks.
- Network Routes directs traffic towards network cryptographic devices and cross domain gateways (these are considered examples of network nodes); however it is assumed that the infrastructure will be constructed in such a way that the network is protected from this component accidentally or maliciously directing traffic across a security domain boundary without the data going through an appropriate barrier.
- For safety critical and command and control services it is assumed that resistance to corruption of the data in transit is not handled by this component in isolation (although some elements, e.g. packet CRC, do fall within its responsibility).

5.4.2.37.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Network Routes:

- Use of Communications - Multiple instances of Network Routes are likely to be used, with one for each node. Instances of Network Routes will communicate and coordinate to manage performance across a network.

Extensions

- This component could utilise extension components for specific dynamic routing algorithms or methods of congestion management.

Exploitation Considerations

- A specific instance of Network Routes only has visibility of which link or hop to use next (i.e. send data out of this interface to get to the destination), but does not have knowledge of the whole network.
- The data traffic may be encrypted or unencrypted when it passes through Network Routes; this will not change the behaviour, except where explicitly stated in a policy based route.
- Routing (how to reach all destinations and having visibility over all networks) and forwarding (the next hop for a specific stream of traffic) information can be considered to be separate. Routing is the responsibility of the Networks component and forwarding is the responsibility of the Network Routes component.
- Network Routes can use a predetermined route, or a dynamically discovered route.

5.4.2.37.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component may result in the inability to transfer data, between, for example, a ground based control station and the air vehicle. This is primarily a concern for UAVs, but may apply to manned air vehicles where some functions are controlled by external users. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) then the air vehicle will have been designed to mitigate a loss of communications. For a UAS this would be achieved by relying on pre-determined automatic or autonomous behaviour. For this failure mode it is concluded that failure of this component may result a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.
- Failure of this component may also corrupt the transfer of data, which could result in the incorrect operation of the air vehicle, potentially resulting in hazardous consequences. However, where safety critical data is being transported it is expected that the source application would have protected the data from corruption using the hashing function provided by the Cryptographic Methods component. The receiving application would only use the data if the hashing function indicated the data was not corrupted (using another instance of the Cryptographic Methods component). Therefore, corruption of the transfer of data would result in loss of "useable" data, for which DAL C is appropriate, as justified in the previous paragraph.

5.4.2.37.6.4 Security Considerations

The indicative security classification is O-S.

This component is responsible for managing the delivery of data over a network, from one Node to the next. There will be an instance of the component at each node in a security domain appropriate for the network and its traffic, however it does not have any understanding of the data it is transporting or its possible use. The confidentiality, integrity and availability requirements of the component will be specific to the Exploiting Platform’s networks and data, however this component should be considered a likely target for a cyber attack and protected as such.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to use of node connections, excessive use of a resource or changes to the routing policies (e.g. whitelisted connections), etc.
- **Supporting Secure Remote Operation** by means of establishing and maintaining the network links necessary.
- Carrying out **System Status and Monitoring**; this component is a first indicator of vulnerabilities and attacks that are typical at a network level (e.g. DoS) by monitoring for and identifying congestion and network flow issues, but it will not understand the cause. Inappropriate next-hop routing may also indicate a Man-In-The-Middle attack is taking place.

This component will protect the availability of data by redirecting network traffic in the event of bandwidth issues or DoS attack based upon externally-set priorities.

The component is expected to at least partially satisfy security enforcing functions by:

- Supporting the segregation of network traffic necessary for **Restricting Access to Data**.

5.4.2.37.7 Services

5.4.2.37.7.1 Service Definitions

5.4.2.37.7.1.1 Routing_Requirement

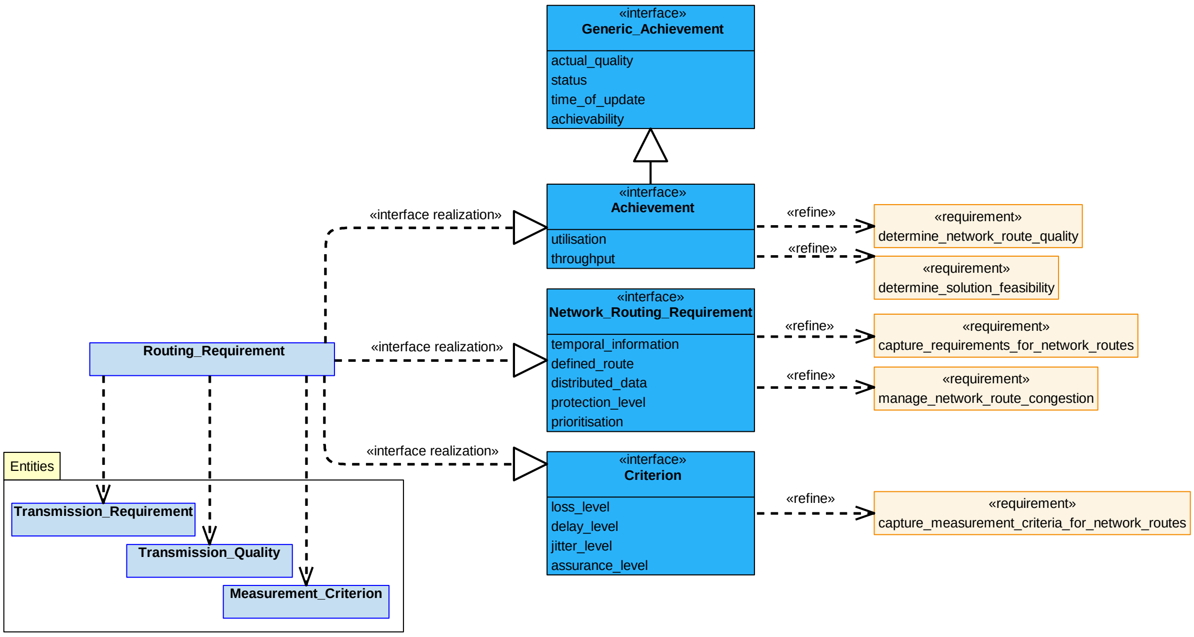


Figure 638: Routing_Requirement Service Definition

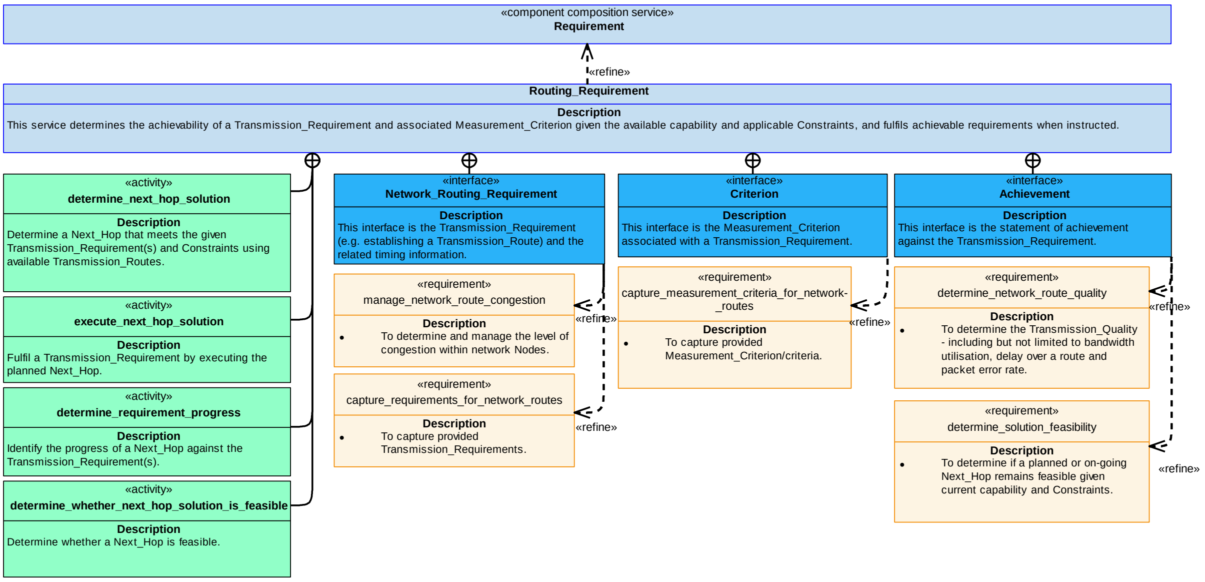

«component composition service»
Requirement
«refine»
Routing_Requirement
Description
This service determines the achievability of a Transmission_Requirement and associated Measurement_Criterion given the available capability and applicable Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_next_hop_solution
Description
Determine a Next_Hop that meets the given Transmission_Requirement(s) and Constraints using available Transmission_Routes.
«activity»
execute_next_hop_solution
Description
Fulfil a Transmission_Requirement by executing the planned Next_Hop.
«activity»
determine_requirement_progress
Description
Identify the progress of a Next_Hop against the Transmission_Requirement(s).
«activity»
determine_whether_next_hop_solution_is_feasible
Description
Determine whether a Next_Hop is feasible.
«interface»
Network_Routing_Requirement
Description
This interface is the Transmission_Requirement (e.g. establishing a Transmission_Route) and the related timing information.
«requirement»
manage_network_route_congestion
Description
To determine and manage the level of congestion within network Nodes.
«refine»
«requirement»
capture_requirements_for_network_routes
Description
To capture provided Transmission_Requirements.
«refine»
«interface»
Criterion
Description
This interface is the Measurement_Criterion associated with a Transmission_Requirement.
«requirement»
capture_measurement_criteria_for_network-_routes
Description
To capture provided Measurement_Criterion/criteria.
«refine»
«interface»
Achievement
Description
This interface is the statement of achievement against the Transmission_Requirement.
«requirement»
determine_network_route_quality
Description
To determine the Transmission_Quality - including but not limited to bandwidth utilisation, delay over a route and packet error rate.
«refine»
«requirement»
determine_solution_feasibility
Description
To determine if a planned or on-going Next_Hop remains feasible given current capability and Constraints.
«refine»

Figure 639: Routing_Requirement Service Policy

Routing_Requirement

This service determines the achievability of a Transmission_Requirement and associated Measurement_Criterion given the available capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Achievement

This interface is the statement of achievement against the Transmission_Requirement.

Attributes

utilisation	The amount of traffic presented to the Transmission_Route.
throughput	The theoretical amount of traffic that can be supported on the Transmission_Route.

Network_Routing_Requirement

This interface is the Transmission_Requirement (e.g. establishing a Transmission_Route) and the related timing information.

Attributes

temporal_information	Timing information related to a Transmission_Requirement, such as time to establish a Transmission_Route, or time to start or end routing of data.
defined_route	Whether a pre-defined Transmission_Route should be used.
distributed_data	The volume and type of data to be distributed across a network.
protection_level	The protection level at which the data must be sent or the Transmission_Route must support.
prioritisation	The priority of the data to be sent along a Transmission_Route.

Criterion

This interface is the Measurement_Criterion associated with a Transmission_Requirement.

Attributes

loss_level	The level of packet loss.
delay_level	The delay in delivery.
jitter_level	The variability in delay of delivery.
assurance_level	The level of assurance of a Transmission_Route, e.g. whether the Transmission_Route is approved for safety critical or control traffic.

Activities

determine_whether_next_hop_solution_is_feasible

Determine whether a Next_Hop is feasible.

determine_next_hop_solution

Determine a Next_Hop that meets the given Transmission_Requirement(s) and Constraints using available Transmission_Routes.

execute_next_hop_solution

Fulfil a Transmission_Requirement by executing the planned Next_Hop.

determine_requirement_progress

Identify the progress of a Next_Hop against the Transmission_Requirement(s).

5.4.2.37.7.1.2 Transmission_Dependency

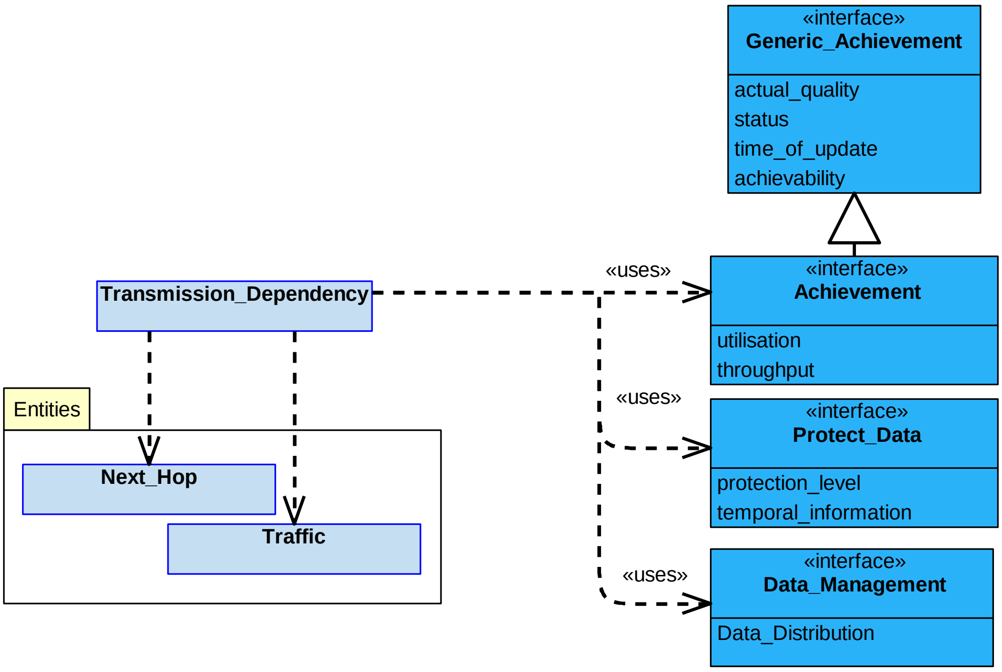


Figure 640: Transmission_Dependency Service Definition

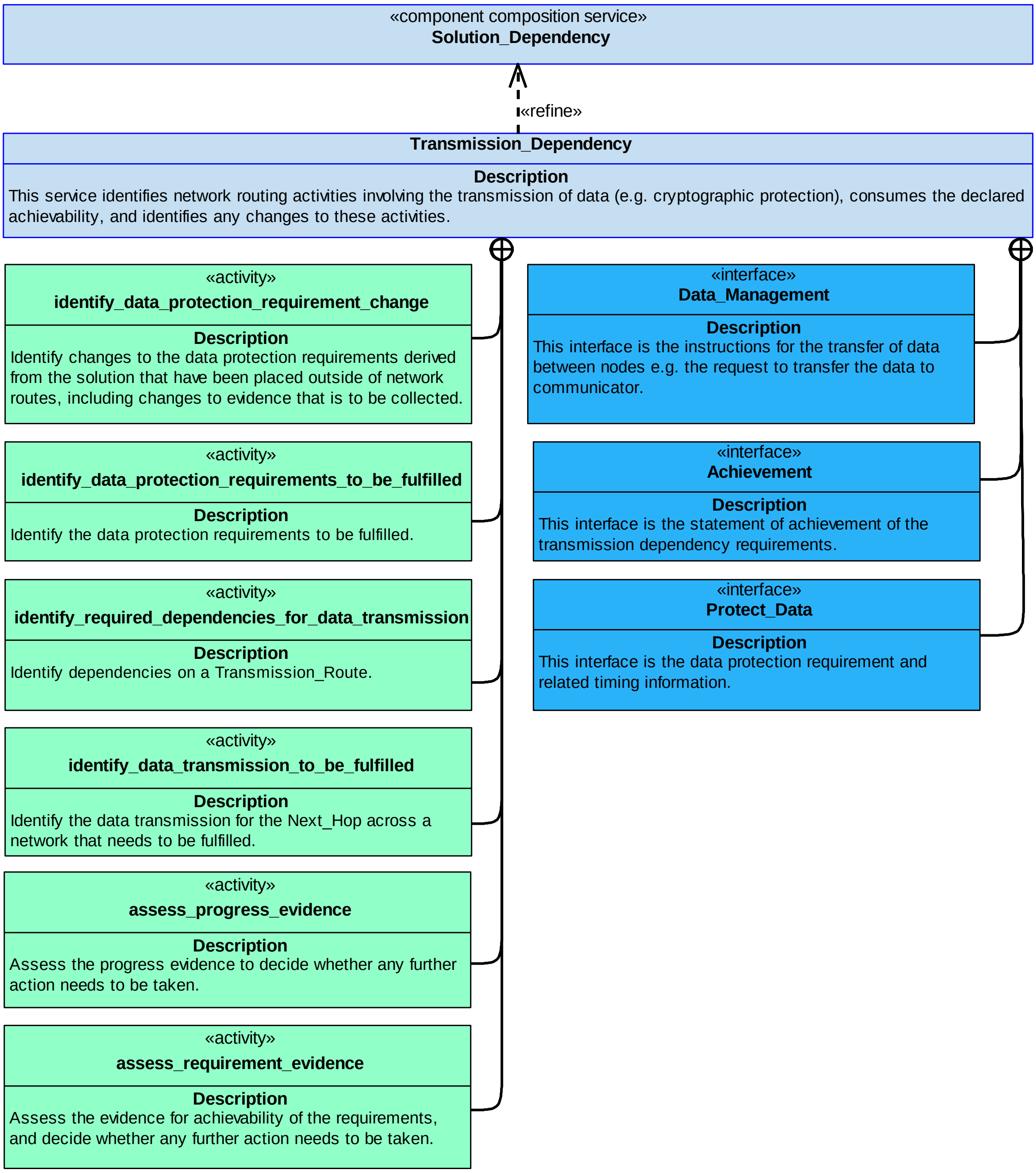

«component composition service»
Solution_Dependency
«refine»
Transmission_Dependency
Description
This service identifies network routing activities involving the transmission of data (e.g. cryptographic protection), consumes the declared achievability, and identifies any changes to these activities.
«activity»
identify_data_protection_requirement_change
Description
Identify changes to the data protection requirements derived from the solution that have been placed outside of network routes, including changes to evidence that is to be collected.
«activity»
identify_data_protection_requirements_to_be_fulfilled
Description
Identify the data protection requirements to be fulfilled.
«activity»
identify_required_dependencies_for_data_transmission
Description
Identify dependencies on a Transmission_Route.
«activity»
identify_data_transmission_to_be_fulfilled
Description
Identify the data transmission for the Next_Hop across a network that needs to be fulfilled.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
assess_requirement_evidence
Description
Assess the evidence for achievability of the requirements, and decide whether any further action needs to be taken.
«interface»
Data_Management
Description
This interface is the instructions for the transfer of data between nodes e.g. the request to transfer the data to communicator.
«interface»
Achievement
Description
This interface is the statement of achievement of the transmission dependency requirements.
«interface»
Protect_Data
Description
This interface is the data protection requirement and related timing information.

Figure 641: Transmission_Dependency Service Policy

Transmission_Dependency

This service identifies network routing activities involving the transmission of data (e.g. cryptographic protection), consumes the declared achievability, and identifies any changes to these activities.

<u>Interfaces</u>

Protect_Data

This interface is the data protection requirement and related timing information.

Attributes

protection_level	The protection level at which the data must be sent.
temporal_information	Timing information requirement, such as for how long the cryptographic transformation should occur.

Achievement

This interface is the statement of achievement of the transmission dependency requirements.

Attributes

utilisation	The amount of traffic presented to the route.
throughput	The theoretical amount of traffic that can be supported on the route.

Data_Management

This interface is the instructions for the transfer of data between nodes e.g. the request to transfer the data to communicator.

Attribute

Data_Distribution	The instructions for where data should be distributed.

Activities

identify_data_protection_requirement_change

Identify changes to the data protection requirements derived from the solution that have been placed outside of network routes, including changes to evidence that is to be collected.

identify_data_protection_requirements_to_be_fulfilled

Identify the data protection requirements to be fulfilled.

assess_requirement_evidence

Assess the evidence for achievability of the requirements, and decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_data_transmission_to_be_fulfilled

Identify the data transmission for the Next_Hop across a network that needs to be fulfilled.

identify_required_dependencies_for_data_transmission

Identify dependencies on a Transmission_Route.

5.4.2.37.7.1.3 Route_Information

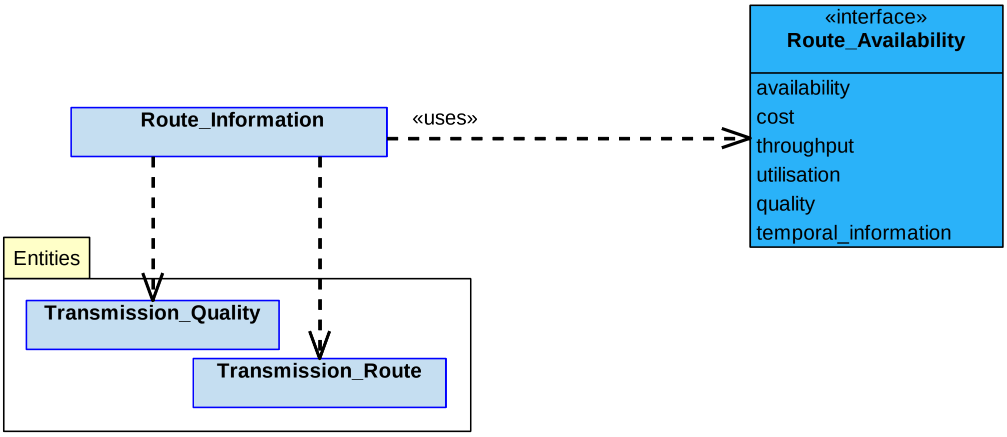


Figure 642: Routes_Information Service Definition

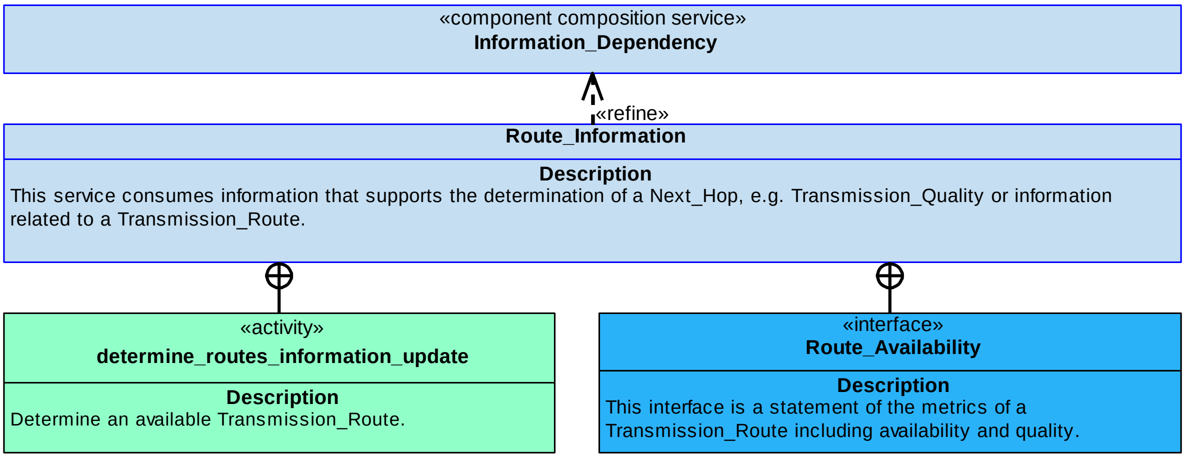

«component composition service»
Information_Dependency
«refine»
Route_Information
Description
This service consumes information that supports the determination of a Next_Hop, e.g. Transmission_Quality or information related to a Transmission_Route.
«activity»
determine_routes_information_update
Description
Determine an available Transmission_Route.
«interface»
Route_Availability
Description
This interface is a statement of the metrics of a Transmission_Route including availability and quality.

Figure 643: Routes_Information Service Policy

Route_Information

This service consumes information that supports the determination of a Next_Hop, e.g. Transmission_Quality or information related to a Transmission_Route.

Interface

Route_Availability

This interface is a statement of the metrics of a Transmission_Route including availability and quality.

Attributes

availability	Whether the Transmission_Route is available.
cost	The cost metric of a Transmission_Route.
throughput	The theoretical amount of traffic that can be supported on the Transmission_Route.
utilisation	The amount of traffic presented to the Transmission_Route.

quality	The Transmission_Quality of the Transmission_Route for example loss level, delay level and jitter level.
temporal_information	The time at which information about a Transmission_Route applies.

Activity

determine_routes_information_update

Determine an available Transmission_Route.

5.4.2.37.7.1.4 Constraint

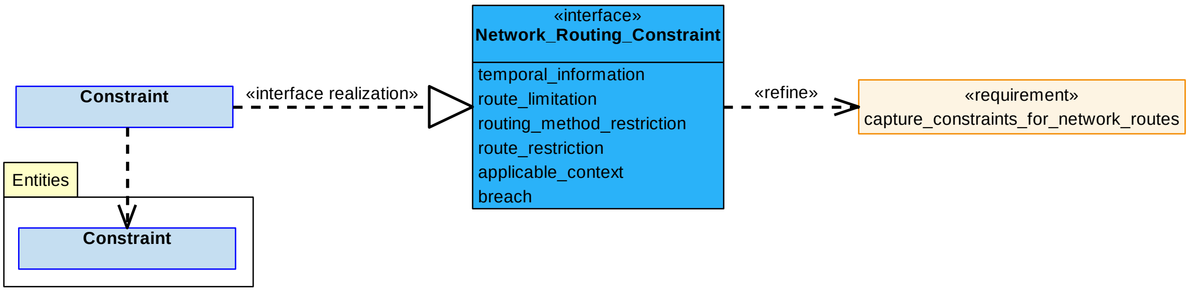


Figure 644: Constraint Service Definition

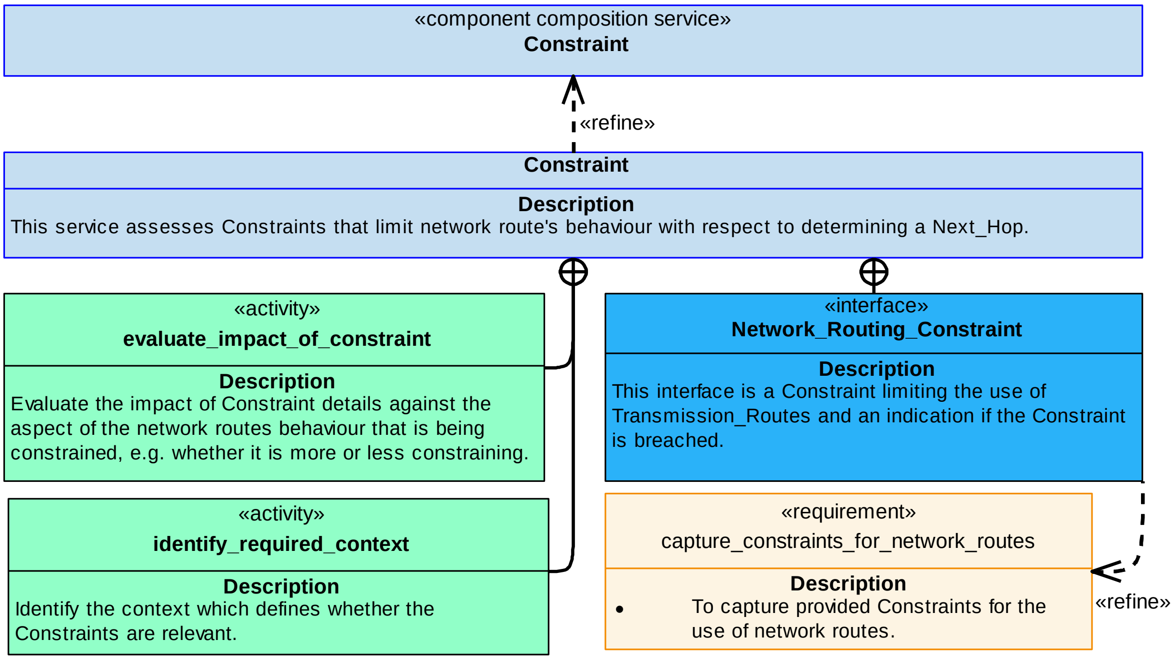

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints that limit network route's behaviour with respect to determining a Next_Hop.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of the network routes behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Network_Routing_Constraint
Description
This interface is a Constraint limiting the use of Transmission_Routes and an indication if the Constraint is breached.
«requirement»
capture_constraints_for_network_routes
Description
To capture provided Constraints for the use of network routes.
«refine»

Figure 645: Constraint Service Policy

Constraint

This service assesses Constraints that limit network route's behaviour with respect to determining a Next_Hop.

Interface

Network_Routing_Constraint

This interface is a Constraint limiting the use of Transmission_Routes and an indication if the Constraint is breached.

Attributes

temporal_information	Timing information that will constrain a solution, such as what times a data can be sent on a Transmission_Route.
route_limitation	A limit on the usage of a Transmission_Route, e.g. important data is limited to higher protection transmission routes.
routing_method_restriction	A restriction to the methods in which data can be sent on a Transmission_Route, such as only being able to send using TCP or a max MTU size.
route_restriction	A restriction to a Transmission_Route, such as constraining the volume of data, e.g. to preserve bandwidth.
applicable_context	The context in which the Constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of the network routes behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.37.7.1.5 Capability

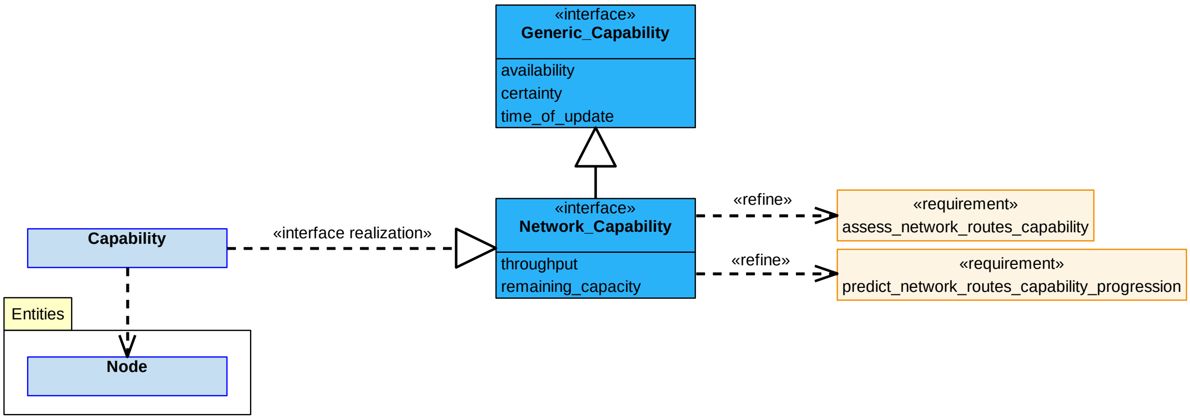


Figure 646: Capability Service Definition

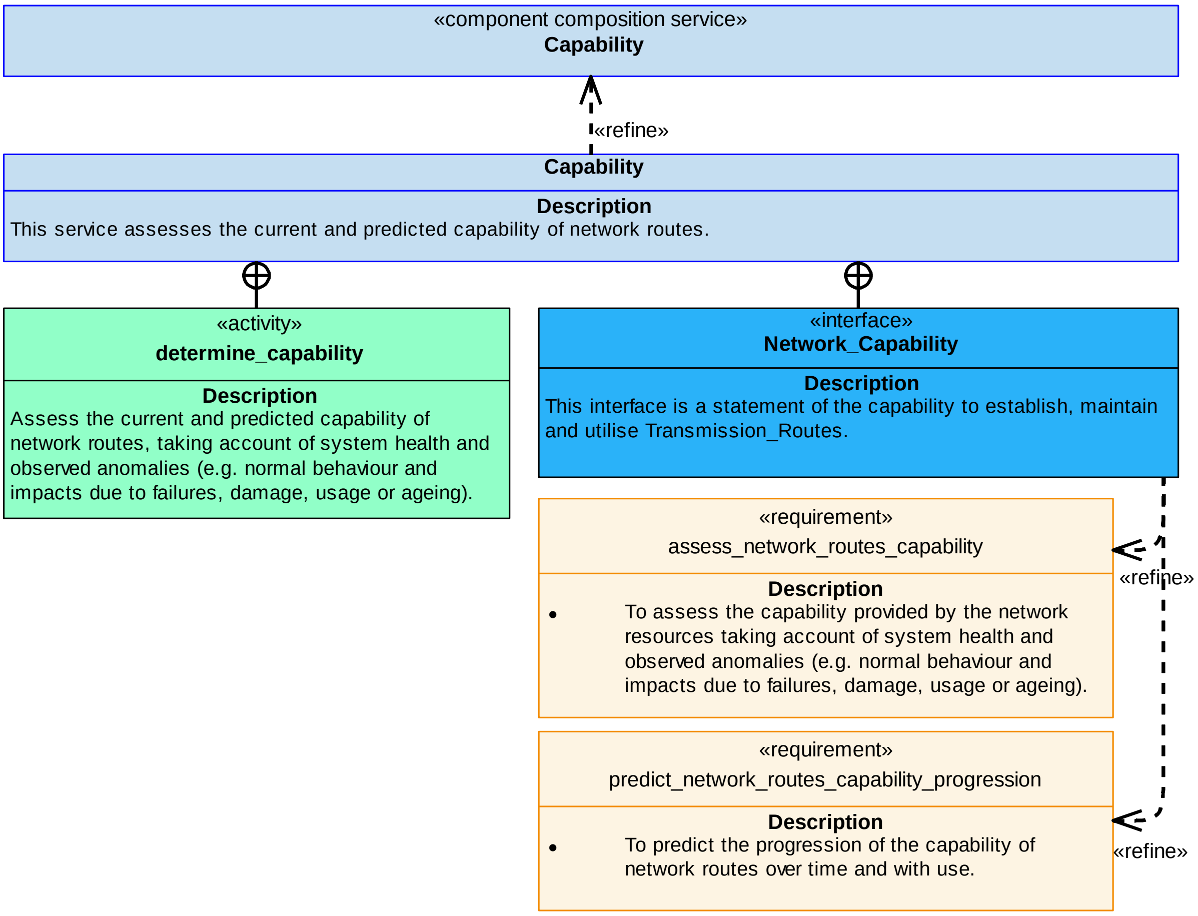

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability of network routes.
«activity»
determine_capability
Description
Assess the current and predicted capability of network routes, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Network_Capability
Description
This interface is a statement of the capability to establish, maintain and utilise Transmission_Routes.
«requirement»
assess_network_routes_capability
Description
To assess the capability provided by the network resources taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_network_routes_capability_progression
Description
To predict the progression of the capability of network routes over time and with use.
«refine»

Figure 647: Capability Service Policy

Capability

This service assesses the current and predicted capability of network routes.

Interface

Network_Capability

This interface is a statement of the capability to establish, maintain and utilise Transmission_Routes.

Attributes

throughput	The theoretical amount of traffic that can be supported on the Transmission_Route.
remaining_capacity	The remaining capacity for traffic for a Transmission_Route (e.g. after considering expected or actual utilisation).

Activity

determine_capability

Assess the current and predicted capability of network routes, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.37.7.1.6 Capability_Evidence

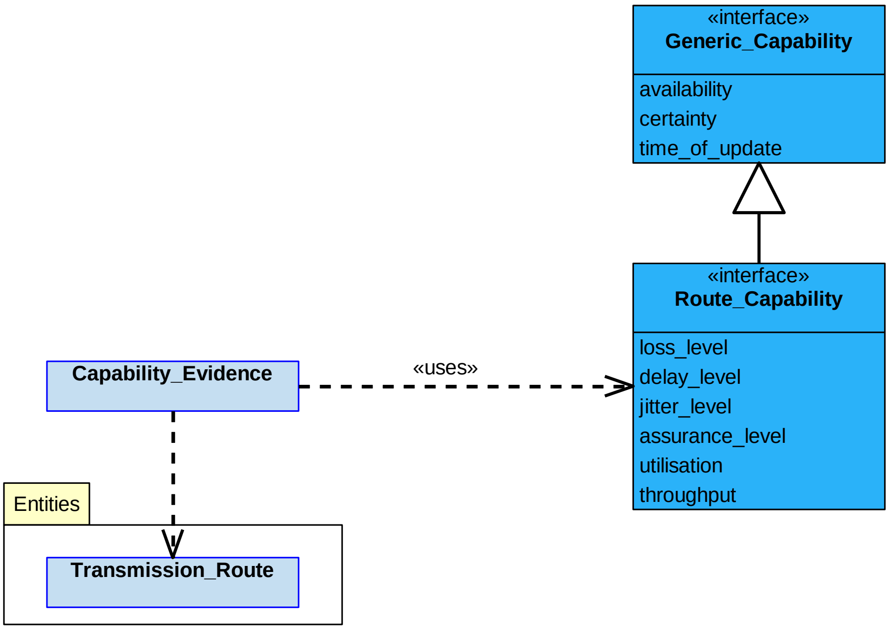


Figure 648: Capability_Evidence Service Definition

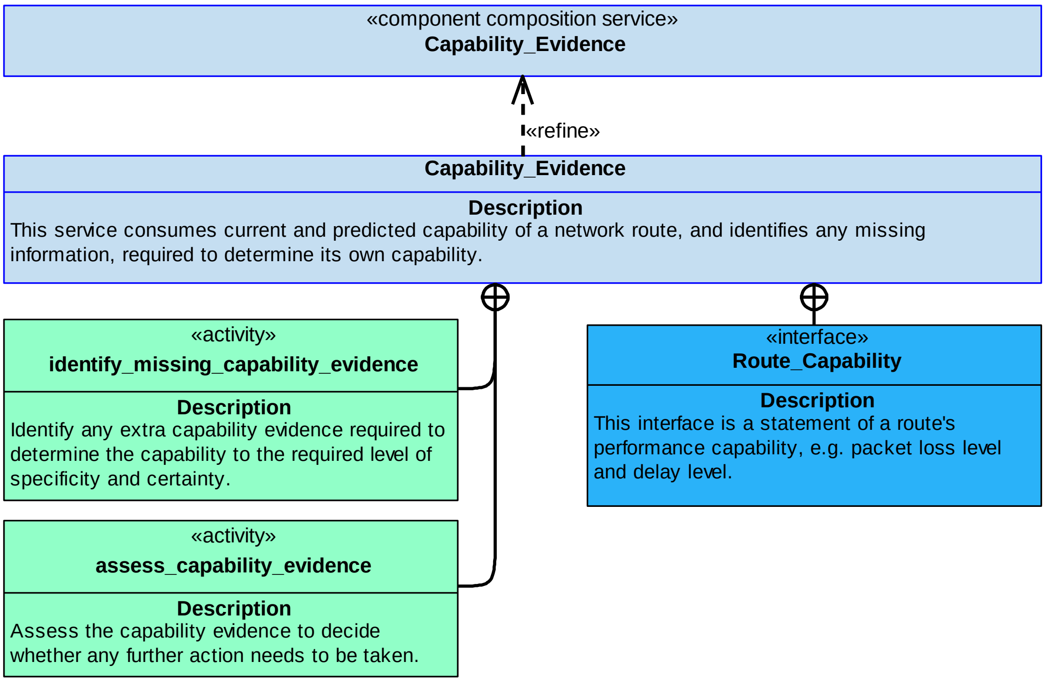

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability of a network route, and identifies any missing information, required to determine its own capability.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«interface»
Route_Capability
Description
This interface is a statement of a route's performance capability, e.g. packet loss level and delay level.

Figure 649: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability of a network route, and identifies any missing information, required to determine its own capability.

Interface

Route_Capability

This interface is a statement of a route's performance capability, e.g. packet loss level and delay level.

Attributes

loss_level	The level of packet loss.
delay_level	The delay in delivery.
jitter_level	The variability in delay of delivery.
assurance_level	The level of assurance of a route, e.g. whether the Transmission_Route is approved for safety critical or control traffic.
utilisation	The amount of traffic presented to the Transmission_Route.
throughput	The amount of traffic currently supported on the Transmission_Route.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.37.7.2 Service Dependencies

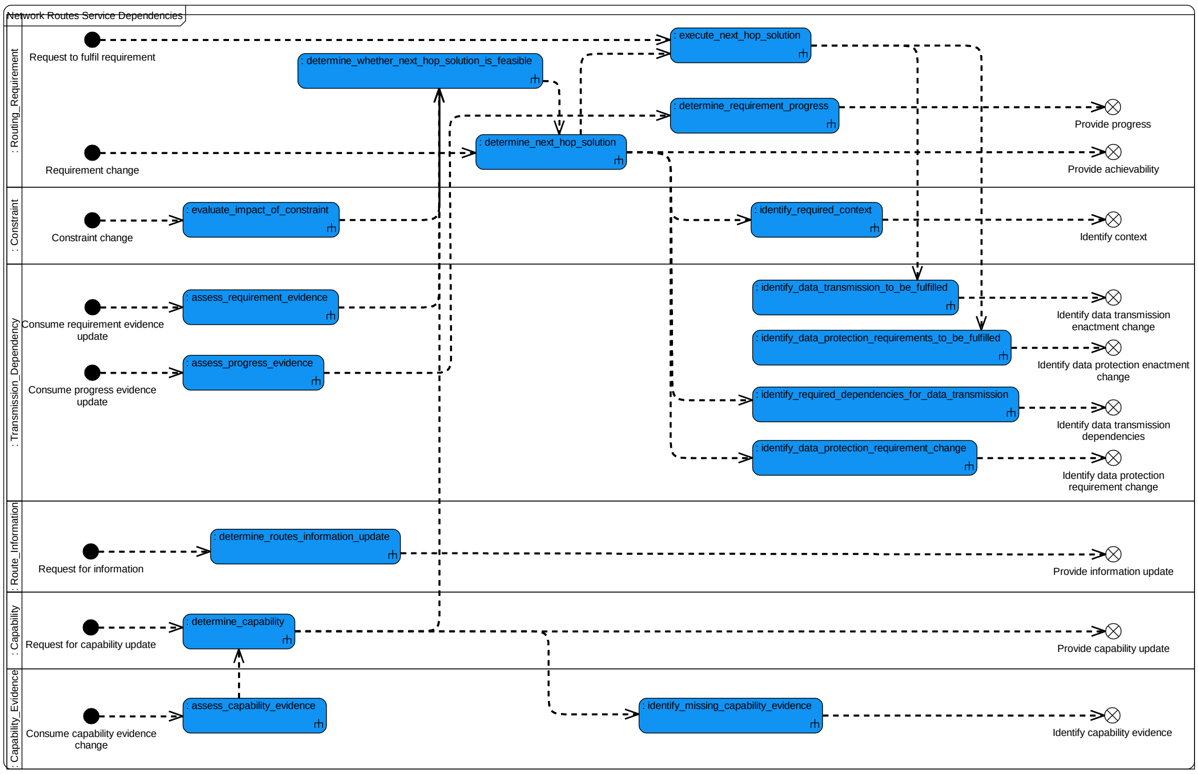

Network Routes Service Dependencies
: Routing_Requirement
Request to fulfil requirement
: determine_whether_next_hop_solution_is_feasible
: execute_next_hop_solution
: determine_requirement_progress
Provide progress
: determine_next_hop_solution
Requirement change
Provide achievability
: Constraint
: evaluate_impact_of_constraint
Constraint change
: identify_required_context
Identify context
: Transmission_Dependency
: assess_requirement_evidence
Consume requirement evidence update
: assess_progress_evidence
Consume progress evidence update
: identify_data_transmission_to_be_fulfilled
Identify data transmission enactment change
: identify_data_protection_requirements_to_be_fulfilled
Identify data protection enactment change
: identify_required_dependencies_for_data_transmission
Identify data transmission dependencies
: identify_data_protection_requirement_change
Identify data protection requirement change
: Route_Information
: determine_routes_information_update
Request for information
Provide information update
: Capability
: determine_capability
Request for capability update
Provide capability update
: Capability_Evidence
: assess_capability_evidence
Consume capability evidence change
: identify_missing_capability_evidence
Identify capability evidence

Figure 650: Network Routes Service Dependencies

5.4.2.38 Networks

5.4.2.38.1 Role

The role of Networks is to set-up, manage and optimise communications networks.

5.4.2.38.2 Overview

Control Architecture

Networks is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Networks coordinates the establishment and termination of end-to-end Connections between Nodes in response to Connectivity_Requirements. The Connections that can be established are dependent on the Capability of the Network_Resources. The allowable Connections will be restricted by the Constraint(s) between different topologies.

Networks will measure the achieved quality of service of Connections against Connectivity_Requirements using given measurement criteria.

Examples of Use

This component may be used where:

- There is a need to plan and/or manage connections between Nodes or across multiple security domains.
- There is a need to determine what connections are available for use by the system, and to initiate the establishment of additional connectivity resources, such as a new links, to meet requirements.

5.4.2.38.3 Service Summary

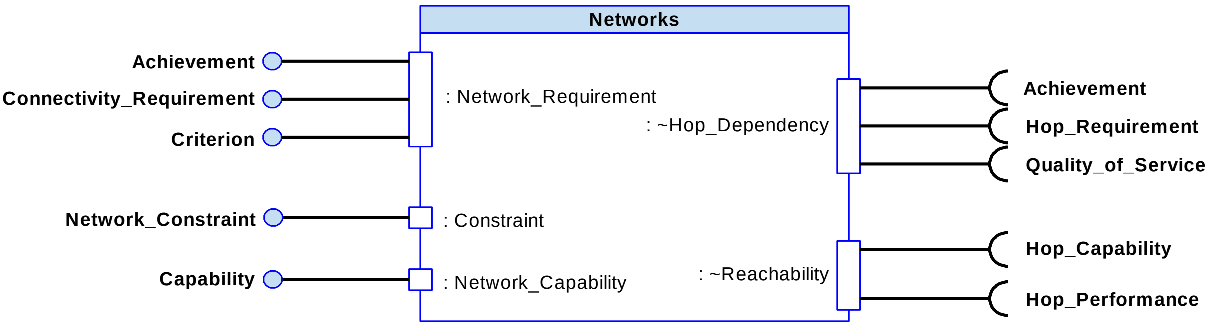


Figure 651: Networks Service Summary

5.4.2.38.4 Responsibilities

capture_network_requirements

- To capture given network Connectivity_Requirements (e.g. the need to establish a new Connection).

capture_network_measurement_criteria

- To capture given Performance/criteria (e.g. bandwidth, latency, or reach) for networks.

capture_network_constraints

- To capture given network Constraints (e.g. integrity, security, or safety).

determine_network_solution

- Determine a network Topology and Connections that support the given network Connectivity_Requirement within Network_Resource limits and the given Constraints.

identify_network_solution_in_progress_remains_feasible

- To identify whether a network Connection in progress remains feasible against particular Connectivity_Requirements and Performance/criteria given current resources.

determine_network_performance

- To determine the load and performance of a network in terms of the availability and usage of Network_Resources.

identify_network_pre-conditions

- To identify Pre-Conditions required for a Connection.

identify_network_change

- To identify divergence from the expected Topology or network performance.

maintain_network

- To establish and maintain a network Topology by management of Network_Resources.

determine_network_solution_quality

- To determine the quality of a Connection.

determine_network_capability

- To determine the Capability to provide networks using available Network_Resources, taking into account observed anomalies.

identify_missing_capability_information

- To identify missing information which could improve the certainty or specificity of network Capability determination.

predict_network_capability

- To predict the progression of network Capability over time and with use.

5.4.2.38.5 Subject Matter Semantics

The subject matter of Networks is the Nodes and Connections that form network topologies.

Exclusions

The subject matter of Networks does not include:

- The routing of specific data flow within the network.
- Physical transmission management including encoding and decoding into electro-magnetic signals.
- Properties of physical platforms such as location and observability between platforms.
- The understanding of the types of data to be passed (or its intended use), aside from priority, importance and security considerations.

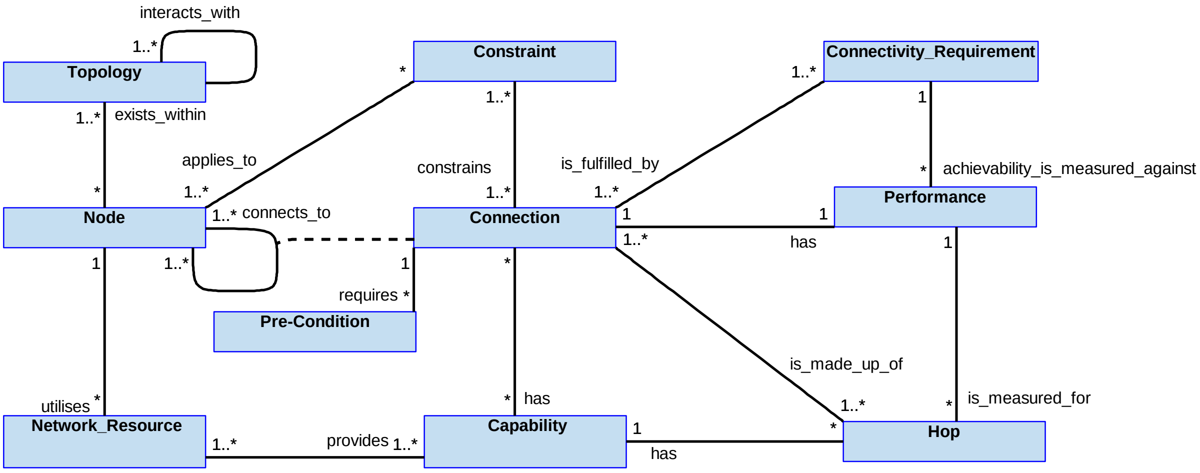


Figure 652: Networks Semantics

5.4.2.38.5.1 Entities

Capability

The capabilities that can theoretically be provided by Network_Resources (e.g. confidentiality, integrity, availability, or certified for flight control traffic).

Connection

The properties and behaviour of the network connection between Nodes. Note that the connection may be direct (a single 'hop' as an end-to-end connection) or indirect (e.g. as a series of 'hops' through intermediate Nodes).

Connectivity_Requirement

A requirement for connectivity between Nodes.

Constraint

A constraint on allowable interactions between members of the Topology, or a rule on Topology connectivity (e.g. between security domains).

Hop

An individual step across the network.

Performance

The performance (e.g. bandwidth, utilisation, latency, reachability, or speed of establishment) for a Connection or its constituent Hops.

Network_Resource

A resource used whilst providing the connectivity (e.g. routers, radios, cryptographic devices, or cross-domain gateways).

Node

A logical location within a managed communications network.

Pre-Condition

A condition that must be satisfied outside this component in order for a connection to be available.

Topology

The community of interest in which connection can exist (e.g. the same deployment or the same security domain).

5.4.2.38.6 Design Rationale

5.4.2.38.6.1 Assumptions

- Networks is responsible for controlling the network Topology, but not for handling or processing the data that traverses those networks. Thus, knowledge of content, permissions and use of the data is unknown to this component, other than how network policies may be applied to topologies.
- This component will have knowledge of traffic cryptographic devices (their location, connectivity and usage rules), will ask for them to be enabled when required, and will monitor them for traffic flow. Security and key material issues will be covered by other components.
- This component will have knowledge of traffic cryptographic devices and Topology changes for multi-level (including high) security domains.

5.4.2.38.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Networks:

- Control Architecture - As an action component, Networks is responsible for coordinating end-to-end Connections between Nodes in response to Connectivity_Requirements.
- Recording and Logging - Networks will log the status of Connections that are established between Nodes.
- Data Driving - This component will need to represent the types of topologies available. Types of supported topologies should be data-driven (at build time) in accordance with this PYRAMID concept. This allows the component to be reusable between multiple Exploiting Programmes and maintainable as behaviours change and technological resources are replaced.

- Use of Communications - Networks is responsible for managing the logical infrastructure with respect to communications.

Exploitation Considerations

- This component is aware of security and safety partitions within a network.
- This component is not directly involved in the movement of data, but manages the setting up of infrastructure for movement of data, including for safety related command and control.
- There is typically one Node per air vehicle or ground station, but this may not be the case.
- This component coordinates underlying infrastructure and Topology changes on multi-level security domains and is likely to require the ability to communicate and coordinate across the security domain boundaries.

5.4.2.38.6.3 Safety Considerations

The indicative IDAL is DAL C*.*

The rationale behind this is:

- Failure of this component may result in the inability to transfer data, between, for example, a ground based control station and the air vehicle. This is primarily a concern for UAVs, but may apply to manned air vehicles where some functions are controlled by external users. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) then the air vehicle will have been designed to mitigate a loss of communications. For a UAS this would be achieved by relying on pre-determined automatic or autonomous behaviour. For this failure mode it is concluded that failure of this component may result a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.
- This component does not handle the data being transferred. Therefore, this component cannot corrupt data.

5.4.2.38.6.4 Security Considerations

The indicative security classification is SNEO.

This component establishes and manages the available network and its infrastructure and Topology. The security domain in which this component is deployed will reflect the network and the data it transports; it does not have an understanding of the data or its use. Whilst some networks may be classified for O-S data, this component is expected to have higher confidentiality requirements as the information it holds on the network would allow a much more targeted attack should it be divulged. Network settings for system boundary protection devices (Firewalls, IPS, IDS, DMZ, etc.) may be determined and set by external sources.

This component is considered cognisant of any security and safety boundaries within a network. In order to coordinate underlying infrastructure and topology changes for multi-level (including high confidentiality, integrity or availability) networks, this component is likely to require the ability to communicate and coordinate across domain boundaries. This component will be key to ensuring availability (and priority) of data within the system thus will require additional rigour in its development and protection from corruption and attack; given its ability to control the network infrastructure, and affect communication flows, this component would be a target for attack and requires protection from such security risks. This may include appropriate corresponding access control protection (e.g. authentication of commands).

The component is expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** detailing use of certain network connections, changes to topology, configurations, traffic flow, etc.
- **Supporting Secure Remote Operation** by means of establishing and maintaining the necessary end-to-end networks.
- Carrying out **System Status and Monitoring**, poor network performance is a possible indicator of jamming or DoS type cyber attack.

The component is expected to at least partially satisfy security enforcing functions relating to:

- The devices required for traffic cryptography necessary for **Encrypting Data**, and will request traffic encryption.
- **Ensuring Separation of Security Domains** by supporting the segregation of differing classifications of network traffic; allowable Connections will be restricted according to any confidentiality and integrity constraints.
- **Preventing Cyber Attacks and Malware**; this component is the decision-maker to counter network-level cyber attack, as such it will protect the availability of data by redirecting network traffic in the event of bandwidth issues or DoS attack.
- **Restricting Access to Data** insofar as this is covered by network topology; this component does not have knowledge of user permissions etc.

5.4.2.38.7 Services

5.4.2.38.7.1 Service Definitions

5.4.2.38.7.1.1 Network_Requirement

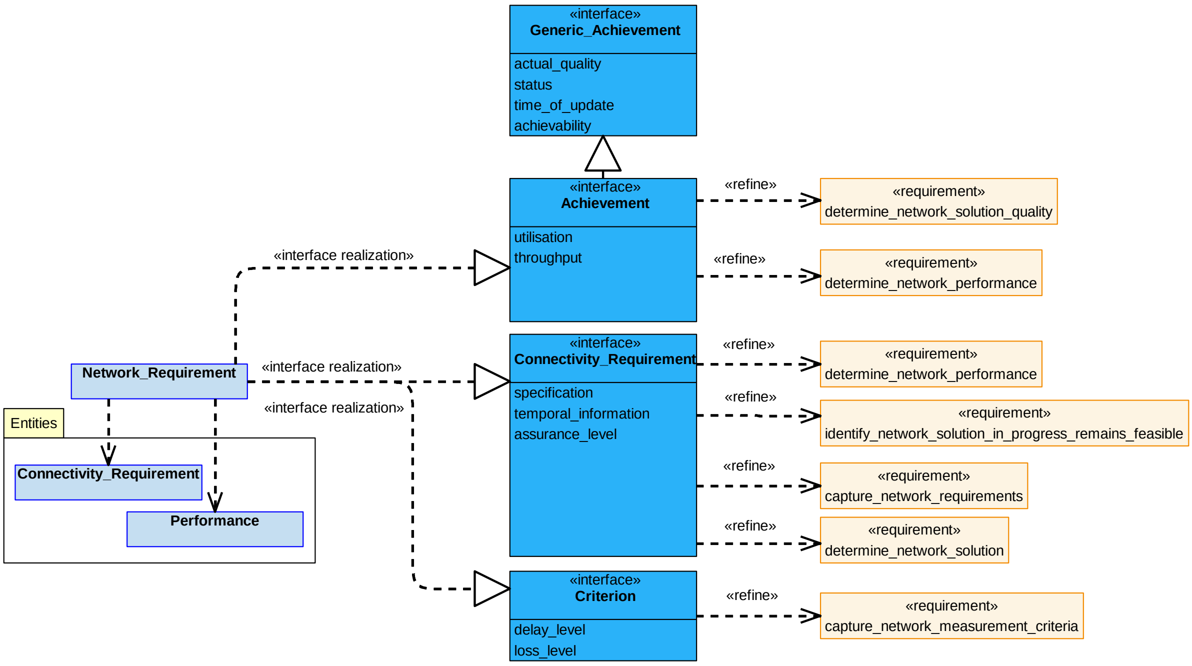


Figure 653: Network_Requirement Service Definition

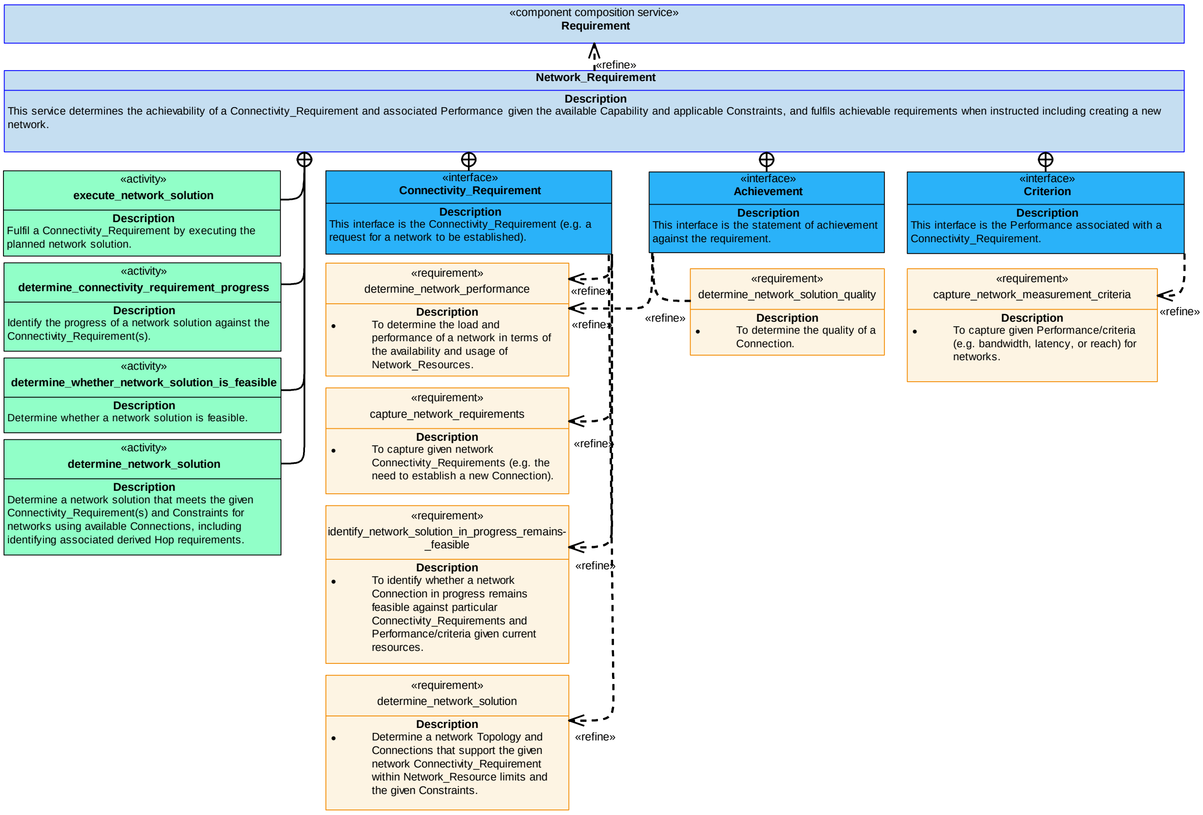

«component composition service»
Requirement
«refine»
Network_Requirement
Description
This service determines the achievability of a Connectivity_Requirement and associated Performance given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed including creating a new network.
«activity»
execute_network_solution
Description
Fulfil a Connectivity_Requirement by executing the planned network solution.
«activity»
determine_connectivity_requirement_progress
Description
Identify the progress of a network solution against the Connectivity_Requirement(s).
«activity»
determine_whether_network_solution_is_feasible
Description
Determine whether a network solution is feasible.
«activity»
determine_network_solution
Description
Determine a network solution that meets the given Connectivity_Requirement(s) and Constraints for networks using available Connections, including identifying associated derived Hop requirements.
«interface»
Connectivity_Requirement
Description
This interface is the Connectivity_Requirement (e.g. a request for a network to be established).
«requirement»
determine_network_performance
Description
To determine the load and performance of a network in terms of the availability and usage of Network_Resources.
«requirement»
capture_network_requirements
Description
To capture given network Connectivity_Requirements (e.g. the need to establish a new Connection).
«requirement»
identify_network_solution_in_progress_remains-_feasible
Description
To identify whether a network Connection in progress remains feasible against particular Connectivity_Requirements and Performance/criteria given current resources.
«requirement»
determine_network_solution
Description
Determine a network Topology and Connections that support the given network Connectivity_Requirement within Network_Resource limits and the given Constraints.
«interface»
Achievement
Description
This interface is the statement of achievement against the requirement.
«requirement»
determine_network_solution_quality
Description
To determine the quality of a Connection.
«interface»
Criterion
Description
This interface is the Performance associated with a Connectivity_Requirement.
«requirement»
capture_network_measurement_criteria
Description
To capture given Performance/criteria (e.g. bandwidth, latency, or reach) for networks.
«refine»
«refine»
«refine»
«refine»
«refine»
«refine»
«refine»

Figure 654: Network_Requirement Service Policy

Network_Requirement

This service determines the achievability of a Connectivity_Requirement and associated Performance given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed including creating a new network.

Interfaces

Connectivity_Requirement

This interface is the Connectivity_Requirement (e.g. a request for a network to be established).

Attributes

specification	What is specified to be connected, e.g. the end of the end-to-end Connection.
temporal_information	Timing information related to a Connectivity_Requirement such as time to establish a Connection, or time to start or end a Connection being used.
assurance_level	The level of assurance required of a connection, e.g. whether the network is approved for safety critical or control traffic.

Achievement

This interface is the statement of achievement against the requirement.

Attributes

utilisation	The amount of traffic presented to a network against the bandwidth.

throughput	The theoretical amount of traffic that can be supported on a network.

Criterion

This interface is the Performance associated with a Connectivity_Requirement.

Attributes

delay_level	The delay in the network including both general routing time and message latency.
loss_level	The level of packet loss in the network.

Activities

execute_network_solution

Fulfil a Connectivity_Requirement by executing the planned network solution.

determine_network_solution

Determine a network solution that meets the given Connectivity_Requirement(s) and Constraints for networks using available Connections, including identifying associated derived Hop requirements.

determine_connectivity_requirement_progress

Identify the progress of a network solution against the Connectivity_Requirement(s).

determine_whether_network_solution_is_feasible

Determine whether a network solution is feasible.

5.4.2.38.7.1.2 Hop_Dependency

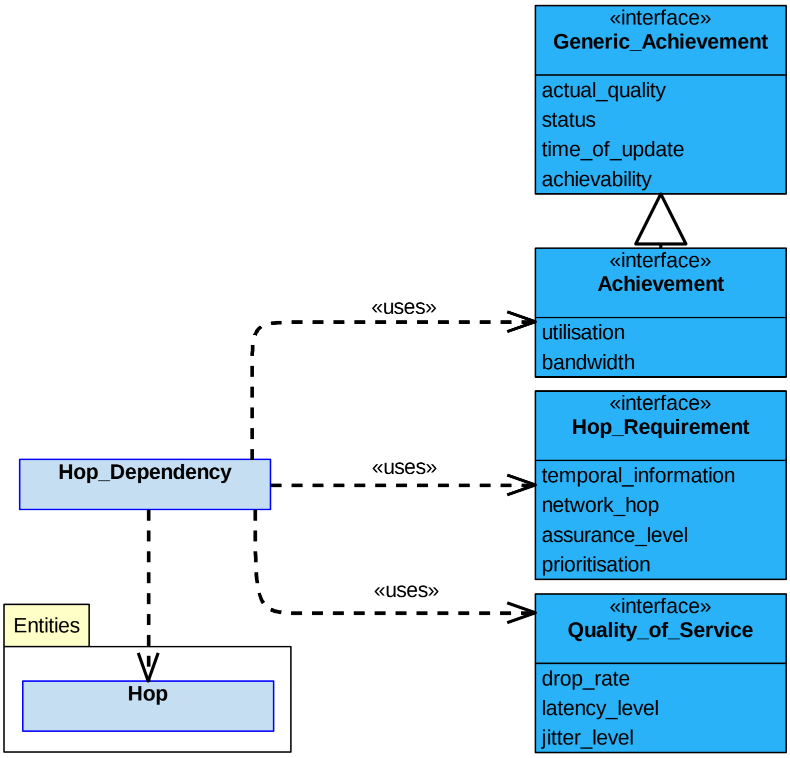


Figure 655: Hop_Dependency Service Definition

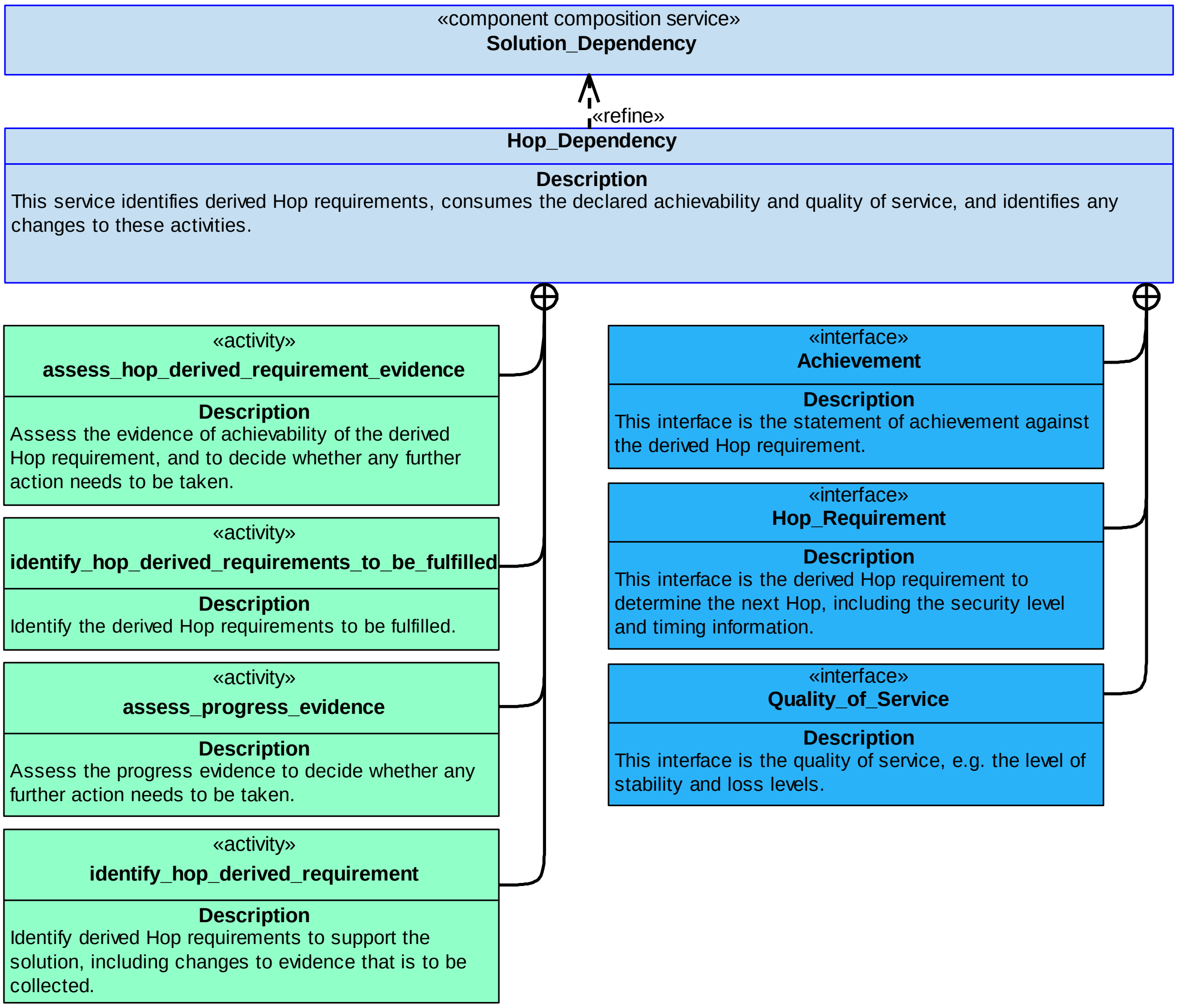

«component composition service»
Solution_Dependency
«refine»
Hop_Dependency
Description
This service identifies derived Hop requirements, consumes the declared achievability and quality of service, and identifies any changes to these activities.
«activity»
assess_hop_derived_requirement_evidence
Description
Assess the evidence of achievability of the derived Hop requirement, and to decide whether any further action needs to be taken.
«activity»
identify_hop_derived_requirements_to_be_fulfilled
Description
Identify the derived Hop requirements to be fulfilled.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
identify_hop_derived_requirement
Description
Identify derived Hop requirements to support the solution, including changes to evidence that is to be collected.
«interface»
Achievement
Description
This interface is the statement of achievement against the derived Hop requirement.
«interface»
Hop_Requirement
Description
This interface is the derived Hop requirement to determine the next Hop, including the security level and timing information.
«interface»
Quality_of_Service
Description
This interface is the quality of service, e.g. the level of stability and loss levels.

Figure 656: Hop_Dependency Service Policy

Hop_Dependency

This service identifies derived Hop requirements, consumes the declared achievability and quality of service, and identifies any changes to these activities.

<u>Interfaces</u>

Achievement

This interface is the statement of achievement against the derived Hop requirement.

<u>Attributes</u>

utilisation	The amount of the network being utilised by a Hop.
bandwidth	The actual amount of traffic supported on a Hop.

Hop_Requirement

This interface is the derived Hop requirement to determine the next Hop, including the security level and timing information.

Attributes

temporal_information	Timing information related to a Hop requirement, such as time to establish a Hop.
network_hop	Whether a pre-defined Hop should be used for a specific destination node, data type or protection level.
assurance_level	The required assurance level of a Hop, e.g. whether the Hop is approved for safety critical or control traffic.
prioritisation	The level of priority of a Hop requirement, e.g. whether this Hop requirement is of high or low importance to be fulfilled.

Quality_of_Service

This interface is the quality of service, e.g. the level of stability and loss levels.

Attributes

drop_rate	The rate of data being dropped in a Hop, e.g. packet loss.
latency_level	The latency of a Hop.
jitter_level	The variability in latency over a Hop.

Activities

assess_hop_derived_requirement_evidence

Assess the evidence of achievability of the derived Hop requirement, and to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_hop_derived_requirements_to_be_fulfilled

Identify the derived Hop requirements to be fulfilled.

identify_hop_derived_requirement

Identify derived Hop requirements to support the solution, including changes to evidence that is to be collected.

5.4.2.38.7.1.3 Constraint

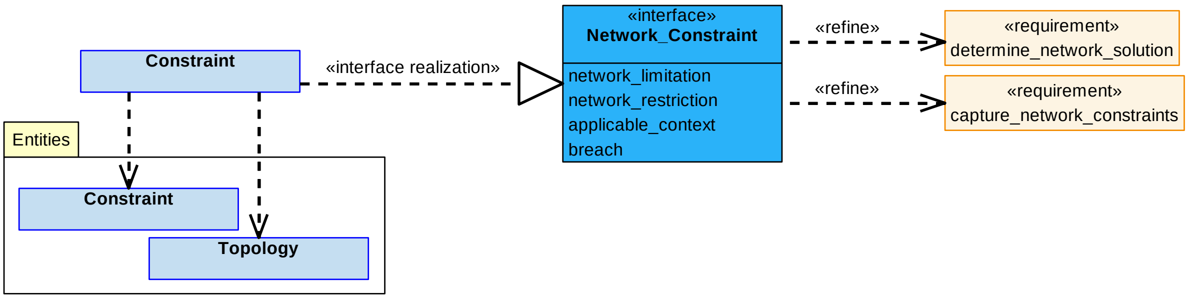


Figure 657: Constraint Service Definition

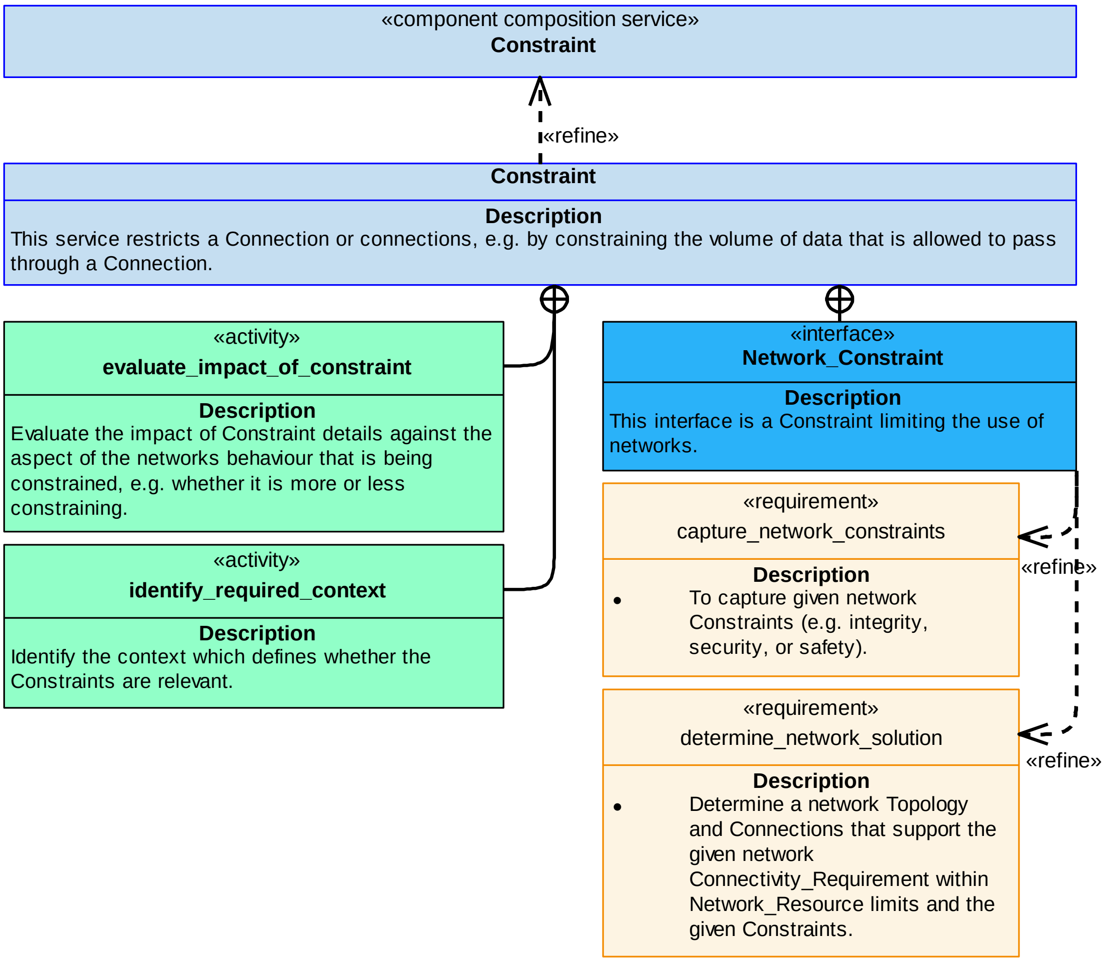

«component composition service»
Constraint
«refine»
Constraint
Description
This service restricts a Connection or connections, e.g. by constraining the volume of data that is allowed to pass through a Connection.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of the networks behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Network_Constraint
Description
This interface is a Constraint limiting the use of networks.
«requirement»
capture_network_constraints
Description
• To capture given network Constraints (e.g. integrity, security, or safety).
«refine»
«requirement»
determine_network_solution
Description
• Determine a network Topology and Connections that support the given network Connectivity_Requirement within Network_Resource limits and the given Constraints.
«refine»

Figure 658: Constraint Service Policy

Constraint

This service restricts a Connection or connections, e.g. by constraining the volume of data that is allowed to pass through a Connection.

Interface

Network_Constraint

This interface is a Constraint limiting the use of networks.

Attributes

network_limitation	A limit on the network's, or section of a network's, usage, e.g. a limit on the use of a hop known to have high latency.
network_restriction	A network, or section of a network, that is not allowed to be utilised, e.g. a network's use being restricted due to it not having the correct protection level.
applicable_context	The context in which the Constraint is applicable.
breach	A statement that a Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of the networks behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.38.7.1.4 Network_Capability

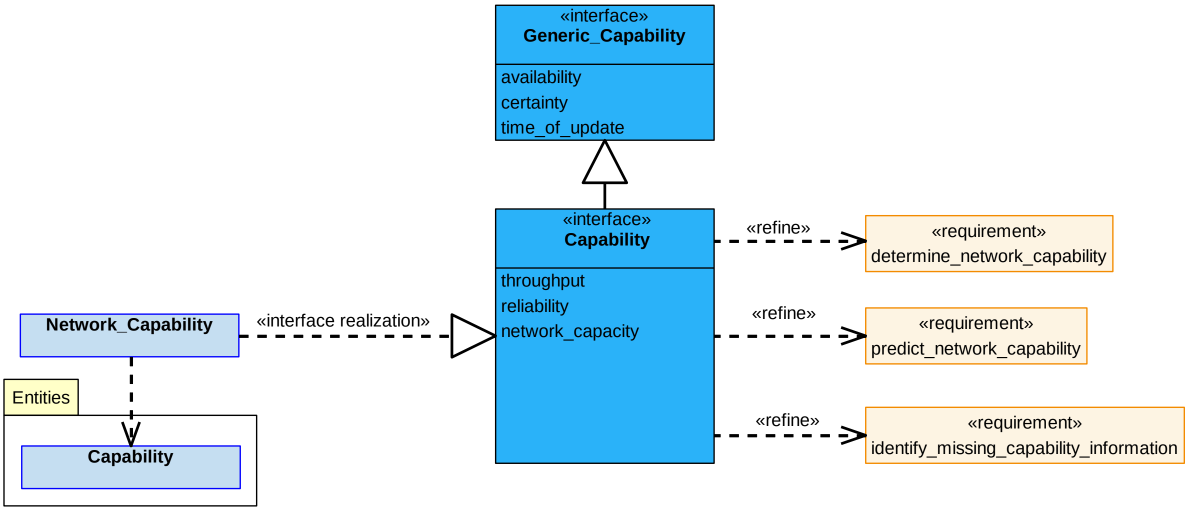


Figure 659: Network_Capability Service Definition

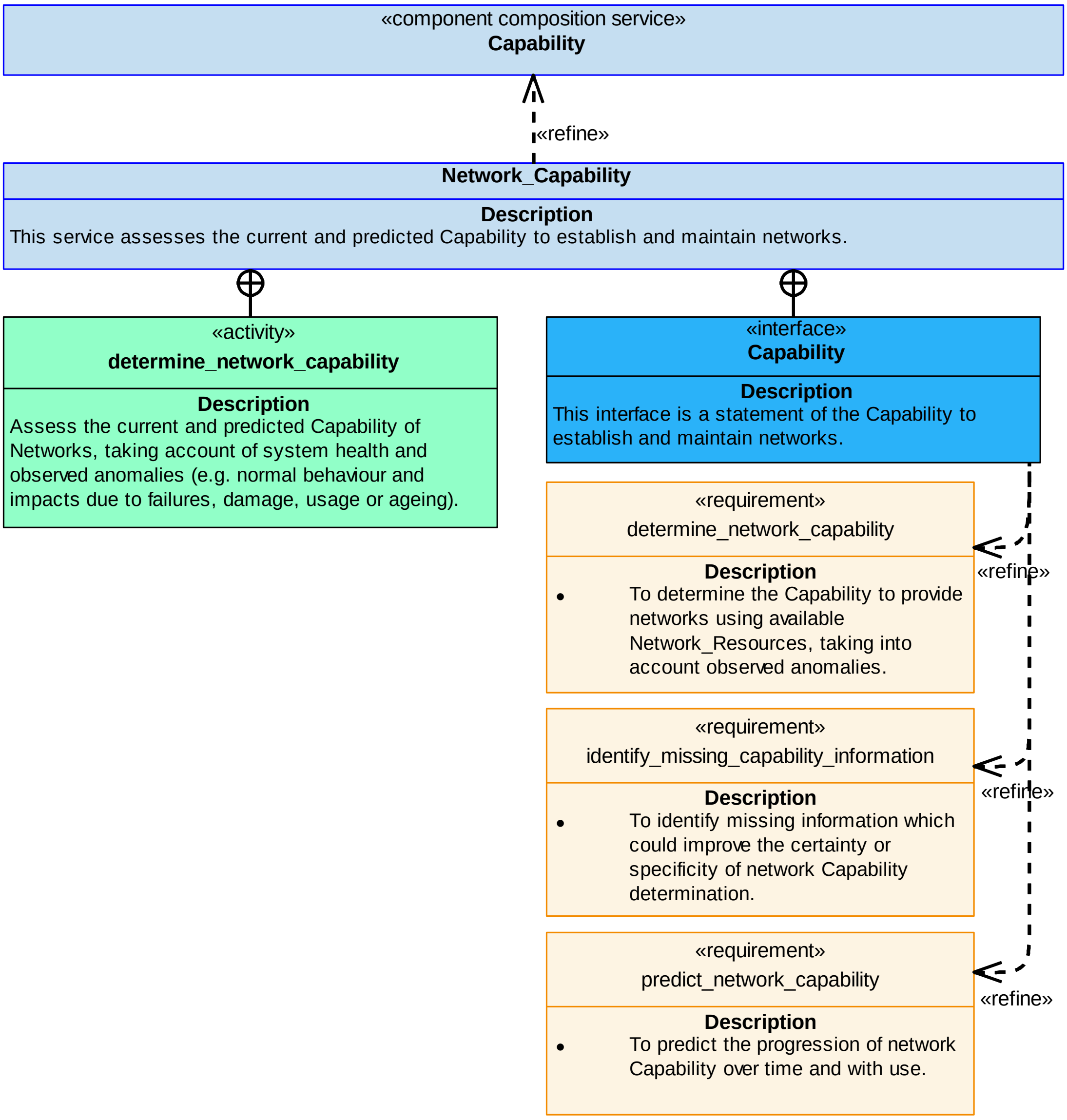

«component composition service»
Capability
«refine»
Network_Capability
Description
This service assesses the current and predicted Capability to establish and maintain networks.
«activity»
determine_network_capability
Description
Assess the current and predicted Capability of Networks, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Capability
Description
This interface is a statement of the Capability to establish and maintain networks.
«requirement»
determine_network_capability
Description
To determine the Capability to provide networks using available Network_Resources, taking into account observed anomalies.
«refine»
«requirement»
identify_missing_capability_information
Description
To identify missing information which could improve the certainty or specificity of network Capability determination.
«refine»
«requirement»
predict_network_capability
Description
To predict the progression of network Capability over time and with use.
«refine»

Figure 660: Network_Capability Service Policy

Network_Capability

This service assesses the current and predicted Capability to establish and maintain networks.

Interface

Capability

This interface is a statement of the Capability to establish and maintain networks.

Attributes

throughput	The theoretical amount of traffic that can be supported on a network.
reliability	The reliability of the network, e.g. whether an end to end Connection is immune to termination, losses and delay.
network_capacity	The amount of the network available to be used, e.g. available bandwidth.

Activity

determine_network_capability

Assess the current and predicted Capability of Networks, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.38.7.1.5 Reachability

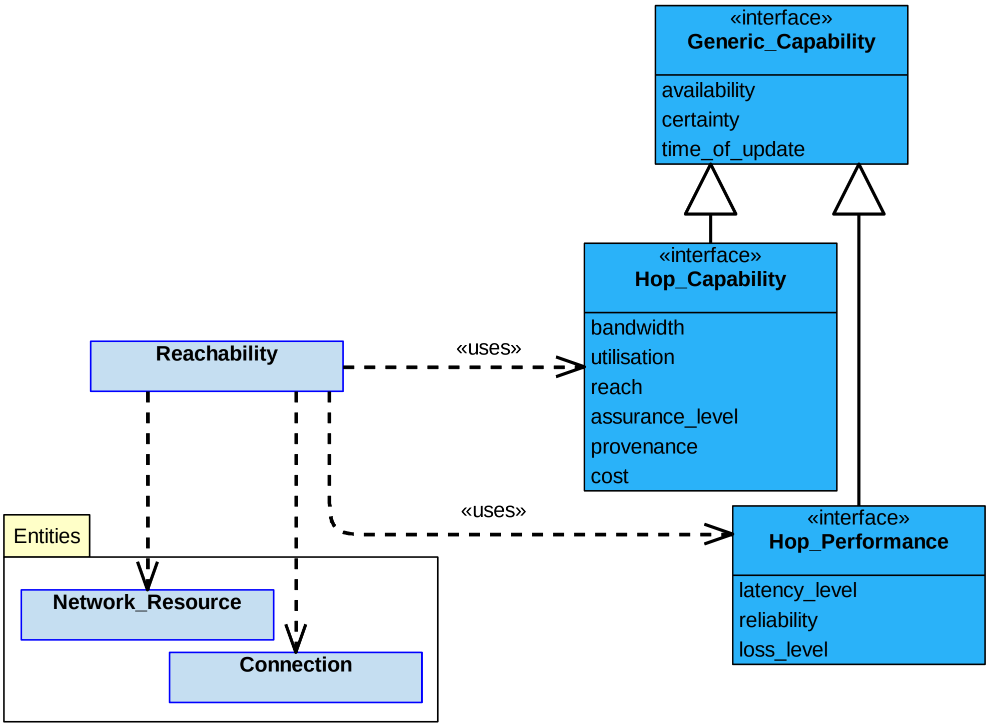


Figure 661: Reachability Service Definition

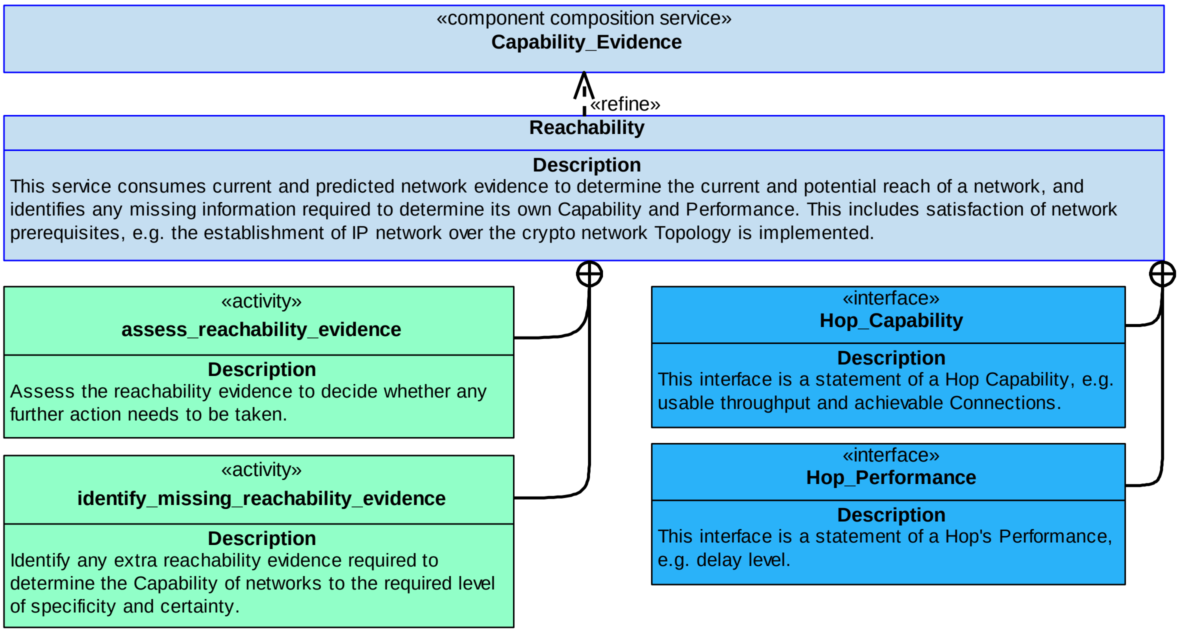

«component composition service»
Capability_Evidence
«refine»
Reachability
Description
This service consumes current and predicted network evidence to determine the current and potential reach of a network, and identifies any missing information required to determine its own Capability and Performance. This includes satisfaction of network prerequisites, e.g. the establishment of IP network over the crypto network Topology is implemented.
«activity»
assess_reachability_evidence
Description
Assess the reachability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_reachability_evidence
Description
Identify any extra reachability evidence required to determine the Capability of networks to the required level of specificity and certainty.
«interface»
Hop_Capability
Description
This interface is a statement of a Hop Capability, e.g. usable throughput and achievable Connections.
«interface»
Hop_Performance
Description
This interface is a statement of a Hop's Performance, e.g. delay level.

Figure 662: Reachability Service Policy

Reachability

This service consumes current and predicted network evidence to determine the current and potential reach of a network, and identifies any missing information required to determine its own Capability and Performance. This includes satisfaction of network prerequisites, e.g. the establishment of IP network over the crypto network Topology is implemented.

Interfaces

Hop_Capability

This interface is a statement of a Hop Capability, e.g. usable throughput and achievable Connections.

Attributes

bandwidth	The actual amount of traffic that can be supported on a Hop.
utilisation	The amount of the network being utilised by a Hop.
reach	Where a Node thinks it is able to go.
assurance_level	The level of assurance of a Hop, e.g. whether the network is approved for safety critical or control traffic.
provenance	Where the information was learnt from, e.g. what routing protocols.
cost	The determined value of cost for the network used by the Hop, often derived from bandwidth.

Hop_Performance

This interface is a statement of a Hop's Performance, e.g. delay level.

Attributes

latency_level	The latency of a Hop.

reliability	The reliability of a Hop, e.g. the certainty that the connection of Hop is not going to terminate unexpectedly.
loss_level	The level of data loss of a Hop, e.g. loss or drop rate.

Activities

assess_reachability_evidence

Assess the reachability evidence to decide whether any further action needs to be taken.

identify_missing_reachability_evidence

Identify any extra reachability evidence required to determine the Capability of networks to the required level of specificity and certainty.

5.4.2.38.7.2 Service Dependencies

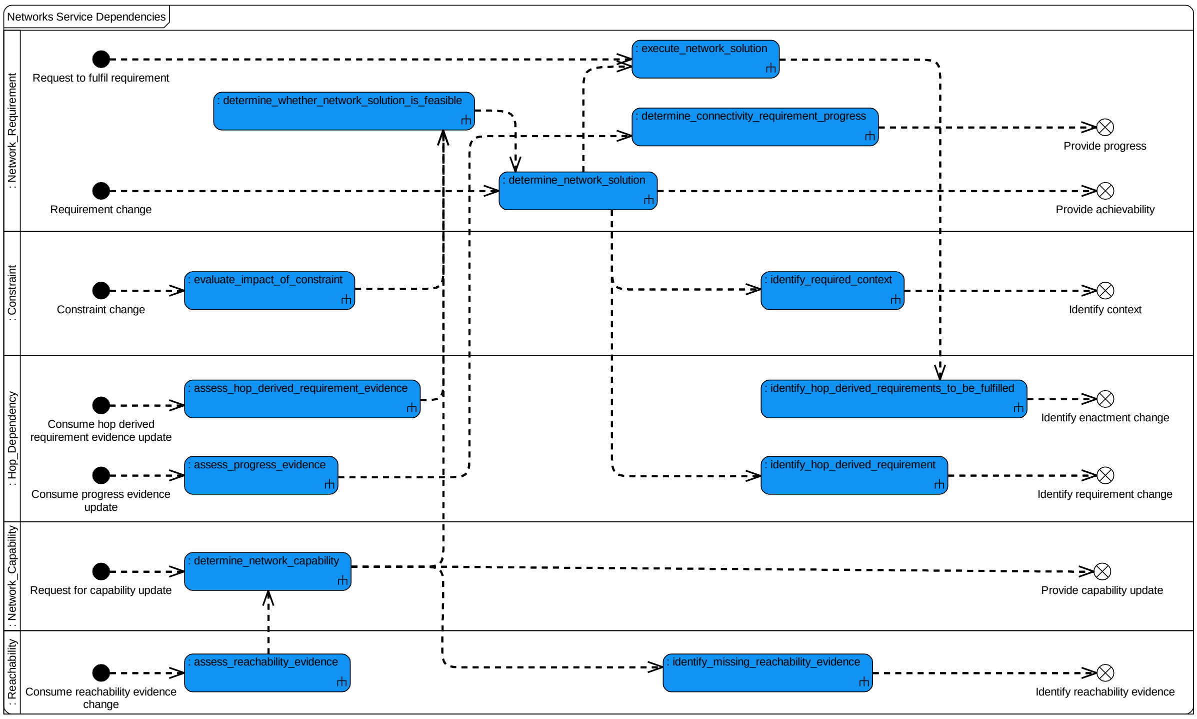

Networks Service Dependencies
: Network_Requirement
: Constraint
: Hop_Dependency
: Network_Capability
: Reachability
Request to fulfil requirement
: execute_network_solution
: determine_whether_network_solution_is_feasible
: determine_connectivity_requirement_progress
Provide progress
Requirement change
: determine_network_solution
Provide achievability
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume hop derived requirement evidence update
: assess_hop_derived_requirement_evidence
: identify_hop_derived_requirements_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_progress_evidence
: identify_hop_derived_requirement
Identify requirement change
Request for capability update
: determine_network_capability
Provide capability update
Consume reachability evidence change
: assess_reachability_evidence
: identify_missing_reachability_evidence
Identify reachability evidence

Figure 663: Networks Service Dependencies

5.4.2.39 Objectives

5.4.2.39.1 Role

The role of Objectives is to coordinate the achievement of objectives through the execution of tasks.

5.4.2.39.2 Overview

Control Architecture

Objectives is the only component in the Objective Layer, as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When an Objective is being planned, the Objective specification is provided to Objectives. Objectives will determine one or more Objective_Solutions which are a coordinated set of Tasks, taking into account any Constraints. Objectives will oversee the execution of an Objective_Solution and, if this Objective_Solution becomes unachievable, Objectives will re-assess.

Examples of Use

Objectives is required whenever the coordination of Tasks by the system to achieve an Objective is necessary. For example:

- Where the system is required to generate and fulfil Tasks from a provided Objective to perform a reconnaissance mission of a specified zone. A mission can be comprised of multiple Objectives handled by the Objectives component.

5.4.2.39.3 Service Summary

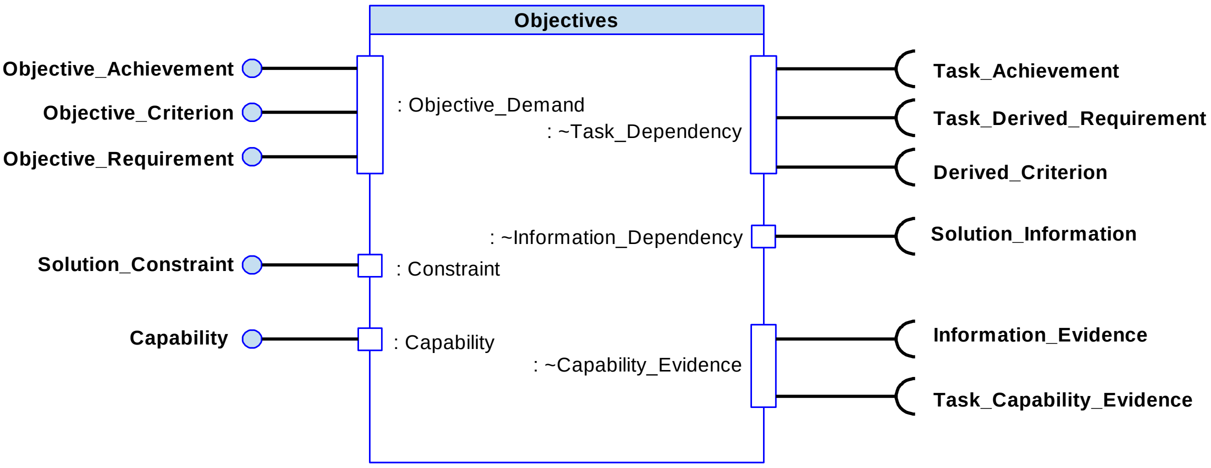


Figure 664: Objectives Service Summary

5.4.2.39.4 Responsibilities

capture_requirements

- To capture Objectives (e.g. objective type, timing and risk profile).

capture_measurement_criteria

- To capture given quality requirements for Objective_Solutions (e.g. quality, risk and robustness).

capture_constraints

- To capture Constraints on how an Objective may be achieved (e.g. operator imposed restrictions or Rules of engagement).

identify_whether_requirement_remains_achievable

- To identify whether an Objective is still achievable given current Flight_Capability and Constraints.

determine_implementation_scheme

- To determine an Objective_Solution to achieve an Objective.

determine_predicted_quality_of_solution

- To evaluate the predicted quality of a proposed Objective_Solution against given quality requirements.

identify_dependencies

- To identify the interdependencies between Tasks required to support the delivery of an Objective_Solution.

satisfy_dependencies_between_tasks

- To satisfy the interdependencies between Tasks through the management of Tasks.

coordinate_objective_enactment

- To coordinate the enactment of an Objective_Solution via the fulfilment of a set of coordinated Tasks.

identify_progress_of_objective

- To identify the progress of an Objective's Objective_Solution.

determine_actual_quality_of_solution

- To evaluate the quality of the delivered Objective_Solution against given quality requirements.

evaluate_implementation_scheme_cost

- To evaluate the costs of a planned Objective_Solution against given measurement criteria.

determine_capability

- To determine Flight_Capability based on Participant_Capability, taking into account observed anomalies.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the capability assessment.

predict_capability_progression

- To predict the progression of available Flight_Capability over time and with use.

5.4.2.39.5 Subject Matter Semantics

The subject matter of Objectives is the coordination of Tasks to fulfil Objectives.

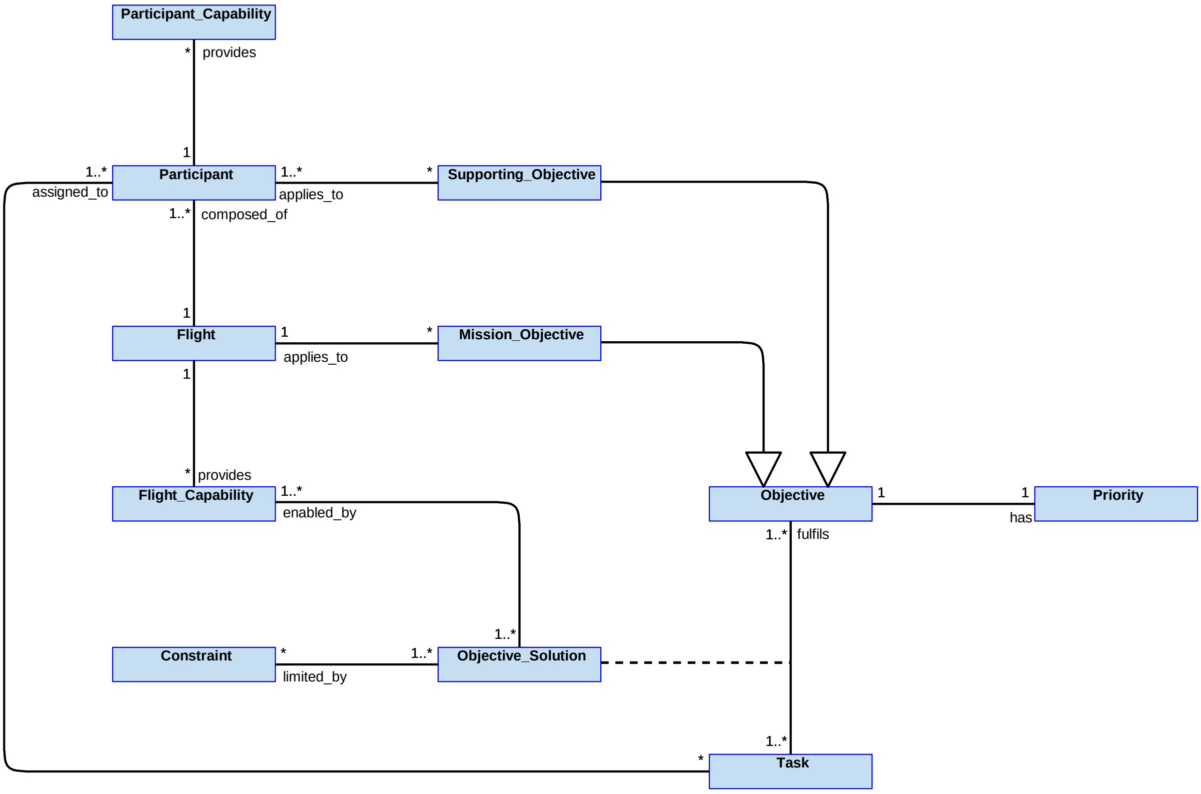


Figure 665: Objectives Semantics

5.4.2.39.5.1 Entities

Constraint

A restriction that cannot be contravened and which may affect the ways in which an Objective can be carried out (e.g. operator imposed restrictions or rules of engagement).

Flight

A collection of one or more Participants that are cooperating to achieve Objectives.

Flight_Capability

The capability of the Flight to carry out Objectives.

This will be based on the collective capability of all the available Participants and their ability to interact (including the capability of the coordinator).

Mission_Objective

This is an Objective that defines a purpose of a mission. This objective contributes to specific strategic goals such as deployment of equipment, maintaining control of an air space, or carrying out surveillance over an area.

Examples include: suppression of enemy air defences (SEAD), intercept and attack enemy air vehicle, provision of close air support, personnel recovery from a hostile location, supply drop of equipment, and relocation to a specified airbase.

Objective

The definition of an immutable goal that contributes to a broader strategic goal. This is expressed in the terms of what needs to be achieved without specifying how it should be achieved. Objectives can either be a Mission_Objective, or a Supporting_Objective.

Objective_Solution

The breakdown of Objectives into allocated Tasks.

Participant

An air vehicle that can contribute to the achievement of an Objective.

Participant_Capability

The capability of a Participant to perform a specific role (e.g. an aircraft fitted with a sensor pod may have an increased capability to carry out search objectives, or an aircraft that has expended all of its weapons won't have any capability to perform an attack).

Priority

A measurement of the relative importance of an Objective in comparison to the other Objectives.

This allows decisions of which Objectives are allowed to proceed where a conflict arises.

Supporting_Objective

This is an Objective that does not directly contribute to a mission goal, but will operate in parallel to a mission.

This objective contributes to broader strategic goals such as the psychological impact of operations or the continued availability of equipment.

Examples include: survivability of the aircraft or visibility of the aircraft (e.g. overt presence or non-detection).

Task

The specification of a goal which needs to be achieved by a Participant (e.g. transit to a location, search an area or attack a target).

Unlike Objectives, these may be updated during the course of a mission. For example, if an equipment failure prevents an aircraft from carrying out a particular task, then the Objective_Solution can be updated to use a different Participant.

5.4.2.39.6 Design Rationale

5.4.2.39.6.1 Assumptions

- The available Flight_Capability will vary between missions and during missions.
- Types of Participant_Capability will vary less often: new types of Participant_Capability may become available within the lifetime of a deployment, but not within a mission.

5.4.2.39.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Objectives:

- Constraint Management - Rules of engagement are an example of Constraints that are considered by Objectives.
- Data Driving - Priority and Constraints could be data-driven using build time data.
- Multi-Vehicle Coordination - An Objectives component may break down Mission_Objectives into a series of Tasks to be fulfilled using more than one flight member.

Extensions

- The breakdown of a particular type of Mission_Objective into Tasks may be guided by the use of configured extension components.

Exploitation Considerations

- Normally a deployment will have a single instance of Objectives. However, in a multi-vehicle deployment, there may be an instance of Objectives on each vehicle to ensure redundancy for flight lead handover.
- Mission_Objectives should contain the key parts of what needs to be achieved without specific details of how they are met. In the example of relocating to an airbase, the reason for the mission is so that the aircraft is available for a future mission. So while the airbase must be specified, which of the runways or taxiways to use shouldn't be specified as part of the objective, as the intent of the mission objective can be met no matter which is used.
- In a multiple-vehicle deployment the Objective_Solution will be managed by a coordinator (e.g. the flight lead). The roles of Participants and the sequence of Tasks will be established and then enacted.

5.4.2.39.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component would mean that the Mission_Objectives were not fulfilled, which is not a safety concern. It is expected that other components in the Control Architecture would ensure that any actions were performed safely (e.g. Interlocks) and perform mitigating actions to accommodate failures or a change in circumstances (e.g. Tasks). However, failure of the component would cause an increase in workload for crew which is considered a "Significant Reduction in Safety Margins" (severity major). Therefore, the indicative IDAL is DAL C.

5.4.2.39.6.4 Security Considerations

The indicative security classification is SNEO.

This component coordinates the Tasks required to achieve Objectives; the details of Exploiting Platform capability (and potentially those it coordinates with), available strategies, targets and control orders are deemed SNEO. Due to the central role in conducting a mission, enhanced measures to ensure ongoing confidentiality, integrity, availability and authenticity are considered appropriate.

The component may be expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** of authorisation success/failures, access and changes to high-value data, etc. for later forensic examination.
- **Maintaining Audit Records** to support non-repudiation of command approval and other events performed in the fulfilment of a mission objective.
- **System Status and Monitoring** through the monitoring of the objectives set and progress against them. Unexpected deviation from a task that contributes to satisfying the mission objectives (e.g. an unexplainable deviation in route) may indicate a cyber adversary has infiltrated the control architecture.

The component may be expected to at least partially satisfy security enforcing functions relating to:

- **Verifying Integrity of Data** through assuring Mission_Objectives have not been tampered with prior to their execution.

5.4.2.39.7 Services

5.4.2.39.7.1 Service Definitions

5.4.2.39.7.1.1 Objective_Demand

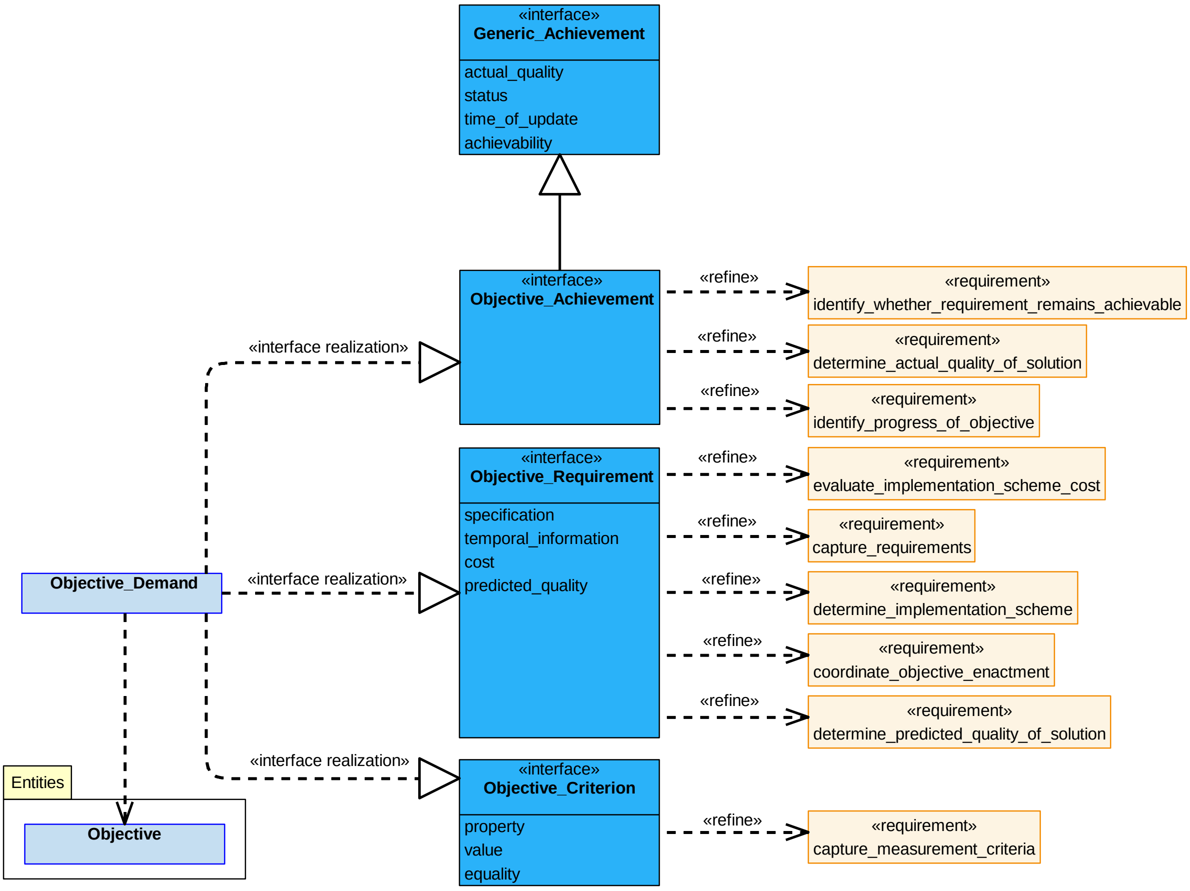


Figure 666: Objective Demand Service Definition

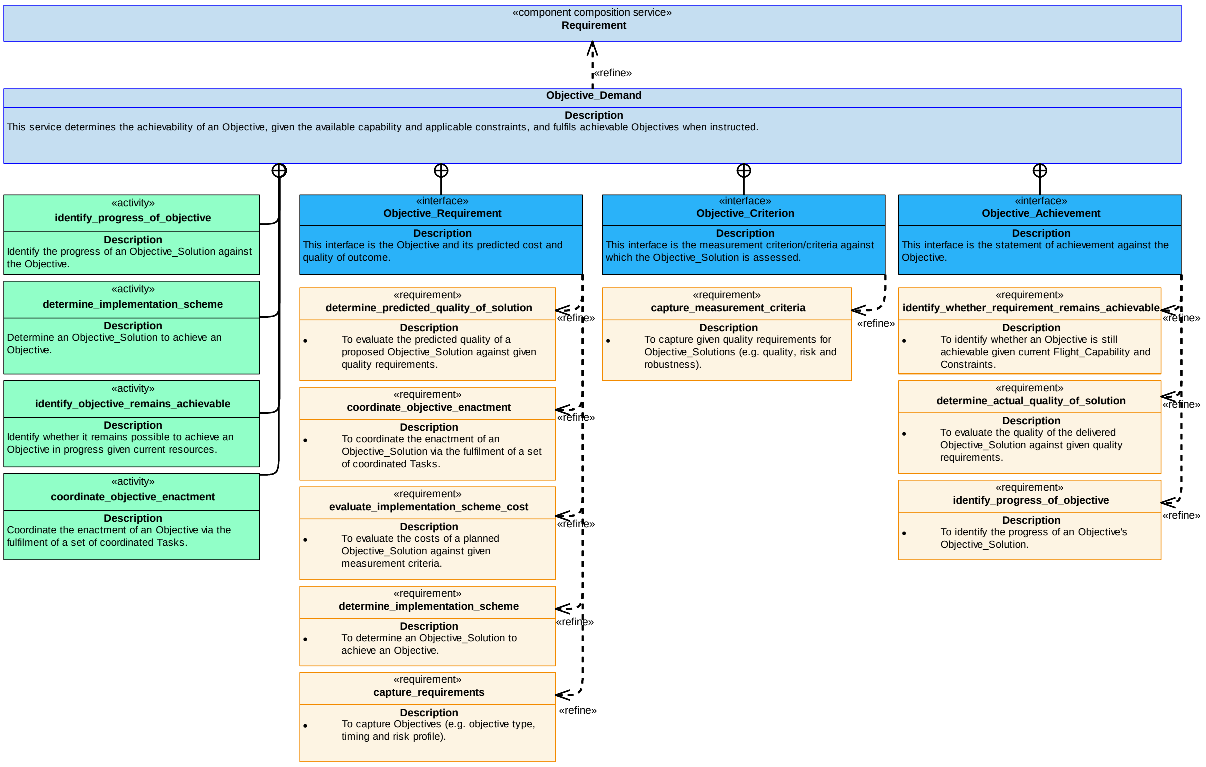

«component composition service»
Requirement
«refine»
Objective_Demand
Description
This service determines the achievability of an Objective, given the available capability and applicable constraints, and fulfils achievable Objectives when instructed.
«activity»
identify_progress_of_objective
Description
Identify the progress of an Objective_Solution against the Objective.
«activity»
determine_implementation_scheme
Description
Determine an Objective_Solution to achieve an Objective.
«activity»
identify_objective_remains_achievable
Description
Identify whether it remains possible to achieve an Objective in progress given current resources.
«activity»
coordinate_objective_enactment
Description
Coordinate the enactment of an Objective via the fulfilment of a set of coordinated Tasks.
«interface»
Objective_Requirement
Description
This interface is the Objective and its predicted cost and quality of outcome.
«requirement»
determine_predicted_quality_of_solution
Description
To evaluate the predicted quality of a proposed Objective_Solution against given quality requirements.
«refine»
«requirement»
coordinate_objective_enactment
Description
To coordinate the enactment of an Objective_Solution via the fulfilment of a set of coordinated Tasks.
«refine»
«requirement»
evaluate_implementation_scheme_cost
Description
To evaluate the costs of a planned Objective_Solution against given measurement criteria.
«refine»
«requirement»
determine_implementation_scheme
Description
To determine an Objective_Solution to achieve an Objective.
«refine»
«requirement»
capture_requirements
Description
To capture Objectives (e.g. objective type, timing and risk profile).
«refine»
«interface»
Objective_Criterion
Description
This interface is the measurement criterion/criteria against which the Objective_Solution is assessed.
«requirement»
capture_measurement_criteria
Description
To capture given quality requirements for Objective_Solutions (e.g. quality, risk and robustness).
«refine»
«interface»
Objective_Achievement
Description
This interface is the statement of achievement against the Objective.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether an Objective is still achievable given current Flight_Capability and Constraints.
«refine»
«requirement»
determine_actual_quality_of_solution
Description
To evaluate the quality of the delivered Objective_Solution against given quality requirements.
«refine»
«requirement»
identify_progress_of_objective
Description
To identify the progress of an Objective's Objective_Solution.
«refine»

Figure 667: Objective Demand Service Policy

Objective_Demand

This service determines the achievability of an Objective, given the available capability and applicable constraints, and fulfils achievable Objectives when instructed.

Interfaces

Objective_Achievement

This interface is the statement of achievement against the Objective.

Objective_Criterion

This interface is the measurement criterion/criteria against which the Objective_Solution is assessed.

Attributes

property	The property to be measured, e.g. number of square miles in which enemy air defences must be suppressed.
value	The measured value of the property, e.g. 50 square miles.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Objective_Requirement

This interface is the Objective and its predicted cost and quality of outcome.

Attributes

specification	The specification of the Objective, for example to suppress air defences across a region of enemy territory.
temporal_information	Timing related requirements for this Objective, e.g. start and end times/duration, or complete by time.
cost	The cost of executing the solution, e.g. resources used or time taken.
predicted_quality	How well the proposed Objective_Solution is predicted to satisfy the requirement.

Activities

identify_progress_of_objective

Identify the progress of an Objective_Solution against the Objective.

determine_implementation_scheme

Determine an Objective_Solution to achieve an Objective.

coordinate_objective_enactment

Coordinate the enactment of an Objective via the fulfilment of a set of coordinated Tasks.

identify_objective_remains_achievable

Identify whether it remains possible to achieve an Objective in progress given current resources.

5.4.2.39.7.1.2 Task_Dependency

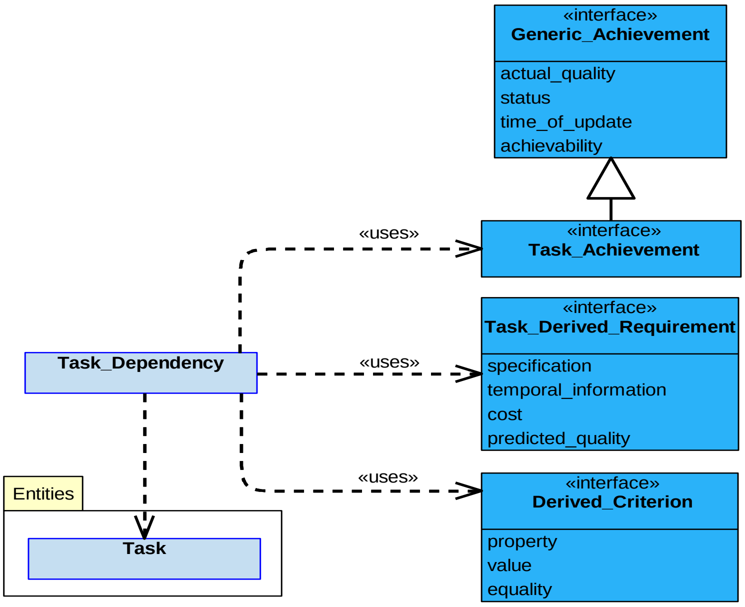


Figure 668: Task Dependency Service Definition

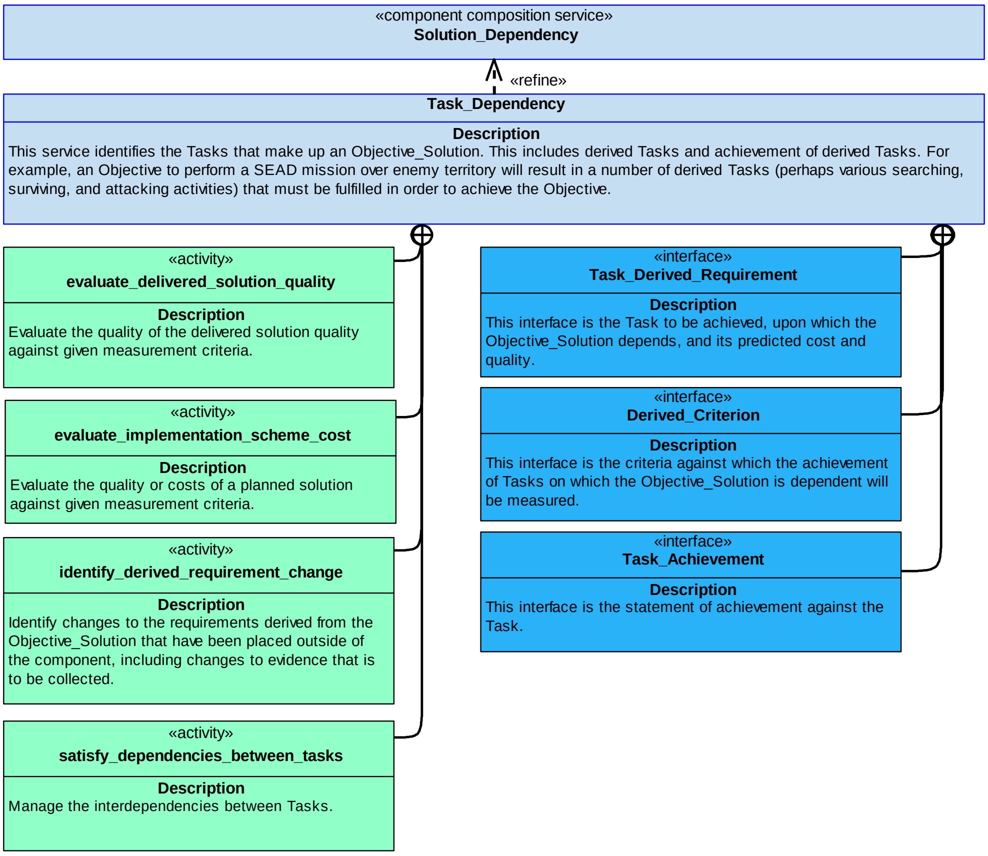

«component composition service»
Solution_Dependency
«refine»
Task_Dependency
Description
This service identifies the Tasks that make up an Objective_Solution. This includes derived Tasks and achievement of derived Tasks. For example, an Objective to perform a SEAD mission over enemy territory will result in a number of derived Tasks (perhaps various searching, surviving, and attacking activities) that must be fulfilled in order to achieve the Objective.
«activity»
evaluate_delivered_solution_quality
Description
Evaluate the quality of the delivered solution quality against given measurement criteria.
«activity»
evaluate_implementation_scheme_cost
Description
Evaluate the quality or costs of a planned solution against given measurement criteria.
«activity»
identify_derived_requirement_change
Description
Identify changes to the requirements derived from the Objective_Solution that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
satisfy_dependencies_between_tasks
Description
Manage the interdependencies between Tasks.
«interface»
Task_Derived_Requirement
Description
This interface is the Task to be achieved, upon which the Objective_Solution depends, and its predicted cost and quality.
«interface»
Derived_Criterion
Description
This interface is the criteria against which the achievement of Tasks on which the Objective_Solution is dependent will be measured.
«interface»
Task_Achievement
Description
This interface is the statement of achievement against the Task.

Figure 669: Task Dependency Service Policy

Task_Dependency

This service identifies the Tasks that make up an Objective_Solution. This includes derived Tasks and achievement of derived Tasks. For example, an Objective to perform a SEAD mission over enemy territory will result in a number of derived Tasks (perhaps various searching, surviving, and attacking activities) that must be fulfilled in order to achieve the Objective.

Interfaces

Task_Derived_Requirement

This interface is the Task to be achieved, upon which the Objective_Solution depends, and its predicted cost and quality.

Attributes

specification	The aim of this Task.
temporal_information	Timing related requirements for this Task, for example start and end times/duration, or complete by time.
cost	The cost of executing the solution, e.g. resources used, time taken.
predicted_quality	How well the planned task solution is predicted to satisfy the requirement.

Derived_Criterion

This interface is the criteria against which the achievement of Tasks on which the Objective_Solution is dependent will be measured.

Attributes

property	The property to be measured, e.g. a quality category required for surveillance imaging.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Task_Achievement

This interface is the statement of achievement against the Task.

Activities

evaluate_delivered_solution_quality

Evaluate the quality of the delivered solution quality against given measurement criteria.

identify_derived_requirement_change

Identify changes to the requirements derived from the Objective_Solution that have been placed outside of the component, including changes to evidence that is to be collected.

satisfy_dependencies_between_tasks

Manage the interdependencies between Tasks.

evaluate_implementation_scheme_cost

Evaluate the quality or costs of a planned solution against given measurement criteria.

5.4.2.39.7.1.3 Information_Dependency

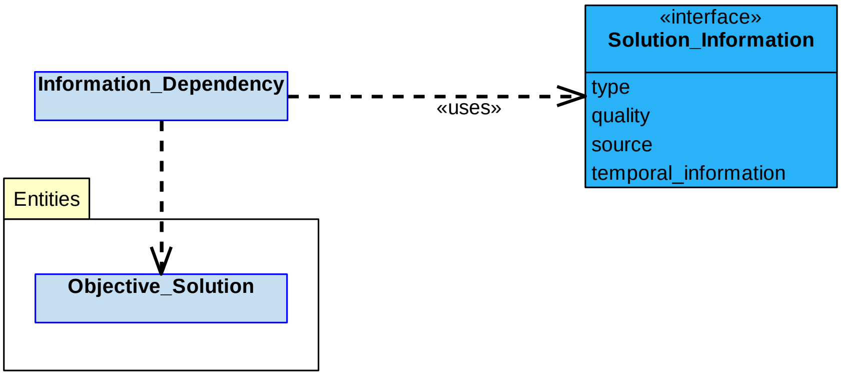


Figure 670: Information Dependency Service Definition

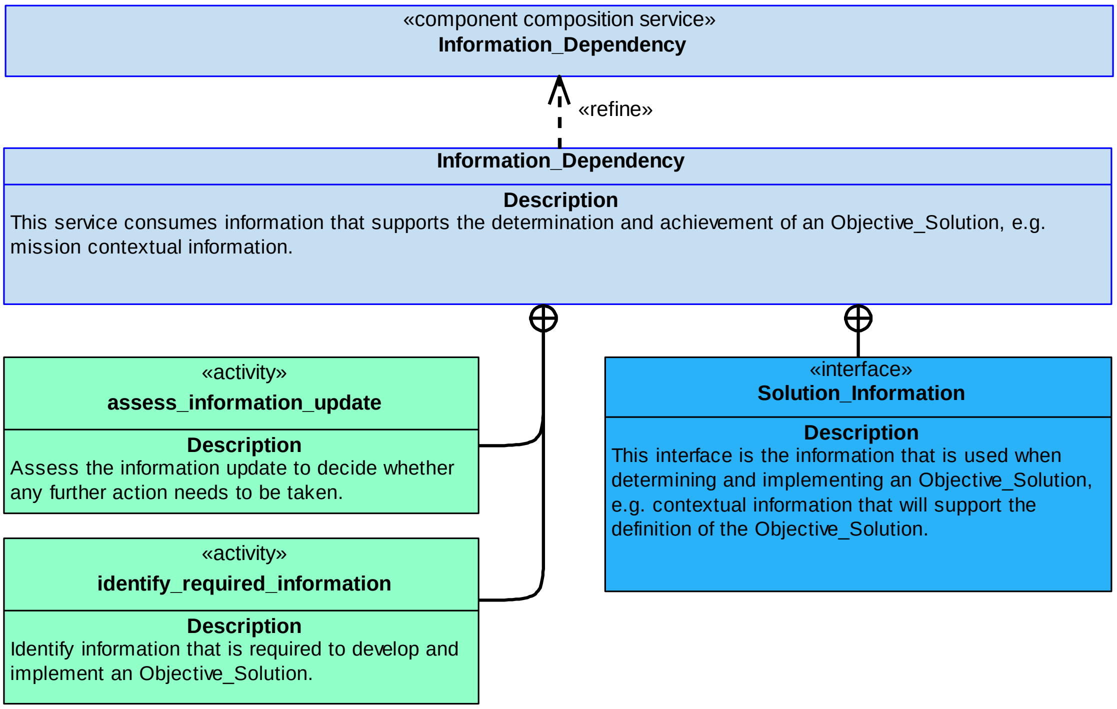

«component composition service»
Information_Dependency
«refine»
Information_Dependency
Description
This service consumes information that supports the determination and achievement of an Objective_Solution, e.g. mission contextual information.
«activity»
assess_information_update
Description
Assess the information update to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify information that is required to develop and implement an Objective_Solution.
«interface»
Solution_Information
Description
This interface is the information that is used when determining and implementing an Objective_Solution, e.g. contextual information that will support the definition of the Objective_Solution.

Figure 671: Information Dependency Service Policy

Information_Dependency

This service consumes information that supports the determination and achievement of an Objective_Solution, e.g. mission contextual information.

Interface

Solution_Information

This interface is the information that is used when determining and implementing an Objective_Solution, e.g. contextual information that will support the definition of the Objective_Solution.

Attributes

type	The type of information, e.g. aircraft availability.
quality	The quality of the information received, e.g. accuracy of reported information.
source	The source of the information, e.g. information has been received from a trusted source.
temporal_information	Information covering timing, such as start and end timing, e.g. for the next half an hour.

Activities

assess_information_update

Assess the information update to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to develop and implement an Objective_Solution.

5.4.2.39.7.1.4 Constraint

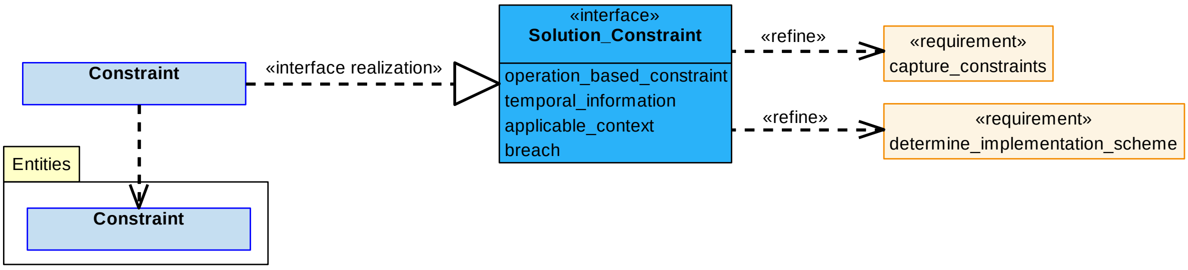


Figure 672: Constraint Service Definition

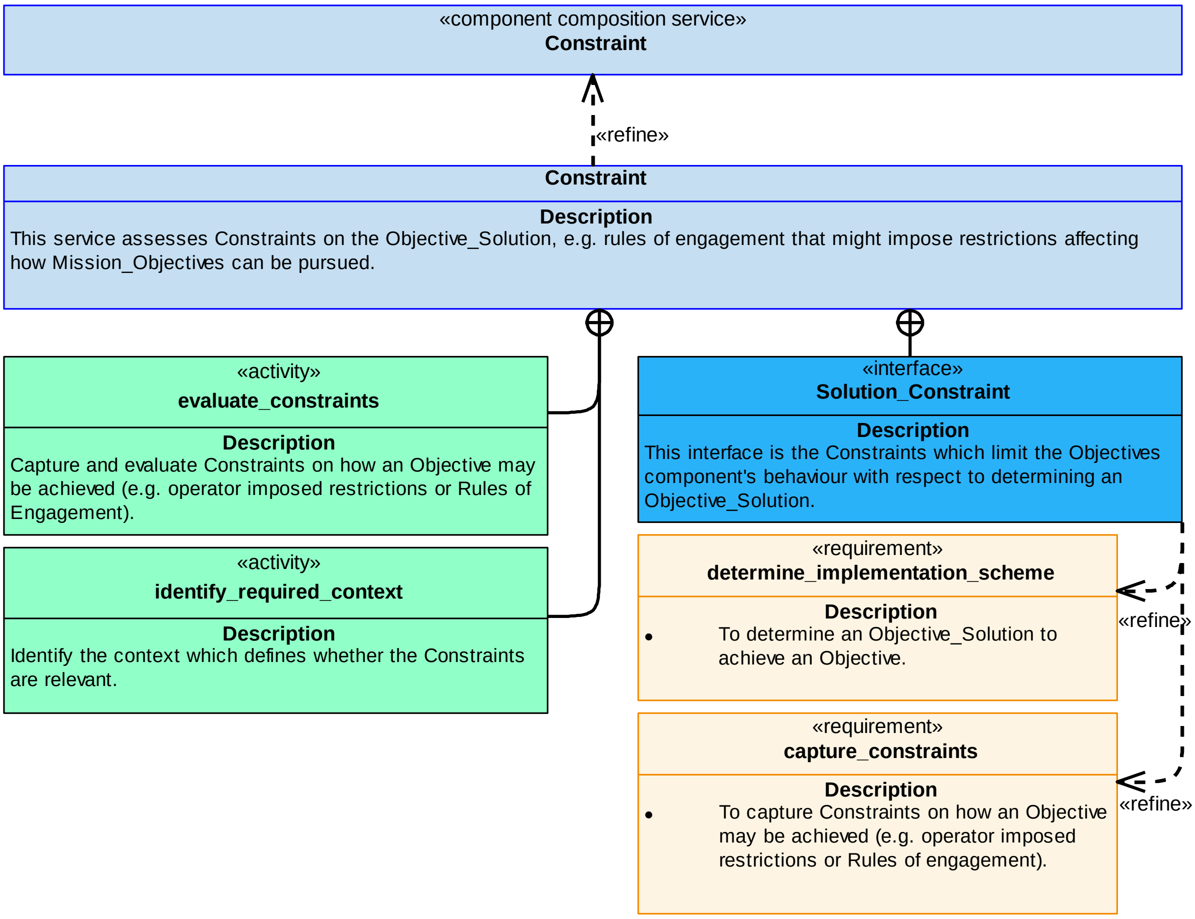

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints on the Objective_Solution, e.g. rules of engagement that might impose restrictions affecting how Mission_Objectives can be pursued.
«activity»
evaluate_constraints
Description
Capture and evaluate Constraints on how an Objective may be achieved (e.g. operator imposed restrictions or Rules of Engagement).
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Solution_Constraint
Description
This interface is the Constraints which limit the Objectives component's behaviour with respect to determining an Objective_Solution.
«requirement»
determine_implementation_scheme
Description
• To determine an Objective_Solution to achieve an Objective.
«refine»
«requirement»
capture_constraints
Description
• To capture Constraints on how an Objective may be achieved (e.g. operator imposed restrictions or Rules of engagement).
«refine»

Figure 673: Constraint Service Policy

Constraint

This service assesses Constraints on the Objective_Solution, e.g. rules of engagement that might impose restrictions affecting how Mission_Objectives can be pursued.

Interface

Solution_Constraint

This interface is the Constraints which limit the Objectives component's behaviour with respect to determining an Objective_Solution.

Attributes

operation_based_constraint	A Constraint that limits the behaviour of the Objectives component, e.g. Rules of Engagement currently in force may dictate that certain Objective_Solutions cannot be carried out in achieving a Mission_Objective.
temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the Constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_constraints

Capture and evaluate Constraints on how an Objective may be achieved (e.g. operator imposed restrictions or Rules of Engagement).

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.39.7.1.5 Capability

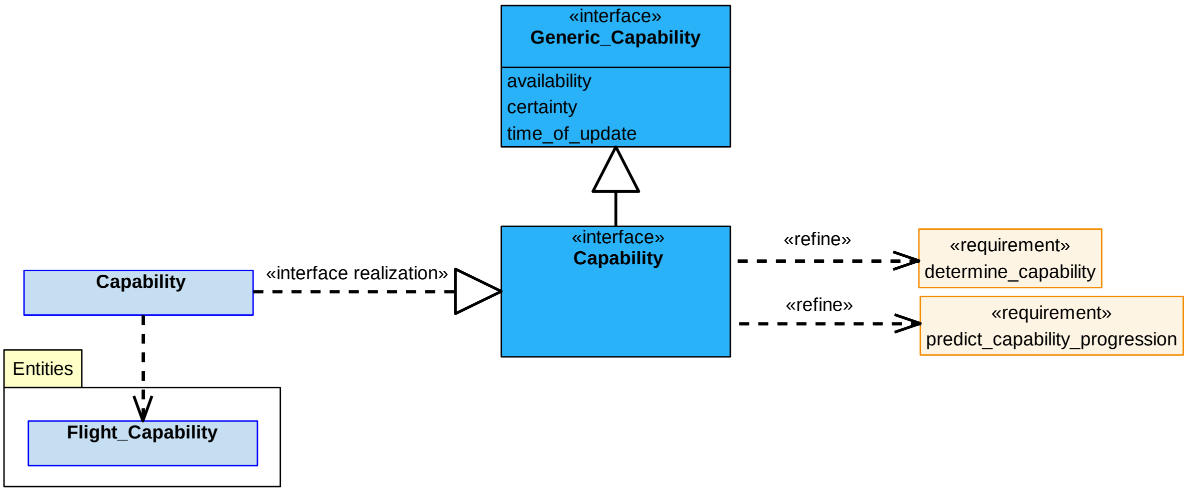


Figure 674: Capability Service Definition

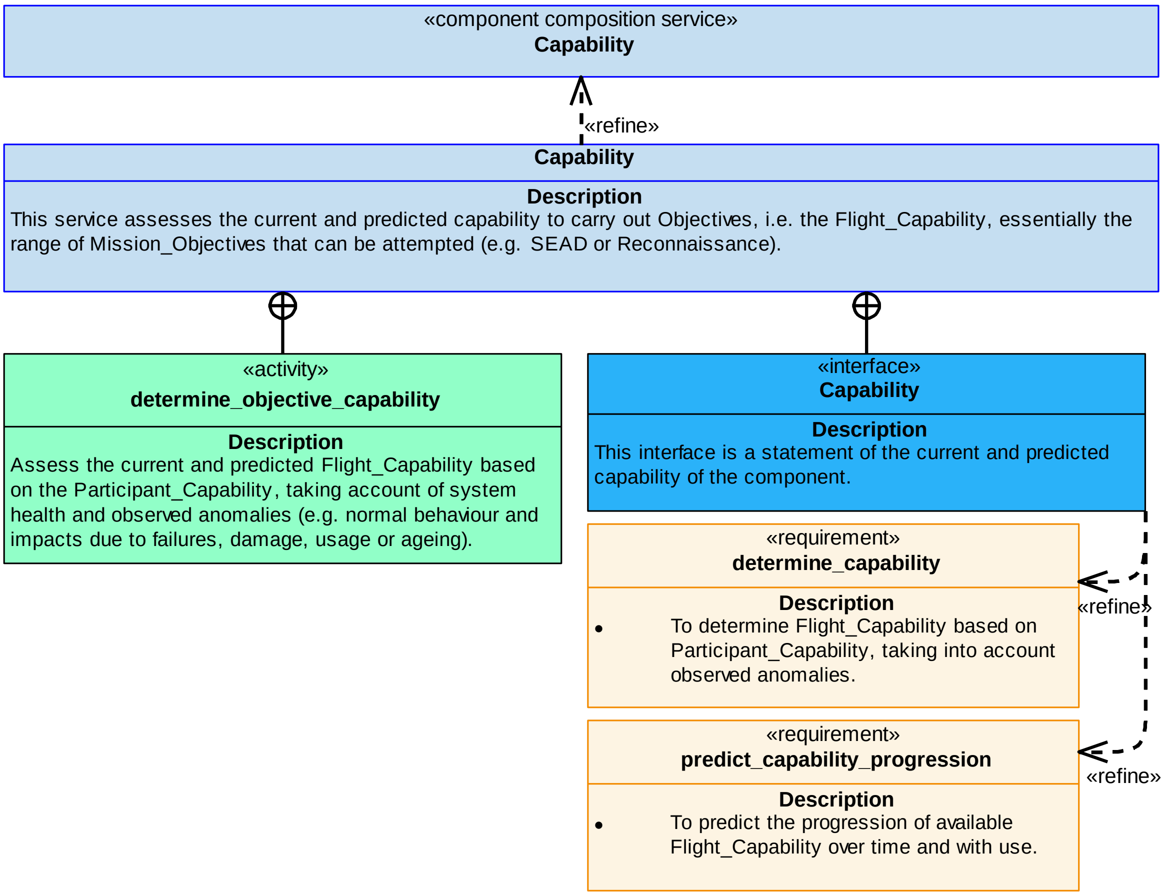

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability to carry out Objectives, i.e. the Flight_Capability, essentially the range of Mission_Objectives that can be attempted (e.g. SEAD or Reconnaissance).
«activity»
determine_objective_capability
Description
Assess the current and predicted Flight_Capability based on the Participant_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Capability
Description
This interface is a statement of the current and predicted capability of the component.
«requirement»
determine_capability
Description
To determine Flight_Capability based on Participant_Capability, taking into account observed anomalies.
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of available Flight_Capability over time and with use.
«refine»

Figure 675: Capability Service Policy

Capability

This service assesses the current and predicted capability to carry out Objectives, i.e. the Flight_Capability, essentially the range of Mission_Objectives that can be attempted (e.g. SEAD or Reconnaissance).

Interface

Capability

This interface is a statement of the current and predicted capability of the component.

Activity

determine_objective_capability

Assess the current and predicted Flight_Capability based on the Participant_Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.39.7.1.6 Capability_Evidence

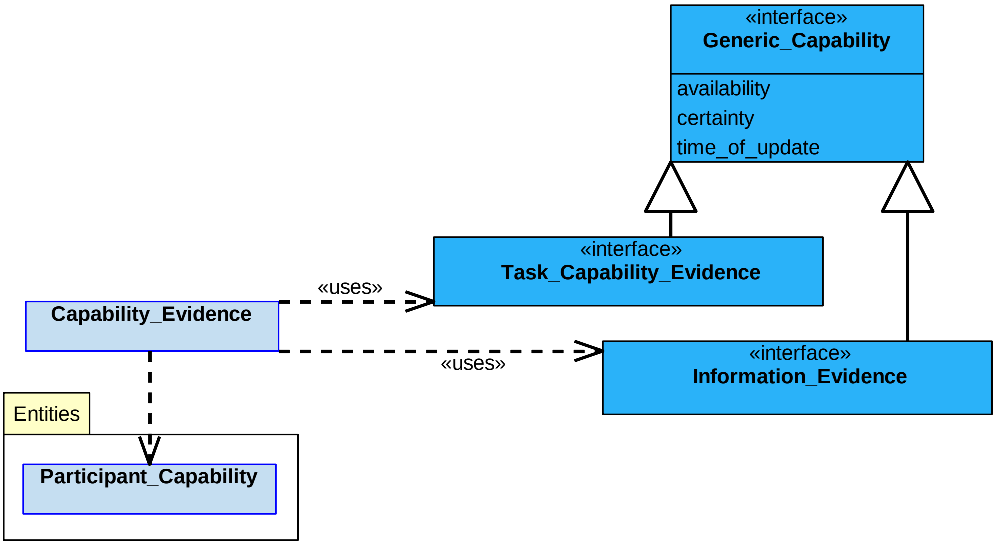


Figure 676: Capability Evidence Service Definition

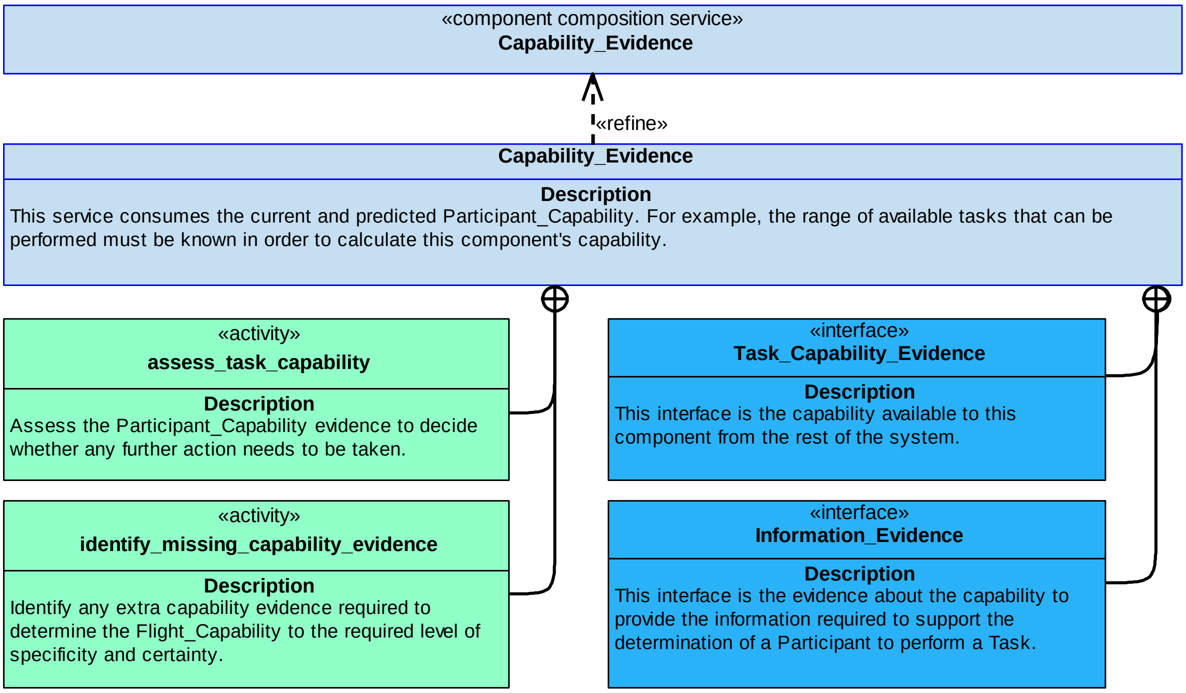

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted Participant_Capability. For example, the range of available tasks that can be performed must be known in order to calculate this component's capability.
«activity»
assess_task_capability
Description
Assess the Participant_Capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Flight_Capability to the required level of specificity and certainty.
«interface»
Task_Capability_Evidence
Description
This interface is the capability available to this component from the rest of the system.
«interface»
Information_Evidence
Description
This interface is the evidence about the capability to provide the information required to support the determination of a Participant to perform a Task.

Figure 677: Capability Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted Participant_Capability. For example, the range of available tasks that can be performed must be known in order to calculate this component's capability.

<u>**Interfaces**</u>

Task_Capability_Evidence

This interface is the capability available to this component from the rest of the system.

Information_Evidence

This interface is the evidence about the capability to provide the information required to support the determination of a Participant to perform a Task.

Activities

assess_task_capability

Assess the Participant_Capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Flight_Capability to the required level of specificity and certainty.

5.4.2.39.7.2 Service Dependencies

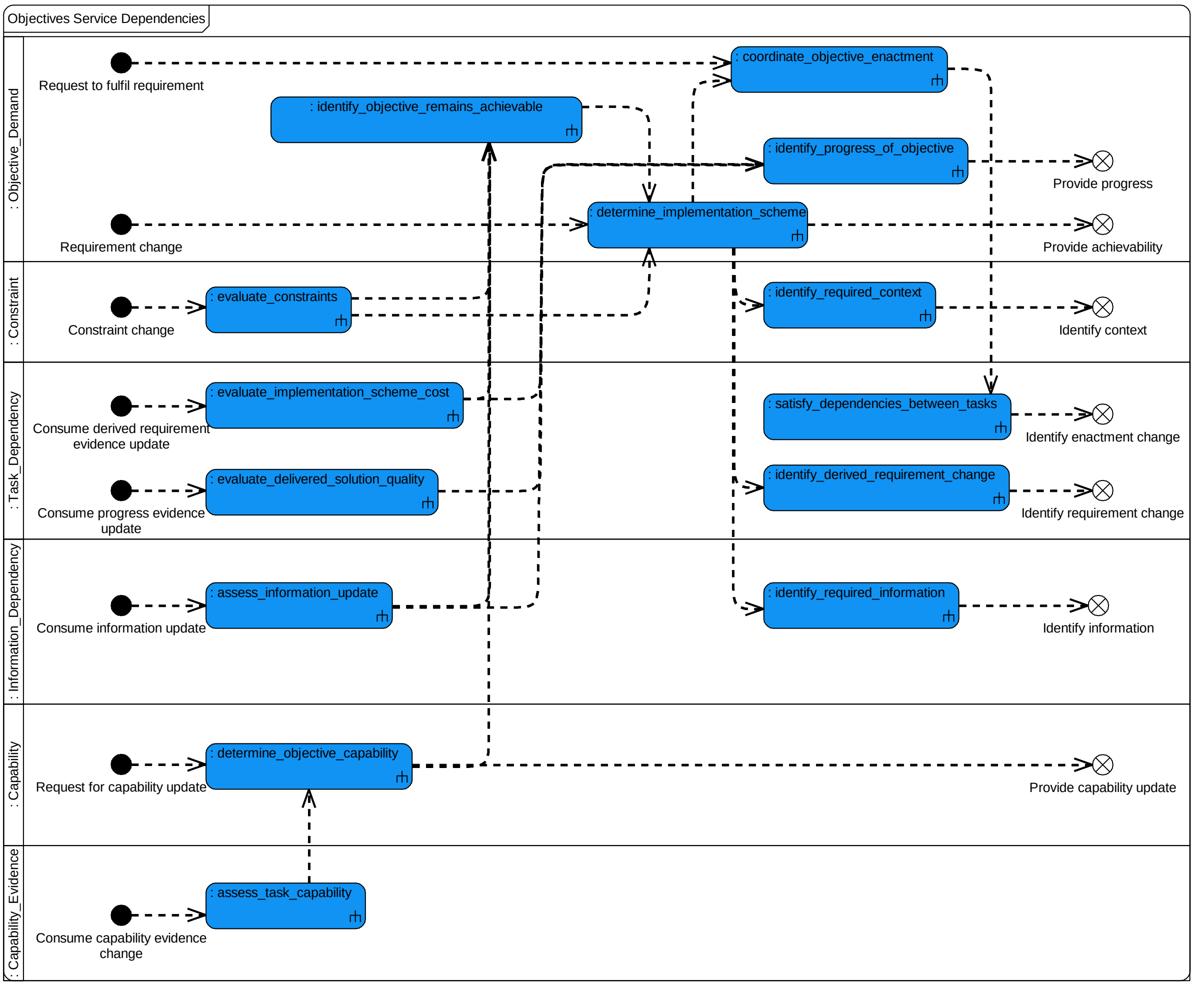

Objectives Service Dependencies
: Objective_Demand
: Constraint
: Task_Dependency
: Information_Dependency
: Capability
: Capability_Evidence
Request to fulfil requirement
: coordinate_objective_enactment
: identify_objective_remains_achievable
: identify_progress_of_objective
Provide progress
Requirement change
: determine_implementation_scheme
Provide achievability
Constraint change
: evaluate_constraints
: identify_required_context
Identify context
Consume derived requirement evidence update
: evaluate_implementation_scheme_cost
: satisfy_dependencies_between_tasks
Identify enactment change
Consume progress evidence update
: evaluate_delivered_solution_quality
: identify_derived_requirement_change
Identify requirement change
Consume information update
: assess_information_update
: identify_required_information
Identify information
Request for capability update
: determine_objective_capability
Provide capability update
Consume capability evidence change
: assess_task_capability

Figure 678: Objectives Service Dependencies

5.4.2.40 Observability

5.4.2.40.1 Role

The role of Observability is to evaluate a subject's observability by an observer.

5.4.2.40.2 Overview

Control Architecture

Observability is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Observability takes information regarding the Observer and its Observation_Means in order to determine the likelihood of observing a given Subject, taking into account any known occlusions, range, etc. Observability can be determined with ownship as the Observer or the Subject, or for two third parties.

Examples of Use

Observability will be used when information is needed about:

- The probability of own vehicle being detected by a third party Observer, e.g. if flying through a region with known SAM coverage.
- The probability of detecting Subjects in an area using given sensors, e.g. for observing enemy land forces using IR sensors or radar.
- The line of sight between two parties flying specified routes.

5.4.2.40.3 Service Summary

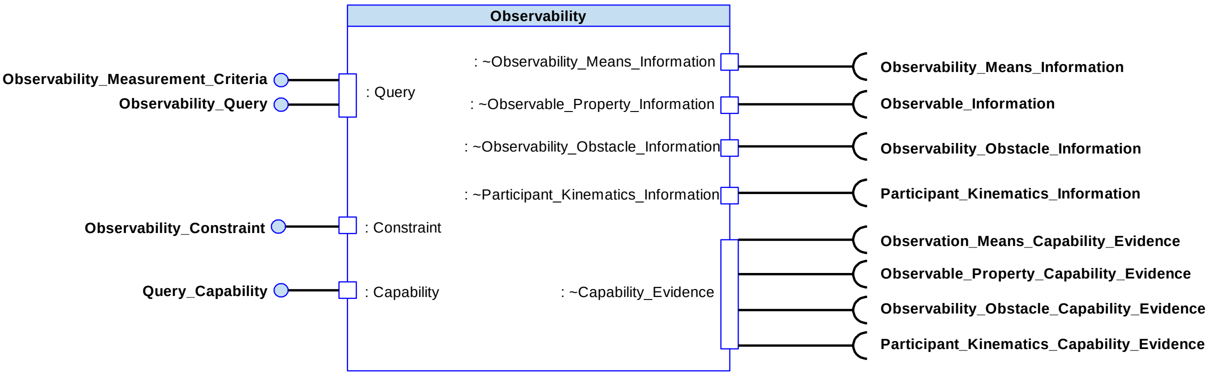

Observability
Observability_Measurement_Criteria
Observability_Query
: Query
Observability_Constraint
: Constraint
Query_Capability
: Capability
: ~Observability_Means_Information
: ~Observable_Property_Information
: ~Observability_Obstacle_Information
: ~Participant_Kinematics_Information
: ~Capability_Evidence
Observability_Means_Information
Observable_Information
Observability_Obstacle_Information
Participant_Kinematics_Information
Observation_Means_Capability_Evidence
Observable_Property_Capability_Evidence
Observability_Obstacle_Capability_Evidence
Participant_Kinematics_Capability_Evidence

Figure 679: Observability Service Summary

5.4.2.40.4 Responsibilities

capture_observability_requirements

- To capture requirements for determining the Observability of a Subject.

capture_observability_measurement_criteria

- To capture Measurement_Criterion/criteria for Observability.

capture_observability_constraints

- To capture Constraints on Observability (e.g. Observation_Means not to be used, effect of range and occlusion on Observation_Means).

determine_observability

- To determine whether an Observer can observe a Subject taking into account Observation_Means, Observability_Obstacles and any Constraints.

determine_observability_threshold

- To determine the Observability_Threshold, for a given Observable_Property and with any Observability_Obstacles, at which a Subject will become observable to a given Observer.

determine_line_of_sight

- To determine Observation_Path between an Observer and a Subject.

determine_quality_of_observability_service

- To determine the quality of Observability against given Measurement_Criterion/criteria.

assess_observability_service_capability

- To assess the Capability to determine Observability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_observability_capability_progression

- To predict the progression of the Observability Capability over time and with use.

5.4.2.40.5 Subject Matter Semantics

The subject matter of Observability is the observability of a Subject in the environment relative to an Observer.

Exclusions

The subject matter of Observability does not include:

- A Subject's signature in isolation, but rather when it is observable relative to an Observer.

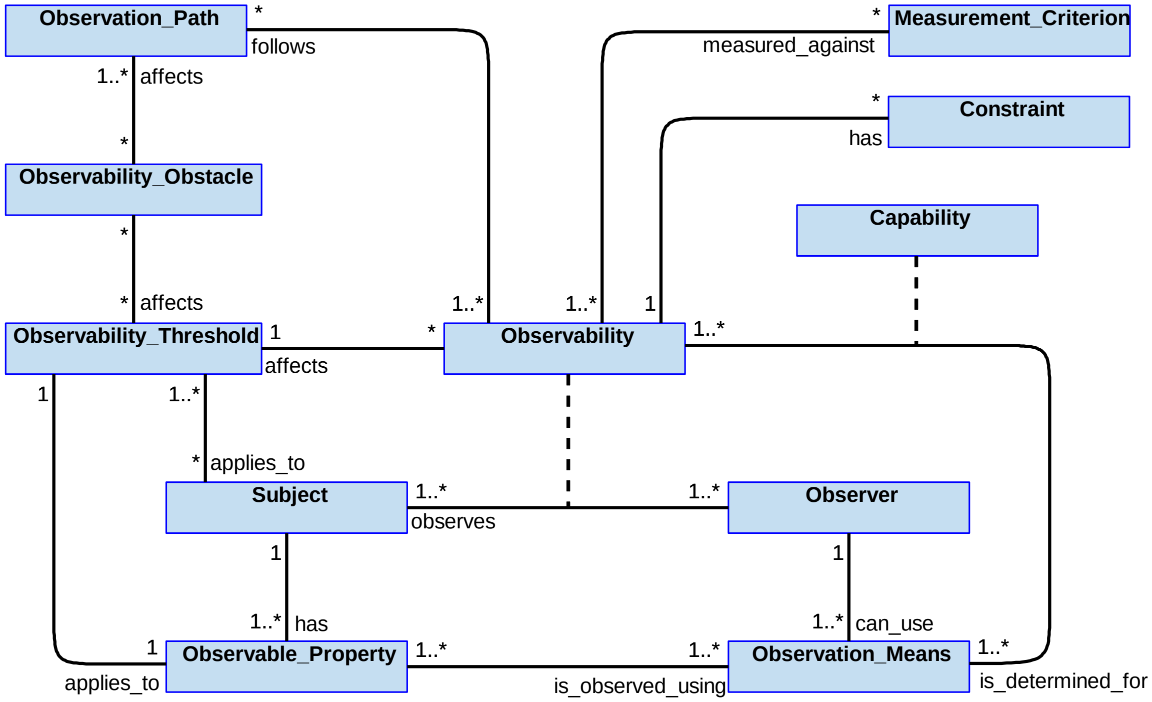


Figure 680: Observability Semantics

5.4.2.40.5.1 Entities

Capability

The capability to determine the Observability of the Subject, or the Observability_Threshold at which a Subject becomes observable.

Constraint

An externally imposed restriction that limits when or how Observability can be determined.

Measurement_Criterion

A criterion which the quality of Observability will be measured against.

Observability

A measure of whether the Observer can detect the Subject.

Observability_Obstacle

A feature that comes in between the Observer and Subject, e.g. weather, terrain, EM clutter.

Observable_Property

An emission or reflection that may be observed (e.g. EM Emission, RCS, or acoustic).

Observability_Threshold

The value, for a given Observable_Property, at which the Subject will become observable to an Observer.

Observation_Means

A method used by the Observer by which observation is to be achieved, e.g. visually, using radar or IR sensor.

Observer

The entity for which the Subject's Observability is being determined.

Subject

The entity for which Observability by the Observer is being determined.

Observation_Path

The path along which an Observer may perceive a Subject (e.g. straight line visual or sonar path).

5.4.2.40.6 Design Rationale

5.4.2.40.6.1 Assumptions

- Observability will have access to information about:
    - The Observable_Propertys of Subjects.
    - The sensing capabilities (Observation_Means) of observers (especially ownship).
    - Observability_Obstacles (e.g. weather or terrain).
    - Information derived from the trajectory of Subjects and Observers.
    - Specific identity of Subjects and Observers.

5.4.2.40.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Observability:

- Data Driving - To mitigate the significant variances in the means of observation, capabilities of sensors and characteristics of the Subject, etc. this information could be data-driven.

Extensions

- The observability of a subject varies according to the Subject, the Observation_Means used by the Observer, its sensor capability and any intervening Observability_Obstacles, etc. Extension components may be useful to apply different algorithms to cover this variability.

Exploitation Considerations

- It may be appropriate for an exploitation to include multiple instances of the Observability component dealing with different Observation_Means (e.g. using radar or IR) or Subject type (e.g. air vehicles or land vehicles), or each may be extension components to a single instance of the component.

5.4.2.40.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component may result in loss of line of sight communications between a UCS and UAV. As loss of communications can occur frequently for reasons outside of the control of the air system (e.g. interference due to weather or satellite infrastructure) then the air vehicle will have been designed to mitigate a loss of communications. For a UAS this would be achieved by relying on pre-determined automatic / autonomous behaviour. For this failure mode it is concluded that failure of this component may result a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.
- Additionally, failure of this component could result in the air vehicle being observed by enemy forces when not intended. Therefore, the air vehicle may be subjected to physical threat from enemy forces (e.g. missile attack). However, this is normally excluded from safety analysis. Therefore, an IDAL no more onerous than DAL C, is considered appropriate for this component.

5.4.2.40.6.4 Security Considerations

The indicative security classification is SNEO.

The component determines whether Subjects are observable using data for emissions, sensors and the intervening medium, it can also be used to determine if a communications link can be set up. The algorithms for determining observability are likely to be O, however the information on the capability of sensors and the signatures/stealth characteristics of different subjects (including the Exploiting Platform) is likely to be SNEO. Where necessary, there may be instances in different security domains, e.g. to cater for different sensors or intelligence data. These instances may need to communicate with each other to provide a full observability assessment. If so, separation will be handled externally to the component. Any loss of integrity or availability in the output of this component may lead to the Exploiting Platform placing itself in a situation where it may be observed by hostile forces. The confidentiality, integrity and availability requirements will need to reflect this. Where algorithms are data-driven, the associated configuration data will also carry appropriate confidentiality requirements.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of observability assessments made during the course of a mission.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.40.7 Services

5.4.2.40.7.1 Service Definitions

5.4.2.40.7.1.1 Query

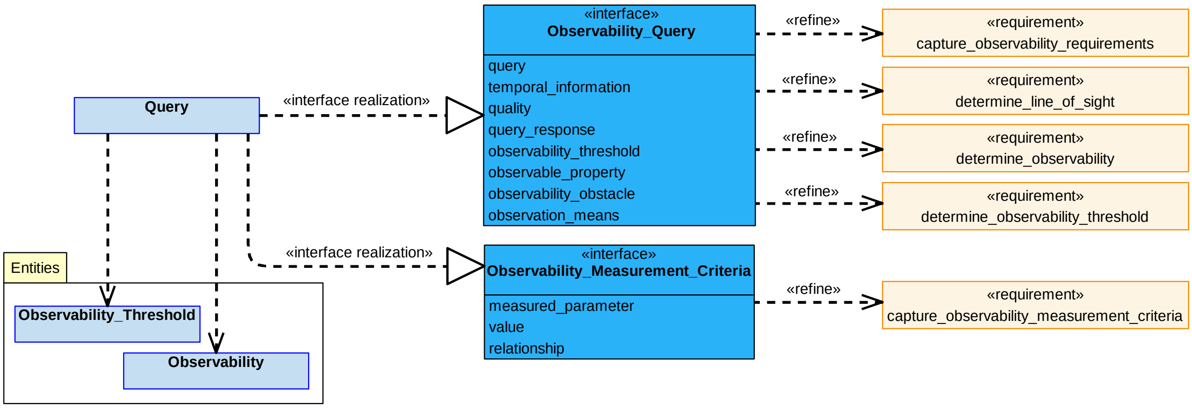
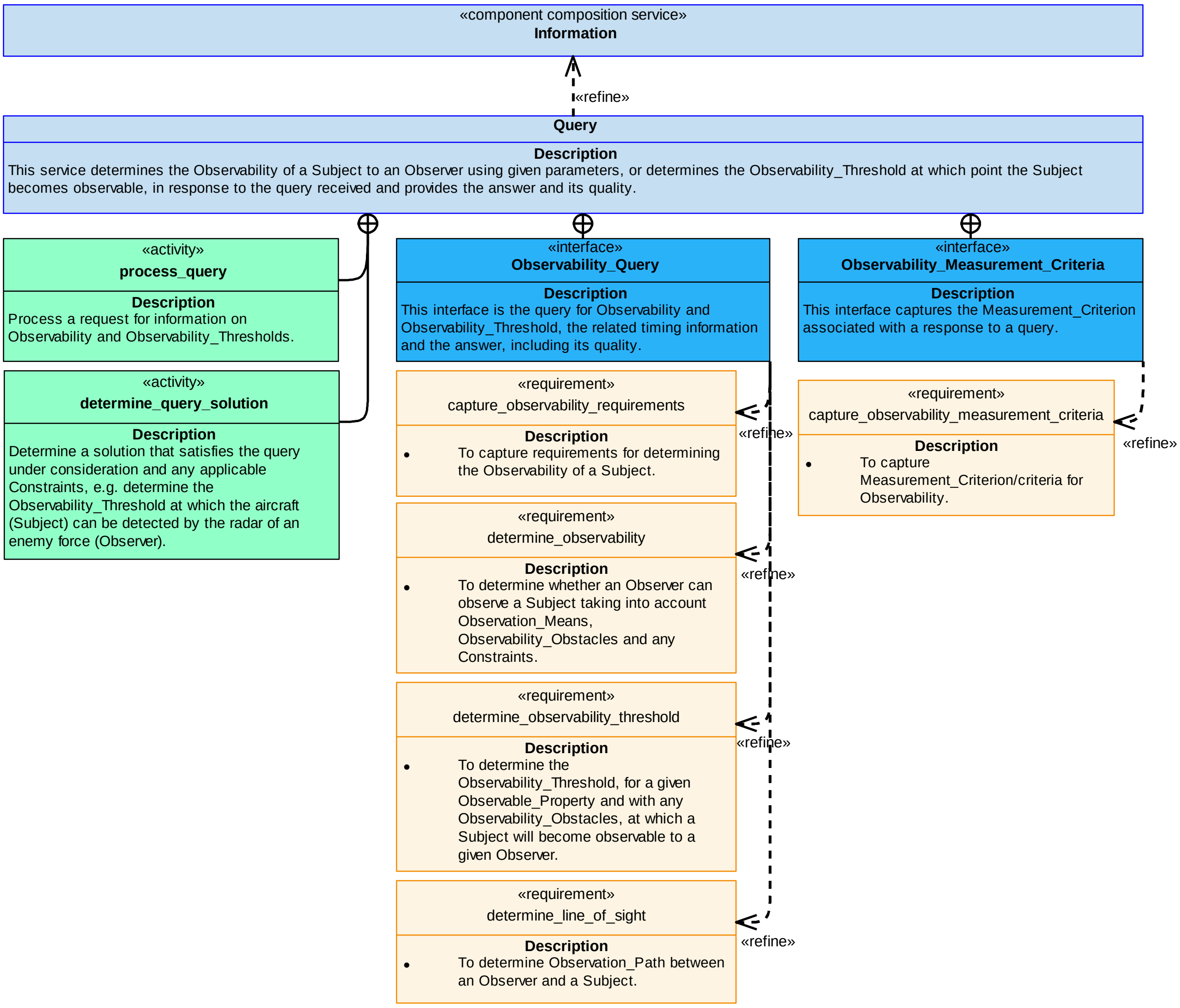


Figure 681: Query Service Definition



This service determines the Observability of a Subject to an Observer using given parameters, or determines the Observability_Threshold at which point the Subject becomes observable, in response to the query received and provides the answer and its quality.

Process a request for information on Observability and Observability_Thresholds.

Determine a solution that satisfies the query under consideration and any applicable Constraints, e.g. determine the Observability_Threshold at which the aircraft (Subject) can be detected by the radar of an enemy force (Observer).

This interface is the query for Observability and Observability_Threshold, the related timing information and the answer, including its quality.

This interface captures the Measurement_Criterion associated with a response to a query.

- To capture requirements for determining the Observability of a Subject.

- To determine whether an Observer can observe a Subject taking into account Observation_Means, Observability_Obstacles and any Constraints.

- To determine the Observability_Threshold, for a given Observable_Property and with any Observability_Obstacles, at which a Subject will become observable to a given Observer.

- To determine Observation_Path between an Observer and a Subject.

- To capture Measurement_Criterion/criteria for Observability.

Figure 682: Query Service Policy

Query

This service determines the Observability of a Subject to an Observer using given parameters, or determines the Observability_Threshold at which point the Subject becomes observable, in response to the query received and provides the answer and its quality.

Interfaces

Observability_Query

This interface is the query for Observability and Observability_Threshold, the related timing information and the answer, including its quality.

Attributes

query	The definition of the Observability query that is to be the determined; e.g. determine the Observability_Threshold at which the aircraft (Subject) can be detected by the radar of an enemy force (Observer).
temporal_information	Information covering timing, such as start and end times and any points in time which define changes in parameters.

quality	The quality of a query response against defined Measurement_Criterion.
query_response	The response to the query, stating whether a specific scenario results in the Observability of the Subject to the Observer, or providing the allowable value for a specified unknown parameter; e.g. the closest distance a Subject can approach an Observer before breaching an Observability_Threshold.
observability_threshold	The Observability_Threshold to be used for the query.
observable_property	The identification of the Observable_Property to be used for the query, allowing relevant information to be gathered.
observability_obstacle	The identification of any Observability_Obstacle(s) to be used for the query, allowing relevant information to be gathered.
observation_means	The identification of the Observation_Means to be used for the query, allowing relevant information to be gathered.

Observability_Measurement_Criteria

This interface captures the Measurement_Criterion associated with a response to a query.

Attributes

measured_parameter	The parameter the Measurement_Criterion is associated with.
value	An absolute value against which the Measurement_Criterion is to be judged (e.g. a given Observable_Property).
relationship	A relationship to a different value against which the Measurement_Criterion is to be judged (e.g. rather than a fixed value, the Measurement_Criterion is to be less than the current Observable_Property of a particular object).

Activities

process_query

Process a request for information on Observability and Observability_Thresholds.

determine_query_solution

Determine a solution that satisfies the query under consideration and any applicable Constraints, e.g. determine the Observability_Threshold at which the aircraft (Subject) can be detected by the radar of an enemy force (Observer).

5.4.2.40.7.1.2 Observability_Means_Information

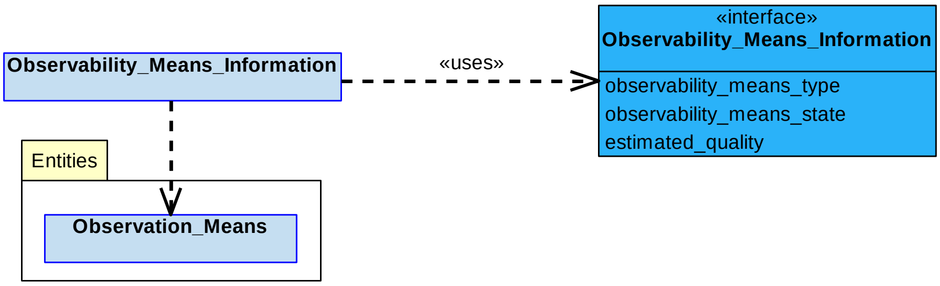

Observability_Means_Information
«uses»
«interface»
Observability_Means_Information
observability_means_type
observability_means_state
estimated_quality
Entities
Observation_Means

Figure 683: Observability_Means_Information Service Definition

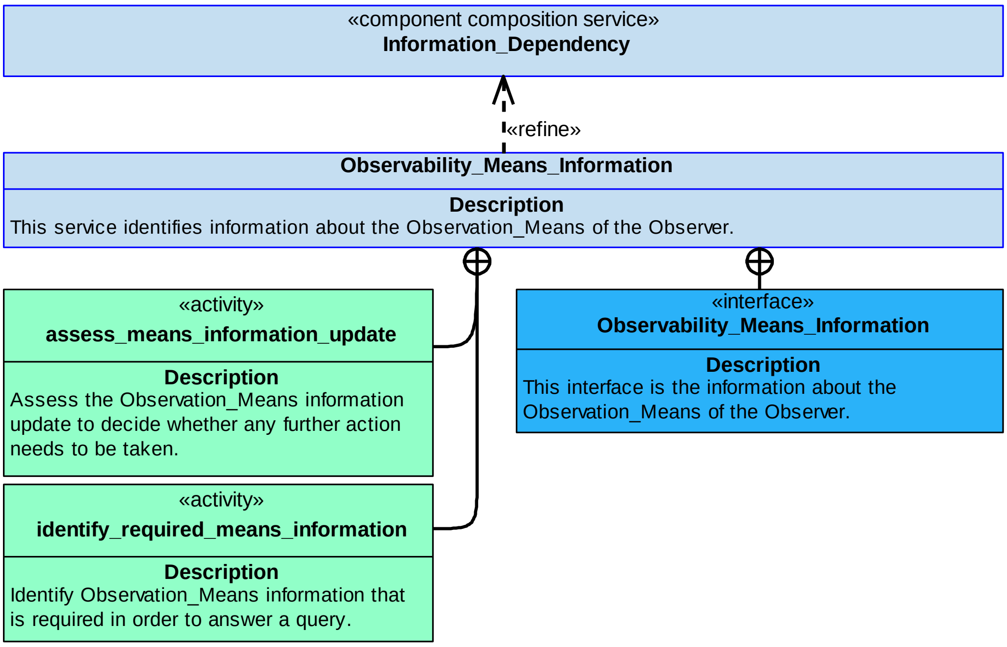

«component composition service»
Information_Dependency
«refine»
Observability_Means_Information
Description
This service identifies information about the Observation_Means of the Observer.
«activity»
assess_means_information_update
Description
Assess the Observation_Means information update to decide whether any further action needs to be taken.
«interface»
Observability_Means_Information
Description
This interface is the information about the Observation_Means of the Observer.
«activity»
identify_required_means_information
Description
Identify Observation_Means information that is required in order to answer a query.

Figure 684: Observability_Means_Information Service Policy

Observability_Means_Information

This service identifies information about the Observation_Means of the Observer.

Interface

Observability_Means_Information

This interface is the information about the Observation_Means of the Observer.

Attributes

observability_means_type	The type of Observation_Means being used by the Observer.
observability_means_state	The current configuration state and capability of the Observation_Means.
estimated_quality	The quality or confidence of the Observation_Means information, primarily for cases where the Observer under consideration is a different platform.

Activities

assess_means_information_update

Assess the Observation_Means information update to decide whether any further action needs to be taken.

identify_required_means_information

Identify Observation_Means information that is required in order to answer a query.

5.4.2.40.7.1.3 Observable_Property_Information

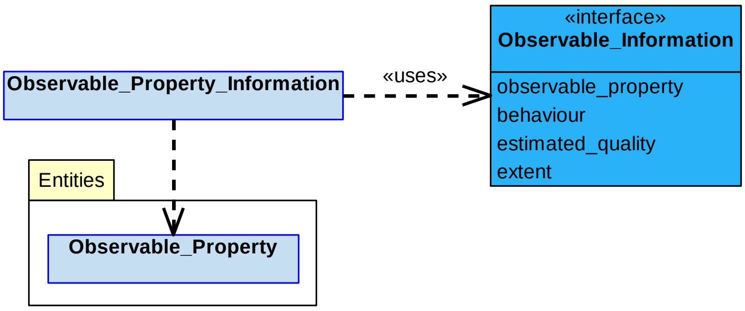


Figure 685: Observable_Property_Information Service Definition

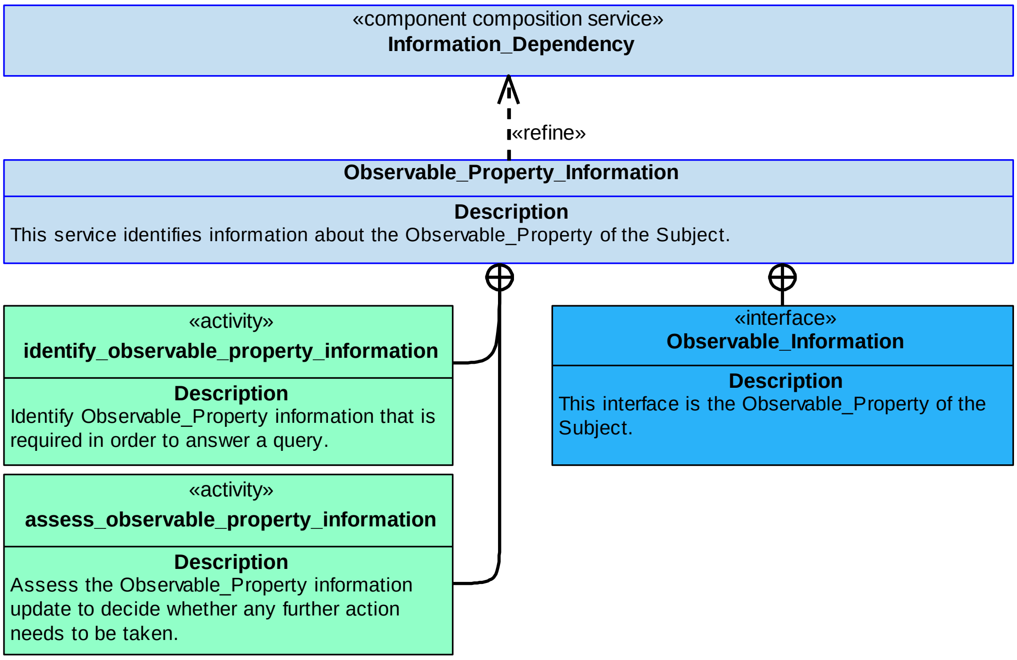

«component composition service»
Information_Dependency
«refine»
Observable_Property_Information
Description
This service identifies information about the Observable_Property of the Subject.
«activity»
identify_observable_property_information
Description
Identify Observable_Property information that is required in order to answer a query.
«activity»
assess_observable_property_information
Description
Assess the Observable_Property information update to decide whether any further action needs to be taken.
«interface»
Observable_Information
Description
This interface is the Observable_Property of the Subject.

Figure 686: Observable_Property_Information Service Policy

Observable_Property_Information

This service identifies information about the Observable_Property of the Subject.

Interface

Observable_Information

This interface is the Observable_Property of the Subject.

Attributes

observable_property	The Observable_Property being used for the Subject.
behaviour	Any aspect of the Subject's known or estimated behaviour which could impact the Observability of its Observable_Property.
estimated_quality	The quality or confidence of the Observable_Property information, primarily for cases where the Subject under consideration is a different platform.

extent	The size and extent of a Subject which could impact the Observability of its Observable_Property.

Activities

identify_observable_property_information

Identify Observable_Property information that is required in order to answer a query.

assess_observable_property_information

Assess the Observable_Property information update to decide whether any further action needs to be taken.

5.4.2.40.7.1.4 Observability_Obstacle_Information

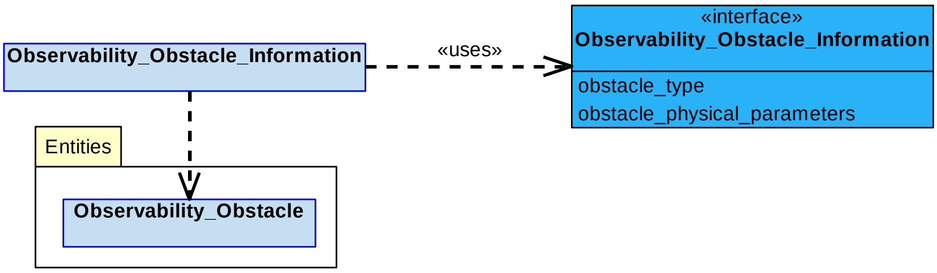

Observability_Obstacle_Information
«uses»
«interface»
Observability_Obstacle_Information
obstacle_type
obstacle_physical_parameters
Entities
Observability_Obstacle

Figure 687: Observability_Obstacle_Information Service Definition

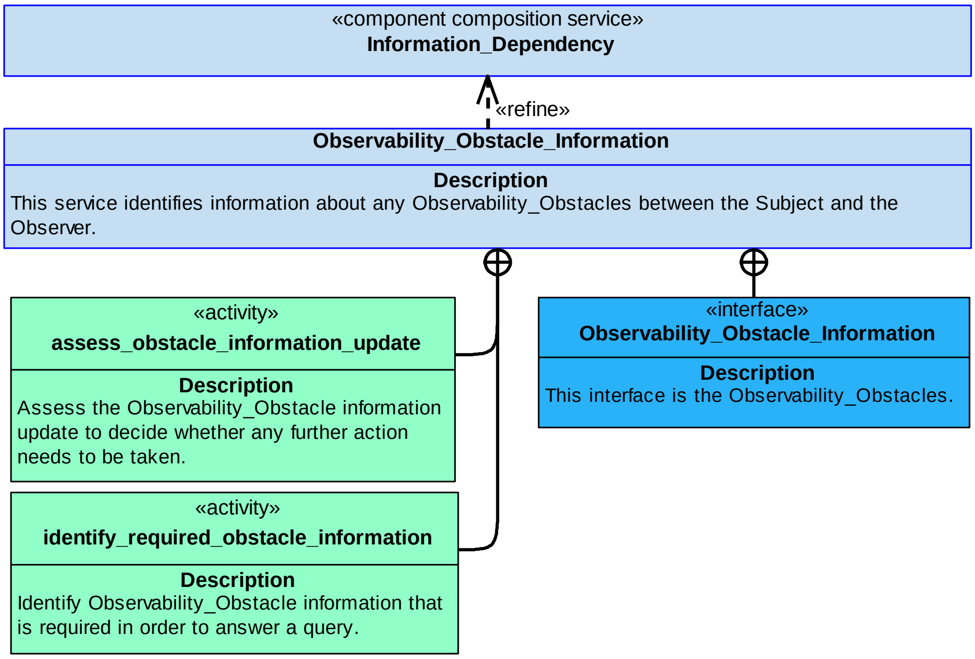

«component composition service»
Information_Dependency
«refine»
Observability_Obstacle_Information
Description
This service identifies information about any Observability_Obstacles between the Subject and the Observer.
«activity»
assess_obstacle_information_update
Description
Assess the Observability_Obstacle information update to decide whether any further action needs to be taken.
«interface»
Observability_Obstacle_Information
Description
This interface is the Observability_Obstacles.
«activity»
identify_required_obstacle_information
Description
Identify Observability_Obstacle information that is required in order to answer a query.

Figure 688: Observability_Obstacle_Information Service Policy

Observability_Obstacle_Information

This service identifies information about any Observability_Obstacles between the Subject and the Observer.

Interface

Observability_Obstacle_Information

This interface is the Observability_Obstacles.

Attributes

obstacle_type	The type of Observability_Obstacle under consideration.
obstacle_physical_parameters	The spatial location (and size) of an Observability_Obstacle

Activities

assess_obstacle_information_update

Assess the Observability_Obstacle information update to decide whether any further action needs to be taken.

identify_required_obstacle_information

Identify Observability_Obstacle information that is required in order to answer a query.

5.4.2.40.7.1.5 Participant_Kinematics_Information

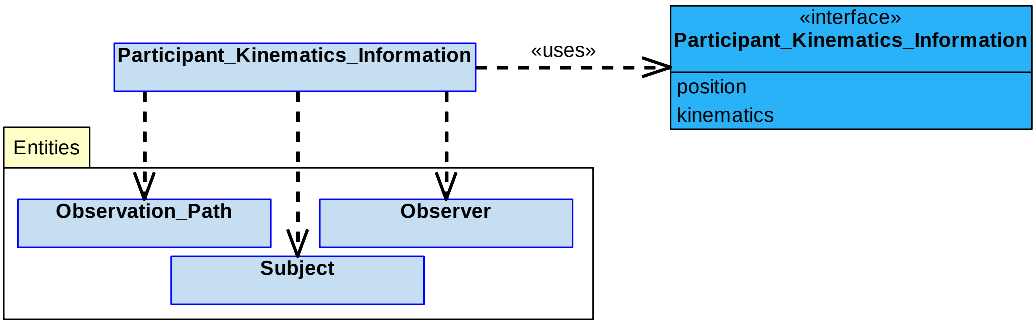


Figure 689: Participant_Kinematics_Information Service Definition

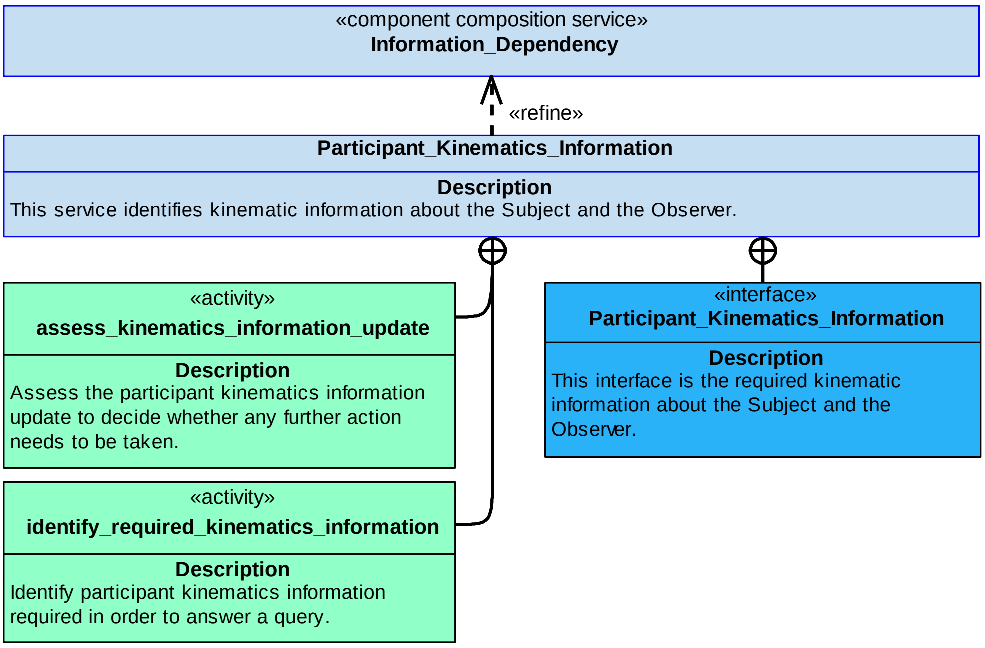

«component composition service»
Information_Dependency
«refine»
Participant_Kinematics_Information
Description
This service identifies kinematic information about the Subject and the Observer.
«activity»
assess_kinematics_information_update
Description
Assess the participant kinematics information update to decide whether any further action needs to be taken.
«interface»
Participant_Kinematics_Information
Description
This interface is the required kinematic information about the Subject and the Observer.
«activity»
identify_required_kinematics_information
Description
Identify participant kinematics information required in order to answer a query.

Figure 690: Participant_Kinematics_Information Service Policy

Participant_Kinematics_Information

This service identifies kinematic information about the Subject and the Observer.

Interface

Participant_Kinematics_Information

This interface is the required kinematic information about the Subject and the Observer.

Attributes

position	The current position of the Observer or Subject.
kinematics	The kinematics of an Observer or Subject, e.g. the path that an Observer is predicted to traverse.

Activities

assess_kinematics_information_update

Assess the participant kinematics information update to decide whether any further action needs to be taken.

identify_required_kinematics_information

Identify participant kinematics information required in order to answer a query.

5.4.2.40.7.1.6 Constraint

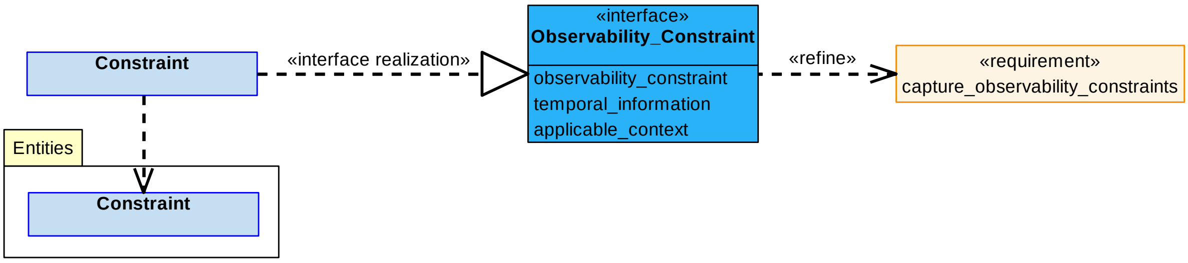


Figure 691: Constraint Service Definition

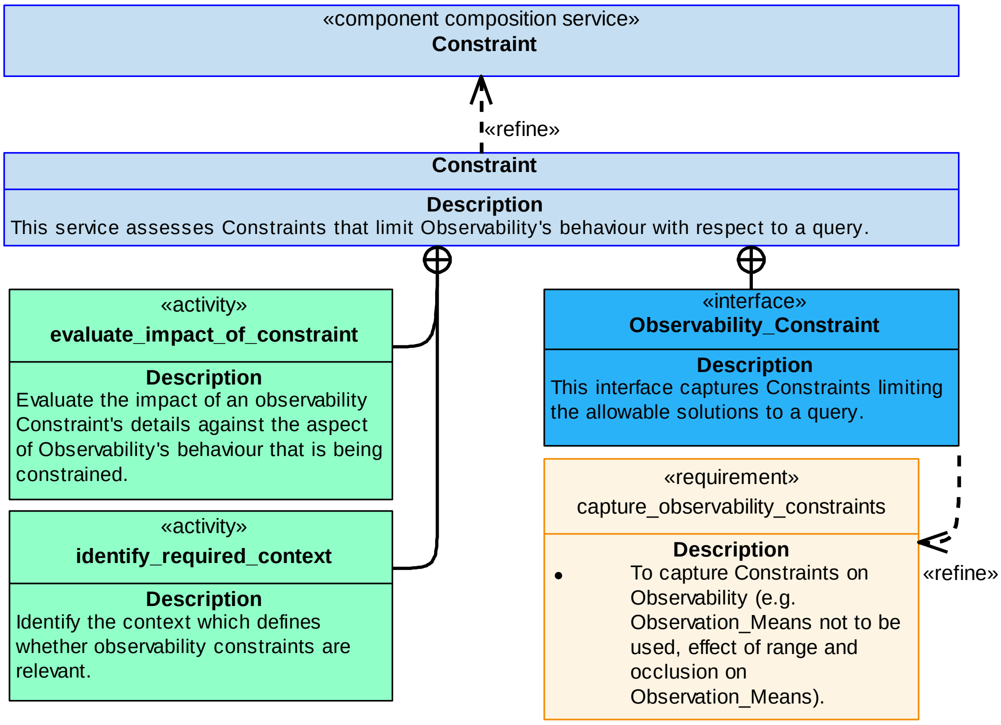

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints that limit Observability's behaviour with respect to a query.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of an observability Constraint's details against the aspect of Observability's behaviour that is being constrained.
«activity»
identify_required_context
Description
Identify the context which defines whether observability constraints are relevant.
«interface»
Observability_Constraint
Description
This interface captures Constraints limiting the allowable solutions to a query.
«requirement»
capture_observability_constraints
Description
To capture Constraints on Observability (e.g. Observation_Means not to be used, effect of range and occlusion on Observation_Means).
«refine»

Figure 692: Constraint Service Policy

Constraint

This service assesses Constraints that limit Observability's behaviour with respect to a query.

Interface

Observability_Constraint

This interface captures Constraints limiting the allowable solutions to a query.

Attributes

observability_constraint	A Constraint that limits observability behaviour, e.g. EM restrictions.
temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the Constraint is applicable.

Activities

evaluate_impact_of_constraint

Evaluate the impact of an observability Constraint's details against the aspect of Observability's behaviour that is being constrained.

identify_required_context

Identify the context which defines whether observability constraints are relevant.

5.4.2.40.7.1.7 Capability

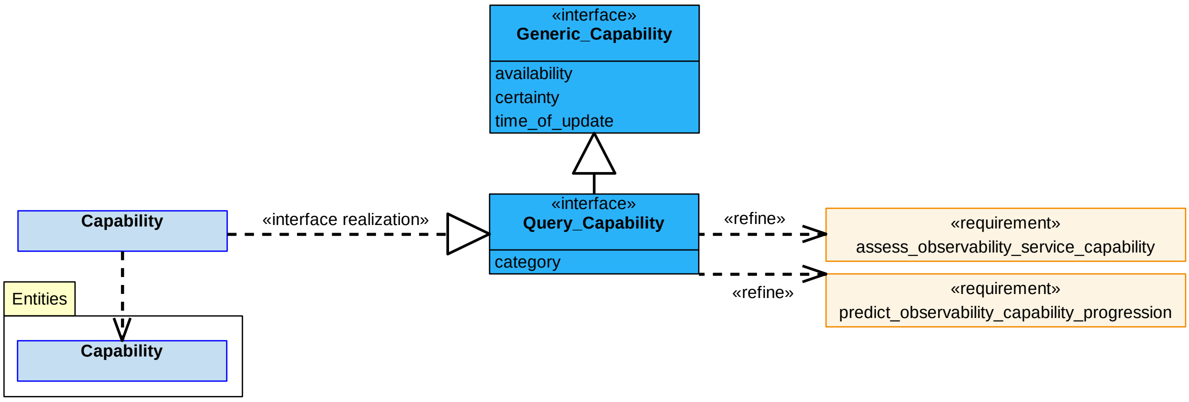


Figure 693: Capability Service Definition

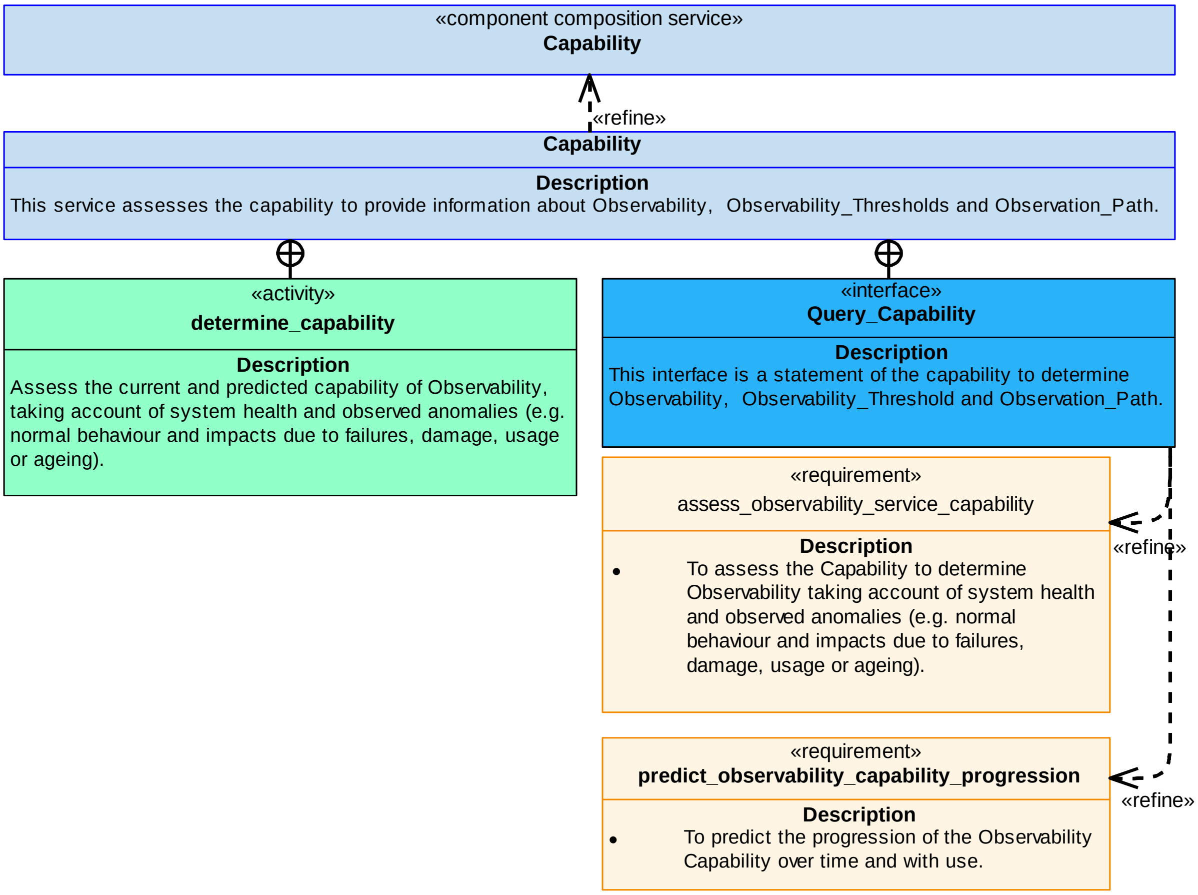

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the capability to provide information about Observability, Observability_Thresholds and Observation_Path.
«activity»
determine_capability
Description
Assess the current and predicted capability of Observability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Query_Capability
Description
This interface is a statement of the capability to determine Observability, Observability_Threshold and Observation_Path.
«requirement»
assess_observability_service_capability
Description
To assess the Capability to determine Observability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_observability_capability_progression
Description
To predict the progression of the Observability Capability over time and with use.
«refine»

Figure 694: Capability Service Policy

Capability

This service assesses the capability to provide information about Observability, Observability_Thresholds and Observation_Path.

Interface

Query_Capability

This interface is a statement of the capability to determine Observability, Observability_Threshold and Observation_Path.

Attribute

category	A category of Observability query, e.g. related to a specific Observer or Observation_Means.

Activity

determine_capability

Assess the current and predicted capability of Observability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.40.7.1.8 Capability_Evidence

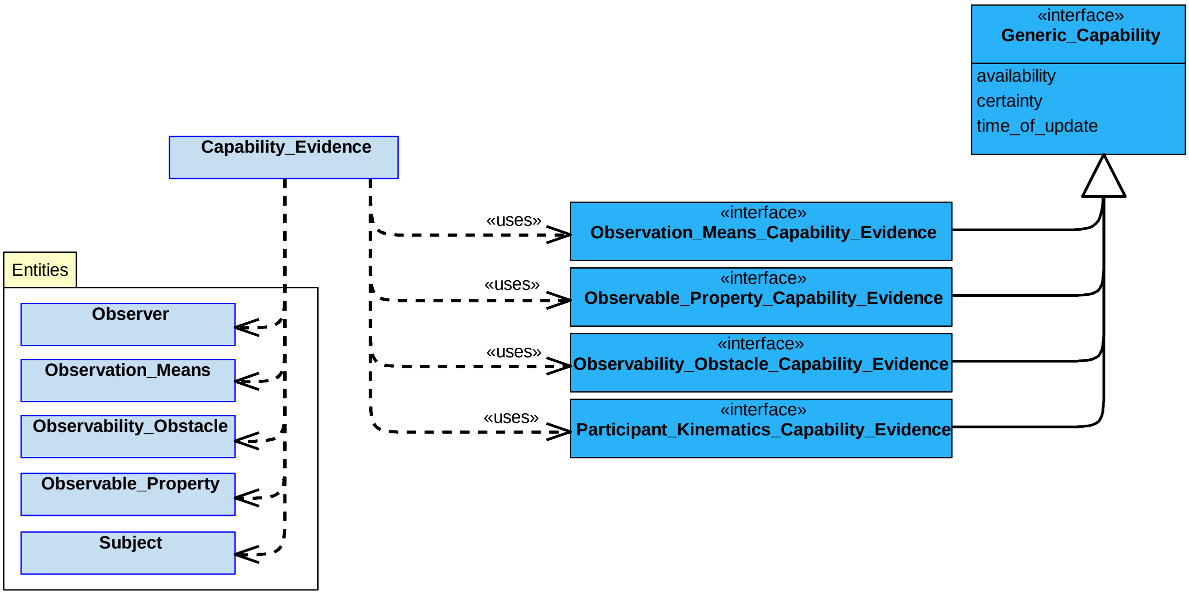


Figure 695: Capability_Evidence Service Definition

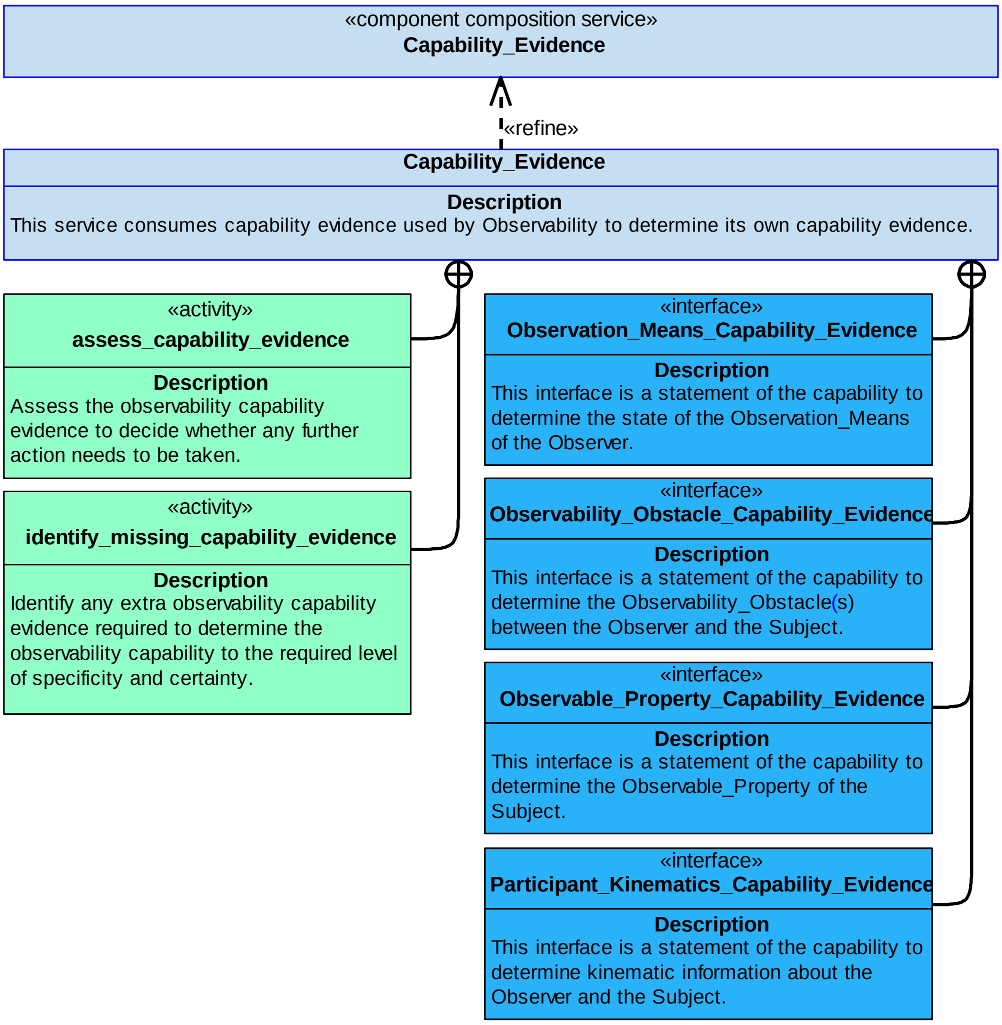

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capability evidence used by Observability to determine its own capability evidence.
«activity»
assess_capability_evidence
Description
Assess the observability capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra observability capability evidence required to determine the observability capability to the required level of specificity and certainty.
«interface»
Observation_Means_Capability_Evidence
Description
This interface is a statement of the capability to determine the state of the Observation_Means of the Observer.
«interface»
Observability_Obstacle_Capability_Evidence
Description
This interface is a statement of the capability to determine the Observability_Obstacle(s) between the Observer and the Subject.
«interface»
Observable_Property_Capability_Evidence
Description
This interface is a statement of the capability to determine the Observable_Property of the Subject.
«interface»
Participant_Kinematics_Capability_Evidence
Description
This interface is a statement of the capability to determine kinematic information about the Observer and the Subject.

Figure 696: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability evidence used by Observability to determine its own capability evidence.

<u>Interfaces</u>

Observation_Means_Capability_Evidence

This interface is a statement of the capability to determine the state of the Observation_Means of the Observer.

Observable_Property_Capability_Evidence

This interface is a statement of the capability to determine the Observable_Property of the Subject.

Observability_Obstacle_Capability_Evidence

This interface is a statement of the capability to determine the Observability_Obstacle(s) between the Observer and the Subject.

Participant_Kinematics_Capability_Evidence

This interface is a statement of the capability to determine kinematic information about the Observer and the Subject.

Activities

assess_capability_evidence

Assess the observability capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra observability capability evidence required to determine the observability capability to the required level of specificity and certainty.

5.4.2.40.7.2 Service Dependencies

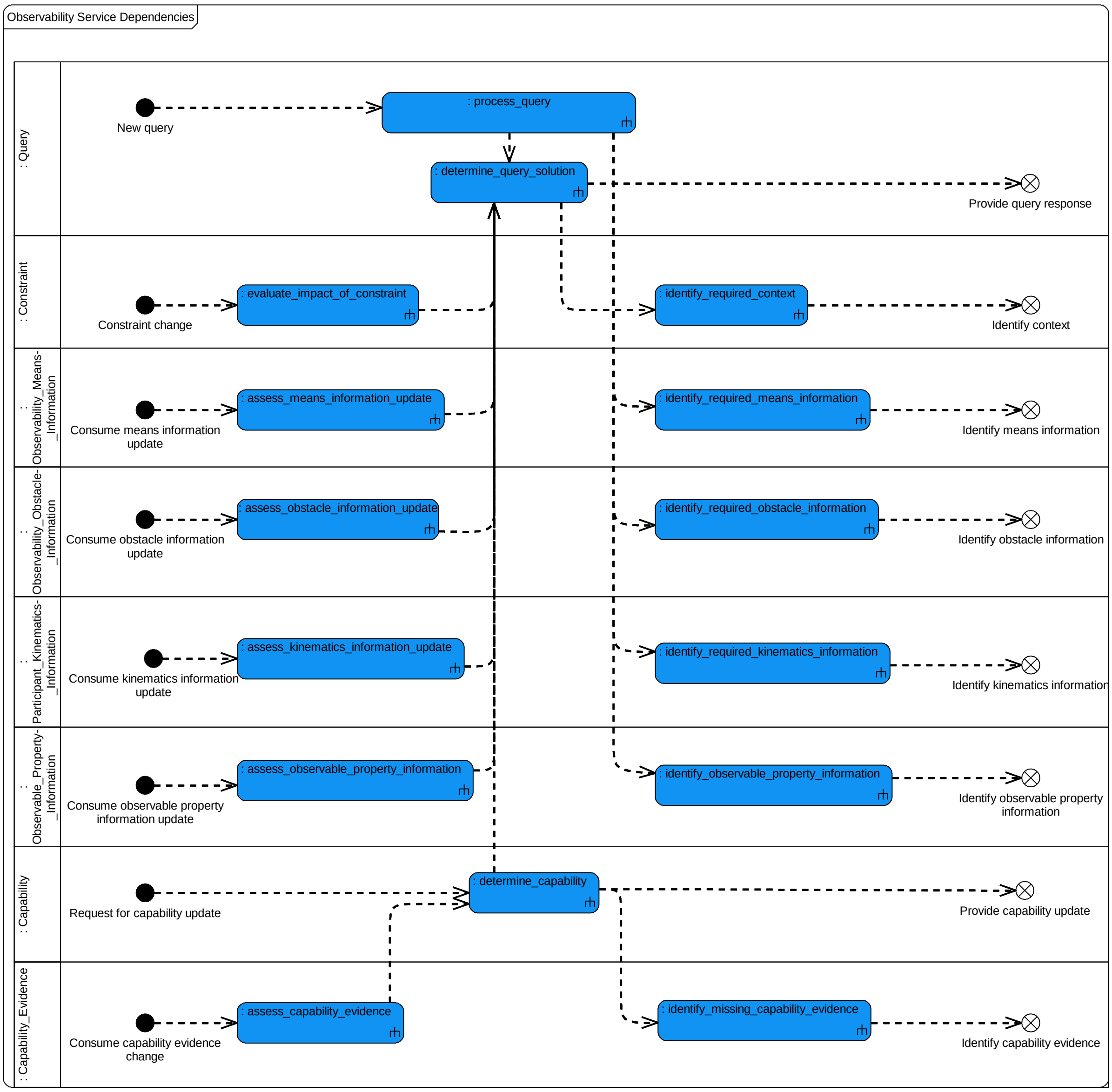

Observability Service Dependencies
: Query
: Constraint
: Observability_Means-_Information
: Observability_Obstacle-_Information
: Participant_Kinematics-_Information
: Observable_Property-_Information
: Capability
: Capability_Evidence
New query
: process_query
: determine_query_solution
Provide query response
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume means information update
: assess_means_information_update
: identify_required_means_information
Identify means information
Consume obstacle information update
: assess_obstacle_information_update
: identify_required_obstacle_information
Identify obstacle information
Consume kinematics information update
: assess_kinematics_information_update
: identify_required_kinematics_information
Identify kinematics information
Consume observable property information update
: assess_observable_property_information
: identify_observable_property_information
Identify observable property information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 697: Observability Service Dependencies

5.4.2.41 Operational Rules and Limits

5.4.2.41.1 Role

The role of Operational Rules and Limits is to derive limits from rules.

5.4.2.41.2 Overview

Control Architecture

Operational Rules and Limits is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Operational Rules and Limits determines any active Limits to be applied; these Limits are constraints placed upon other components.

Limit_Breach is reported by an external component when a Limit has been broken.

Rule_Breaches are calculated internally based upon Limit_Breaches.

Components may request hypothetical Limit information. For example, EMCON at another location.

Examples of Use

Operational Rules and Limits could be used for the following purposes:

- To limit a system capable of autonomous or semi-autonomous behaviour such that courses of action adopted or recommended are within the extant Rules of Engagement.
- To translate an EMCON policy into limits which are suitable for application to a system.

5.4.2.41.3 Service Summary

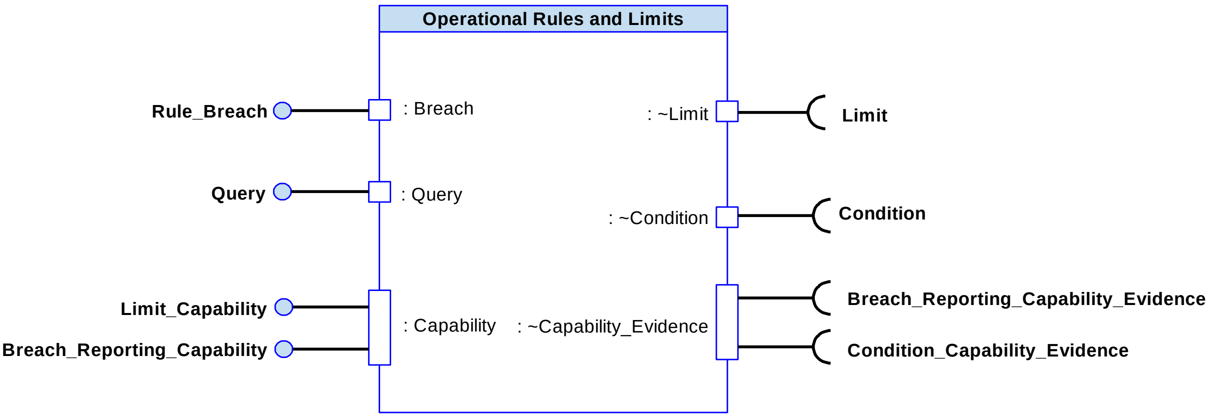

Operational Rules and Limits
Rule_Breach
: Breach
: ~Limit
Limit
Query
: Query
: ~Condition
Condition
Limit_Capability
Breach_Reporting_Capability
: Capability
: ~Capability_Evidence
Breach_Reporting_Capability_Evidence
Condition_Capability_Evidence

Figure 698: Operational Rules and Limits Service Summary

5.4.2.41.4 Responsibilities

capture_rules

- To capture Operational_Rules.

determine_applicable_rules

- To determine which Operational_Rules are applicable under given conditions related to current or forecast Conditions.

determine_applicable_limits

- To determine which Limits are applicable under applicable rules.

determine_breach_of_rule

- To determine which Operational_Rules are broken following a Limit_Breach.

assess_capability

- To assess the Capability to provide Limits and report Rule_Breaches.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of Capability over time and with use.

5.4.2.41.5 Subject Matter Semantics

The subject matter of Operational Rules and Limits is the status and applicability of Operational_Rules, and Limits arising from them.

Exclusions

The subject matter of Operational Rules and Limits does not include:

- Ensuring adherence with constraints that are identified for the system from the Operational_Rules.

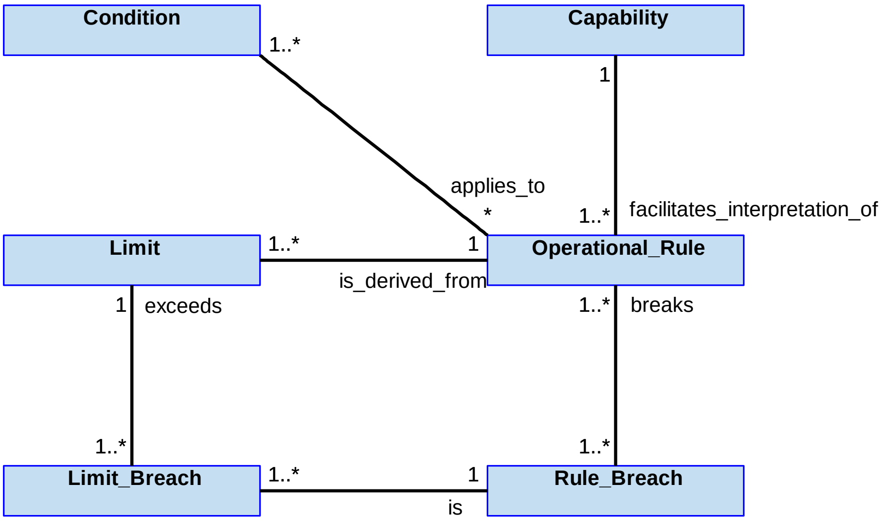


Figure 699: Operational Rules and Limits Semantics

5.4.2.41.5.1 Entities

Capability

The range of activities involved in being able to provide limits and report breaches of operational rules.

Condition

An operating parameter which could be either current or hypothetical, e.g. the Exploiting Platform's current speed or location, or nearby tactical objects.

Limit

An instance of a restriction on the size, amount or usage of something within the system, e.g. a torque limit setting for an engine or whether use of a particular weapons system is permitted.

Limit_Breach

An externally reported exceedance of a limit, e.g. the airspeed limit being set at 500 knots and the Exploiting Platform is travelling at 700 knots.

Operational_Rule

A regulation or principle governing conduct or procedure.

Rule_Breach

An indication that a rule has been broken, e.g. an over-speed has occurred.

5.4.2.41.6 Design Rationale

5.4.2.41.6.1 Assumptions

- Operational Rules and Limits is not responsible for determining the legitimacy of Operational_Rule sources.
- Operational Rules and Limits will not determine whether the implementation of a Limit results in a reduction in system capability.
- As discussed in the Constraint Management PYRAMID concept, Operational Rules and Limits will not manage solution-based constraints. These will be managed across the architecture.
- The relationship between Operational_Rules and Limits will be defined using a controlled process outside the scope of the PRA.

5.4.2.41.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Operational Rules and Limits:

- Constraint Management - Operational Rules and Limits follows the PYRAMID concept.

- Data Driving - The Limits maintained by the Operational Rules and Limits component for use throughout the system are intended to be data-driven at build time.

Extensions

- It may be appropriate to use extension components for different sets of rules.

Exploitation Considerations

- Restricting knowledge of the Operational_Rules and their applicability to this one component minimises the number of components exposed to potentially highly classified Operational_Rules.

5.4.2.41.6.3 Safety Considerations

The indicative IDAL is DAL A*.*

The rationale behind this is:

- Failure of this component would lead to incorrect Operational_Rule based constraints being used within the system. It is assumed that some of these constraints are used to achieve an acceptable level of safety and prevent hazards with potentially catastrophic consequences. Therefore, an indicative IDAL of DAL A is considered appropriate.

Where instances of this component are used to prevent hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.41.6.4 Security Considerations

The indicative security classification is SNEO.

This component determines all of the Operational_Rules and subsequent Limits that will apply to the Exploiting Platform, and these rules will include (amongst others) the current RoE, therefore the indicative security classification is considered to be SNEO. As the applicable limits can constrain the ability of the Exploiting Platform to conduct its mission objectives, the integrity of the component can be considered critical to the success of the mission.

The component is expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** to support forensic examination.
- **Maintaining Audit Records** that show when particular rules and limits came into effect, and when they were lifted.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **System Status and Monitoring** to identify where rules and limits have been breached. The unexpected breaching of an applied limit may indicate the presence of malware subverting the normal behaviour of the system.

Supports security enforcing functions by:

- Identifying the operational rules to be used when **Applying EMCON Rules** (enforced by another component).

5.4.2.41.7 Services

5.4.2.41.7.1 Service Definitions

5.4.2.41.7.1.1 Breach

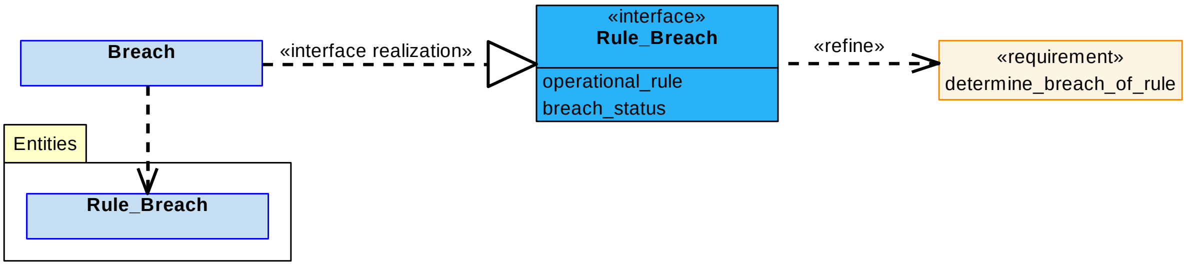


Figure 700: Breach Service Definition

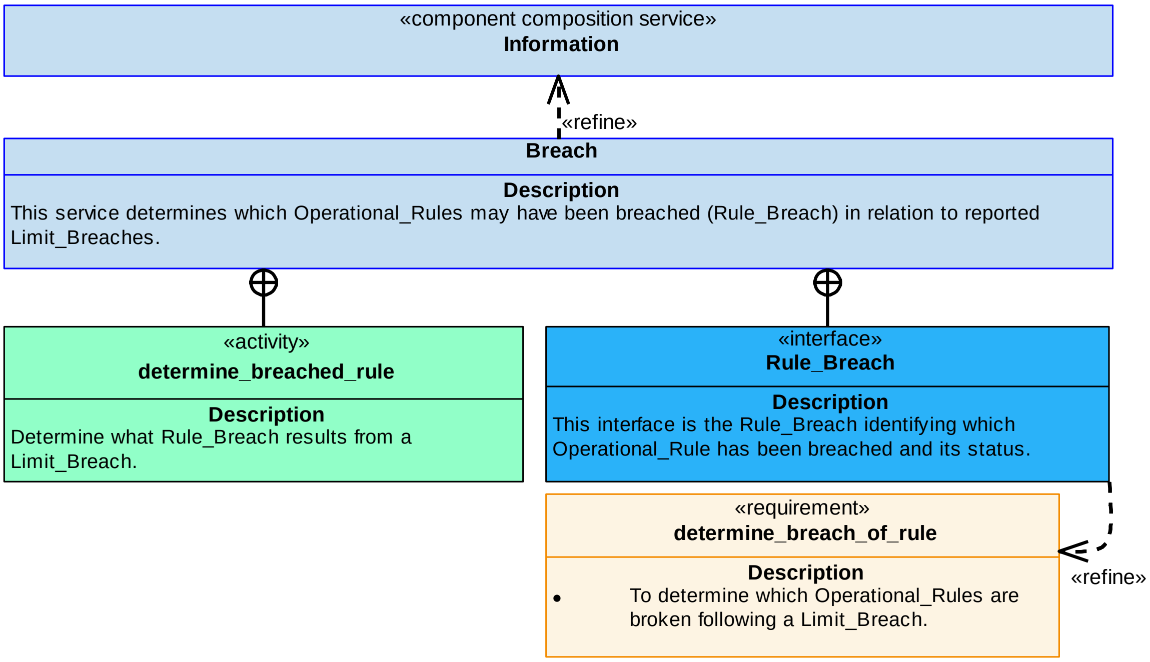

«component composition service»
Information
«refine»
Breach
Description
This service determines which Operational_Rules may have been breached (Rule_Breach) in relation to reported Limit_Breaches.
«activity»
determine_breached_rule
Description
Determine what Rule_Breach results from a Limit_Breach.
«interface»
Rule_Breach
Description
This interface is the Rule_Breach identifying which Operational_Rule has been breached and its status.
«requirement»
determine_breach_of_rule
Description
To determine which Operational_Rules are broken following a Limit_Breach.
«refine»

Figure 701: Breach Service Policy

Breach

This service determines which Operational_Rules may have been breached (Rule_Breach) in relation to reported Limit_Breaches.

Interface

Rule_Breach

This interface is the Rule_Breach identifying which Operational_Rule has been breached and its status.

Attributes

operational_rule	The Operational_Rules which relate to a reported Limit_Breach.
breach_status	The current state of a breach, e.g. breached, cleared, risk of breach, or not being used.

Activity

determine_breached_rule

Determine what Rule_Breach results from a Limit_Breach.

5.4.2.41.7.1.2 Query

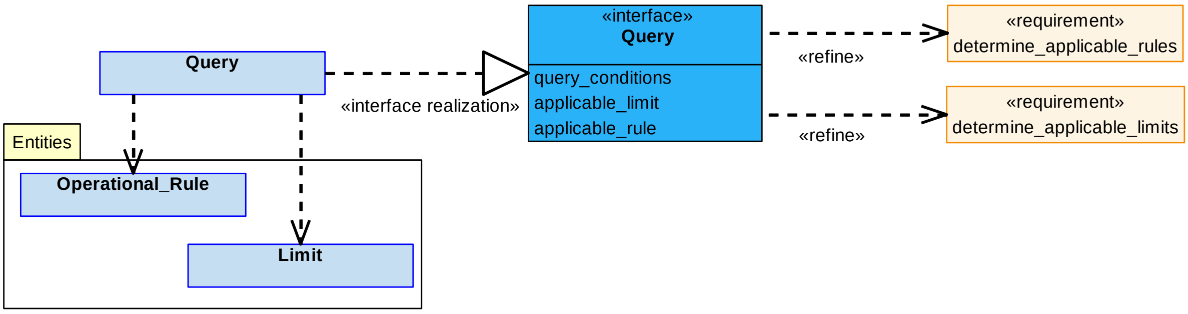


Figure 702: Query Service Definition

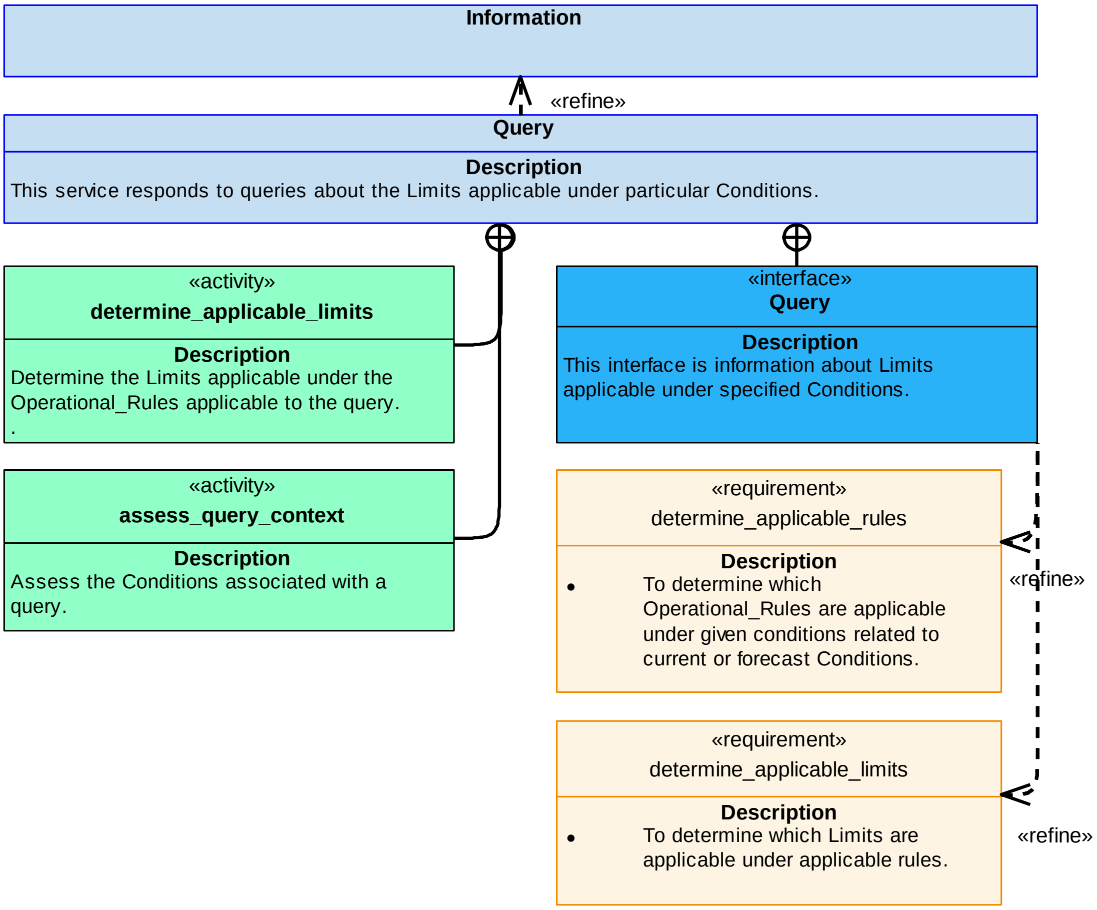

Information
«refine»
Query
Description
This service responds to queries about the Limits applicable under particular Conditions.
«activity»
determine_applicable_limits
Description
Determine the Limits applicable under the Operational_Rules applicable to the query.
.
«interface»
Query
Description
This interface is information about Limits applicable under specified Conditions.
«activity»
assess_query_context
Description
Assess the Conditions associated with a query.
«requirement»
determine_applicable_rules
Description
To determine which Operational_Rules are applicable under given conditions related to current or forecast Conditions.
«refine»
«requirement»
determine_applicable_limits
Description
To determine which Limits are applicable under applicable rules.
«refine»

Figure 703: Query Service Policy

Query

This service responds to queries about the Limits applicable under particular Conditions.

Interface

Query

This interface is information about Limits applicable under specified Conditions.

Attributes

query_conditions	The Conditions for which applicable Operational_Rules and Limits are to be determined.
applicable_limit	The definition of the applicable Limit, e.g. a maximum speed of 50 m/s.
applicable_rule	The definition of the applicable rule.

Activities

determine_applicable_limits

Determine the Limits applicable under the Operational_Rules applicable to the query.

.

assess_query_context

Assess the Conditions associated with a query.

5.4.2.41.7.1.3 Limit

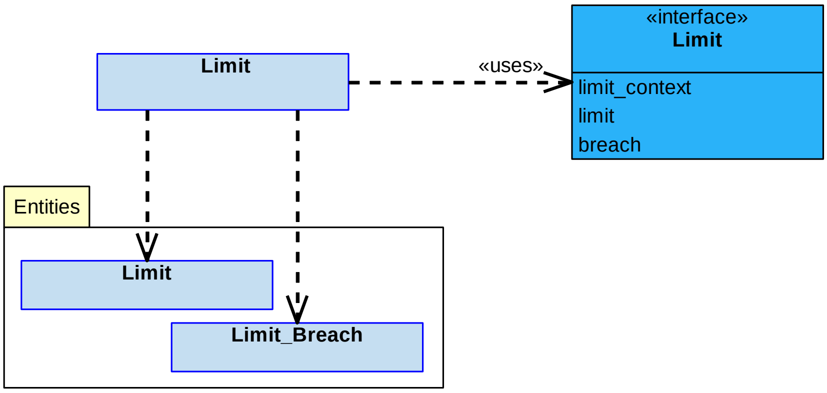


Figure 704: Limit Service Definition

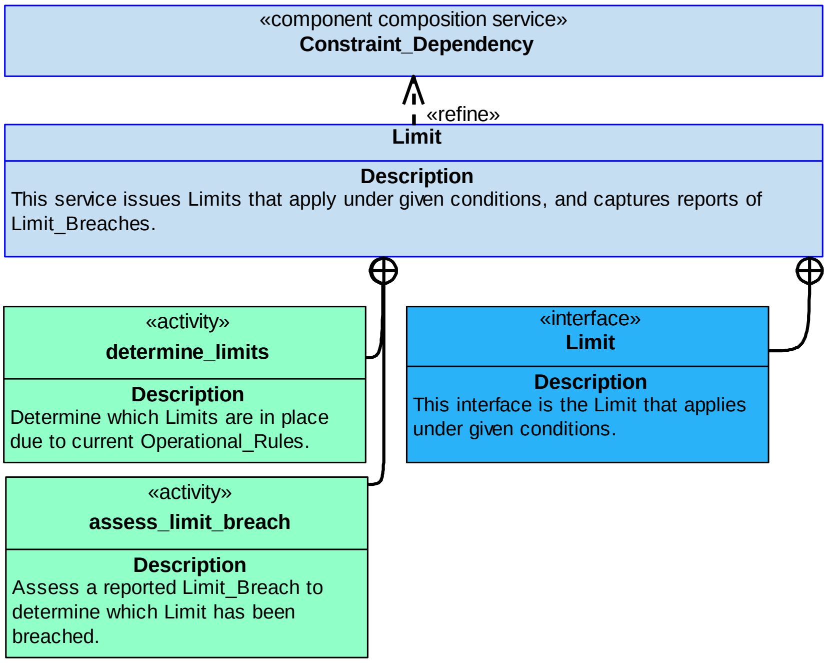

«component composition service»
Constraint_Dependency
«refine»
Limit
Description
This service issues Limits that apply under given conditions, and captures reports of Limit_Breaches.
«activity»
determine_limits
Description
Determine which Limits are in place due to current Operational_Rules.
«interface»
Limit
Description
This interface is the Limit that applies under given conditions.
«activity»
assess_limit_breach
Description
Assess a reported Limit_Breach to determine which Limit has been breached.

Figure 705: Limit Service Policy

Limit

This service issues Limits that apply under given conditions, and captures reports of Limit_Breaches.

<u>Interface</u>

Limit

This interface is the Limit that applies under given conditions.

Attributes

limit_context	The conditions under which the Limit is applicable, e.g. location, timeframe, or altitude.
limit	The definition of the Limit, e.g. a maximum speed of 50 m/s.
breach	A statement that the limit has been breached, or is likely to be breached if enforced.

Activities

determine_limits

Determine which Limits are in place due to current Operational_Rules.

assess_limit_breach

Assess a reported Limit_Breach to determine which Limit has been breached.

5.4.2.41.7.1.4 Condition

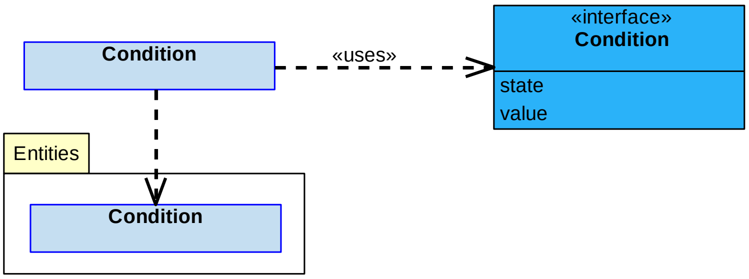


Figure 706: Condition Service Definition

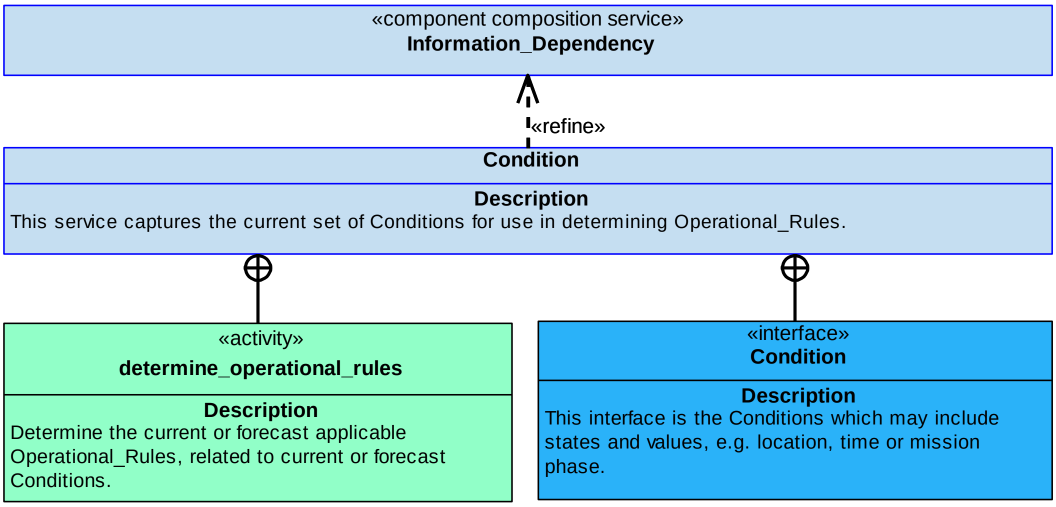

«component composition service»
Information_Dependency
«refine»
Condition
Description
This service captures the current set of Conditions for use in determining Operational_Rules.
«activity»
determine_operational_rules
Description
Determine the current or forecast applicable Operational_Rules, related to current or forecast Conditions.
«interface»
Condition
Description
This interface is the Conditions which may include states and values, e.g. location, time or mission phase.

Figure 707: Condition Service Policy

Condition

This service captures the current set of Conditions for use in determining Operational_Rules.

Interface

Condition

This interface is the Conditions which may include states and values, e.g. location, time or mission phase.

Attributes

state	A current state, e.g. mission phase.
value	A current value, e.g. an altitude of 1,000ft or a remaining fuel status of 50%.

Activity

determine_operational_rules

Determine the current or forecast applicable Operational_Rules, related to current or forecast Conditions.

5.4.2.41.7.1.5 Capability

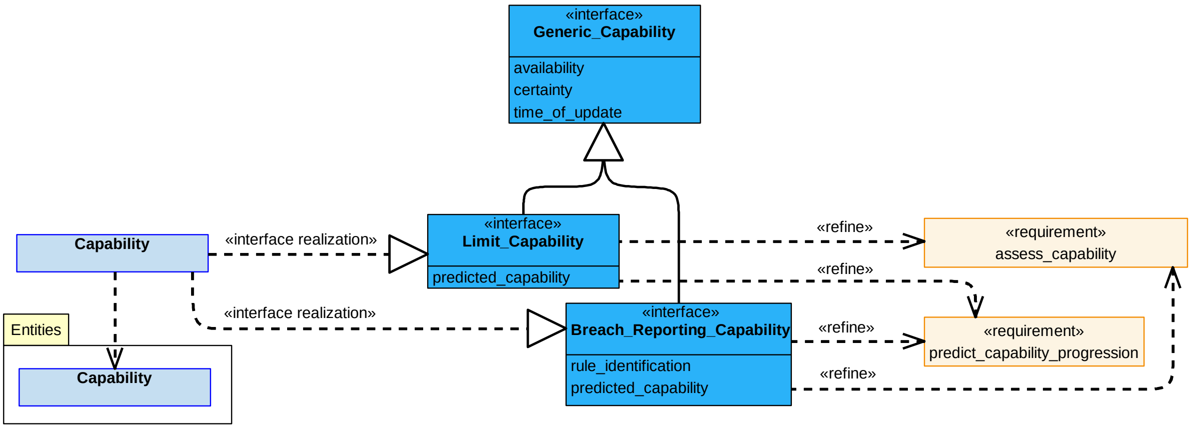


Figure 708: Capability Service Definition

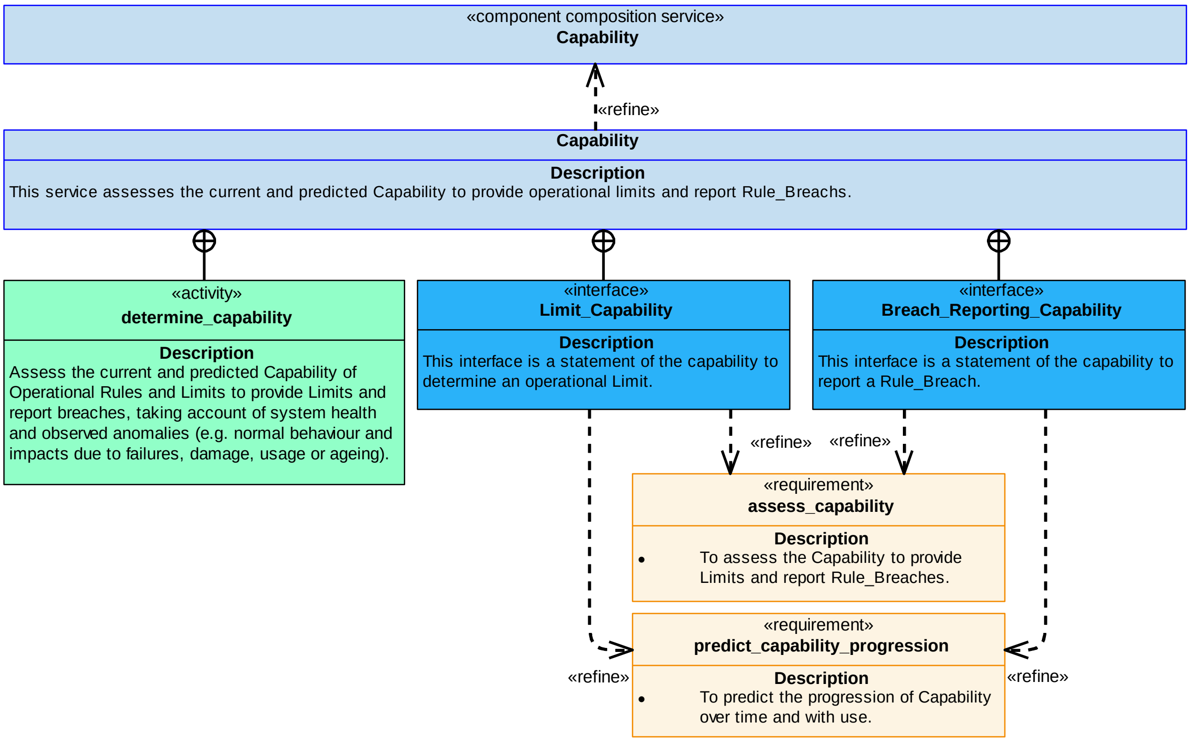

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to provide operational limits and report Rule_Breachs.
«activity»
determine_capability
Description
Assess the current and predicted Capability of Operational Rules and Limits to provide Limits and report breaches, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Limit_Capability
Description
This interface is a statement of the capability to determine an operational Limit.
«interface»
Breach_Reporting_Capability
Description
This interface is a statement of the capability to report a Rule_Breach.
«refine»
«refine»
«requirement»
assess_capability
Description
To assess the Capability to provide Limits and report Rule_Breaches.
«requirement»
predict_capability_progression
Description
To predict the progression of Capability over time and with use.
«refine»
«refine»

Figure 709: Capability Service Policy

Capability

This service assesses the current and predicted Capability to provide operational limits and report Rule_Breachs.

Interfaces

Breach_Reporting_Capability

This interface is a statement of the capability to report a Rule_Breach.

Attributes

rule_identification	Identifier to match the Operational_Rule that is being reported on.
predicted_capability	A measure of how the Rule_Breach reporting capability is expected to change over time.

Limit_Capability

This interface is a statement of the capability to determine an operational Limit.

Attribute

predicted_capability	A measure of how the ability to provide Limit values is expected to change over time.

Activity

determine_capability

Assess the current and predicted Capability of Operational Rules and Limits to provide Limits and report breaches, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.41.7.1.6 Capability_Evidence

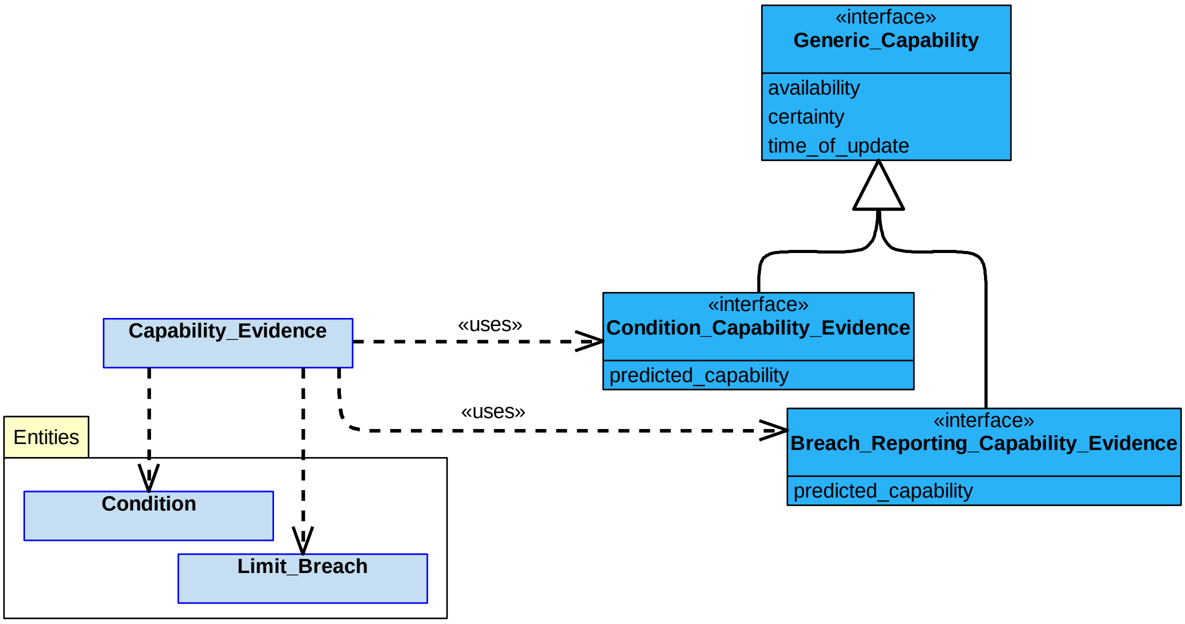


Figure 710: Capability_Evidence Service Definition

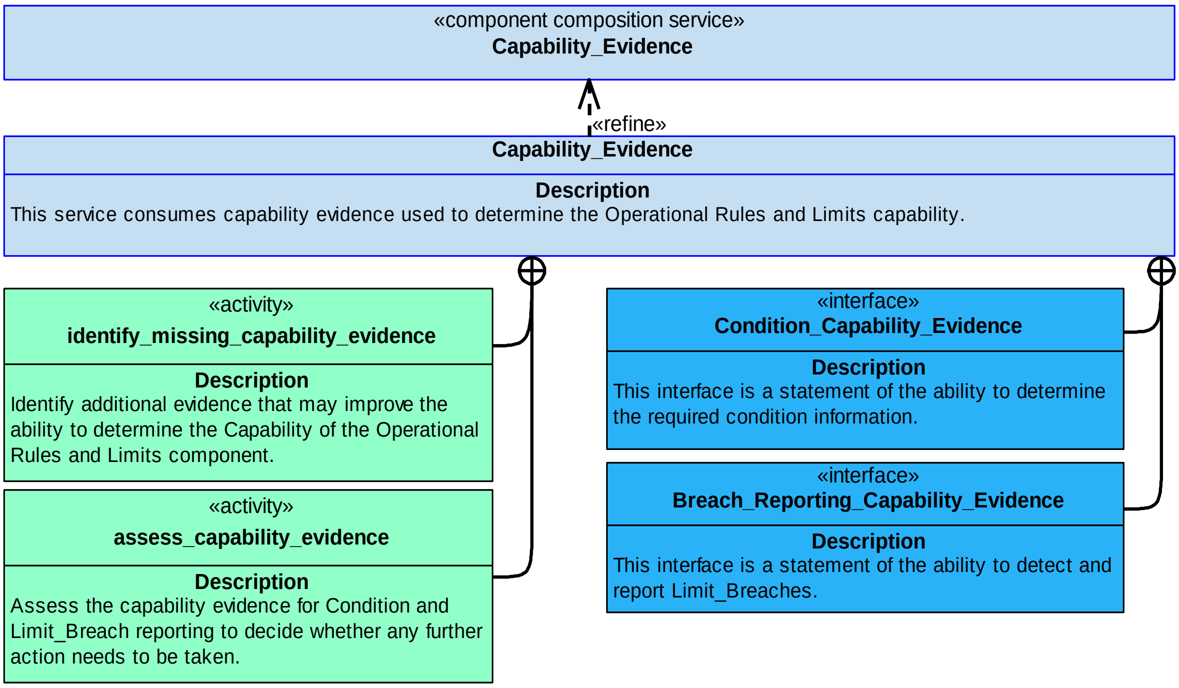

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capability evidence used to determine the Operational Rules and Limits capability.
«activity»
identify_missing_capability_evidence
Description
Identify additional evidence that may improve the ability to determine the Capability of the Operational Rules and Limits component.
«activity»
assess_capability_evidence
Description
Assess the capability evidence for Condition and Limit_Breach reporting to decide whether any further action needs to be taken.
«interface»
Condition_Capability_Evidence
Description
This interface is a statement of the ability to determine the required condition information.
«interface»
Breach_Reporting_Capability_Evidence
Description
This interface is a statement of the ability to detect and report Limit_Breaches.

Figure 711: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability evidence used to determine the Operational Rules and Limits capability.

Interfaces

Breach_Reporting_Capability_Evidence

This interface is a statement of the ability to detect and report Limit_Breaches.

Attribute

predicted_capability	A measure of how the ability to monitor for Limit_Breaches is expected to change over time.

Condition_Capability_Evidence

This interface is a statement of the ability to determine the required condition information.

Attribute

predicted_capability	A measure of how the ability to determine the condition is expected to change over time.

Activities

identify_missing_capability_evidence

Identify additional evidence that may improve the ability to determine the Capability of the Operational Rules and Limits component.

assess_capability_evidence

Assess the capability evidence for Condition and Limit_Breach reporting to decide whether any further action needs to be taken.

5.4.2.41.7.2 Service Dependencies

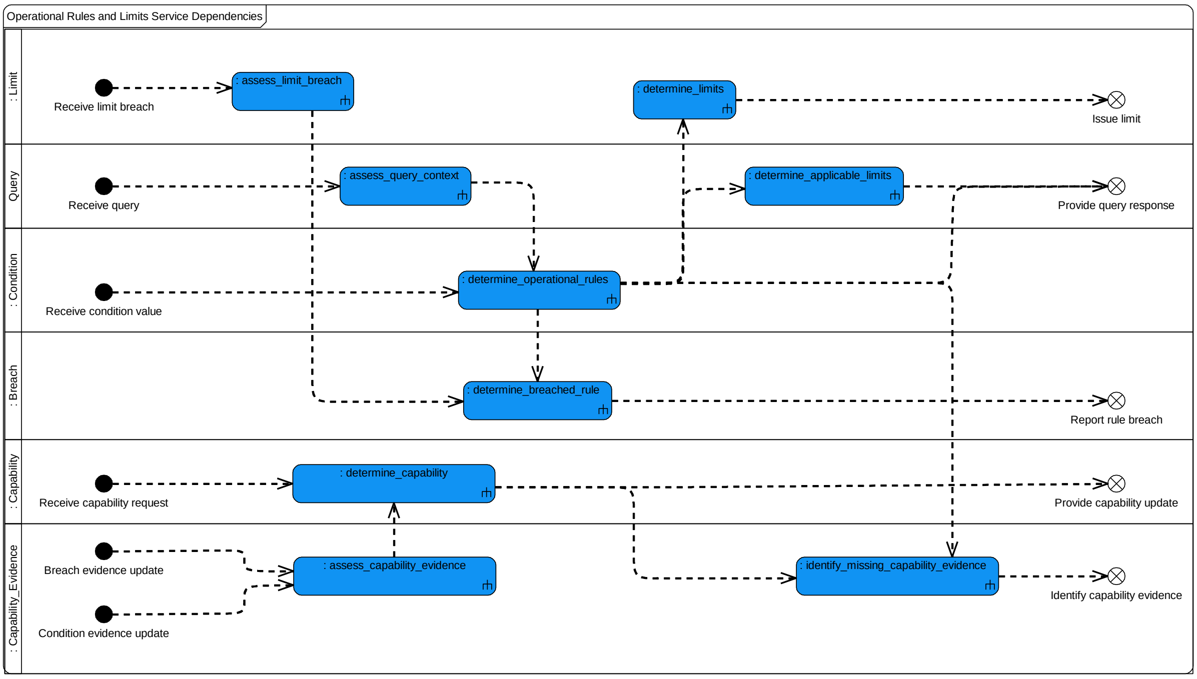

Operational Rules and Limits Service Dependencies
: Limit
Query
: Condition
: Breach
: Capability
: Capability_Evidence
Receive limit breach
: assess_limit_breach
: determine_limits
Issue limit
Receive query
: assess_query_context
: determine_applicable_limits
Provide query response
Receive condition value
: determine_operational_rules
: determine_breached_rule
Report rule breach
Receive capability request
: determine_capability
Provide capability update
Breach evidence update
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence
Condition evidence update

Figure 712: Operational Rules and Limits Service Dependencies

Assess the hypothetical Conditions associated with a query.

5.4.2.42 Pointing

5.4.2.42.1 Role

The role of Pointing is to determine and control the orientation of equipment that can be directed in its orientation.

5.4.2.42.2 Overview

Control Architecture

Pointing is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When there is a requirement to orientate a piece of equipment on the Exploiting Platform, Pointing will determine how to achieve the desired orientation, and put a demand on a resource component to attain the orientation required. Pointing can also monitor the Controllable_Element (e.g. an LDP) being used whilst it is performing its demand.

Examples of Use

Pointing will be used for the positioning of all Controllable_Elements that can be directed in their orientation, such as:

- LDPs.
- Turrets.
- Directional antennas.

5.4.2.42.3 Service Summary

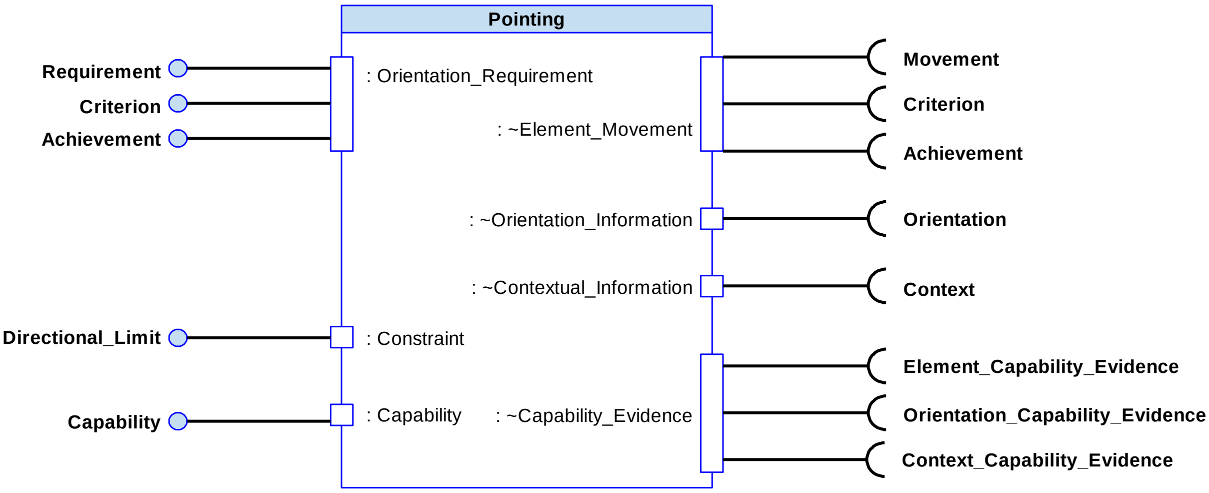


Figure 713: Pointing Service Summary

5.4.2.42.4 Responsibilities

capture_orientation_requirements

- To capture the orientation Requirements (e.g. required position or turn rate).

capture_orientation_measurement_criteria

- To capture provided criteria that an Orientation_Solution will be measured against.

capture_orientation_constraints

- To capture provided Constraints for Orientation actions.

determine_orientation_solution

- To determine an Orientation_Solution that meets the given Requirements and Constraints.

determine_current_orientation

- To determine the current Orientation of a Controllable_Element.

identify_orientation_solution_in_progress_remains_feasible

- To identify whether an Orientation_Solution in progress remains feasible.

coordinate_orientation_solution

- To coordinate the Orientation_Solution to ensure the individual Controllable_Element is oriented correctly (either mechanically or electronically).

identify_progress_of_orientation_solution

- To identify the progress of an Orientation_Solution against the Requirements.

determine_quality_of_orientation_solution

- To determine the quality of a proposed Orientation_Solution against given Measurement_Criterion/criteria.

determine_quality_of_deliverables

- To determine the quality of the outcomes generated by executing an Orientation_Solution, measured against given Requirements and Measurement_Criterion/criteria.

assess_orientation_capability

- To assess the Orientation_Capability available to the Pointing component taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Orientation_Capability assessment.

predict_capability_progression

- To predict the progression of the Orientation_Capability over time and with use.

5.4.2.42.5 Subject Matter Semantics

The subject matter of Pointing is the orientation of Controllable_Elements.

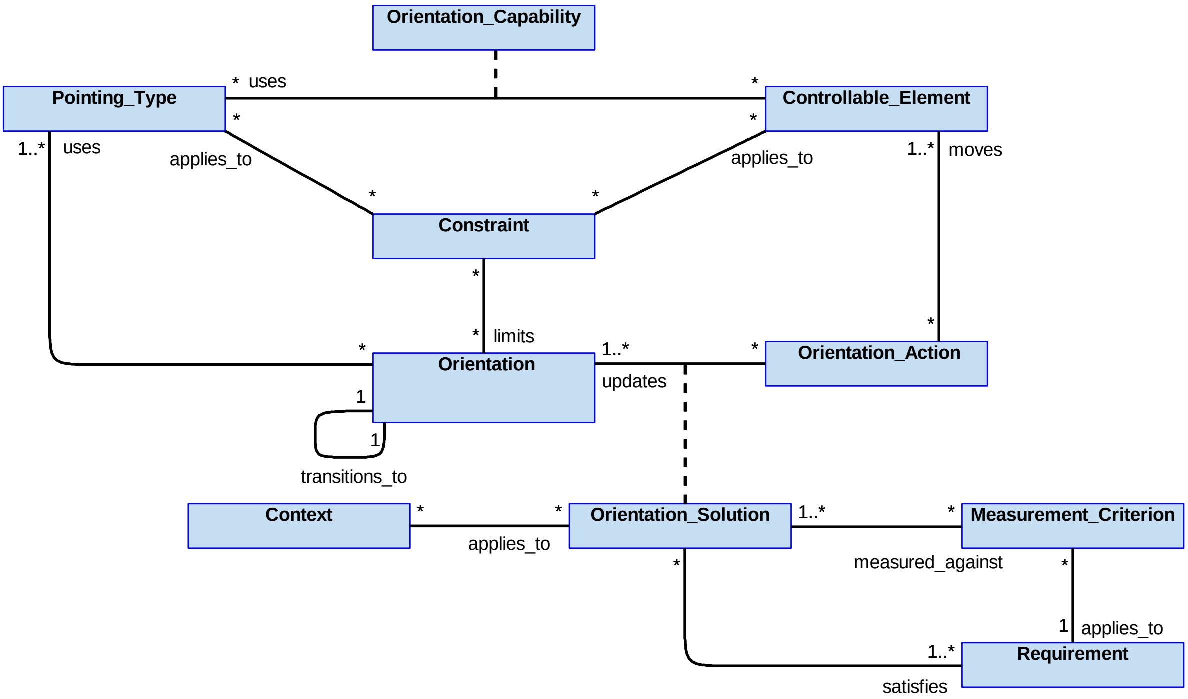


Figure 714: Pointing Semantics

5.4.2.42.5.1 Entities

Context

Information about the context in which the pointing is being carried out, e.g. any conditions that need accounting for.

Constraint

An externally placed limit on where a Controllable_Element can point, or how it gets there, e.g. preventing any diversion from current orientation of an element, to keep variable geometry from being infringed or limiting the rate of rotation.

Controllable_Element

An element (e.g. LDP or turret) that is capable of being oriented.

Measurement_Criterion

A criterion used to determine quality or progress.

Orientation

The direction in which the Controllable_Element (e.g. LDP or turret) is pointing relative to an identified datum.

Orientation_Capability

The range of Pointing_Types that the component is able to utilise with its available Controllable_Elements.

Orientation_Action

An action that can be used to achieve or maintain orientation.

Orientation_Solution

A selected set of actions and parameters that can be used to achieve or maintain the Orientation in which a Controllable_Element is pointing.

Pointing_Type

A type of movement that can be used to orientate, e.g. the rotation of a turret or the steering of the aircraft.

Requirement

A requirement to achieve or maintain the Orientation of a Controllable_Element, e.g. to point an antenna towards a transmitter or to maintain an LDP pointing at a tactical object.

5.4.2.42.6 Design Rationale

5.4.2.42.6.1 Assumptions

- Pointing is only responsible for developing and maintaining the requirements to direct a Controllable_Element for functional reasons. The implementation of this will be enacted by other components, for example Mechanical Positioning.
- Any safety related limits for a Controllable_Element's position or use have been agreed outside this component. This component needs to be cognisant of these limits and comply with them. However, these limits maybe enforced by other components (for example Interlocks), interfacing equipment or mechanical stops.
- The pointing of the vehicle may be required to orientate some types of equipment (e.g. fixed or limited-control equipment). Where steering cues are provided by this component, they are expected to be arbitrated by the system as per any other steering input.
- Where the equipment to be oriented does not have free 360 degree movement, an Orientation_Solution may include the coordination of a number of elements to align that equipment.
- Context may include offset information to cover the effects of gravity or atmospheric conditions, etc. on the pointed resource.
- This component will not require operational performance information for the asset being pointed, only those related to its movement (e.g. range and speed of movement).
- Sufficient feedback on the achieved orientation is provided by sensors so that the effector's reported orientation does not have to be relied on.

5.4.2.42.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Pointing:

- Data Driving - This PYRAMID concept will allow the component to be configured to orient different types of Controllable_Element.
- Interaction with Equipment - This PYRAMID concept details how interaction with new and different pieces of equipment can occur at any appropriate level, including this component, if that level is appropriate for the equipment in question.

Extensions

- The Pointing component could be used to implement both electronic and mechanical pointing through the use of extension components.

Exploitation Considerations

- There could be multiple instances or variants of Pointing for different types of pointing (e.g. two or three axis rotation pointing).

5.4.2.42.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

- Failure of this component could cause the incorrect geolocation of an object that is subsequently targeted by weapons or misdirection of support to a weapon post release from the host air vehicle (e.g. laser designation). This could cause the weapon to strike a location not intended by the crew, resulting in unintended harm to third parties. This drives a DAL B indicative IDAL.
- Failure of this component could also cause turrets or other mechanical devices to move. This could cause harm to ground crew (expected to be no worse than major injury - i.e. critical severity). However, it is expected that the ability to move mechanical devices when ground crew would be at risk would be inhibited independently of this component (e.g. using the Interlocks component).

Where instances of this component contribute to hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.42.6.4 Security Considerations

The indicative security classification is O-S.

The component is responsible for the pointing of equipment. Without needing knowledge of the equipment’s capabilities, this component is considered to be O-S, although in some instances knowing the range of motion may drive a higher classification. Additionally, if the component needs own vehicle positioning data in order to calculate pointing angles etc. this would also drive a higher classification. The component may be in a higher classification security domain depending on the equipment in question. The integrity and availability of this component can have an impact on the combat effectiveness of the Exploiting Platform, e.g. unauthorised movement or the inability to direct a laser designator pod to point at the area of interest will prevent the accumulation of the required information. Integrity and availability will need to be protected according to the equipment being directed by the component.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to pointing commands during the mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.
- Performing **System Status and Monitoring** including feedback on requested and actual pointing direction, with deviation from the expected position being a possible sign of cyber activity.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.42.7 Services

5.4.2.42.7.1 Service Definitions

5.4.2.42.7.1.1 Orientation_Requirement

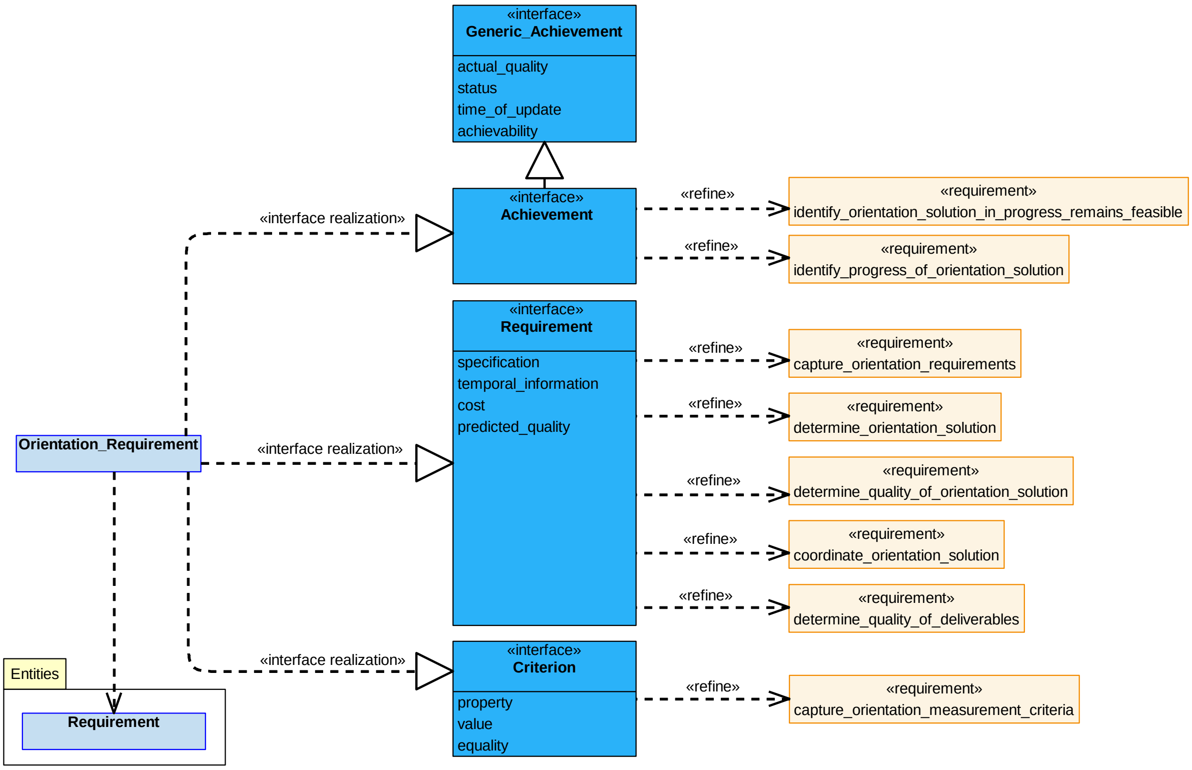


Figure 715: Orientation_Requirement Service Definition

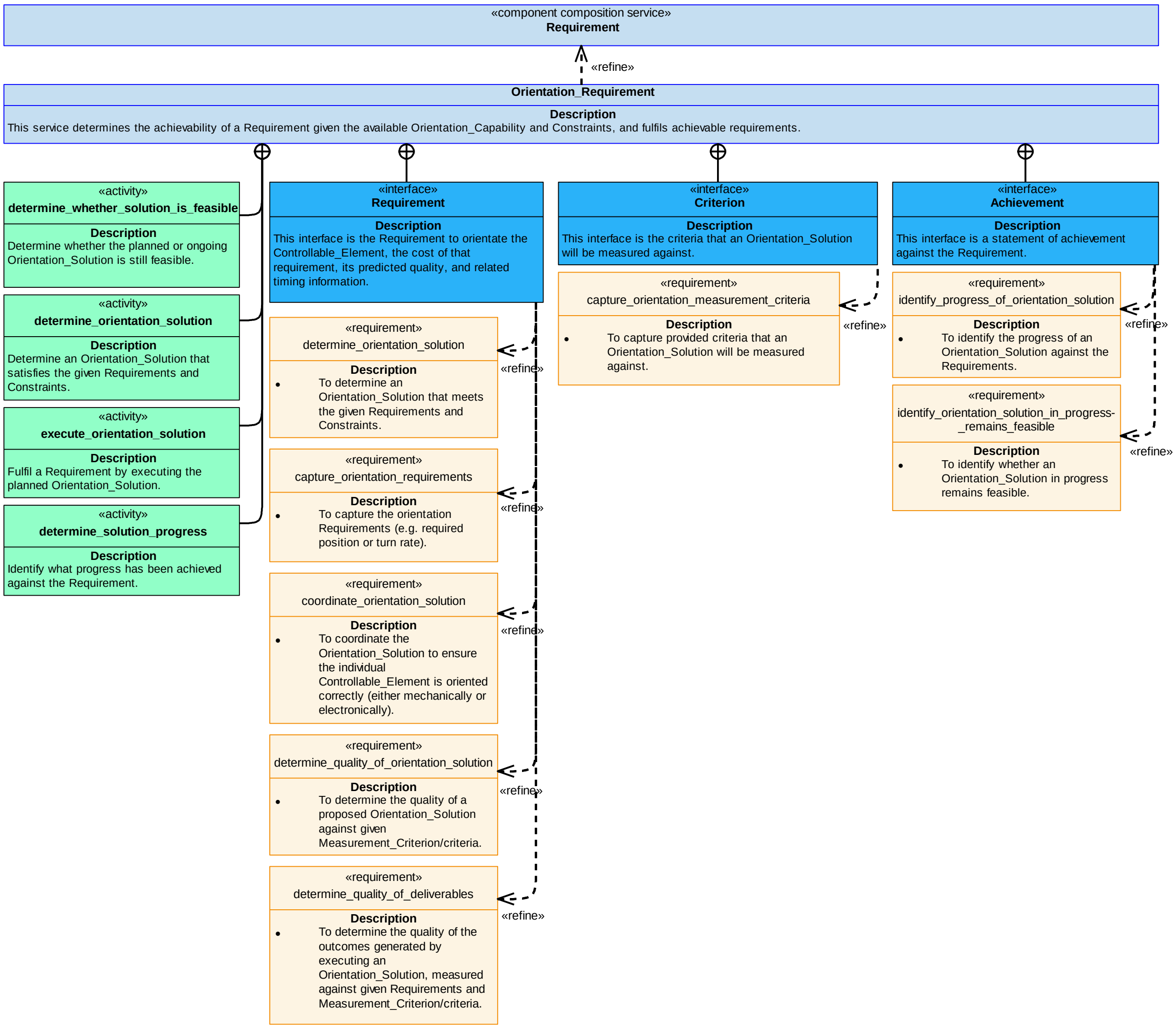

«component composition service»
Requirement
«refine»
Orientation_Requirement
Description
This service determines the achievability of a Requirement given the available Orientation_Capability and Constraints, and fulfils achievable requirements.
«activity»
determine_whether_solution_is_feasible
Description
Determine whether the planned or ongoing Orientation_Solution is still feasible.
«activity»
determine_orientation_solution
Description
Determine an Orientation_Solution that satisfies the given Requirements and Constraints.
«activity»
execute_orientation_solution
Description
Fulfil a Requirement by executing the planned Orientation_Solution.
«activity»
determine_solution_progress
Description
Identify what progress has been achieved against the Requirement.
«interface»
Requirement
Description
This interface is the Requirement to orientate the Controllable_Element, the cost of that requirement, its predicted quality, and related timing information.
«requirement»
determine_orientation_solution
Description
To determine an Orientation_Solution that meets the given Requirements and Constraints.
«refine»
«requirement»
capture_orientation_requirements
Description
To capture the orientation Requirements (e.g. required position or turn rate).
«refine»
«requirement»
coordinate_orientation_solution
Description
To coordinate the Orientation_Solution to ensure the individual Controllable_Element is oriented correctly (either mechanically or electronically).
«refine»
«requirement»
determine_quality_of_orientation_solution
Description
To determine the quality of a proposed Orientation_Solution against given Measurement_Criterion/criteria.
«refine»
«requirement»
determine_quality_of_deliverables
Description
To determine the quality of the outcomes generated by executing an Orientation_Solution, measured against given Requirements and Measurement_Criterion/criteria.
«refine»
«interface»
Criterion
Description
This interface is the criteria that an Orientation_Solution will be measured against.
«requirement»
capture_orientation_measurement_criteria
Description
To capture provided criteria that an Orientation_Solution will be measured against.
«refine»
«interface»
Achievement
Description
This interface is a statement of achievement against the Requirement.
«requirement»
identify_progress_of_orientation_solution
Description
To identify the progress of an Orientation_Solution against the Requirements.
«refine»
«requirement»
identify_orientation_solution_in_progress-_remains_feasible
Description
To identify whether an Orientation_Solution in progress remains feasible.
«refine»

Figure 716: Orientation_Requirement Service Policy

Orientation_Requirement

This service determines the achievability of a Requirement given the available Orientation_Capability and Constraints, and fulfils achievable requirements.

Interfaces

Requirement

This interface is the Requirement to orientate the Controllable_Element, the cost of that requirement, its predicted quality, and related timing information.

Attributes

specification	The definition of the Requirement, e.g. to keep a sensor pointing at a specific location or the required rate of travel.
temporal_information	Information covering timing, such as the orientation start and end times.
cost	The cost of pointing the Controllable_Element, e.g. resources expended.
predicted_quality	How well the Orientation_Solution is predicted to meet the Requirement.

Achievement

This interface is a statement of achievement against the Requirement.

Criterion

This interface is the criteria that an Orientation_Solution will be measured against.

Attributes

property	The property to be measured, e.g. the degree of accuracy in pointing at a specific location.
value	The measured value of the property, e.g. seconds of variation.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

determine_orientation_solution

Determine an Orientation_Solution that satisfies the given Requirements and Constraints.

determine_solution_progress

Identify what progress has been achieved against the Requirement.

execute_orientation_solution

Fulfil a Requirement by executing the planned Orientation_Solution.

determine_whether_solution_is_feasible

Determine whether the planned or ongoing Orientation_Solution is still feasible.

5.4.2.42.7.1.2 Element_Movement

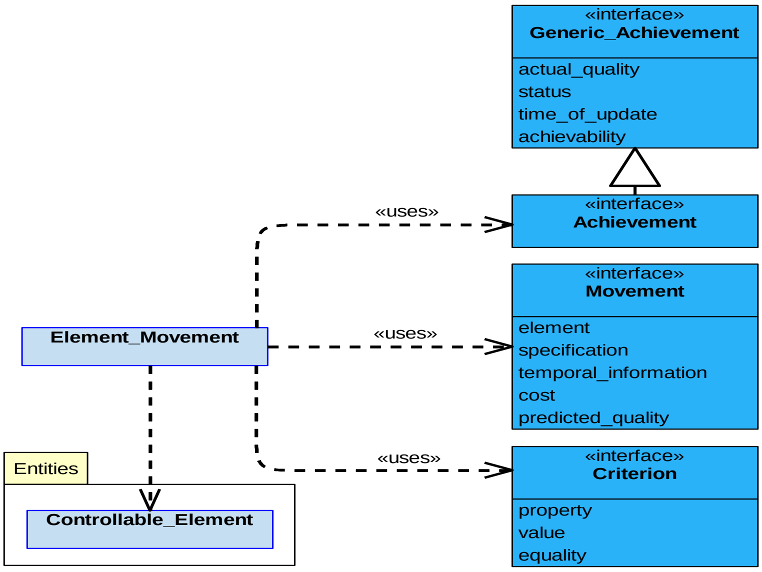


Figure 717: Element_Movement Service Definition

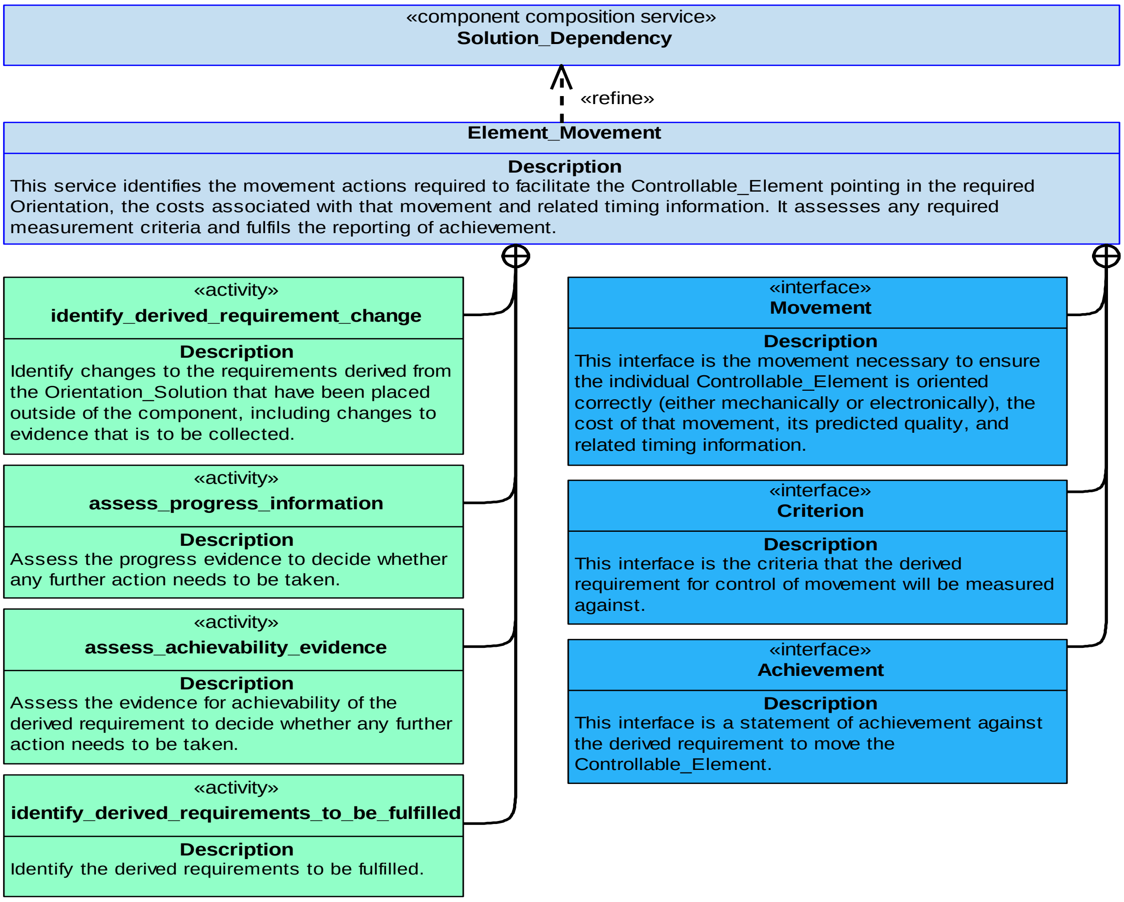

«component composition service»
Solution_Dependency
«refine»
Element_Movement
Description
This service identifies the movement actions required to facilitate the Controllable_Element pointing in the required Orientation, the costs associated with that movement and related timing information. It assesses any required measurement criteria and fulfils the reporting of achievement.
«activity»
identify_derived_requirement_change
Description
Identify changes to the requirements derived from the Orientation_Solution that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
assess_progress_information
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
assess_achievability_evidence
Description
Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.
«activity»
identify_derived_requirements_to_be_fulfilled
Description
Identify the derived requirements to be fulfilled.
«interface»
Movement
Description
This interface is the movement necessary to ensure the individual Controllable_Element is oriented correctly (either mechanically or electronically), the cost of that movement, its predicted quality, and related timing information.
«interface»
Criterion
Description
This interface is the criteria that the derived requirement for control of movement will be measured against.
«interface»
Achievement
Description
This interface is a statement of achievement against the derived requirement to move the Controllable_Element.

Figure 718: Element_Movement Service Policy

Element_Movement

This service identifies the movement actions required to facilitate the Controllable_Element pointing in the required Orientation, the costs associated with that movement and related timing information. It assesses any required measurement criteria and fulfils the reporting of achievement.

Interfaces

Movement

This interface is the movement necessary to ensure the individual Controllable_Element is oriented correctly (either mechanically or electronically), the cost of that movement, its predicted quality, and related timing information.

Attributes

element	The specific element being moved.
specification	The specification of the movement, e.g. the direction, amount of deflection or speed.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used.
predicted_quality	How well the planned movement is predicted to satisfy the requirement.

Achievement

This interface is a statement of achievement against the derived requirement to move the Controllable_Element.

Criterion

This interface is the criteria that the derived requirement for control of movement will be measured against.

Attributes

property	The property to be measured, e.g. the speed of rotation to achieve the desired orientation.
value	The measured value of the property, e.g. x radians per second.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

assess_progress_information

Assess the progress evidence to decide whether any further action needs to be taken.

identify_derived_requirements_to_be_fulfilled

Identify the derived requirements to be fulfilled.

identify_derived_requirement_change

Identify changes to the requirements derived from the Orientation_Solution that have been placed outside of the component, including changes to evidence that is to be collected.

assess_achievability_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

5.4.2.42.7.1.3 Orientation_Information

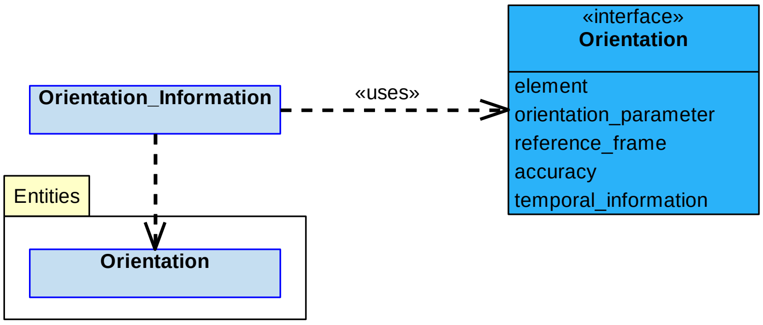


Figure 719: Orientation_Information Service Definition

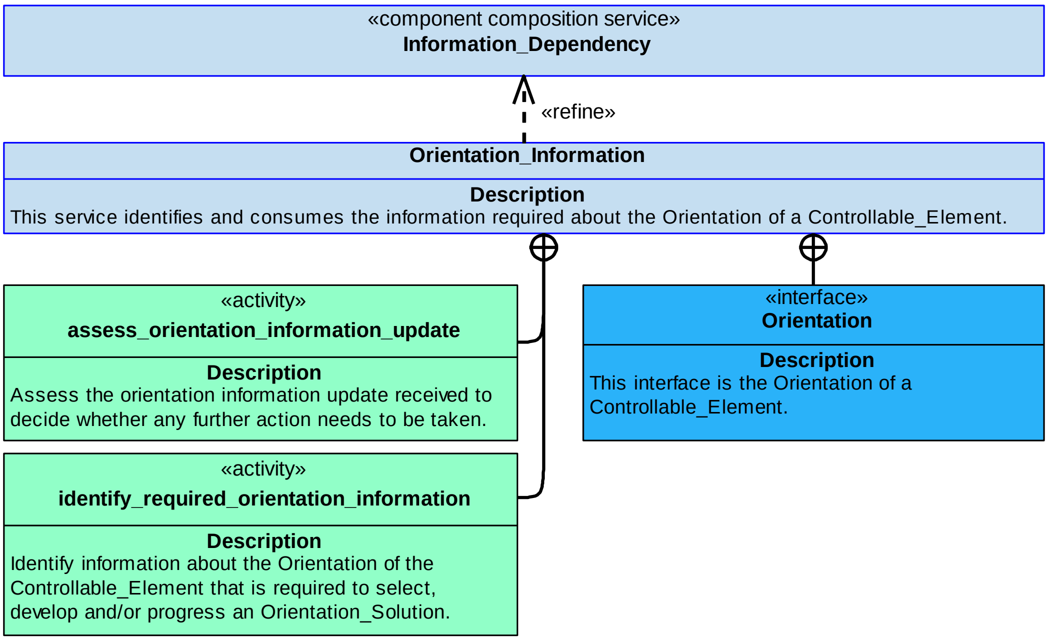

«component composition service»
Information_Dependency
«refine»
Orientation_Information
Description
This service identifies and consumes the information required about the Orientation of a Controllable_Element.
«activity»
assess_orientation_information_update
Description
Assess the orientation information update received to decide whether any further action needs to be taken.
«activity»
identify_required_orientation_information
Description
Identify information about the Orientation of the Controllable_Element that is required to select, develop and/or progress an Orientation_Solution.
«interface»
Orientation
Description
This interface is the Orientation of a Controllable_Element.

Figure 720: Orientation_Information Service Policy

Orientation_Information

This service identifies and consumes the information required about the Orientation of a Controllable_Element.

Interface

Orientation

This interface is the Orientation of a Controllable_Element.

Attributes

element	The specific element the information is about.
orientation_parameter	A parameter describing the Orientation of the Controllable_Element.
reference_frame	The reference frame applicable to the reported information.
accuracy	The level of accuracy in the reported information.
temporal_information	Information covering the timing of the information being reported.

Activities

identify_required_orientation_information

Identify information about the Orientation of the Controllable_Element that is required to select, develop and/or progress an Orientation_Solution.

assess_orientation_information_update

Assess the orientation information update received to decide whether any further action needs to be taken.

5.4.2.42.7.1.4 Contextual_Information

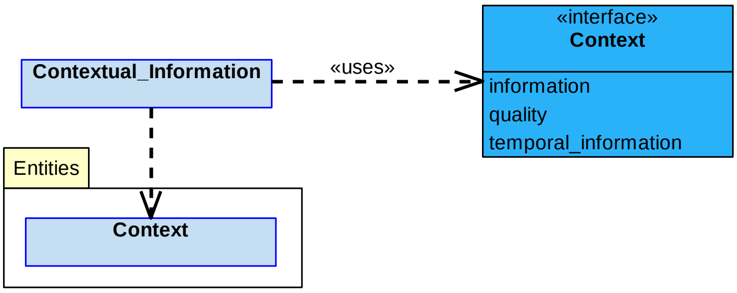


Figure 721: Contextual_Information Service Definition

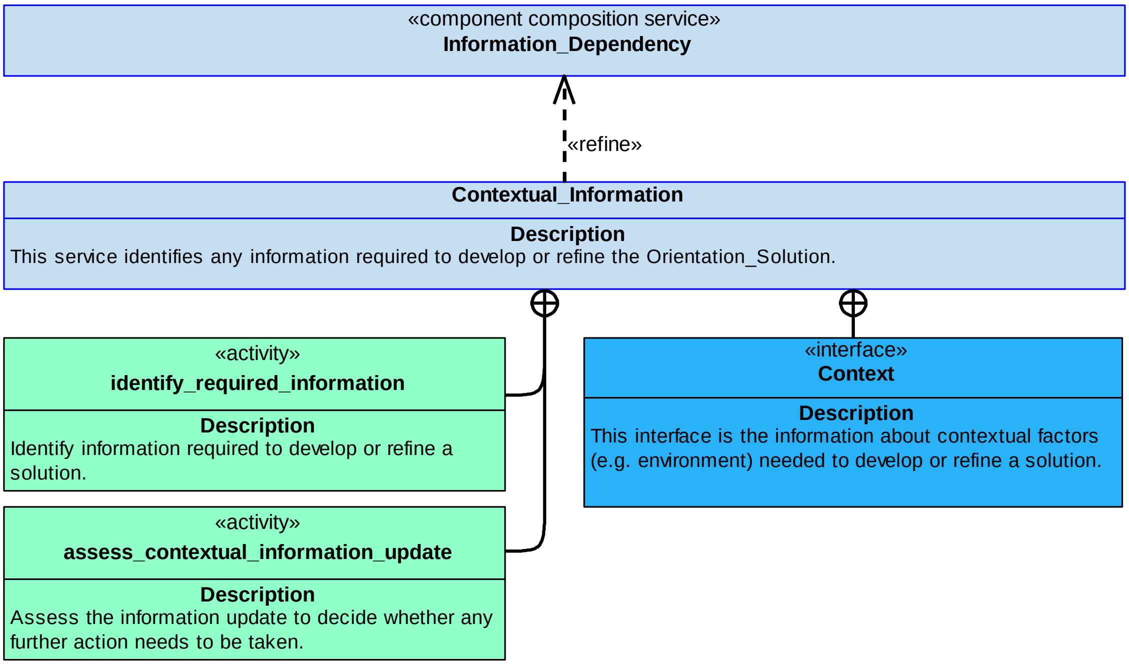

«component composition service»
Information_Dependency
«refine»
Contextual_Information
Description
This service identifies any information required to develop or refine the Orientation_Solution.
«activity»
identify_required_information
Description
Identify information required to develop or refine a solution.
«interface»
Context
Description
This interface is the information about contextual factors (e.g. environment) needed to develop or refine a solution.
«activity»
assess_contextual_information_update
Description
Assess the information update to decide whether any further action needs to be taken.

Figure 722: Contextual_Information Service Policy

Contextual_Information

This service identifies any information required to develop or refine the Orientation_Solution.

Interface

Context

This interface is the information about contextual factors (e.g. environment) needed to develop or refine a solution.

Attributes

information	The contextual information, e.g. speed or direction of ownship.
quality	The quality of the reported information.
temporal_information	The timing of the information being reported.

Activities

identify_required_information

Identify information required to develop or refine a solution.

assess_contextual_information_update

Assess the information update to decide whether any further action needs to be taken.

5.4.2.42.7.1.5 Constraint

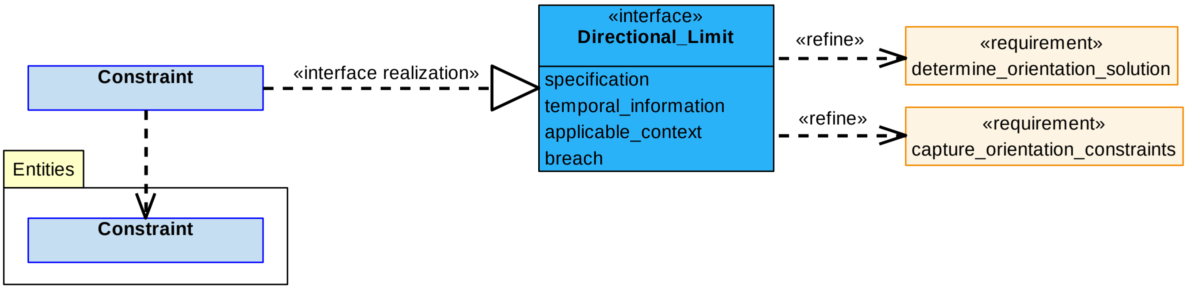


Figure 723: Constraint Service Definition

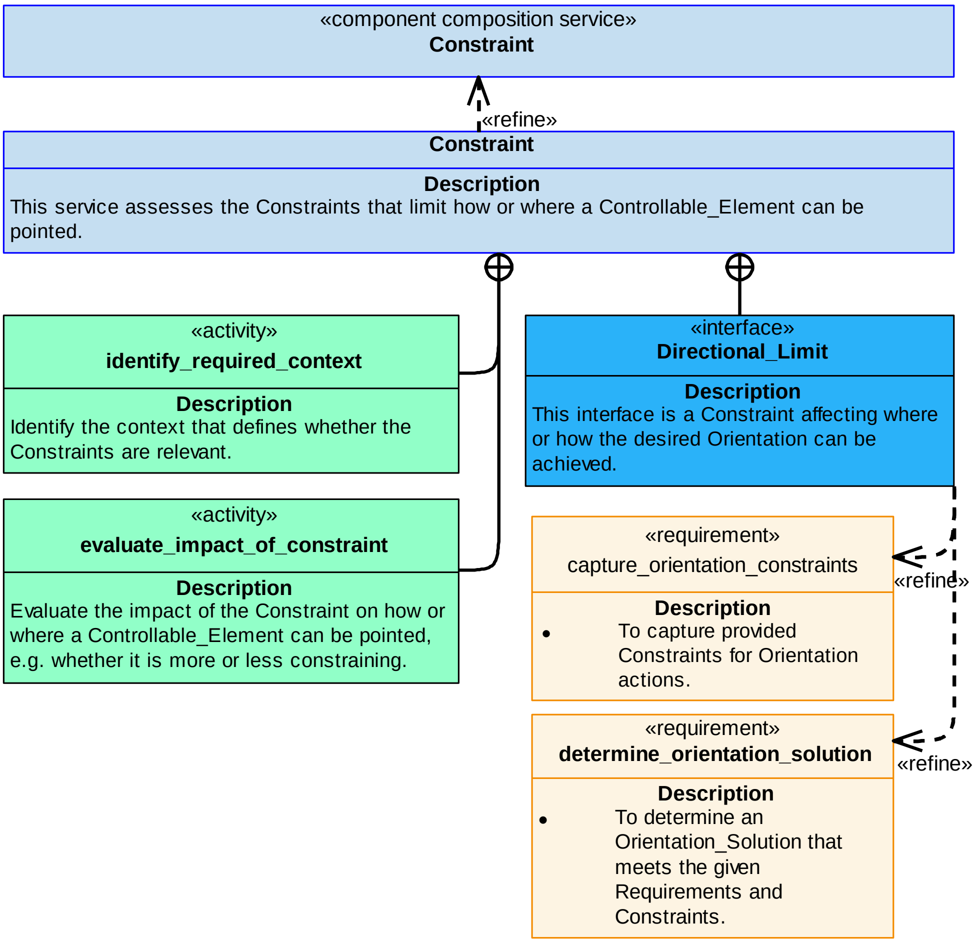

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Constraints that limit how or where a Controllable_Element can be pointed.
«activity»
identify_required_context
Description
Identify the context that defines whether the Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of the Constraint on how or where a Controllable_Element can be pointed, e.g. whether it is more or less constraining.
«interface»
Directional_Limit
Description
This interface is a Constraint affecting where or how the desired Orientation can be achieved.
«requirement»
capture_orientation_constraints
Description
To capture provided Constraints for Orientation actions.
«refine»
«requirement»
determine_orientation_solution
Description
To determine an Orientation_Solution that meets the given Requirements and Constraints.
«refine»

Figure 724: Constraint Service Policy

Constraint

This service assesses the Constraints that limit how or where a Controllable_Element can be pointed.

Interface

Directional_Limit

This interface is a Constraint affecting where or how the desired Orientation can be achieved.

Attributes

specification	The detail of the limit.
temporal_information	Timing information on when the limit is applicable, e.g. start and end times.
applicable_context	The context within which the limit is applicable.
breach	A statement that the limit has been breached.

Activities

identify_required_context

Identify the context that defines whether the Constraints are relevant.

evaluate_impact_of_constraint

Evaluate the impact of the Constraint on how or where a Controllable_Element can be pointed, e.g. whether it is more or less constraining.

5.4.2.42.7.1.6 Capability

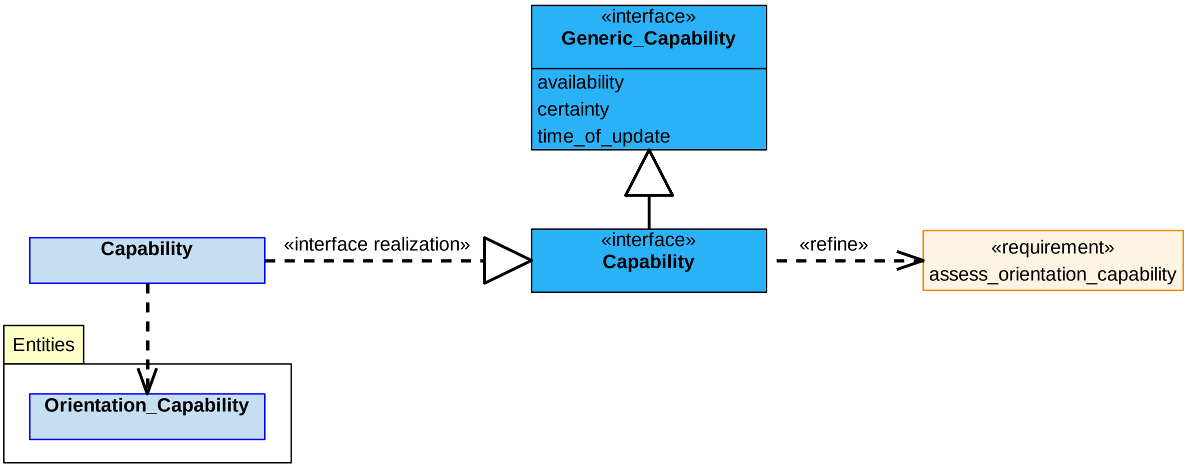


Figure 725: Capability Service Definition

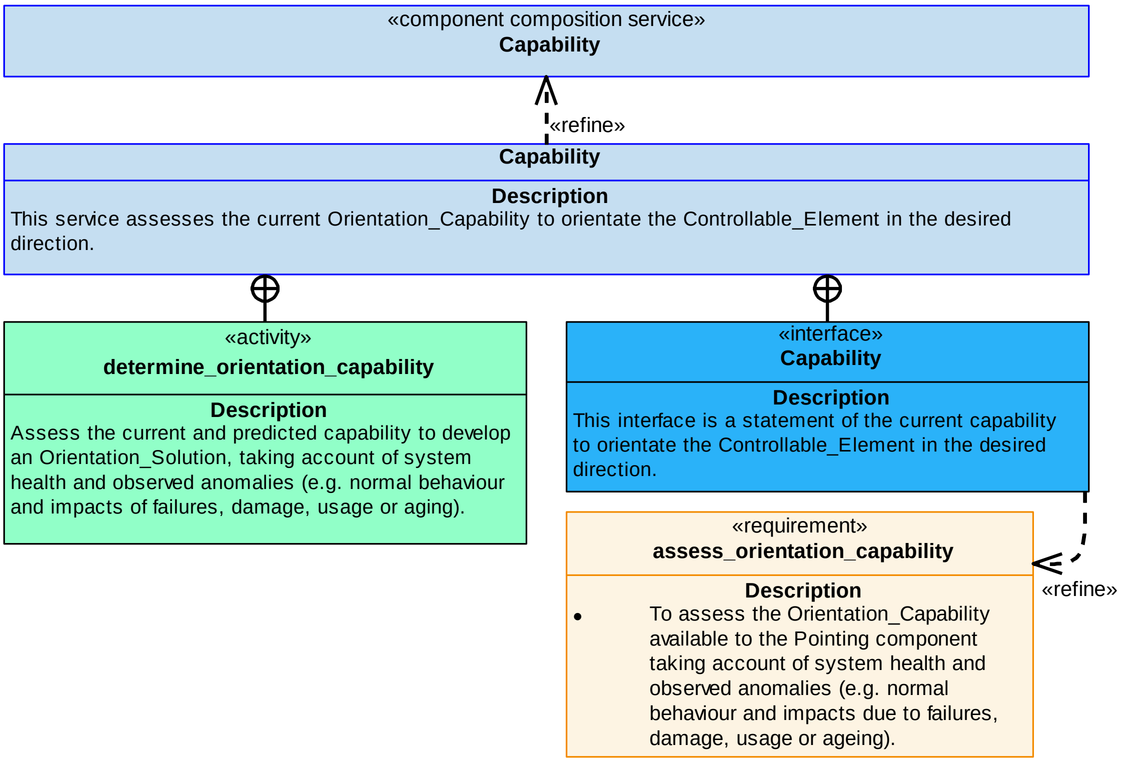

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current Orientation_Capability to orientate the Controllable_Element in the desired direction.
«activity»
determine_orientation_capability
Description
Assess the current and predicted capability to develop an Orientation_Solution, taking account of system health and observed anomalies (e.g. normal behaviour and impacts of failures, damage, usage or aging).
«interface»
Capability
Description
This interface is a statement of the current capability to orientate the Controllable_Element in the desired direction.
«requirement»
assess_orientation_capability
Description
To assess the Orientation_Capability available to the Pointing component taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 726: Capability Service Policy

Capability

This service assesses the current Orientation_Capability to orientate the Controllable_Element in the desired direction.

Interface

Capability

This interface is a statement of the current capability to orientate the Controllable_Element in the desired direction.

Activity

determine_orientation_capability

Assess the current and predicted capability to develop an Orientation_Solution, taking account of system health and observed anomalies (e.g. normal behaviour and impacts of failures, damage, usage or aging).

5.4.2.42.7.1.7 Capability_Evidence

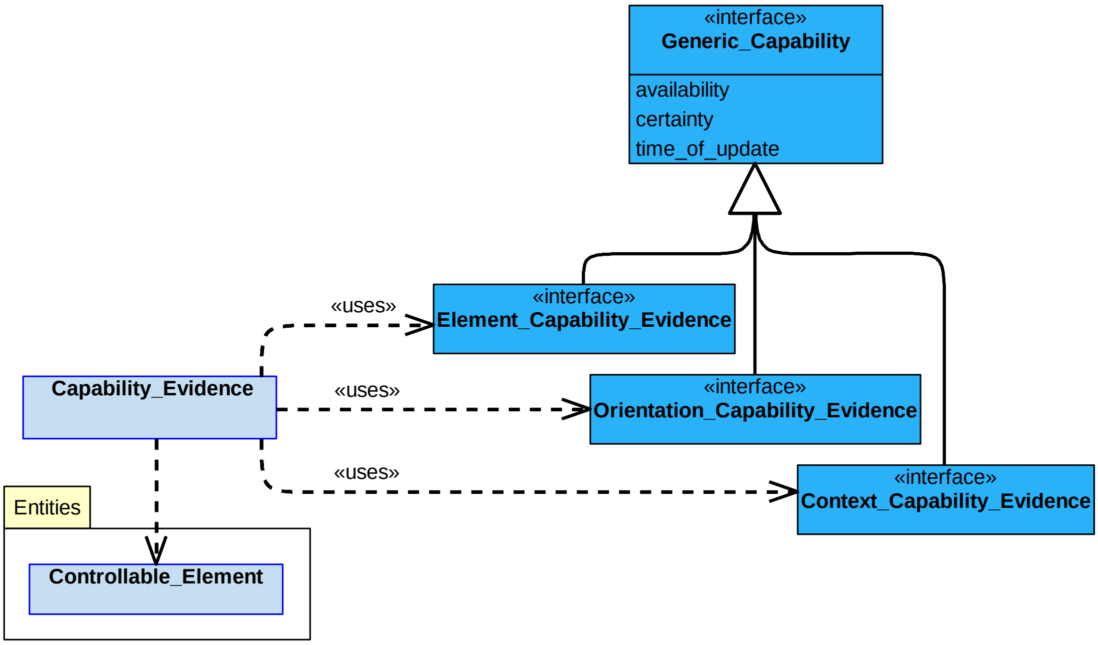
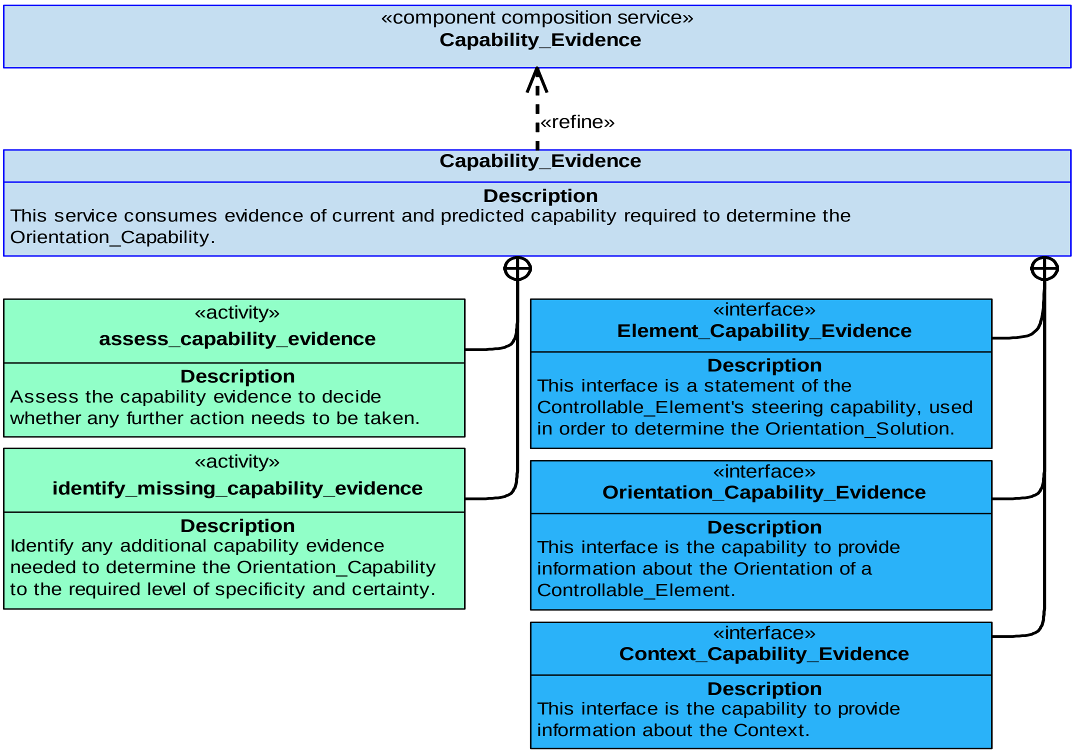


Figure 727: Capability_Evidence Service Definition


«component composition service»
Capability_Evidence
«refine»
Capability_Evidence

Description
This service consumes evidence of current and predicted capability required to determine the Orientation_Capability.

«activity»
assess_capability_evidence

Description
Assess the capability evidence to decide whether any further action needs to be taken.

«activity»
identify_missing_capability_evidence

Description
Identify any additional capability evidence needed to determine the Orientation_Capability to the required level of specificity and certainty.

Description
This interface is a statement of the Controllable_Element's steering capability, used in order to determine the Orientation_Solution.

Description
This interface is the capability to provide information about the Orientation of a Controllable_Element.

Description
This interface is the capability to provide information about the Context.

Figure 728: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence of current and predicted capability required to determine the Orientation_Capability.

Interfaces

Element_Capability_Evidence

This interface is a statement of the Controllable_Element's steering capability, used in order to determine the Orientation_Solution.

Orientation_Capability_Evidence

This interface is the capability to provide information about the Orientation of a Controllable_Element.

Context_Capability_Evidence

This interface is the capability to provide information about the Context.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any additional capability evidence needed to determine the Orientation_Capability to the required level of specificity and certainty.

5.4.2.42.7.2 Service Dependencies

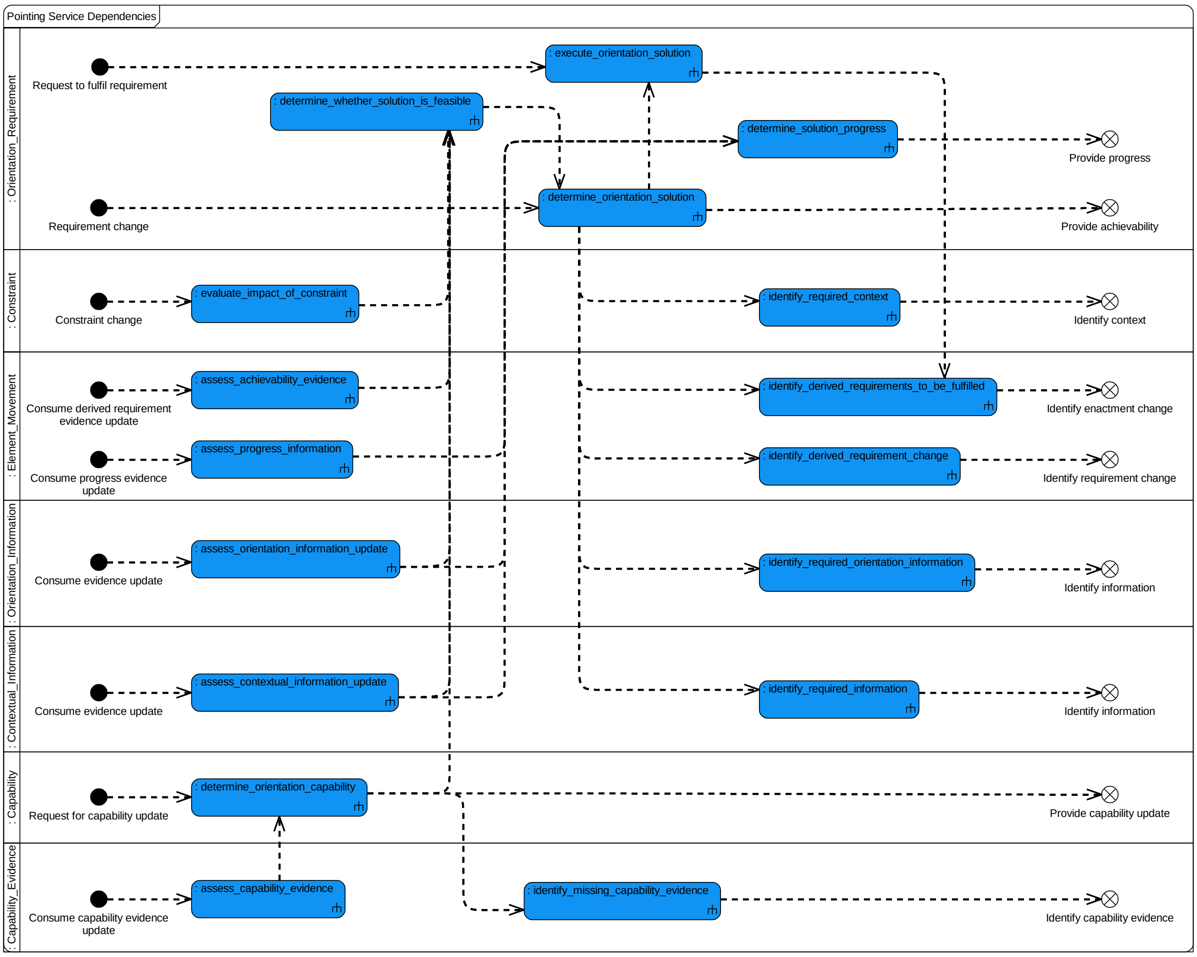

Pointing Service Dependencies
: Orientation_Requirement
Request to fulfil requirement
: execute_orientation_solution
: determine_whether_solution_is_feasible
: determine_solution_progress
Provide progress
Requirement change
: determine_orientation_solution
Provide achievability
: Constraint
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
: Element_Movement
Consume derived requirement evidence update
: assess_achievability_evidence
: identify_derived_requirements_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_progress_information
: identify_derived_requirement_change
Identify requirement change
: Orientation_Information
Consume evidence update
: assess_orientation_information_update
: identify_required_orientation_information
Identify information
: Contextual_Information
Consume evidence update
: assess_contextual_information_update
: identify_required_information
Identify information
: Capability
Request for capability update
: determine_orientation_capability
Provide capability update
: Capability_Evidence
Consume capability evidence update
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 729: Pointing Service Dependencies

5.4.2.43 Power

5.4.2.43.1 Role

The role of Power is to ensure the availability of power by managing power sources and controlling distribution to power sinks that require it.

5.4.2.43.2 Overview

Control Architecture

Power is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

The Power component gathers the requirements for power from the Power_Sinks on the system, identifies the available resource from the Power_Sources and manages the subsequent distribution across the Grid in order to satisfy Equipment_Constraint.

Examples of Use

Power will be used where:

- Management of available power resources (e.g. electrical, pneumatic or hydraulic) and the distribution grid is required to meet the varying demands of a number of consumers.

5.4.2.43.3 Service Summary

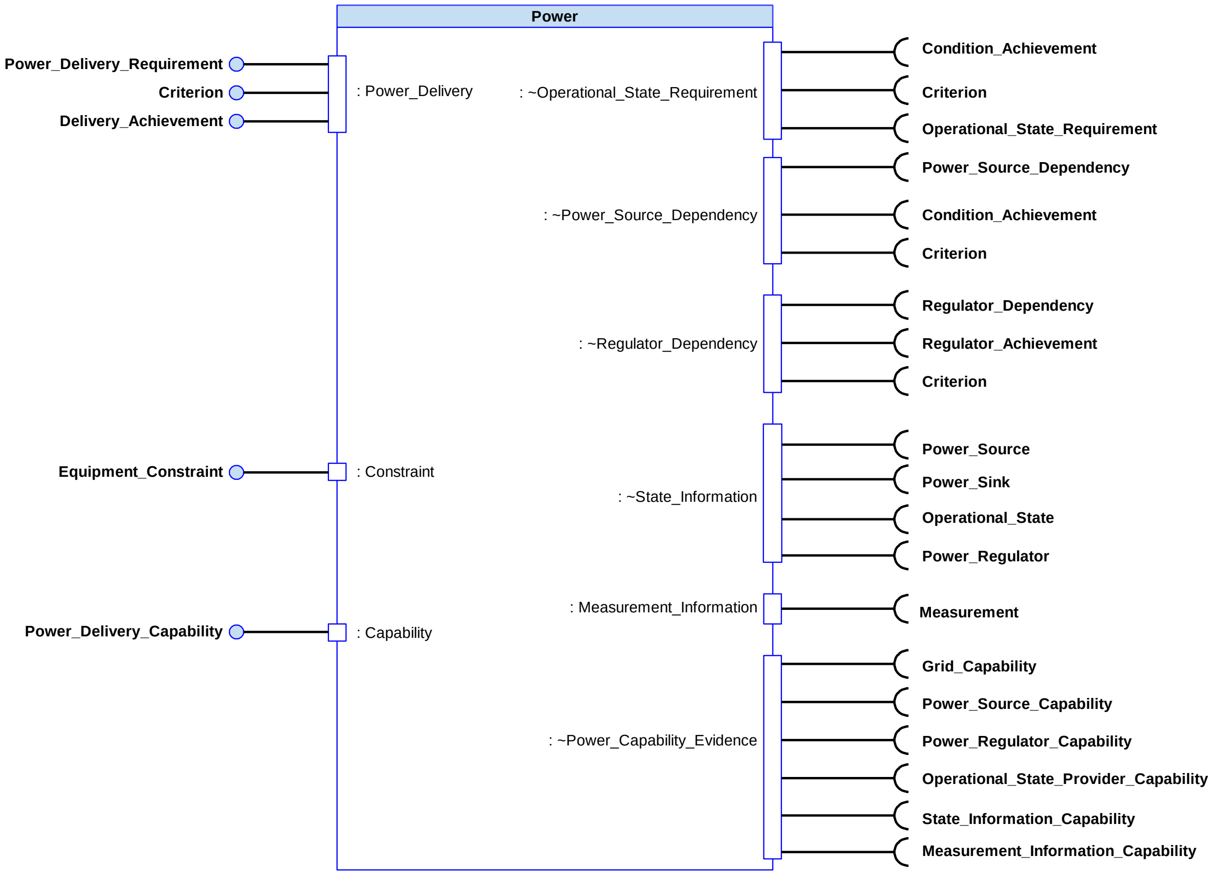


Figure 730: Power Service Summary

5.4.2.43.4 Responsibilities

capture_power_solution_requirements

- To capture Power_Requirements for Power_Solutions.

capture_measurement_criteria_for_power_solution

- To capture provided Measurement_Criterion/criteria for power solutions.

capture_power_solution_constraints

- To capture provided Equipment_Constraints for use of power, such as the restriction of the use of a specific Power_Source.

identify_whether_power_requirement_remains_achievable

- To identify if a Power_Requirement is still achievable given current resources.

determine_power_solution

- To determine the appropriate Power_Solution to meet captured Power_Requirements and Equipment_Constraints.

determine_quality_of_designed_solution

- To determine the Designed_Quality of the Power_Solution against one or more given Measurement_Criterion.

coordinate_power_resources

- To coordinate the use of Power_Sources, Power_Sinks and Power_Regulators to meet the requirements of a Power_Solution.

identify_progress_of_power_solution

- To identify the progress of a Power_Solution against the Power_Requirements.

determine_quality_of_delivered_solution

- To determine the Delivered_Quality of the delivered Power_Solution measured against Power_Requirement in terms of given Measurement_Criterion.

assess_power_capability

- To assess the current capability to manage Power_Sources, Power_Sinks and Power_Regulators.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the power capability assessment.

predict_power_capability_progression

- To predict the progression of power capability over time and with use.

5.4.2.43.5 Subject Matter Semantics

The subject matter of Power is the resources that can be used to coordinate and distribute power from resources (e.g. electrical, pneumatic or hydraulic power) to fulfil the needs of those that require it.

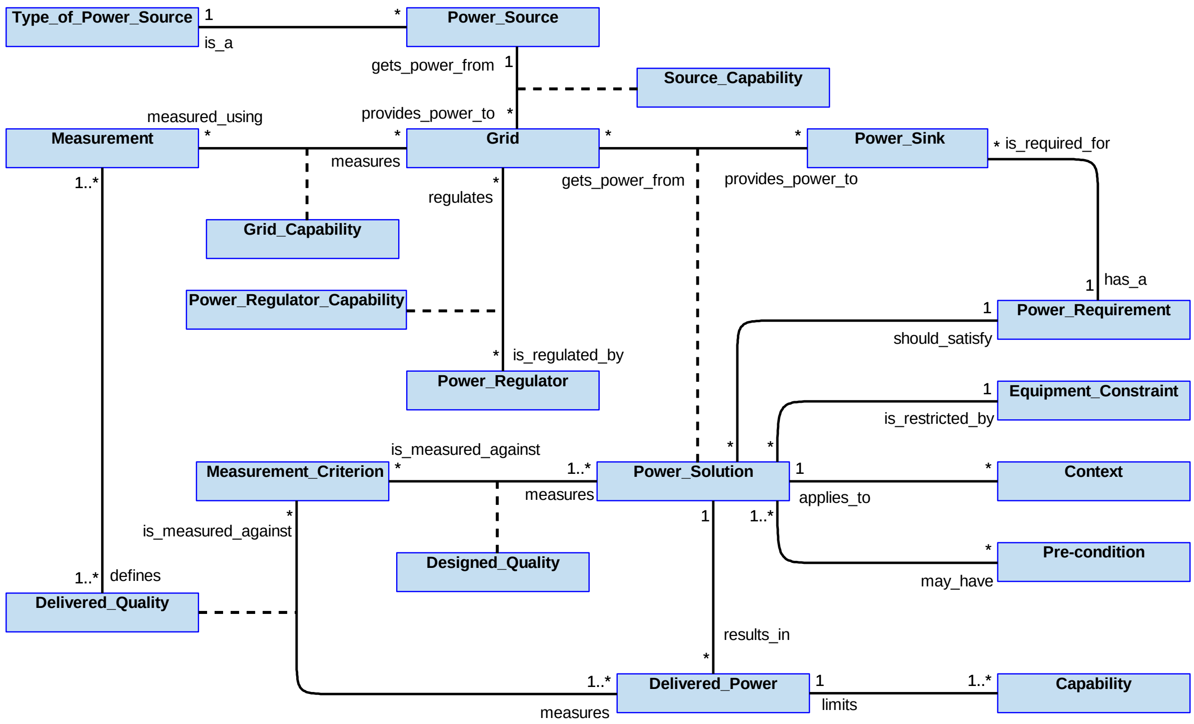


Figure 731: Power Semantics

5.4.2.43.5.1 Entities

Capability

The capability to provide the required power to a Power_Sink.

Context

The situation within which a Power_Solution is being derived.

Delivered_Power

The power that is currently being delivered.

Delivered_Quality

A measure, against a given Measurement_Criterion, of how well the Delivered_Power meets the requirements.

Designed_Quality

A measure, against a given Measurement_Criterion, of how well the Power_Solution meets the requirements.

Equipment_Constraint

A constraint on the way that the solution may provide its capability, e.g. equipment limitations.

Grid

A network over which power can be distributed. This includes network infrastructure, e.g. filters or wiring.

Grid_Capability

The capability to support power flow.

Measurement

A measurement of a property of the Grid (e.g. current or voltage).

Measurement_Criterion

A criterion that needs to be evaluated when determining if a solution satisfies the requirements (e.g. voltage, in-rush current, back EMF or frequency).

Power_Solution

The solution to providing Power_Sinks with power from the Grid.

Power_Regulator

Regulator devices from switches, power converters, and RCDs to more complex devices.

Power_Regulator_Capability

The capability of a Power_Regulator to support power levels on a segment of Grid.

Power_Requirement

The power demand profile specification, required in order that a Power_Sink might fulfil its desired operations (e.g. base power vs dynamic power).

Power_Sink

Any element that consumes power.

Power_Source

Any element that provides power.

Pre-condition

A condition that must be satisfied outside this component, e.g. propulsion system demands to meet the needs of required power generation.

Source_Capability

The Power Profile that can be provided from the Power_Source.

Type_of_Power_Source

The type of element that is providing the energy from conversions (e.g. electrical, hydraulic or pneumatic power).

5.4.2.43.6 Design Rationale

5.4.2.43.6.1 Assumptions

- Power for an Exploiting Platform is provided by a Grid providing the resource from one or more Power_Sources to one or more Power_Sinks.

- Having knowledge of power consumption may provide a means of estimating the performance of a system drawing power. It may also provide knowledge of the current capability of a system, for example if a system is powered or can be powered.

5.4.2.43.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Power:

- Resource Management - As it applies to the pooled power resource.
- Data Driving - The capabilities and requirements of the possible Power_Sources and permitted Power_Sinks may be data-driven.

Extensions

- The use of extension components for Power may be appropriate to accommodate some aspects of the different resources (e.g. electrical, pneumatic or hydraulic power) the component may be used for. The capabilities and requirements of the Power_Sources and Power_Sinks are more likely to be handled by data driving as these are considered more likely to change over time.

Exploitation Considerations

- Some elements may change between a Power_Source and Power_Sink at different points in time (e.g. a battery is a sink when charging and a source when providing electrical power).
- Power will be aware of the power requirements of Power_Sinks in different operating modes, although it is not aware of what these different modes represent other than a varying power demand. These requirements could vary as equipment evolves over time, the use of data driving should be considered to accommodate this variation.
- Power will be aware of the power that can be produced from a Power_Source in different operating modes, although it is not aware of what these different modes represent other than increased or decreased power availability. This performance could vary as equipment evolves over time, the use of data driving should be considered to accommodate this variation.
- It is expected that not all Power_Sources (and possibly Power_Regulators) will be directly controlled by this component. Where they are not directly controlled by Power, Solution_Dependency services will be required.

5.4.2.43.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component could fail to distribute available power to critical equipment. In the case of an air vehicle, loss of power to flight equipment would result in an uncontrolled crash, the result of which is likely to be loss of the air vehicle and fatalities.

- The component may also cause equipment to be powered-up when it is not safe (e.g. during maintenance) or to fail to remove power from Power_Sinks that are in a dangerous state (e.g. in response to safety warnings), potentially causing harm to ground crew.

5.4.2.43.6.4 Security Considerations

The indicative security classification is O-S.

This component is responsible for the control of power to parts of the Exploiting Platform, containing a virtual mapping of Power_Sources and Power_Sinks without knowing the purpose of those sinks; this is expected to drive an indicative security classification of O-S. Where the power requirements may indicate use of specific equipment or reveal potential capabilities or performance, there may need to be a greater degree of confidentiality assigned. If the integrity of demands for, and availability of power is compromised, the combat effectiveness of the Exploiting Platform may be reduced, e.g. through loss of available power to sensors or weapons. The component is considered a legitimate target for cyber attack and due to the risk to integrity and availability, appropriate protection is required. This is one of a series of components that will assist in identifying if form and fit integrity has been interfered with.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Information** relating to possible tamper events.
- **Maintaining Audit Records** to support accountability of power usage.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** of the demanded power against that supplied, identifying unexpected power requests, etc.
- Providing **Warnings and Notifications** of power loss, etc.

The component is involved in satisfying security enforcing functions relating to:

- **Detecting Security Breaches** through identifying conditioning states that may indicate the physical security has been compromised (e.g. an item is drawing excessive power).

5.4.2.43.7 Services

5.4.2.43.7.1 Service Definitions

5.4.2.43.7.1.1 Power_Delivery

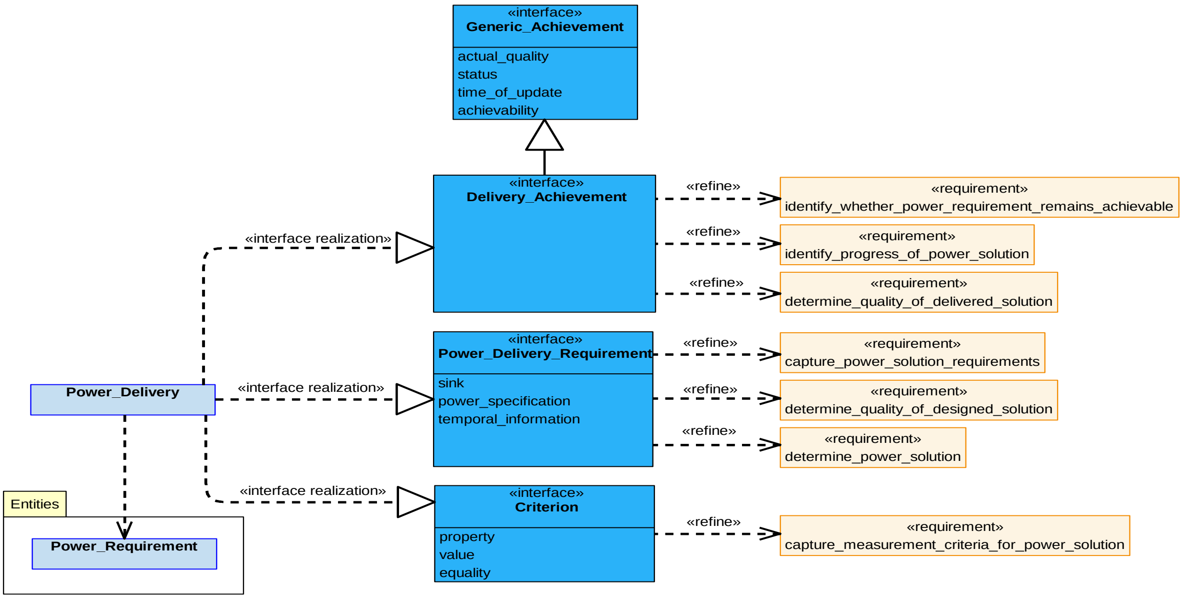
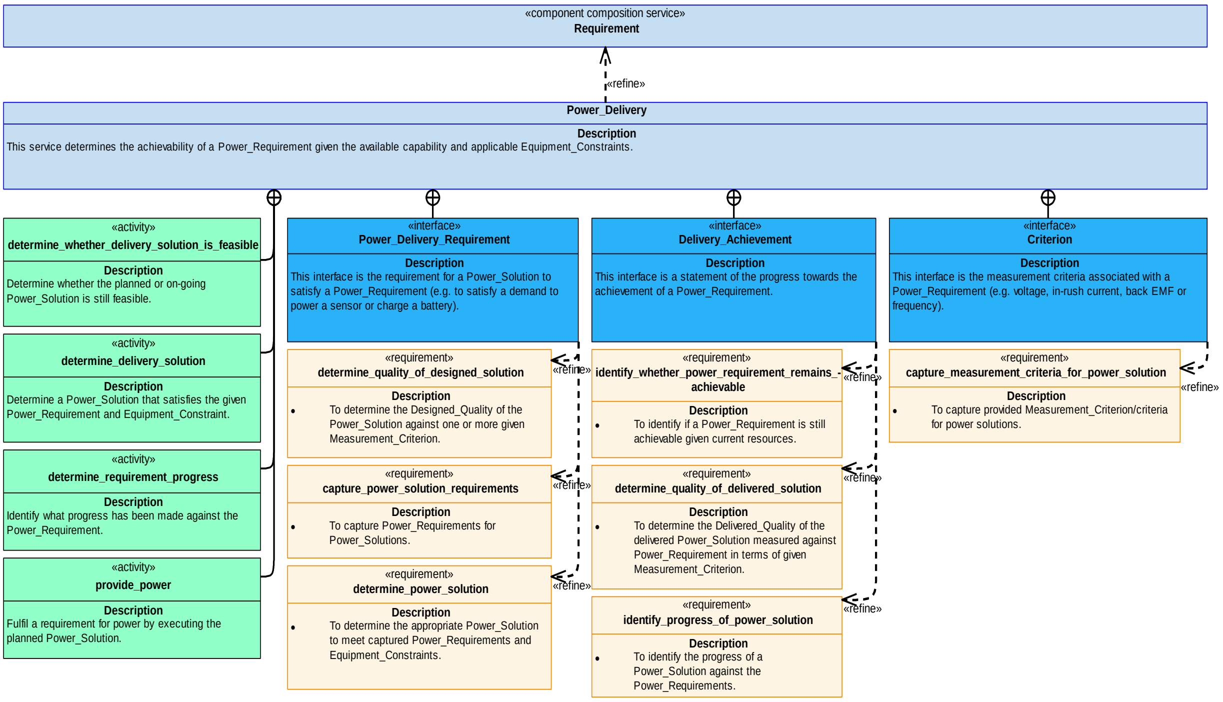


Figure 732: Power_Delivery Service Definition



Power_Delivery

Description
This service determines the achievability of a Power_Requirement given the available capability and applicable Equipment_Constraints.

determine_whether_delivery_solution_is_feasible

Description
Determine whether the planned or on-going Power_Solution is still feasible.

determine_delivery_solution

Description
Determine a Power_Solution that satisfies the given Power_Requirement and Equipment_Constraint.

determine_requirement_progress

Description
Identify what progress has been made against the Power_Requirement.

provide_power

Description
Fulfil a requirement for power by executing the planned Power_Solution.

Power_Delivery_Requirement

Description
This interface is the requirement for a Power_Solution to satisfy a Power_Requirement (e.g. to satisfy a demand to power a sensor or charge a battery).

determine_quality_of_designed_solution

Description
- To determine the Designed_Quality of the Power_Solution against one or more given Measurement_Criterion.

capture_power_solution_requirements

Description
- To capture Power_Requirements for Power_Solutions.

determine_power_solution

Description
- To determine the appropriate Power_Solution to meet captured Power_Requirements and Equipment_Constraints.

Delivery_Achievement

Description
This interface is a statement of the progress towards the achievement of a Power_Requirement.

identify_whether_power_requirement_remains_-achievable

Description
- To identify if a Power_Requirement is still achievable given current resources.

determine_quality_of_delivered_solution

Description
- To determine the Delivered_Quality of the delivered Power_Solution measured against Power_Requirement in terms of given Measurement_Criterion.

identify_progress_of_power_solution

Description
- To identify the progress of a Power_Solution against the Power_Requirements.

Criterion

Description
This interface is the measurement criteria associated with a Power_Requirement (e.g. voltage, in-rush current, back EMF or frequency).

capture_measurement_criteria_for_power_solution

Description
- To capture provided Measurement_Criterion/criteria for power solutions.

Figure 733: Power_Delivery Service Policy

Power_Delivery

This service determines the achievability of a Power_Requirement given the available capability and applicable Equipment_Constraints.

Interfaces

Power_Delivery_Requirement

This interface is the requirement for a Power_Solution to satisfy a Power_Requirement (e.g. to satisfy a demand to power a sensor or charge a battery).

Attributes

sink	The Power_Sink that will be satisfied by the Power_Solution.
power_specification	The definition of the Power_Requirement (e.g. to provide a base level of power to a piece of equipment, or ensure that a battery remains charged).
temporal_information	Information covering timing, such as when and for how long the power is required.

Criterion

This interface is the measurement criteria associated with a Power_Requirement (e.g. voltage, in-rush current, back EMF or frequency).

Attributes

property	The property to be measured, such as the voltage.
value	The measured value of the property, e.g. 240V.
equality	The relationship between the value and any limit on the property, e.g. less than, or equal to.

Delivery_Achievement

This interface is a statement of the progress towards the achievement of a Power_Requirement.

Activities

provide_power

Fulfil a requirement for power by executing the planned Power_Solution.

determine_delivery_solution

Determine a Power_Solution that satisfies the given Power_Requirement and Equipment_Constraint.

determine_whether_delivery_solution_is_feasible

Determine whether the planned or on-going Power_Solution is still feasible.

determine_requirement_progress

Identify what progress has been made against the Power_Requirement.

5.4.2.43.7.1.2 Operational_State_Requirement

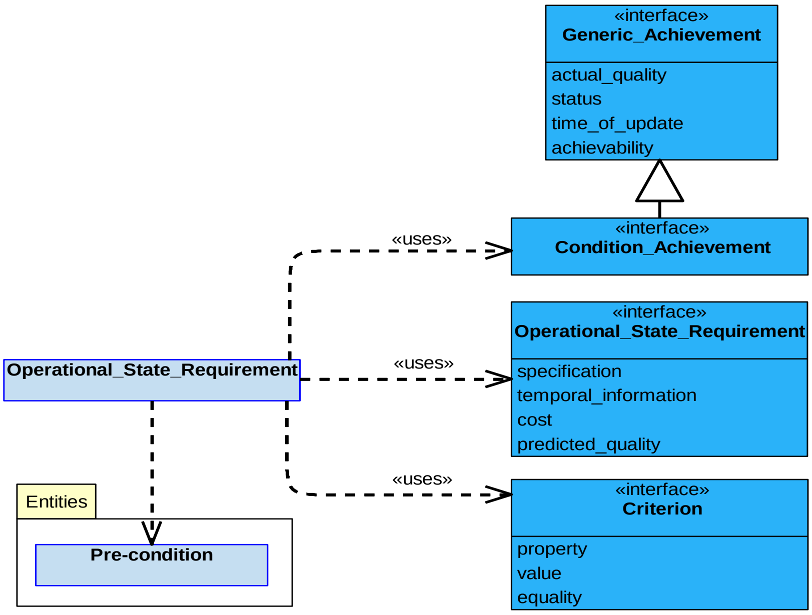


Figure 734: Operational_State_Requirement Service Definition

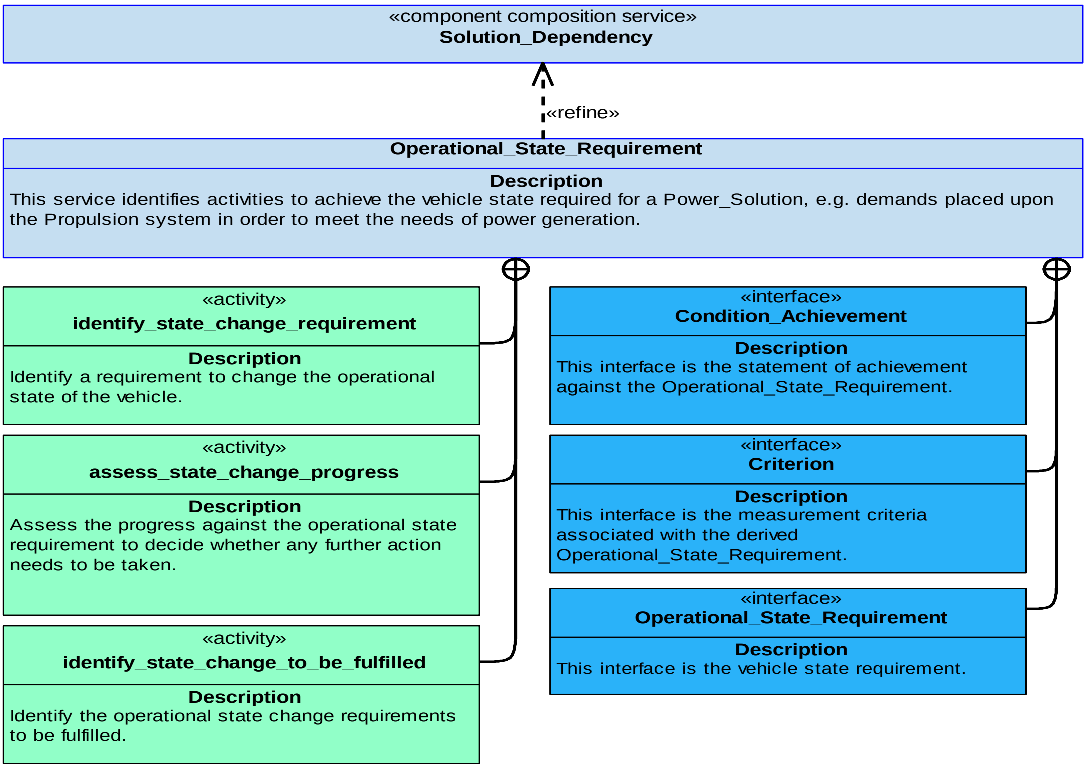

«component composition service»
Solution_Dependency
«refine»
Operational_State_Requirement
Description
This service identifies activities to achieve the vehicle state required for a Power_Solution, e.g. demands placed upon the Propulsion system in order to meet the needs of power generation.
«activity»
identify_state_change_requirement
Description
Identify a requirement to change the operational state of the vehicle.
«activity»
assess_state_change_progress
Description
Assess the progress against the operational state requirement to decide whether any further action needs to be taken.
«activity»
identify_state_change_to_be_fulfilled
Description
Identify the operational state change requirements to be fulfilled.
«interface»
Condition_Achievement
Description
This interface is the statement of achievement against the Operational_State_Requirement.
«interface»
Criterion
Description
This interface is the measurement criteria associated with the derived Operational_State_Requirement.
«interface»
Operational_State_Requirement
Description
This interface is the vehicle state requirement.

Figure 735: Operational_State_Requirement Service Policy

Operational_State_Requirement

This service identifies activities to achieve the vehicle state required for a Power_Solution, e.g. demands placed upon the Propulsion system in order to meet the needs of power generation.

Interfaces

Operational_State_Requirement

This interface is the vehicle state requirement.

Attributes

specification	The required operational state.
temporal_information	Information covering timing, such as start and end times of the derived requirement.
cost	The cost of executing the solution, e.g. resources used or time taken.
predicted_quality	How well the proposed operational state solution is predicted to satisfy the requirement.

Condition_Achievement

This interface is the statement of achievement against the Operational_State_Requirement.

Criterion

This interface is the measurement criteria associated with the derived Operational_State_Requirement.

Attributes

property	The property of the operational state to be measured.
value	The value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

identify_state_change_to_be_fulfilled

Identify the operational state change requirements to be fulfilled.

assess_state_change_progress

Assess the progress against the operational state requirement to decide whether any further action needs to be taken.

identify_state_change_requirement

Identify a requirement to change the operational state of the vehicle.

5.4.2.43.7.1.3 Power_Source_Dependency

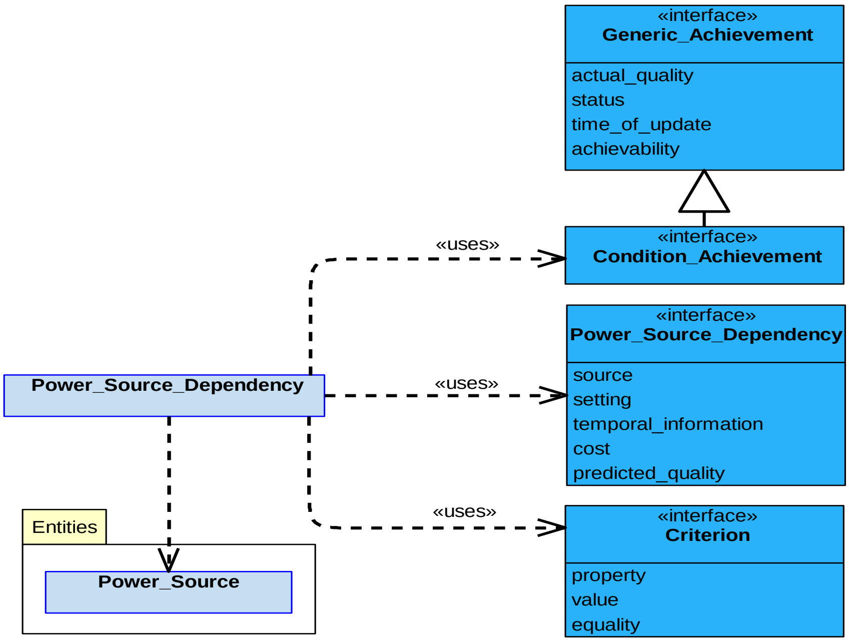


Figure 736: Power_Source_Dependency Service Definition

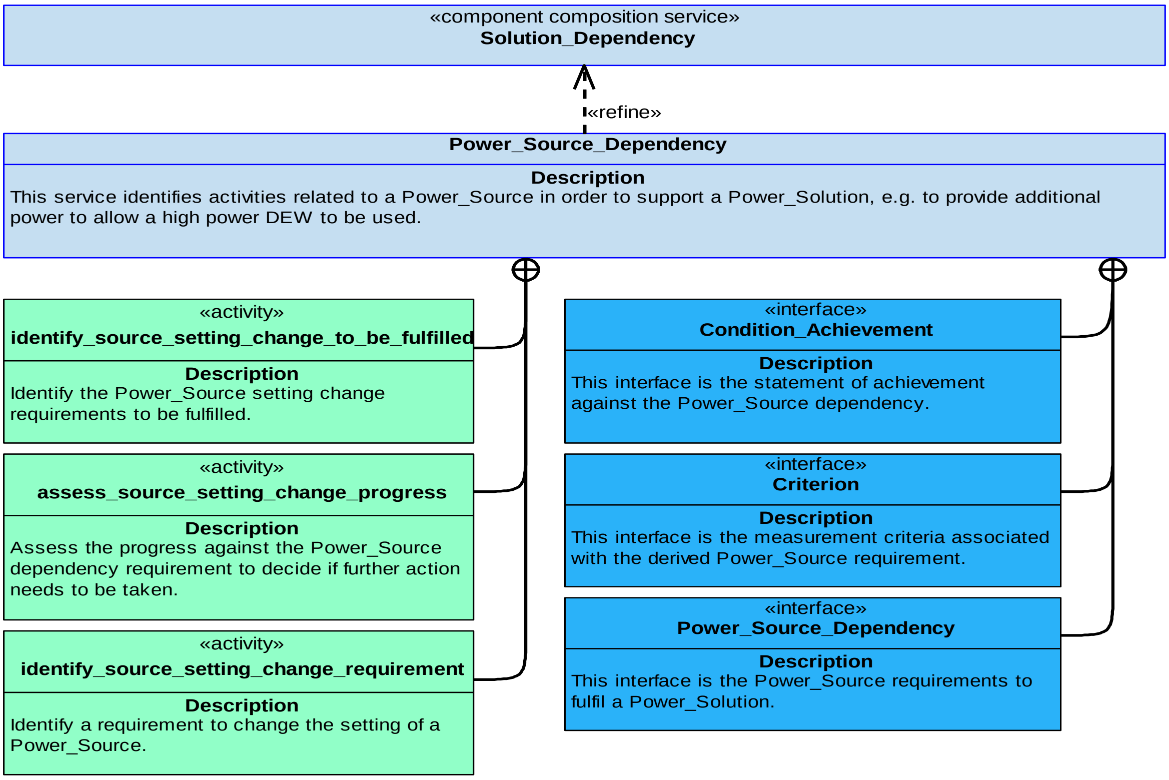

«component composition service»
Solution_Dependency
«refine»
Power_Source_Dependency
Description
This service identifies activities related to a Power_Source in order to support a Power_Solution, e.g. to provide additional power to allow a high power DEW to be used.
«activity»
identify_source_setting_change_to_be_fulfilled
Description
Identify the Power_Source setting change requirements to be fulfilled.
«activity»
assess_source_setting_change_progress
Description
Assess the progress against the Power_Source dependency requirement to decide if further action needs to be taken.
«activity»
identify_source_setting_change_requirement
Description
Identify a requirement to change the setting of a Power_Source.
«interface»
Condition_Achievement
Description
This interface is the statement of achievement against the Power_Source dependency.
«interface»
Criterion
Description
This interface is the measurement criteria associated with the derived Power_Source requirement.
«interface»
Power_Source_Dependency
Description
This interface is the Power_Source requirements to fulfil a Power_Solution.

Figure 737: Power_Source_Dependency Service Policy

Power_Source_Dependency

This service identifies activities related to a Power_Source in order to support a Power_Solution, e.g. to provide additional power to allow a high power DEW to be used.

Interfaces

Power_Source_Dependency

This interface is the Power_Source requirements to fulfil a Power_Solution.

Attributes

source	The specific Power_Source.
setting	The required setting of the Power_Source.
temporal_information	Information covering timing, such as start and end times of the requirement on the Power_Source.
cost	The cost of the Power_Source meeting the requirements, e.g. resources used or time taken.
predicted_quality	How well the proposed Power_Source is predicted to satisfy the requirement.

Condition_Achievement

This interface is the statement of achievement against the Power_Source dependency.

Criterion

This interface is the measurement criteria associated with the derived Power_Source requirement.

Attributes

property	The Power_Source property to be measured.
value	The value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

identify_source_setting_change_to_be_fulfilled

Identify the Power_Source setting change requirements to be fulfilled.

assess_source_setting_change_progress

Assess the progress against the Power_Source dependency requirement to decide if further action needs to be taken.

identify_source_setting_change_requirement

Identify a requirement to change the setting of a Power_Source.

5.4.2.43.7.1.4 Regulator_Dependency

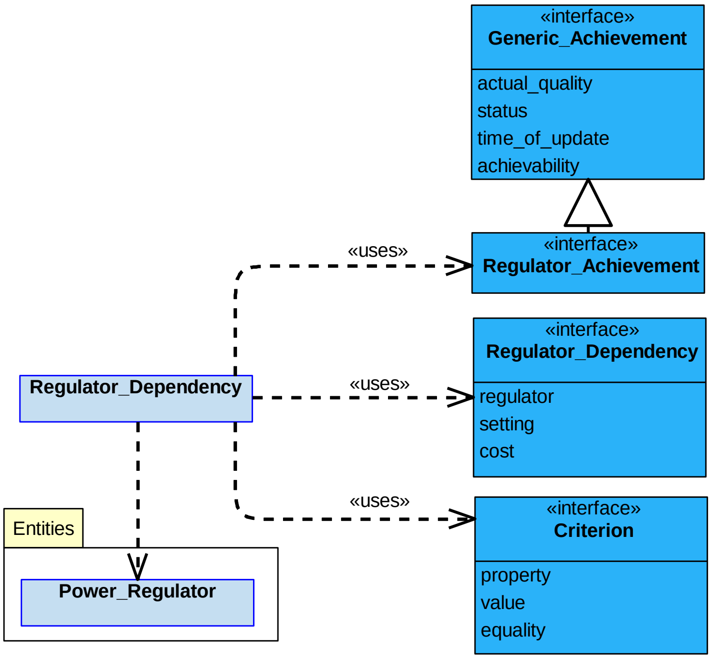


Figure 738: Regulator_Dependency Service Definition

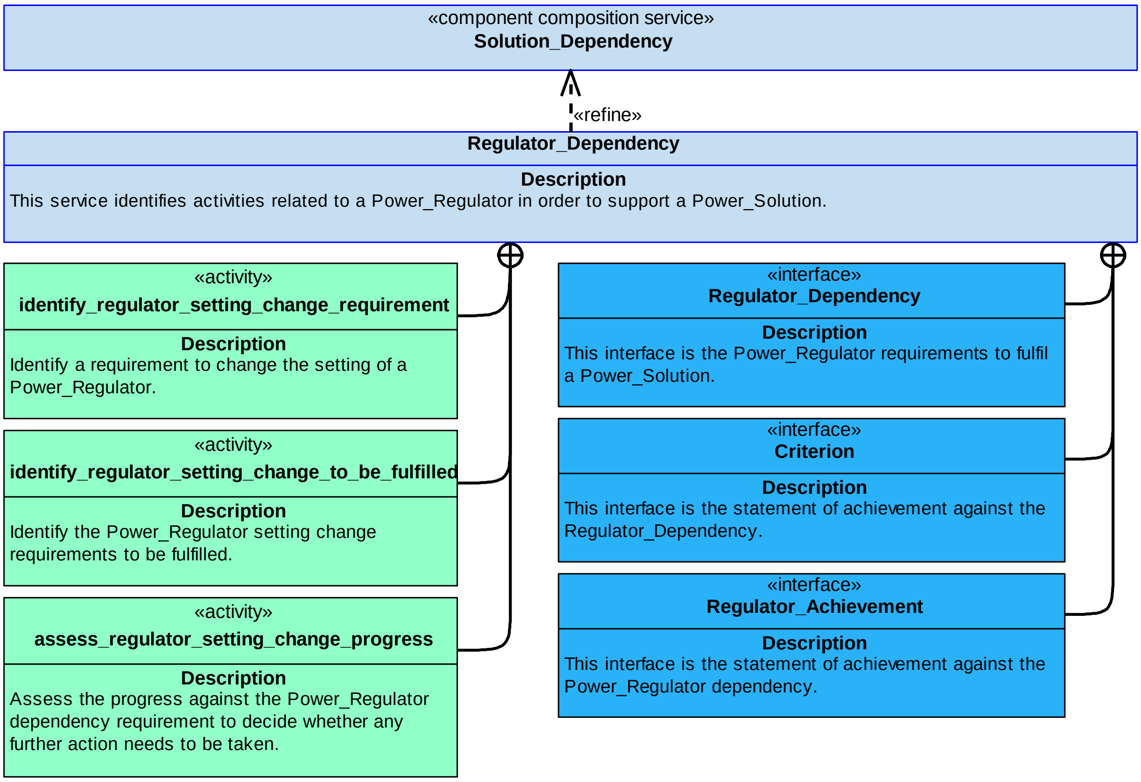

«component composition service»
Solution_Dependency
«refine»
Regulator_Dependency
Description
This service identifies activities related to a Power_Regulator in order to support a Power_Solution.
«activity»
identify_regulator_setting_change_requirement
Description
Identify a requirement to change the setting of a Power_Regulator.
«activity»
identify_regulator_setting_change_to_be_fulfilled
Description
Identify the Power_Regulator setting change requirements to be fulfilled.
«activity»
assess_regulator_setting_change_progress
Description
Assess the progress against the Power_Regulator dependency requirement to decide whether any further action needs to be taken.
«interface»
Regulator_Dependency
Description
This interface is the Power_Regulator requirements to fulfil a Power_Solution.
«interface»
Criterion
Description
This interface is the statement of achievement against the Regulator_Dependency.
«interface»
Regulator_Achievement
Description
This interface is the statement of achievement against the Power_Regulator dependency.

Figure 739: Regulator_Dependency Service Policy

Regulator_Dependency

This service identifies activities related to a Power_Regulator in order to support a Power_Solution.

Interfaces

Regulator_Dependency

This interface is the Power_Regulator requirements to fulfil a Power_Solution.

Attributes

regulator	The specific Power_Regulator.
setting	The required setting of the Power_Regulator.
cost	The cost of the Power_Regulator meeting the requirements, e.g. resources used or time taken.

Criterion

This interface is the statement of achievement against the Regulator_Dependency.

Attributes

property	The Power_Regulator property to be measured.
value	The value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Regulator_Achievement

This interface is the statement of achievement against the Power_Regulator dependency.

Activities

identify_regulator_setting_change_to_be_fulfilled

Identify the Power_Regulator setting change requirements to be fulfilled.

assess_regulator_setting_change_progress

Assess the progress against the Power_Regulator dependency requirement to decide whether any further action needs to be taken.

identify_regulator_setting_change_requirement

Identify a requirement to change the setting of a Power_Regulator.

5.4.2.43.7.1.5 State_Information

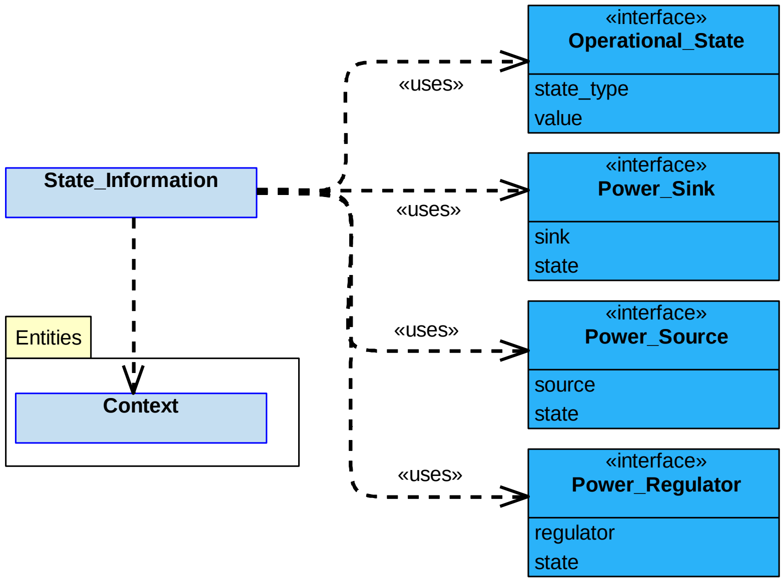
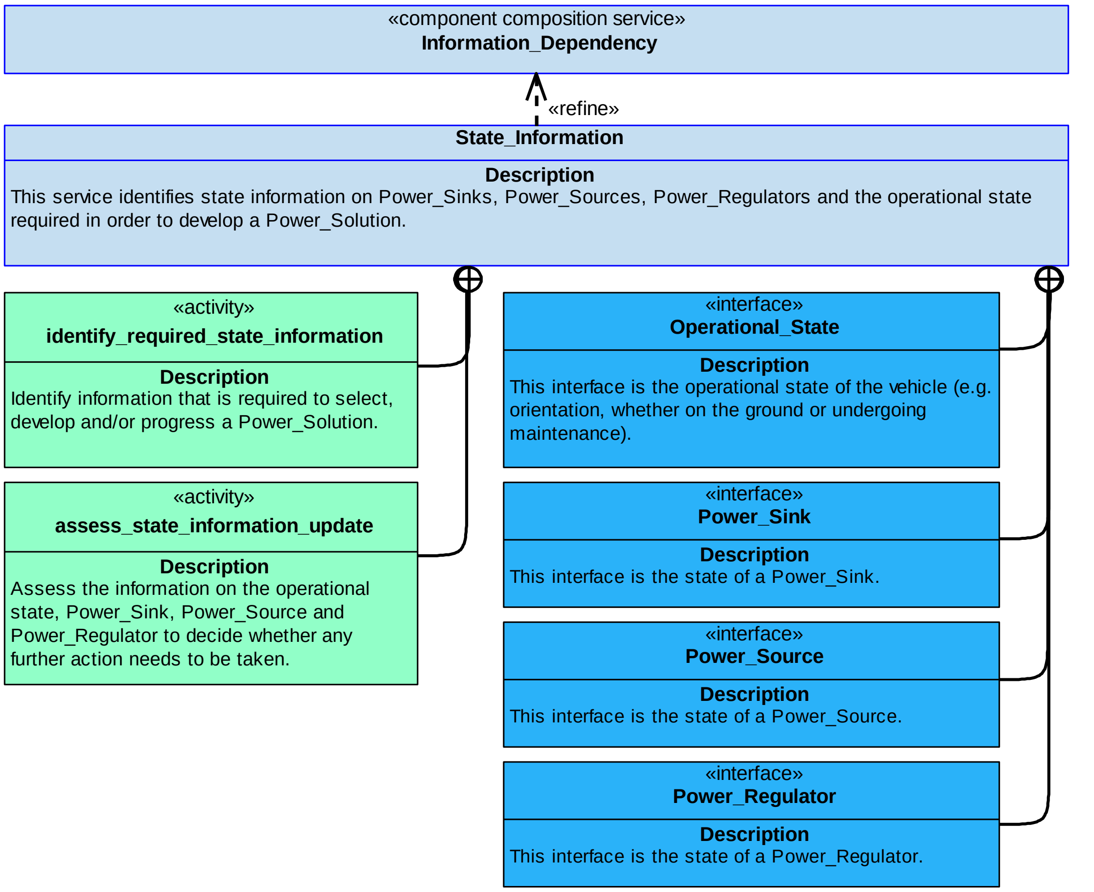


Figure 740: State_Information Service Definition



State_Information

Description

This service identifies state information on Power_Sinks, Power_Sources, Power_Regulators and the operational state required in order to develop a Power_Solution.

identify_required_state_information

Description

Identify information that is required to select, develop and/or progress a Power_Solution.

assess_state_information_update

Description

Assess the information on the operational state, Power_Sink, Power_Source and Power_Regulator to decide whether any further action needs to be taken.

Operational_State

Description

This interface is the operational state of the vehicle (e.g. orientation, whether on the ground or undergoing maintenance).

Power_Sink

Description

This interface is the state of a Power_Sink.

Power_Source

Description

This interface is the state of a Power_Source.

Power_Regulator

Description

This interface is the state of a Power_Regulator.

Figure 741: State_Information Service Policy

State_Information

This service identifies state information on Power_Sinks, Power_Sources, Power_Regulators and the operational state required in order to develop a Power_Solution.

Interfaces

Operational_State

This interface is the operational state of the vehicle (e.g. orientation, whether on the ground or undergoing maintenance).

Attributes

state_type	The type of information relating to the operational state.
value	The value of the state type.

Power_Sink

This interface is the state of a Power_Sink.

Attributes

sink	The specific Power_Sink.
state	A state of a Power_Sink.

Power_Source

This interface is the state of a Power_Source.

Attributes

source	The specific Power_Source.
state	A state of a Power_Source.

Power_Regulator

This interface is the state of a Power_Regulator.

Attributes

regulator	The specific Power_Regulator.
state	A state of a Power_Regulator.

Activities

assess_state_information_update

Assess the information on the operational state, Power_Sink, Power_Source and Power_Regulator to decide whether any further action needs to be taken.

identify_required_state_information

Identify information that is required to select, develop and/or progress a Power_Solution.

5.4.2.43.7.1.6 Measurement_Information

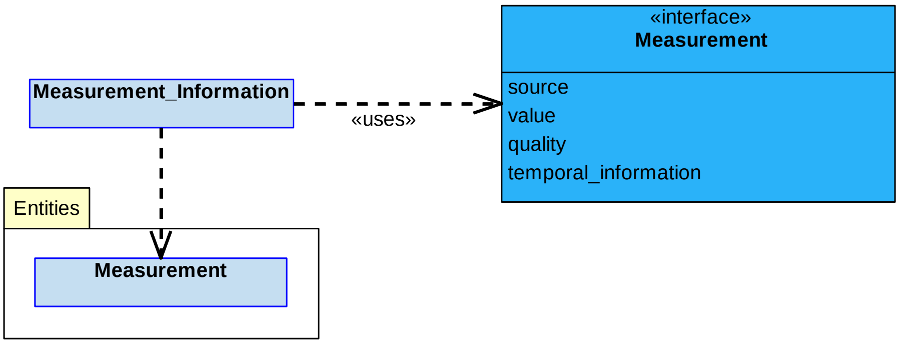


Figure 742: Measurement_Information Service Definition

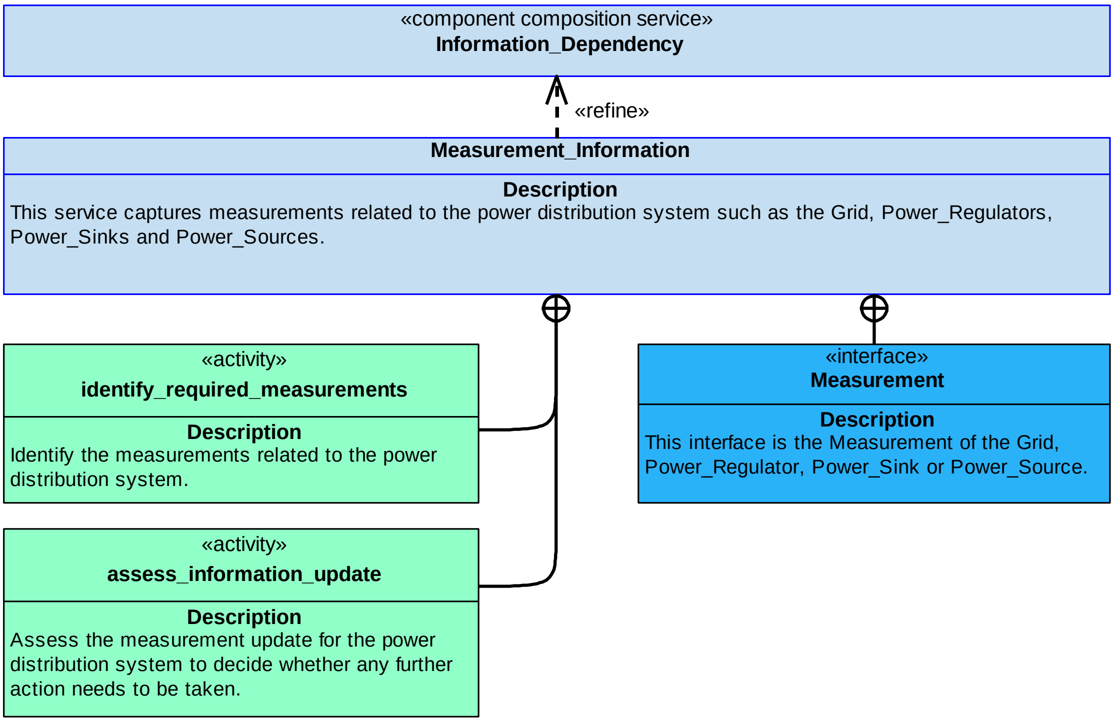

«component composition service»
Information_Dependency
«refine»
Measurement_Information
Description
This service captures measurements related to the power distribution system such as the Grid, Power_Regulators, Power_Sinks and Power_Sources.
«activity»
identify_required_measurements
Description
Identify the measurements related to the power distribution system.
«interface»
Measurement
Description
This interface is the Measurement of the Grid, Power_Regulator, Power_Sink or Power_Source.
«activity»
assess_information_update
Description
Assess the measurement update for the power distribution system to decide whether any further action needs to be taken.

Figure 743: Measurement_Information Service Policy

Measurement_Information

This service captures measurements related to the power distribution system such as the Grid, Power_Regulators, Power_Sinks and Power_Sources.

<u>Interface</u>

Measurement

This interface is the Measurement of the Grid, Power_Regulator, Power_Sink or Power_Source.

<u>Attributes</u>

source	The source of the Measurement, e.g. a specific sensor and its location.
value	The Measurement value.

quality	The quality of the provided Measurement value, e.g. accuracy and precision.
temporal_information	Information covering the timing of the information being reported.

Activities

identify_required_measurements

Identify the measurements related to the power distribution system.

assess_information_update

Assess the measurement update for the power distribution system to decide whether any further action needs to be taken.

5.4.2.43.7.1.7 Constraint

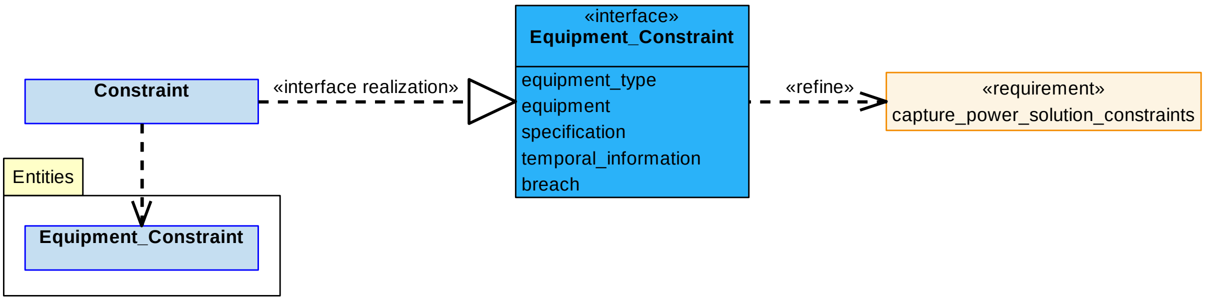
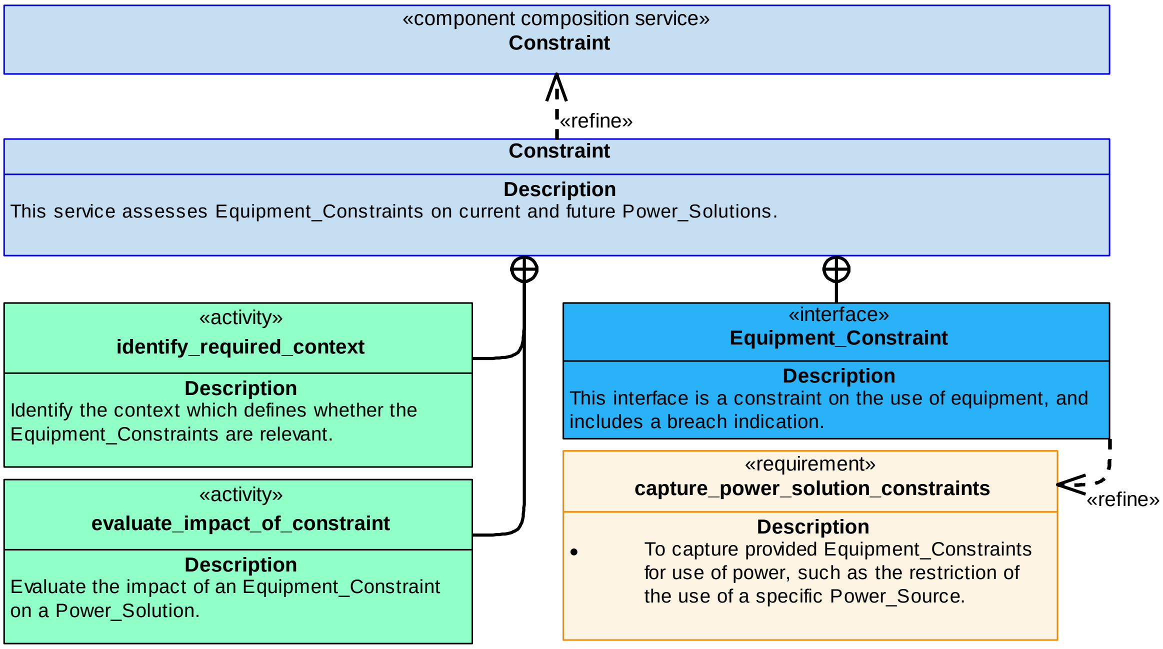


Figure 744: Constraint Service Definition



«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Equipment_Constraints on current and future Power_Solutions.
«activity»
identify_required_context
Description
Identify the context which defines whether the Equipment_Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of an Equipment_Constraint on a Power_Solution.
«interface»
Equipment_Constraint
Description
This interface is a constraint on the use of equipment, and includes a breach indication.
«requirement»
capture_power_solution_constraints
Description
• To capture provided Equipment_Constraints for use of power, such as the restriction of the use of a specific Power_Source.
«refine»

Figure 745: Constraint Service Policy

Constraint

This service assesses Equipment_Constraints on current and future Power_Solutions.

Interface

Equipment_Constraint

This interface is a constraint on the use of equipment, and includes a breach indication.

Attributes

equipment_type	The type of equipment that is restricted for use in a Power_Solution, e.g. engines and batteries.
equipment	The specific piece of equipment that is restricted by the constraint.
specification	The specification of the constraint, e.g. a limit on the amount of power that can be drawn from a battery.
temporal_information	Information covering timing of the constraint, such as start time and duration, or end time.
breach	A statement that the constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of an Equipment_Constraint on a Power_Solution.

identify_required_context

Identify the context which defines whether the Equipment_Constraints are relevant.

5.4.2.43.7.1.8 Capability

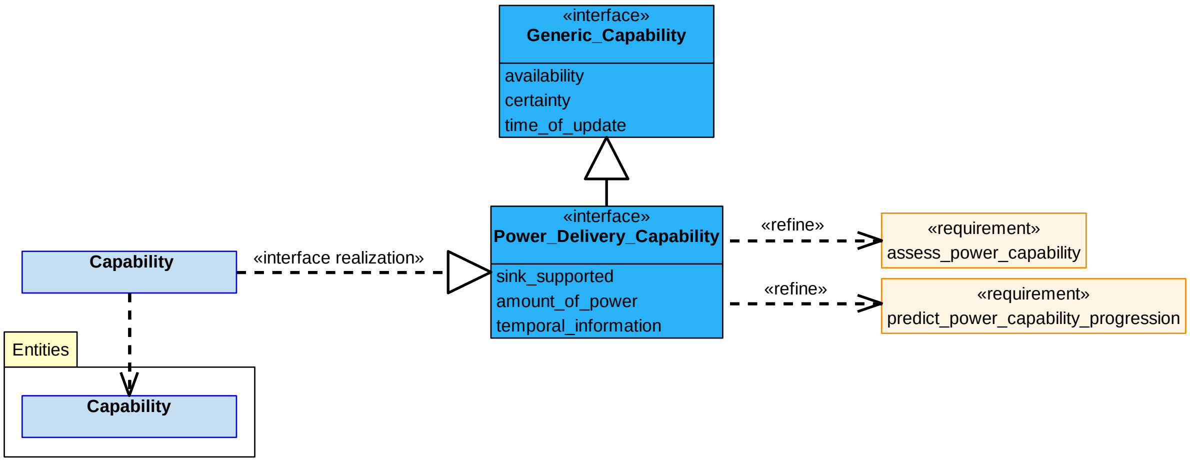


Figure 746: Capability Service Definition

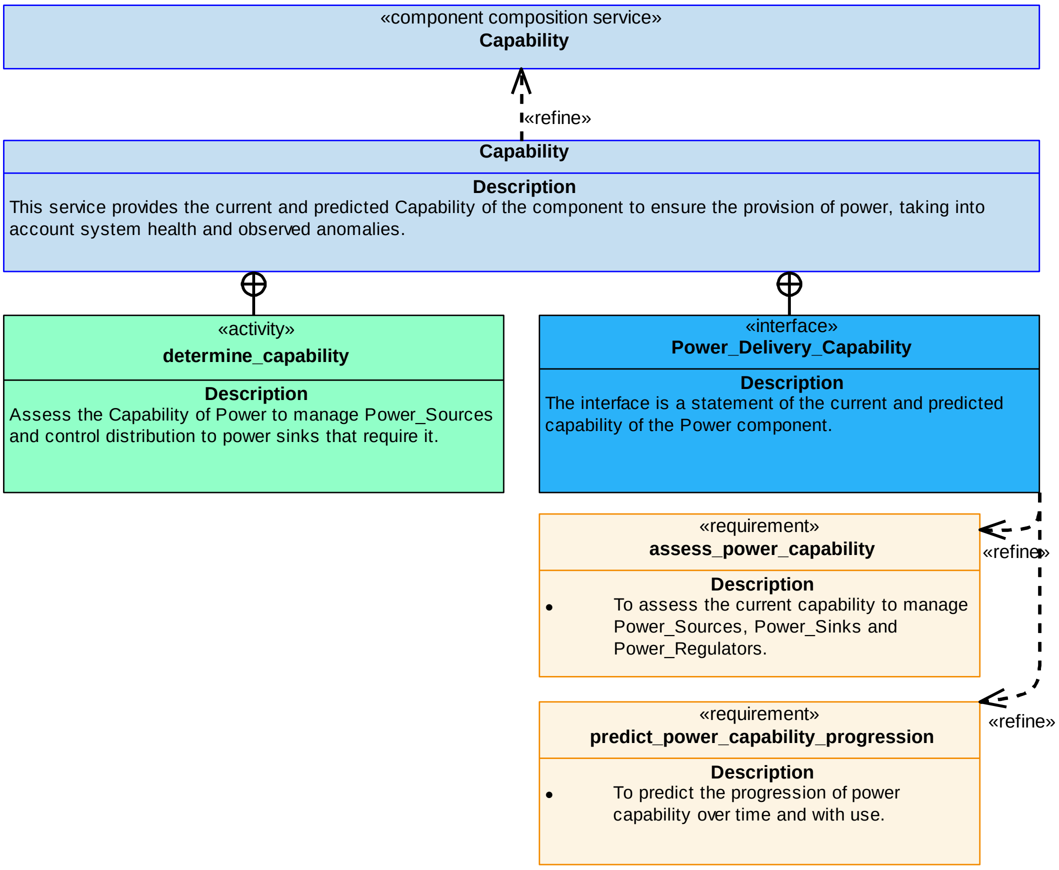

«component composition service»
Capability
«refine»
Capability
Description
This service provides the current and predicted Capability of the component to ensure the provision of power, taking into account system health and observed anomalies.
«activity»
determine_capability
Description
Assess the Capability of Power to manage Power_Sources and control distribution to power sinks that require it.
«interface»
Power_Delivery_Capability
Description
The interface is a statement of the current and predicted capability of the Power component.
«requirement»
assess_power_capability
Description
To assess the current capability to manage Power_Sources, Power_Sinks and Power_Regulators.
«refine»
«requirement»
predict_power_capability_progression
Description
To predict the progression of power capability over time and with use.
«refine»

Figure 747: Capability Service Policy

Capability

This service provides the current and predicted Capability of the component to ensure the provision of power, taking into account system health and observed anomalies.

Interface

Power_Delivery_Capability

The interface is a statement of the current and predicted capability of the Power component.

Attributes

sink_supported	The Power_Sinks that can be provided with power.
amount_of_power	The amount of power that can be provided.
temporal_information	Information covering timing of the Capability, such as for how long the Capability is likely to exist.

Activity

determine_capability

Assess the Capability of Power to manage Power_Sources and control distribution to power sinks that require it.

5.4.2.43.7.1.9 Power_Capability_Evidence

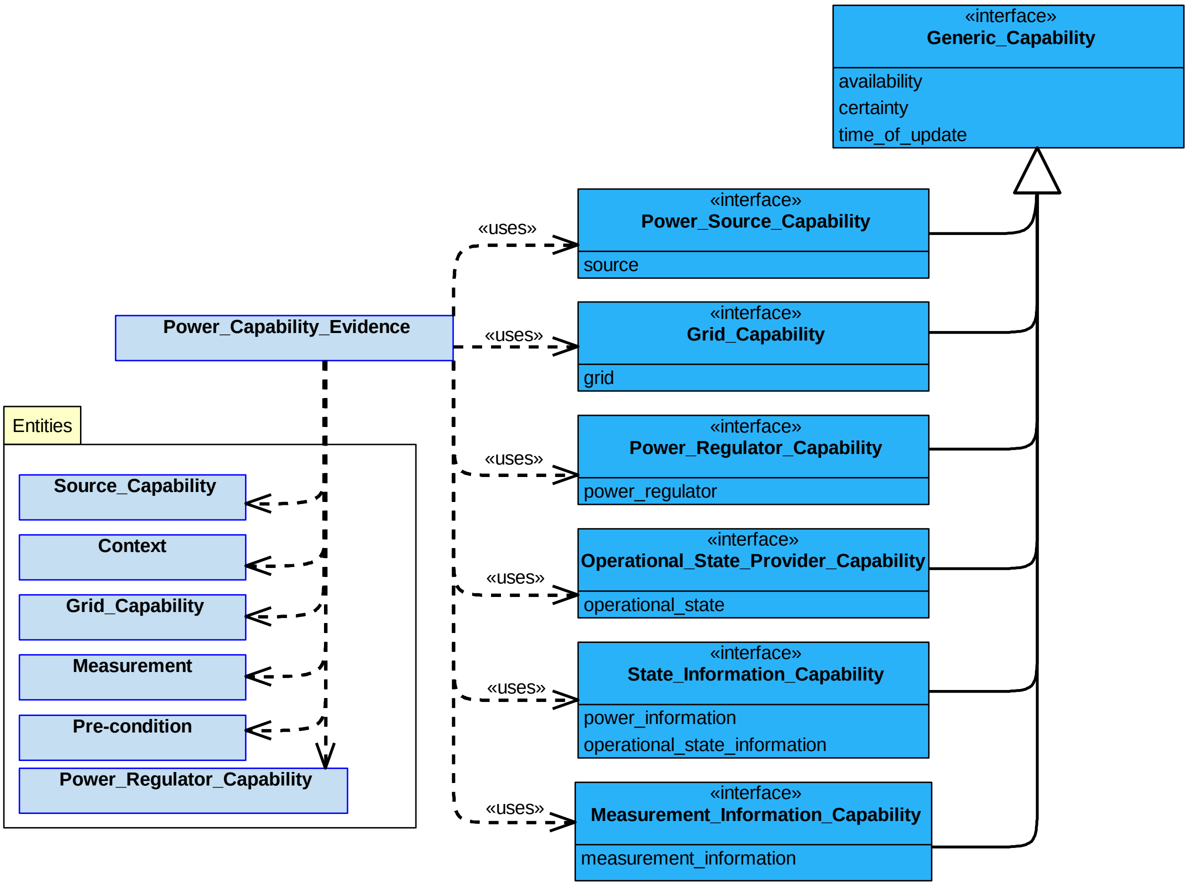


Figure 748: Power_Capability_Evidence Service Definition

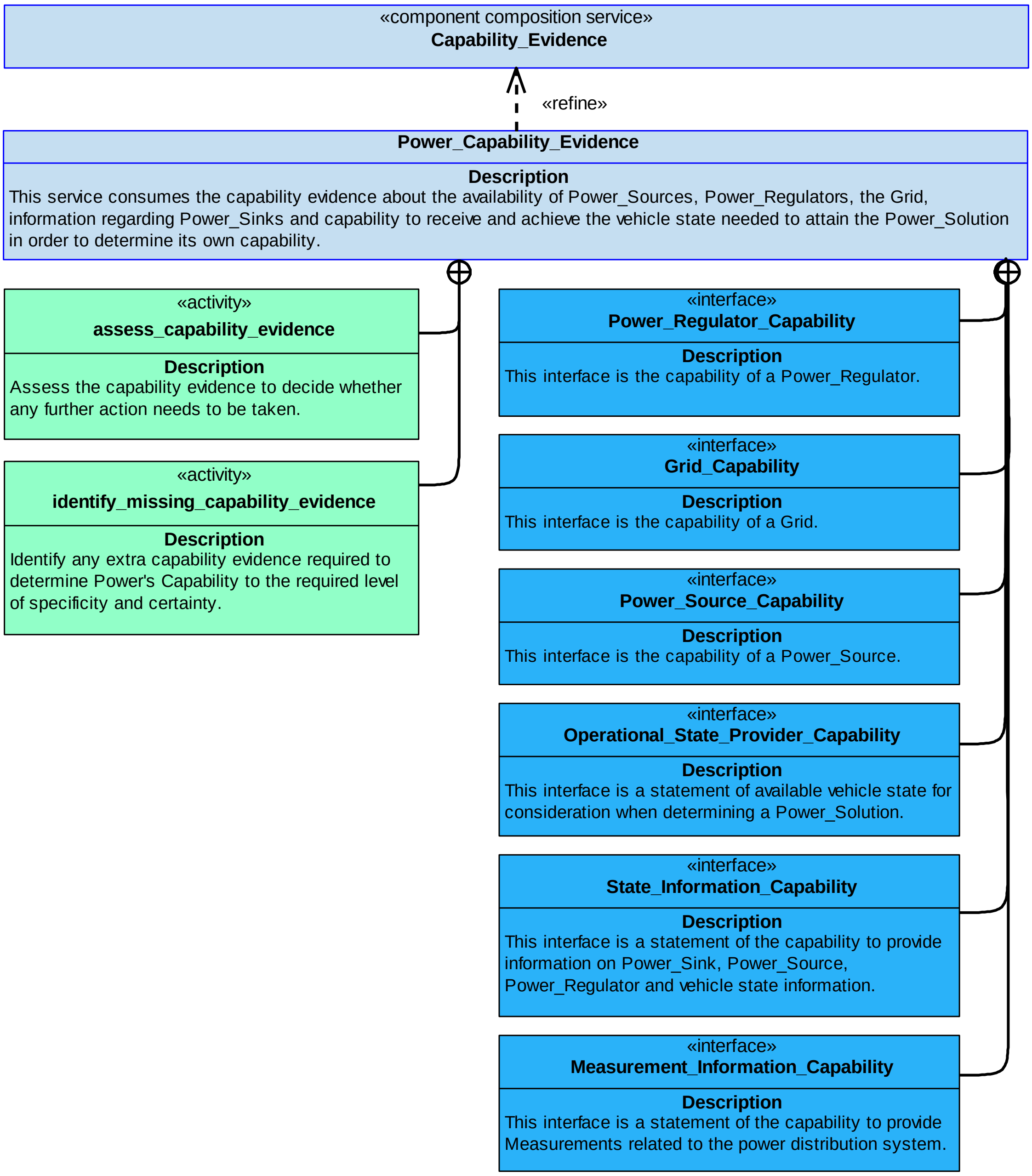

«component composition service»
Capability_Evidence
«refine»
Power_Capability_Evidence
Description
This service consumes the capability evidence about the availability of Power_Sources, Power_Regulators, the Grid, information regarding Power_Sinks and capability to receive and achieve the vehicle state needed to attain the Power_Solution in order to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine Power's Capability to the required level of specificity and certainty.
«interface»
Power_Regulator_Capability
Description
This interface is the capability of a Power_Regulator.
«interface»
Grid_Capability
Description
This interface is the capability of a Grid.
«interface»
Power_Source_Capability
Description
This interface is the capability of a Power_Source.
«interface»
Operational_State_Provider_Capability
Description
This interface is a statement of available vehicle state for consideration when determining a Power_Solution.
«interface»
State_Information_Capability
Description
This interface is a statement of the capability to provide information on Power_Sink, Power_Source, Power_Regulator and vehicle state information.
«interface»
Measurement_Information_Capability
Description
This interface is a statement of the capability to provide Measurements related to the power distribution system.

Figure 749: Power_Capability_Evidence Service Policy

Power_Capability_Evidence

This service consumes the capability evidence about the availability of Power_Sources, Power_Regulators, the Grid, information regarding Power_Sinks and capability to receive and achieve the vehicle state needed to attain the Power_Solution in order to determine its own capability.

Interfaces

Power_Source_Capability

This interface is the capability of a Power_Source.

Attribute

source	The specific Power_Source.

Grid_Capability

This interface is the capability of a Grid.

Attribute

grid	The specific Grid, or portion of a Grid.

Power_Regulator_Capability

This interface is the capability of a Power_Regulator.

Attribute

power_regulator	The specific Power_Regulator.

State_Information_Capability

This interface is a statement of the capability to provide information on Power_Sink, Power_Source, Power_Regulator and vehicle state information.

Attributes

power_information	The specific Power_Sink, Power_Source and Power_Regulator to which a statement applies.
operational_state_information	The specific category of vehicle state to which a statement applies.

Operational_State_Provider_Capability

This interface is a statement of available vehicle state for consideration when determining a Power_Solution.

Attribute

operational_state	The specific vehicle state for consideration when determining a Power_Solution.

Measurement_Information_Capability

This interface is a statement of the capability to provide Measurements related to the power distribution system.

Attribute

measurement_information	The specific Measurement or type of Measurement to which the statement applies.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine Power's Capability to the required level of specificity and certainty.

5.4.2.43.7.2 Service Dependencies

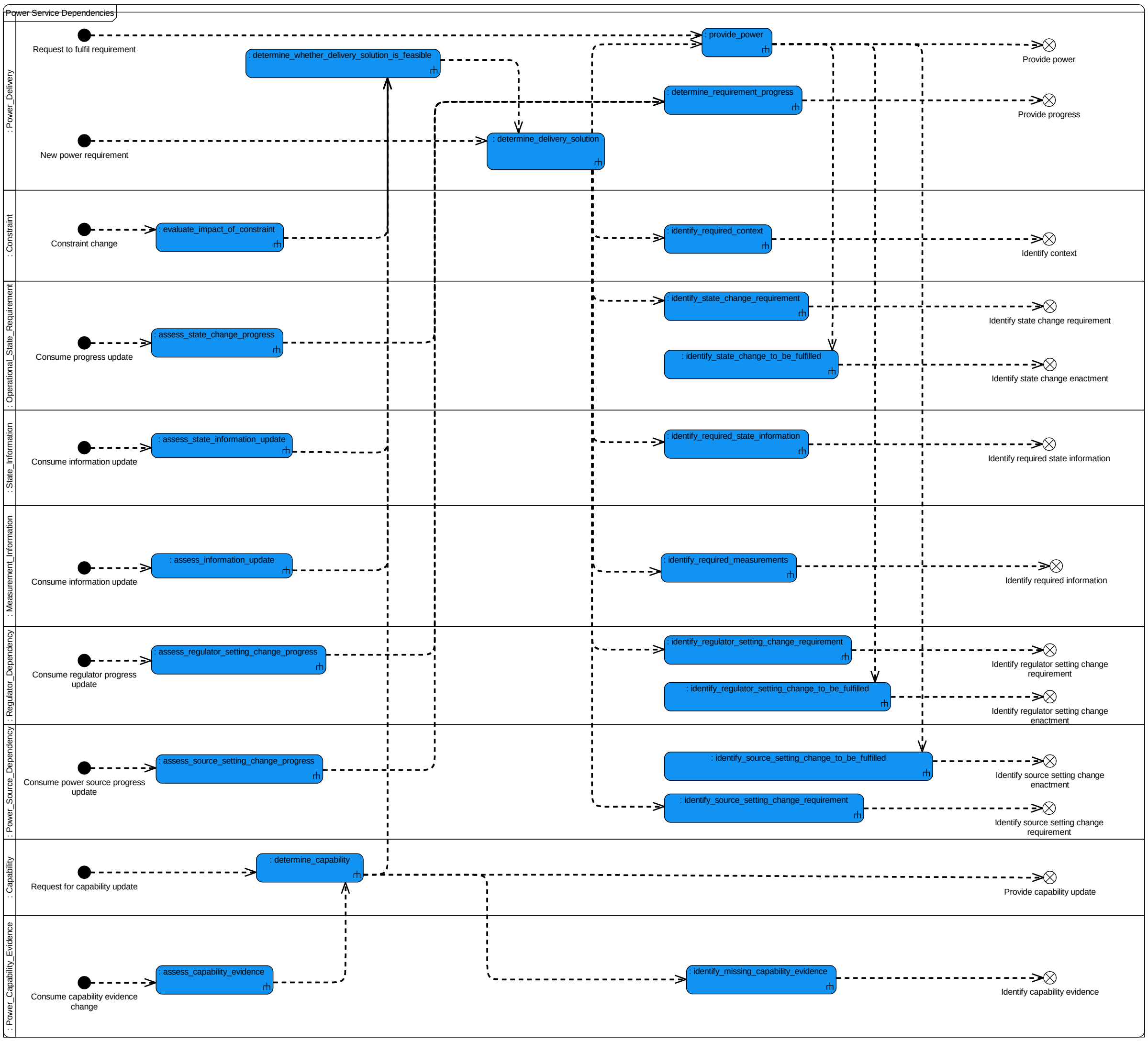

Power Service Dependencies
: Power_Delivery
Request to fulfil requirement
: determine_whether_delivery_solution_is_feasible
: provide_power
Provide power
: determine_requirement_progress
Provide progress
New power requirement
: determine_delivery_solution
: Constraint
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
: Operational_State_Requirement
: identify_state_change_requirement
Identify state change requirement
Consume progress update
: assess_state_change_progress
: identify_state_change_to_be_fulfilled
Identify state change enactment
: State_Information
Consume information update
: assess_state_information_update
: identify_required_state_information
Identify required state information
: Measurement_Information
Consume information update
: assess_information_update
: identify_required_measurements
Identify required information
: Regulator_Dependency
Consume regulator progress update
: assess_regulator_setting_change_progress
: identify_regulator_setting_change_requirement
Identify regulator setting change requirement
: identify_regulator_setting_change_to_be_fulfilled
Identify regulator setting change enactment
: Power_Source_Dependency
Consume power source progress update
: assess_source_setting_change_progress
: identify_source_setting_change_to_be_fulfilled
Identify source setting change enactment
: identify_source_setting_change_requirement
Identify source setting change requirement
: Capability
Request for capability update
: determine_capability
Provide capability update
: Power_Capability_Evidence
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 750: Power Service Dependencies

5.4.2.44 Propulsion

5.4.2.44.1 Role

The role of Propulsion is to control the propulsion of an air vehicle and the state of the propulsion units.

5.4.2.44.2 Overview

Control Architecture

Propulsion is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When there is a need for thrust to either move the vehicle or to provide directional control using thrust, a requirement will be placed on Propulsion to provide said thrust. Propulsion will control a Propulsion_Unit such that the thrust demand is met.

Examples of Use

- Propulsion will be required within a system incorporating an engine to enable a vehicle to move itself.

5.4.2.44.3 Service Summary

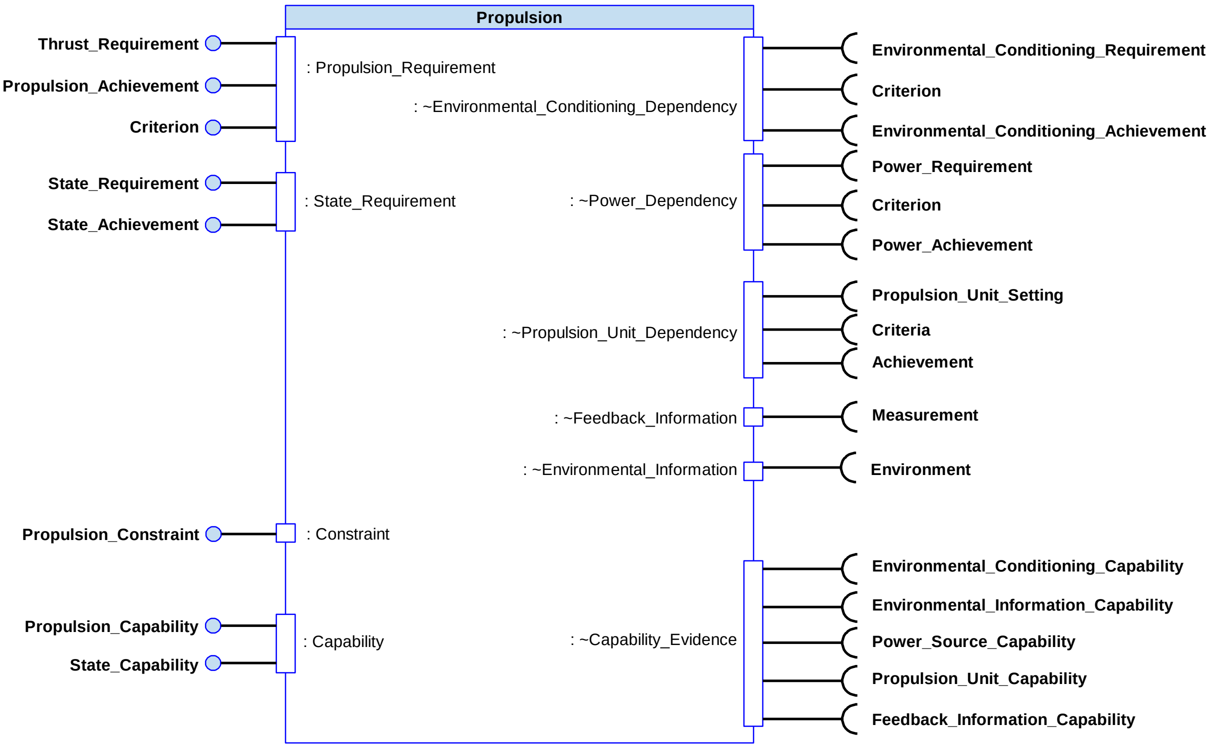


Figure 751: Propulsion Service Summary

5.4.2.44.4 Responsibilities

capture_propulsion_requirements

- To capture Thrust_Requirements.

capture_state_requirements

- To capture State_Requirements.

capture_measurement_criteria_for_propulsion_solution

- To capture provided Measurement_Criterion for propulsion solutions.

capture_propulsion_constraints

- To capture provided Constraints for use of propulsion resources.

identify_whether_thrust_requirement_remains_achievable

- To identify if a Thrust_Requirement is still achievable given current resources.

identify_whether_state_requirement_remains_achievable

- To identify if a State_Requirement is still achievable given current resources.

control_propulsion

- To control the Propulsion_Unit_Setting(s) for the output required.

identify_progress_of_delivered_thrust

- To identify the progress of Thrust against the Thrust_Requirements.

determine_quality_of_propulsion_unit_setting

- To determine the Designed_Thrust_Quality of the Propulsion_Unit_Setting against one or more given Measurement_Criterion.

determine_quality_of_delivered_thrust

- To determine the Delivered_Thrust_Quality of the Thrust measured against Thrust_Requirement in terms of given Measurement_Criterion.

assess_capability

- To assess Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_propulsion_information

- To identify missing information which could improve the certainty or specificity of Capability assessment.

predict_propulsion_capability

- To predict the progression of Capability over time and with use (e.g. in the presence of degrading failures).

5.4.2.44.5 Subject Matter Semantics

The subject matter of Propulsion is the control of Propulsion_Unit state and the provision of thrust by Propulsion_Units.

Exclusions

The subject matter of Propulsion does not include:

- The use of a Propulsion_Unit for any reason other than the provision of thrust.

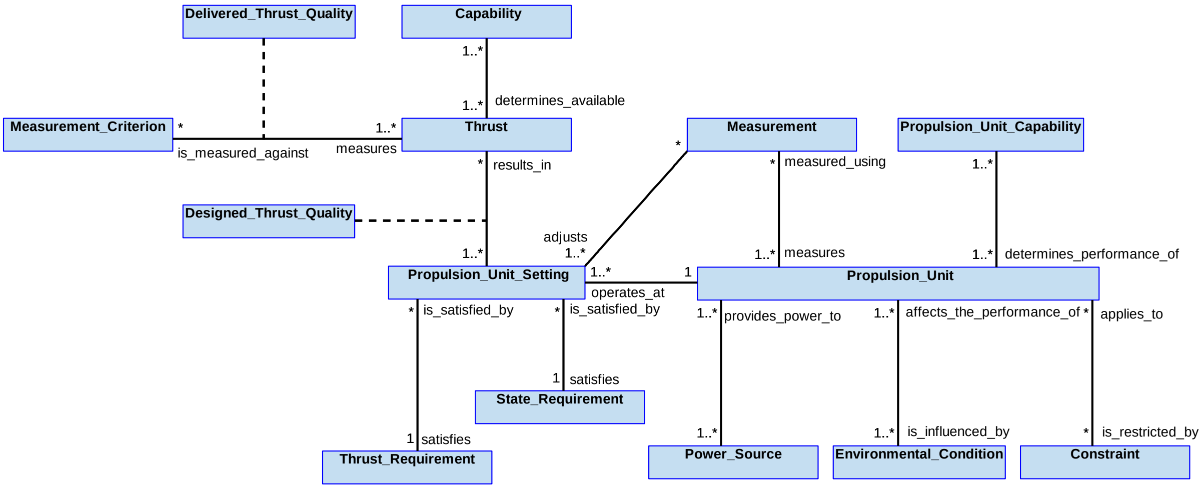


Figure 752: Propulsion Semantics

5.4.2.44.5.1 Entities

Capability

The capability to control the state of Propulsion_Units and the propulsion of an Exploiting Platform.

Constraint

An externally imposed restriction on a Propulsion_Unit, such as a restriction on the use of resources needed to use the Propulsion_Unit.

Propulsion_Unit

An entity that is capable of providing propulsion. It may comprise a number of sub-units, which are individually addressable (e.g. nozzles or fuel pumps).

Propulsion_Unit_Setting

The setting of a Propulsion_Unit. If a Propulsion_Unit comprises a number of sub-units, the Propulsion_Unit_Setting will be the compound of the settings of the sub-units.

State_Requirement

A requirement for a specific state of a Propulsion_Unit, such as maintain idle speed.

Thrust_Requirement

A demand for a specific amount of thrust.

Environmental_Condition

A characteristic surrounding the Exploiting Platform and its equipment (e.g. air pressure). This can be localised to a specific part of the equipment structure, such as a jet engine's turbine blades.

Measurement_Criterion

A criterion that needs to be evaluated when determining if the delivered Thrust satisfies the Thrust_Requirement (e.g. thrust magnitude, thrust direction or torque).

Thrust

The propulsive force that moves an Exploiting Platform. Thrust has a magnitude, direction and duration.

Delivered_Thrust_Quality

A measure, against a given Measurement_Criterion, of how well the Thrust meets the Thrust_Requirement.

Measurement

A measurement from which a property of a propulsion unit can be determined, such as a flow rate or a pressure measurement.

Propulsion_Unit_Capability

The capability of the Propulsion_Unit. This will take into account, for example, the health of hardware components and the ability of the Propulsion_Unit to change state.

Designed_Thrust_Quality

A measure, against a given Measurement_Criterion, of how well the Propulsion_Unit_Setting meets the requirements.

Power_Source

A source of raw or stored energy made available at a specified rate (e.g. electrical power or fuel).

5.4.2.44.6 Design Rationale

5.4.2.44.6.1 Assumptions

- The component captures propulsion demands; it is assumed that theoretical performance data can be derived from these demands.
- Propulsion requirements are enacted instantaneously.

5.4.2.44.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Propulsion:

- Data Driving - It is recommended that the properties of a specific mark/sub-variant of a Propulsion_Unit should be captured in data rather than by extensions or bespoke versions of the Propulsion component.

Extensions

- A variety of propulsion technologies exist. To accommodate this, the use of extensions should be considered. See Component Extensions.

Exploitation Considerations

- Where there are multiple Exploiting Platforms in a combat air system, such as when disposable platforms are used, each individual platform will use one or more instances of the Propulsion component. This is because each separate platform may have different propulsion characteristics, and each platform should have the capability to control propulsion locally.

5.4.2.44.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

Failure of this component may cause either:

- Loss of or reduced thrust. In some cases this may not lead to fatalities (the pilot may eject or the Exploiting Platform may perform a CTT in a location that minimises the risk to third parties). In other cases the Exploiting Platform may be reliant on Propulsion to provide power for directional control (i.e. limited emergency power duration) then loss of thrust could also result in an uncontrolled crash of the Exploiting Platform.
- Excess thrust may result in exceedance of the flight envelope of the Exploiting Platform and/or not being able to adhere to the required flightpath of the Exploiting Platform. This could lead to loss of structural integrity of the Exploiting Platform and/or an uncontrolled crash.
- Inability to accurately control engine thrust, resulting in an inability to accurately follow a planned flightpath.

In each case the result is likely to be loss of the Exploiting Platform and fatalities.

5.4.2.44.6.4 Security Considerations

The indicative security classification is SNEO.

This component is responsible for the control of the propulsion systems. It will have knowledge of propulsion performance data which is considered SNEO. Due to its role, this component is considered a possible target for a cyber attack, and will have rigorous requirements to ensure its integrity and availability.

The component may be expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to management of the vehicles propulsion systems.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

- Performing **System Status and Monitoring** with unexpected behaviour being an indicator the system may have been compromised.

The component is not expected to directly implement security enforcing functions.

5.4.2.44.7 Services

5.4.2.44.7.1 Service Definitions

5.4.2.44.7.1.1 Propulsion_Requirement

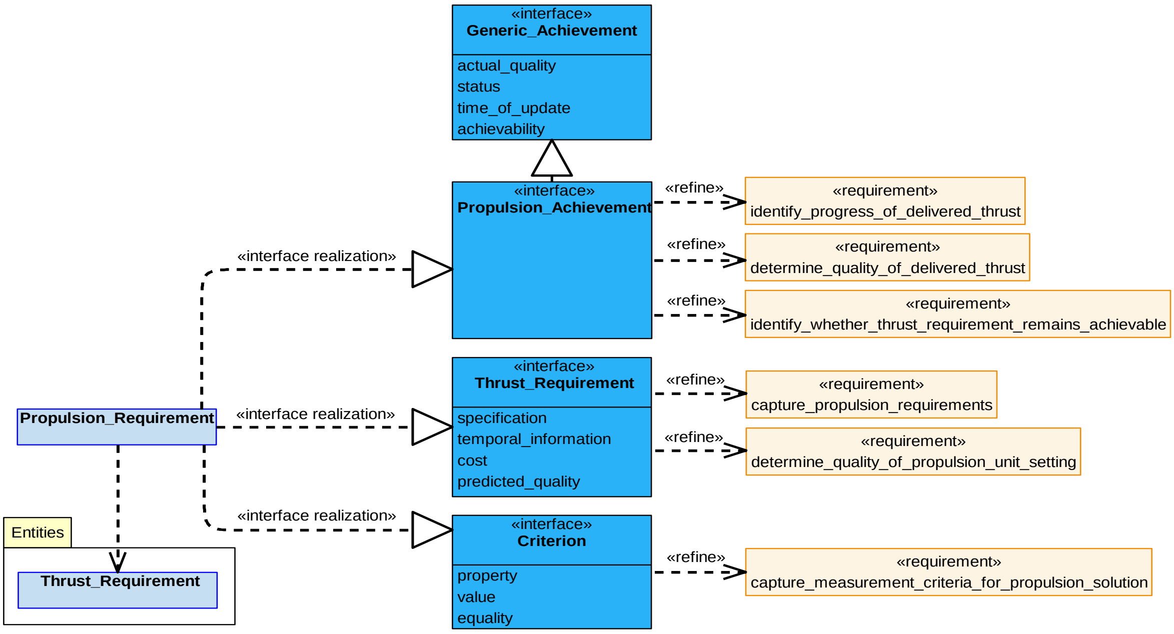


Figure 753: Propulsion_Requirement Service Definition

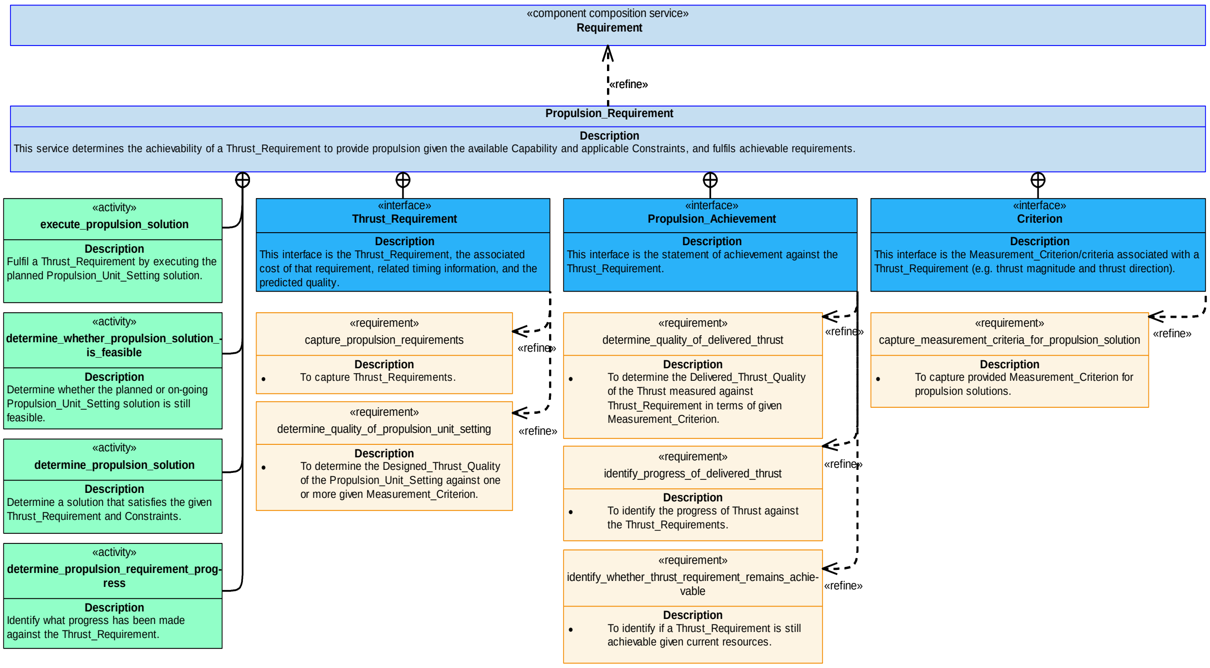

«component composition service»
Requirement
«refine»
Propulsion_Requirement
Description
This service determines the achievability of a Thrust_Requirement to provide propulsion given the available Capability and applicable Constraints, and fulfils achievable requirements.
«activity»
execute_propulsion_solution
Description
Fulfil a Thrust_Requirement by executing the planned Propulsion_Unit_Setting solution.
«activity»
determine_whether_propulsion_solution_is_feasible
Description
Determine whether the planned or on-going Propulsion_Unit_Setting solution is still feasible.
«activity»
determine_propulsion_solution
Description
Determine a solution that satisfies the given Thrust_Requirement and Constraints.
«activity»
determine_propulsion_requirement_progress
Description
Identify what progress has been made against the Thrust_Requirement.
«interface»
Thrust_Requirement
Description
This interface is the Thrust_Requirement, the associated cost of that requirement, related timing information, and the predicted quality.
«interface»
Propulsion_Achievement
Description
This interface is the statement of achievement against the Thrust_Requirement.
«interface»
Criterion
Description
This interface is the Measurement_Criterion/criteria associated with a Thrust_Requirement (e.g. thrust magnitude and thrust direction).
«requirement»
capture_propulsion_requirements
Description
To capture Thrust_Requirements.
«requirement»
determine_quality_of_propulsion_unit_setting
Description
To determine the Designed_Thrust_Quality of the Propulsion_Unit_Setting against one or more given Measurement_Criterion.
«requirement»
determine_quality_of_delivered_thrust
Description
To determine the Delivered_Thrust_Quality of the Thrust measured against Thrust_Requirement in terms of given Measurement_Criterion.
«requirement»
identify_progress_of_delivered_thrust
Description
To identify the progress of Thrust against the Thrust_Requirements.
«requirement»
identify_whether_thrust_requirement_remains_achievable
Description
To identify if a Thrust_Requirement is still achievable given current resources.
«requirement»
capture_measurement_criteria_for_propulsion_solution
Description
To capture provided Measurement_Criterion for propulsion solutions.
«refine»
«refine»
«refine»
«refine»
«refine»
«refine»

Figure 754: Propulsion_Requirement Service Policy

Propulsion_Requirement

This service determines the achievability of a Thrust_Requirement to provide propulsion given the available Capability and applicable Constraints, and fulfils achievable requirements.

Interfaces

Thrust_Requirement

This interface is the Thrust_Requirement, the associated cost of that requirement, related timing information, and the predicted quality.

Attributes

specification	The definition of the Thrust_Requirement, e.g. to provide a specified amount of thrust vectored in a specific direction.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of providing Thrust, for example: fuel flow rate required.
predicted_quality	How well the planned Propulsion_Unit_Setting is predicted to satisfy the Thrust_Requirement.

Propulsion_Achievement

This interface is the statement of achievement against the Thrust_Requirement.

Criterion

This interface is the Measurement_Criterion/criteria associated with a Thrust_Requirement (e.g. thrust magnitude and thrust direction).

Attributes

property	The property to be measured, e.g. Thrust.
value	The measured value of the property, e.g. 50kN.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

execute_propulsion_solution

Fulfil a Thrust_Requirement by executing the planned Propulsion_Unit_Setting solution.

determine_whether_propulsion_solution_is_feasible

Determine whether the planned or on-going Propulsion_Unit_Setting solution is still feasible.

determine_propulsion_solution

Determine a solution that satisfies the given Thrust_Requirement and Constraints.

determine_propulsion_requirement_progress

Identify what progress has been made against the Thrust_Requirement.

5.4.2.44.7.1.2 State_Requirement

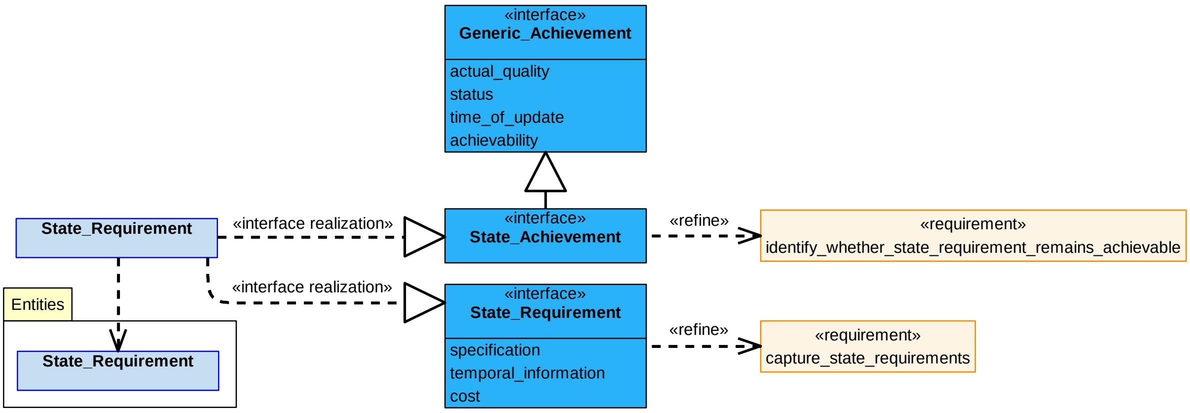


Figure 755: State_Requirement Service Definition

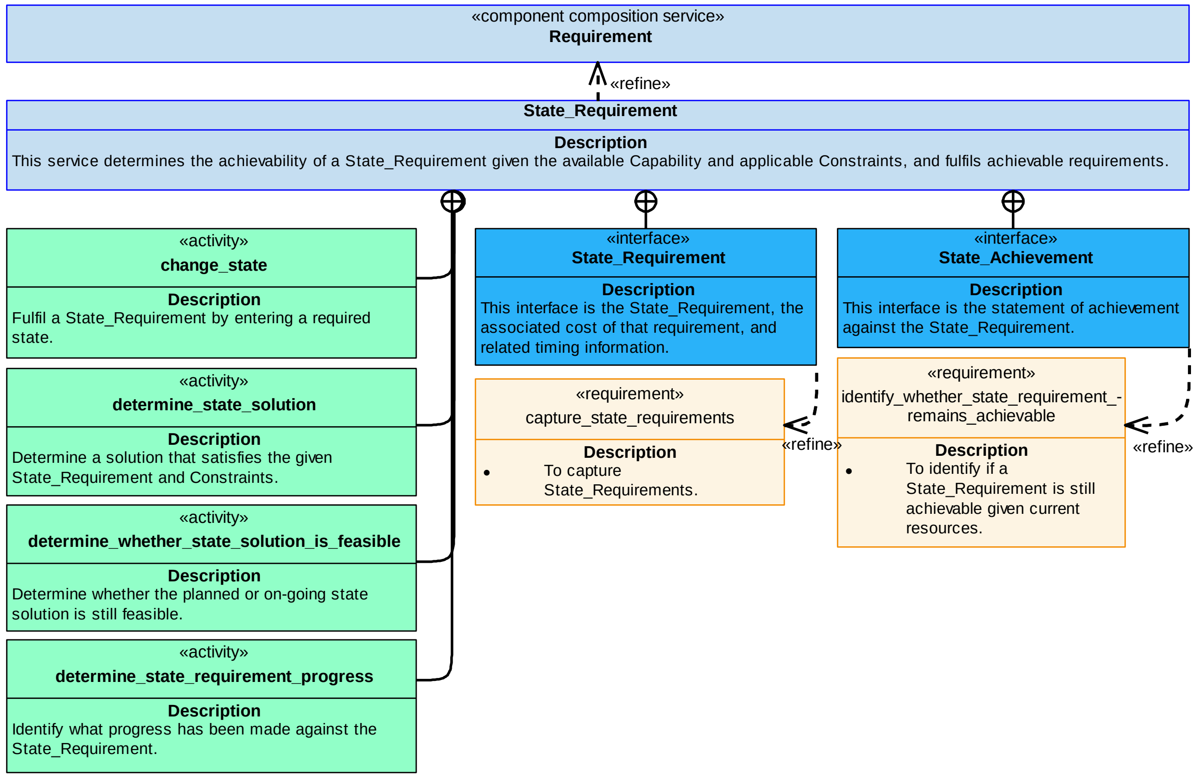

«component composition service»
Requirement
«refine»
State_Requirement
Description
This service determines the achievability of a State_Requirement given the available Capability and applicable Constraints, and fulfils achievable requirements.
«activity»
change_state
Description
Fulfil a State_Requirement by entering a required state.
«activity»
determine_state_solution
Description
Determine a solution that satisfies the given State_Requirement and Constraints.
«activity»
determine_whether_state_solution_is_feasible
Description
Determine whether the planned or on-going state solution is still feasible.
«activity»
determine_state_requirement_progress
Description
Identify what progress has been made against the State_Requirement.
«interface»
State_Requirement
Description
This interface is the State_Requirement, the associated cost of that requirement, and related timing information.
«requirement»
capture_state_requirements
Description
• To capture State_Requirements.
«refine»
«interface»
State_Achievement
Description
This interface is the statement of achievement against the State_Requirement.
«requirement»
identify_whether_state_requirement_-remains_achievable
Description
• To identify if a State_Requirement is still achievable given current resources.
«refine»

Figure 756: State_Requirement Service Policy

State_Requirement

This service determines the achievability of a State_Requirement given the available Capability and applicable Constraints, and fulfils achievable requirements.

Interfaces

State_Requirement

This interface is the State_Requirement, the associated cost of that requirement, and related timing information.

Attributes

specification	The definition of the State_Requirement, e.g. maintain the engine at idle speed.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of meeting the State_Requirement, e.g. fuel flow rate required to maintain idle speed.

State_Achievement

This interface is the statement of achievement against the State_Requirement.

Activities

change_state

Fulfil a State_Requirement by entering a required state.

determine_state_solution

Determine a solution that satisfies the given State_Requirement and Constraints.

determine_whether_state_solution_is_feasible

Determine whether the planned or on-going state solution is still feasible.

determine_state_requirement_progress

Identify what progress has been made against the State_Requirement.

5.4.2.44.7.1.3 Environmental_Conditioning_Dependency

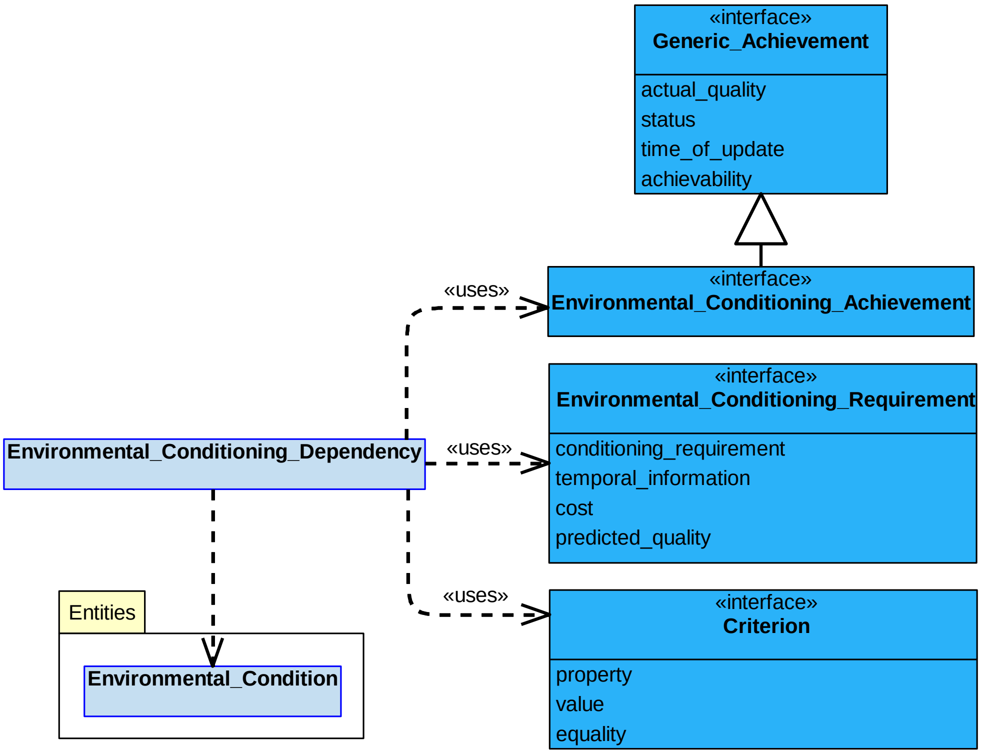


Figure 757: Environmental_Conditioning_Dependency Service Definition

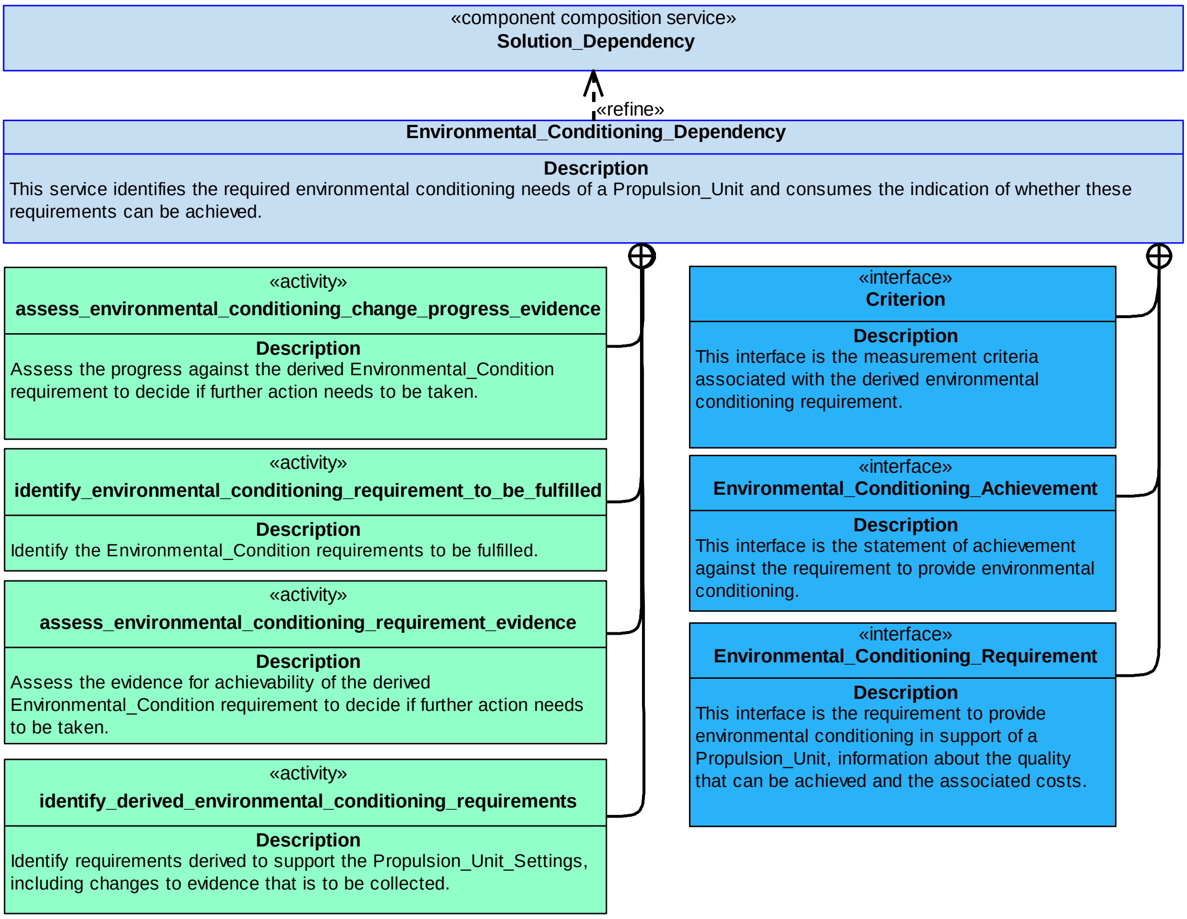

«component composition service»
Solution_Dependency
«refine»
Environmental_Conditioning_Dependency
Description
This service identifies the required environmental conditioning needs of a Propulsion_Unit and consumes the indication of whether these requirements can be achieved.
«activity»
assess_environmental_conditioning_change_progress_evidence
Description
Assess the progress against the derived Environmental_Condition requirement to decide if further action needs to be taken.
«activity»
identify_environmental_conditioning_requirement_to_be_fulfilled
Description
Identify the Environmental_Condition requirements to be fulfilled.
«activity»
assess_environmental_conditioning_requirement_evidence
Description
Assess the evidence for achievability of the derived Environmental_Condition requirement to decide if further action needs to be taken.
«activity»
identify_derived_environmental_conditioning_requirements
Description
Identify requirements derived to support the Propulsion_Unit_Settings, including changes to evidence that is to be collected.
«interface»
Criterion
Description
This interface is the measurement criteria associated with the derived environmental conditioning requirement.
«interface»
Environmental_Conditioning_Achievement
Description
This interface is the statement of achievement against the requirement to provide environmental conditioning.
«interface»
Environmental_Conditioning_Requirement
Description
This interface is the requirement to provide environmental conditioning in support of a Propulsion_Unit, information about the quality that can be achieved and the associated costs.

Figure 758: Environmental_Conditioning_Dependency Service Policy

Environmental_Conditioning_Dependency

This service identifies the required environmental conditioning needs of a Propulsion_Unit and consumes the indication of whether these requirements can be achieved.

Interfaces

Criterion

This interface is the measurement criteria associated with the derived environmental conditioning requirement.

Attributes

property	The Environmental_Condition property to be measured.
value	The value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Environmental_Conditioning_Requirement

This interface is the requirement to provide environmental conditioning in support of a Propulsion_Unit, information about the quality that can be achieved and the associated costs.

Attributes

conditioning_requirement	The need for conditioning required to control an Environmental_Condition (e.g. cooling or pressurisation).
temporal_information	Information covering timing, such as start and end times.
cost	The cost of the Environmental_Condition meeting the requirements, e.g. resources used.
predicted_quality	How well the proposed environmental conditioning can be satisfied.

Environmental_Conditioning_Achievement

This interface is the statement of achievement against the requirement to provide environmental conditioning.

Activities

identify_environmental_conditioning_requirement_to_be_fulfilled

Identify the Environmental_Condition requirements to be fulfilled.

assess_environmental_conditioning_requirement_evidence

Assess the evidence for achievability of the derived Environmental_Condition requirement to decide if further action needs to be taken.

assess_environmental_conditioning_change_progress_evidence

Assess the progress against the derived Environmental_Condition requirement to decide if further action needs to be taken.

identify_derived_environmental_conditioning_requirements

Identify requirements derived to support the Propulsion_Unit_Settings, including changes to evidence that is to be collected.

5.4.2.44.7.1.4 Power_Dependency

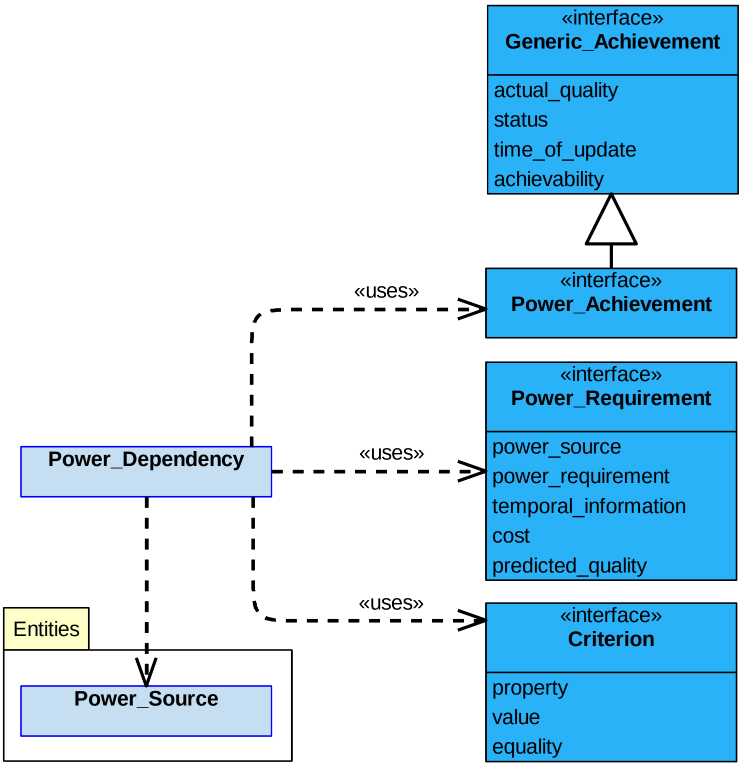


Figure 759: Power_Dependency Service Definition

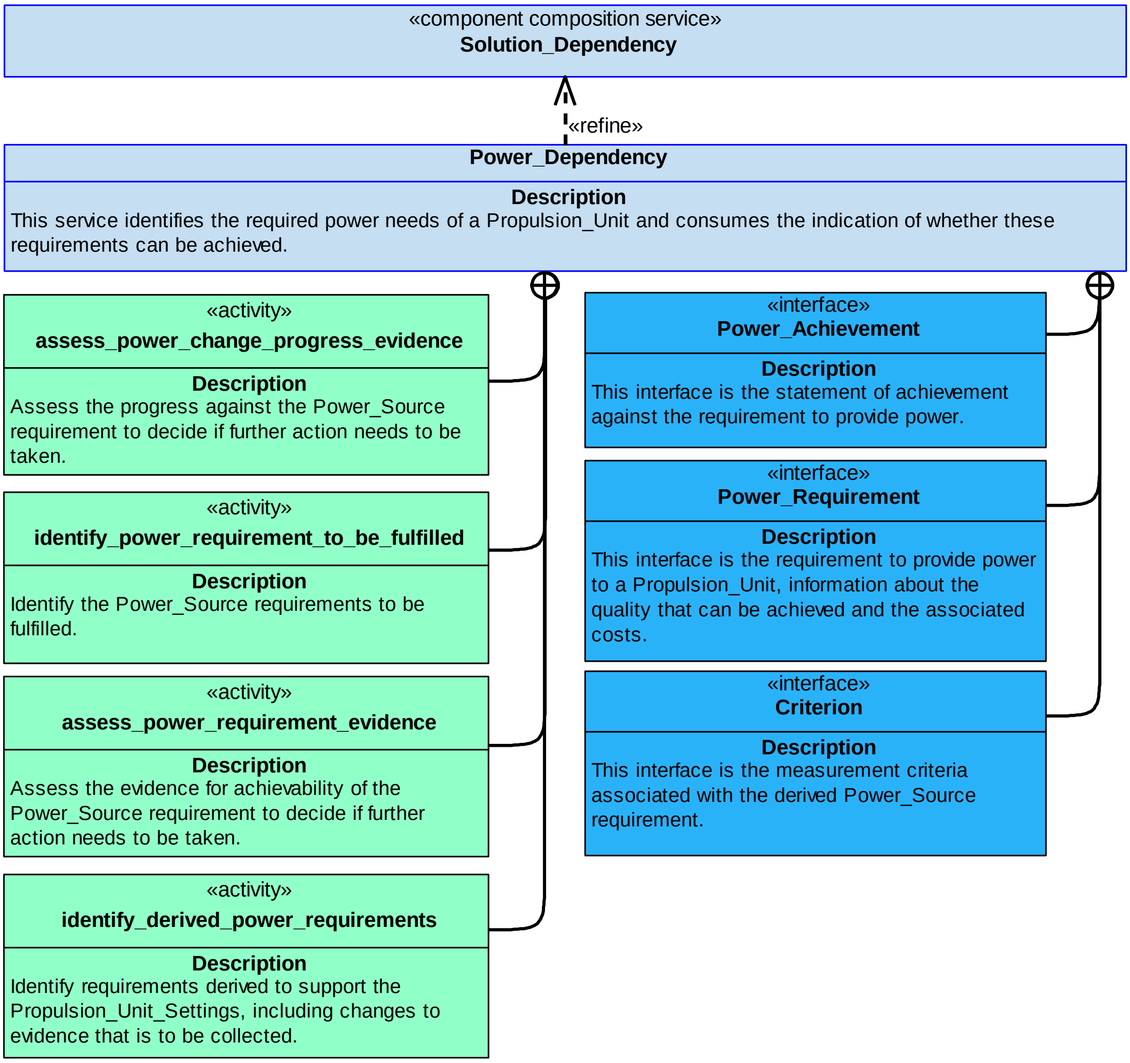

«component composition service»
Solution_Dependency
«refine»
Power_Dependency
Description
This service identifies the required power needs of a Propulsion_Unit and consumes the indication of whether these requirements can be achieved.
«activity»
assess_power_change_progress_evidence
Description
Assess the progress against the Power_Source requirement to decide if further action needs to be taken.
«activity»
identify_power_requirement_to_be_fulfilled
Description
Identify the Power_Source requirements to be fulfilled.
«activity»
assess_power_requirement_evidence
Description
Assess the evidence for achievability of the Power_Source requirement to decide if further action needs to be taken.
«activity»
identify_derived_power_requirements
Description
Identify requirements derived to support the Propulsion_Unit_Settings, including changes to evidence that is to be collected.
«interface»
Power_Achievement
Description
This interface is the statement of achievement against the requirement to provide power.
«interface»
Power_Requirement
Description
This interface is the requirement to provide power to a Propulsion_Unit, information about the quality that can be achieved and the associated costs.
«interface»
Criterion
Description
This interface is the measurement criteria associated with the derived Power_Source requirement.

Figure 760: Power_Dependency Service Policy

Power_Dependency

This service identifies the required power needs of a Propulsion_Unit and consumes the indication of whether these requirements can be achieved.

Interfaces

Power_Achievement

This interface is the statement of achievement against the requirement to provide power.

Criterion

This interface is the measurement criteria associated with the derived Power_Source requirement.

Attributes

property	The Power_Source property to be measured.
value	The value of the property.

equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Power_Requirement

This interface is the requirement to provide power to a Propulsion_Unit, information about the quality that can be achieved and the associated costs.

Attributes

power_source	The Power_Source available for consumption (e.g. liquid fuel, electricity, or wind).
power_requirement	The required rate of energy transfer.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of meeting the Power_Source requirement, e.g. resources used.
predicted_quality	How well the Power_Source is predicted to satisfy the requirement.

Activities

identify_power_requirement_to_be_fulfilled

Identify the Power_Source requirements to be fulfilled.

assess_power_requirement_evidence

Assess the evidence for achievability of the Power_Source requirement to decide if further action needs to be taken.

assess_power_change_progress_evidence

Assess the progress against the Power_Source requirement to decide if further action needs to be taken.

identify_derived_power_requirements

Identify requirements derived to support the Propulsion_Unit_Settings, including changes to evidence that is to be collected.

5.4.2.44.7.1.5 Propulsion_Unit_Dependency

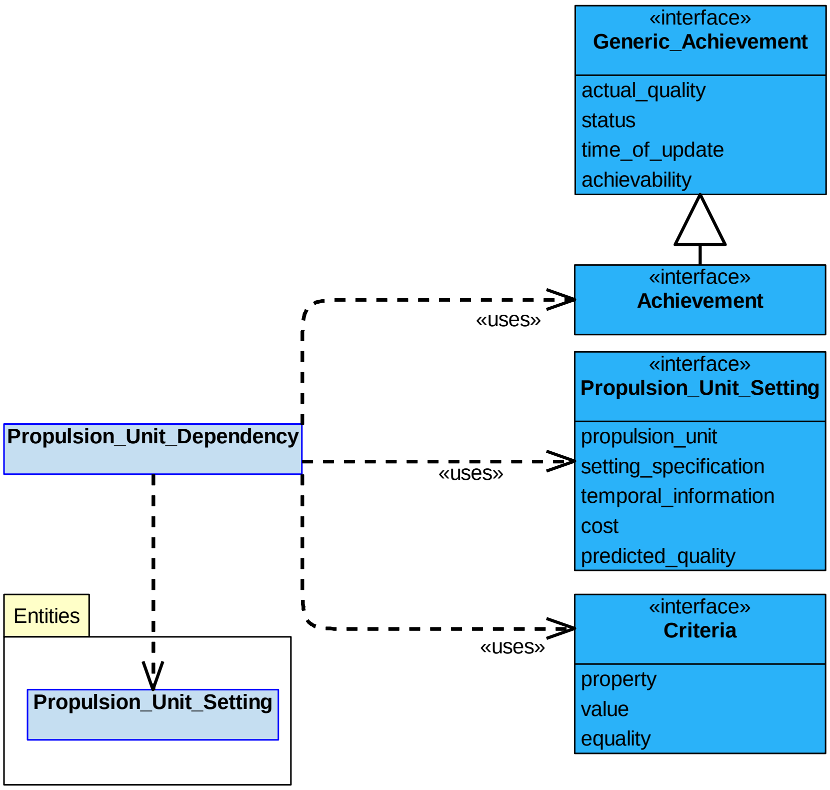
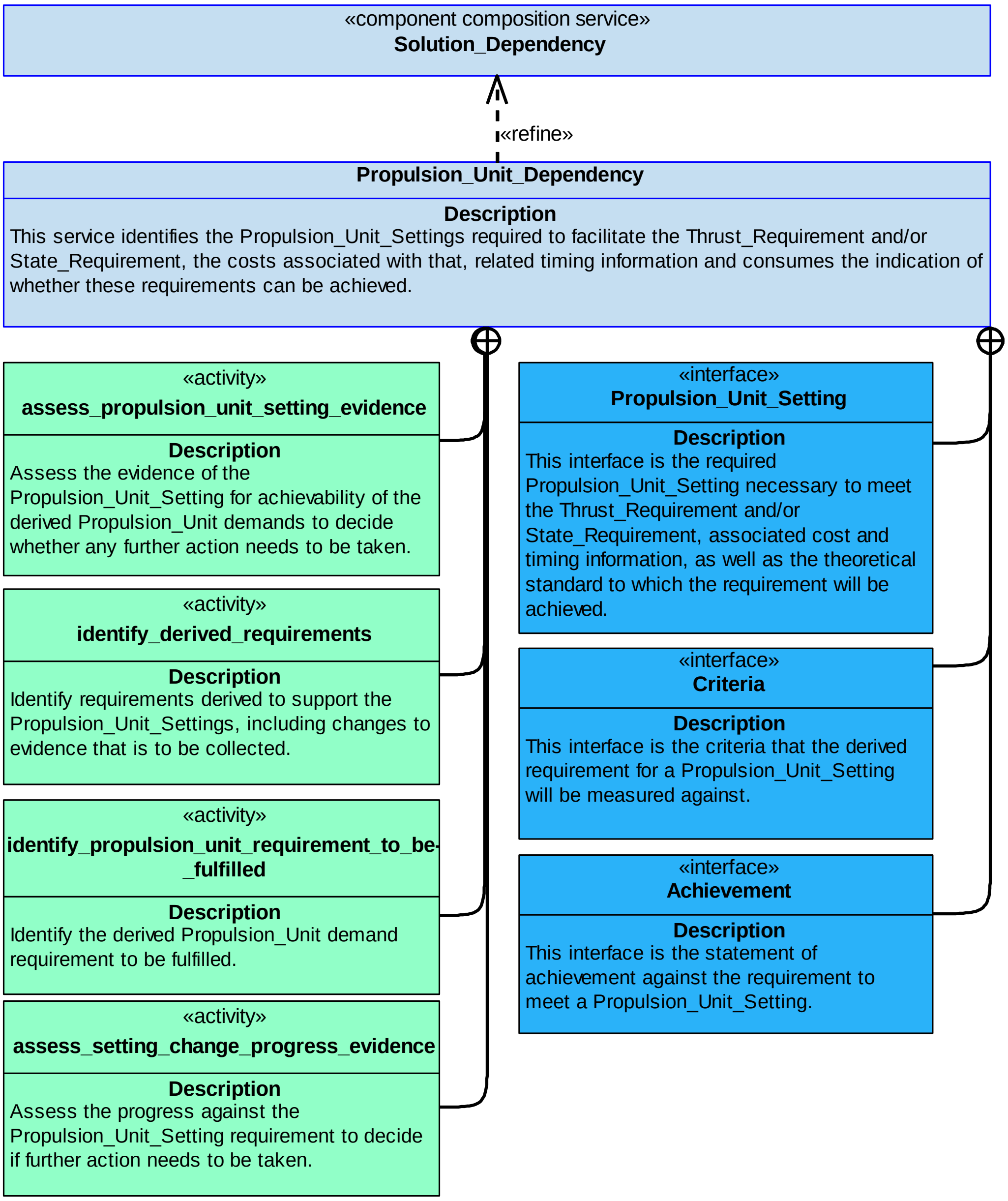


Figure 761: Propulsion_Unit_Dependency Service Definition Diagram


«component composition service»
Solution_Dependency
«refine»
Propulsion_Unit_Dependency

Description
This service identifies the Propulsion_Unit_Settings required to facilitate the Thrust_Requirement and/or State_Requirement, the costs associated with that, related timing information and consumes the indication of whether these requirements can be achieved.

«activity»
assess_propulsion_unit_setting_evidence

Description
Assess the evidence of the Propulsion_Unit_Setting for achievability of the derived Propulsion_Unit demands to decide whether any further action needs to be taken.

Description
Identify requirements derived to support the Propulsion_Unit_Settings, including changes to evidence that is to be collected.

Description
Identify the derived Propulsion_Unit demand requirement to be fulfilled.

Description
Assess the progress against the Propulsion_Unit_Setting requirement to decide if further action needs to be taken.

Description
This interface is the required Propulsion_Unit_Setting necessary to meet the Thrust_Requirement and/or State_Requirement, associated cost and timing information, as well as the theoretical standard to which the requirement will be achieved.

Description
This interface is the criteria that the derived requirement for a Propulsion_Unit_Setting will be measured against.

Description
This interface is the statement of achievement against the requirement to meet a Propulsion_Unit_Setting.

Figure 762: Propulsion_Unit_Dependency Service Policy Diagram

Propulsion_Unit_Dependency

This service identifies the Propulsion_Unit_Settings required to facilitate the Thrust_Requirement and/or State_Requirement, the costs associated with that, related timing information and consumes the indication of whether these requirements can be achieved.

<u>Interfaces</u>

Criteria

This interface is the criteria that the derived requirement for a Propulsion_Unit_Setting will be measured against.

Attributes

property	The Propulsion_Unit property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than or equal to.

Achievement

This interface is the statement of achievement against the requirement to meet a Propulsion_Unit_Setting.

Propulsion_Unit_Setting

This interface is the required Propulsion_Unit_Setting necessary to meet the Thrust_Requirement and/or State_Requirement, associated cost and timing information, as well as the theoretical standard to which the requirement will be achieved.

Attributes

propulsion_unit	The specific Propulsion_Unit the demand is placed on.
setting_specification	The required setting of the Propulsion_Unit.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used.
predicted_quality	How well the Propulsion_Unit_Setting is predicted to satisfy the requirement.

Activities

identify_propulsion_unit_requirement_to_be_fulfilled

Identify the derived Propulsion_Unit demand requirement to be fulfilled.

identify_derived_requirements

Identify requirements derived to support the Propulsion_Unit_Settings, including changes to evidence that is to be collected.

assess_propulsion_unit_setting_evidence

Assess the evidence of the Propulsion_Unit_Setting for achievability of the derived Propulsion_Unit demands to decide whether any further action needs to be taken.

assess_setting_change_progress_evidence

Assess the progress against the Propulsion_Unit_Setting requirement to decide if further action needs to be taken.

5.4.2.44.7.1.6 Environmental_Information

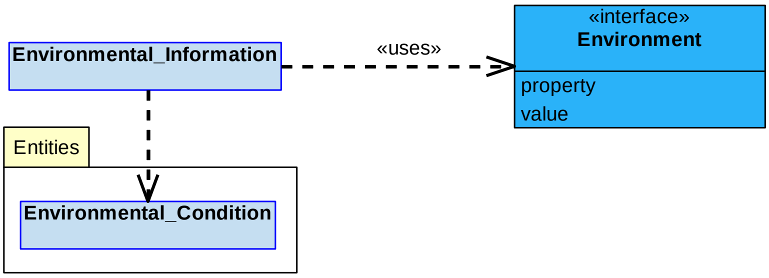


Figure 763: Environmental_Information Service Definition

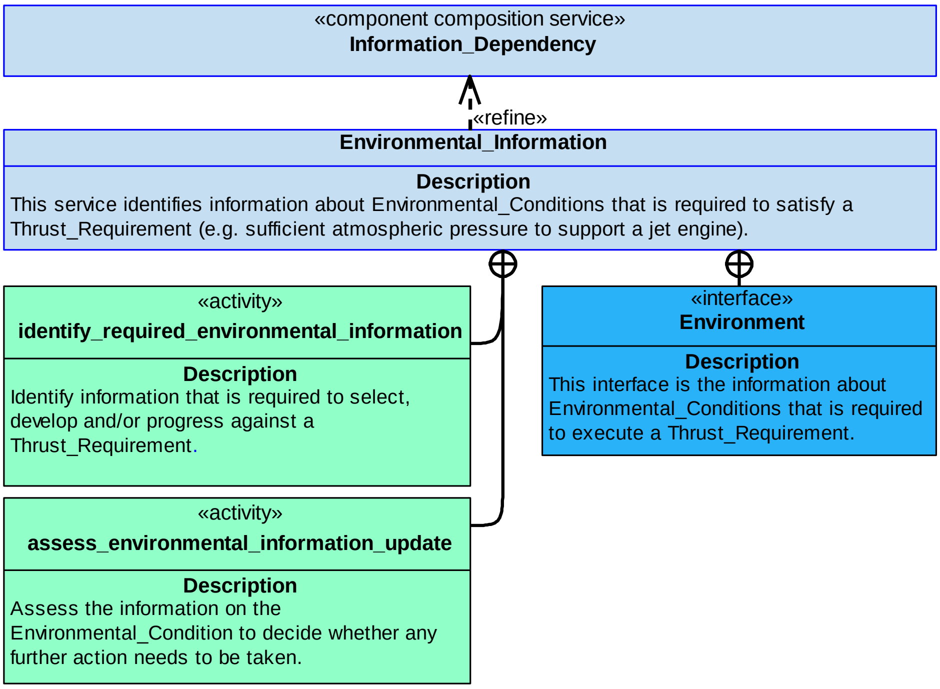

«component composition service»
Information_Dependency
«refine»
Environmental_Information
Description
This service identifies information about Environmental_Conditions that is required to satisfy a Thrust_Requirement (e.g. sufficient atmospheric pressure to support a jet engine).
«activity»
identify_required_environmental_information
Description
Identify information that is required to select, develop and/or progress against a Thrust_Requirement.
«interface»
Environment
Description
This interface is the information about Environmental_Conditions that is required to execute a Thrust_Requirement.
«activity»
assess_environmental_information_update
Description
Assess the information on the Environmental_Condition to decide whether any further action needs to be taken.

Figure 764: Environmental_Information Service Policy

Environmental_Information

This service identifies information about Environmental_Conditions that is required to satisfy a Thrust_Requirement (e.g. sufficient atmospheric pressure to support a jet engine).

Interface

Environment

This interface is the information about Environmental_Conditions that is required to execute a Thrust_Requirement.

Attributes

property	A characteristic of the Environmental_Condition.
value	The value of the property of an Environmental_Condition.

Activities

assess_environmental_information_update

Assess the information on the Environmental_Condition to decide whether any further action needs to be taken.

identify_required_environmental_information

Identify information that is required to select, develop and/or progress against a Thrust_Requirement.

5.4.2.44.7.1.7 Feedback_Information

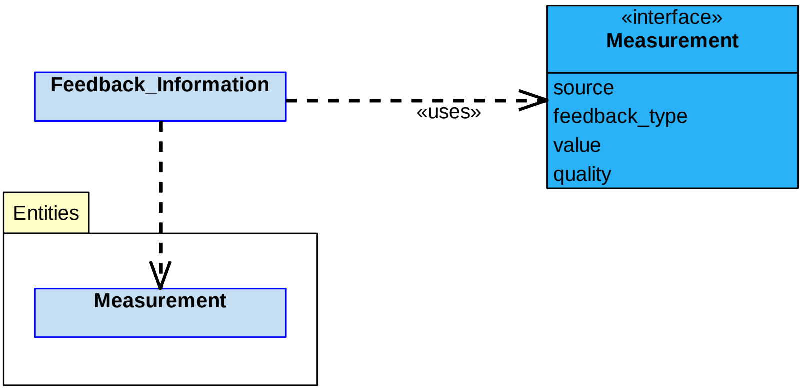


Figure 765: Feedback_Information Service Definition Diagram

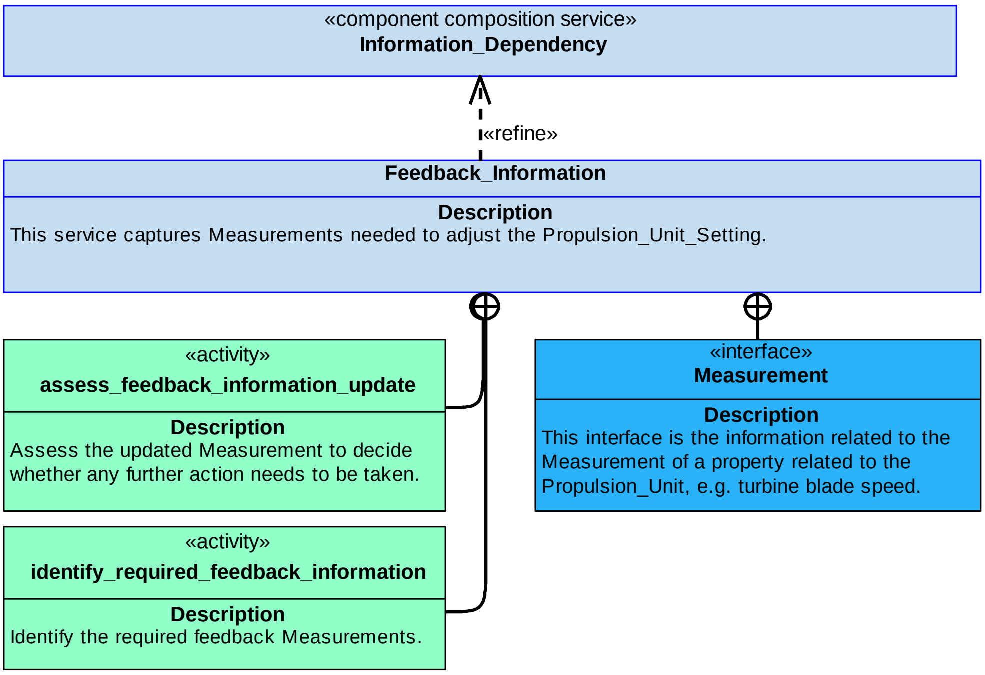

«component composition service»
Information_Dependency
«refine»
Feedback_Information
Description
This service captures Measurements needed to adjust the Propulsion_Unit_Setting.
«activity»
assess_feedback_information_update
Description
Assess the updated Measurement to decide whether any further action needs to be taken.
«interface»
Measurement
Description
This interface is the information related to the Measurement of a property related to the Propulsion_Unit, e.g. turbine blade speed.
«activity»
identify_required_feedback_information
Description
Identify the required feedback Measurements.

Figure 766: Feedback_Information Service Policy Diagram

Feedback_Information

This service captures Measurements needed to adjust the Propulsion_Unit_Setting.

Interface

Measurement

This interface is the information related to the Measurement of a property related to the Propulsion_Unit, e.g. turbine blade speed.

Attributes

source	The source of the Measurement, e.g. a specific sensor and its location.
feedback_type	The property being measured, e.g. flow rate or pressure.
value	The measured value of feedback type.
quality	The quality (e.g. accuracy and certainty) in the provided response.

Activities

assess_feedback_information_update

Assess the updated Measurement to decide whether any further action needs to be taken.

identify_required_feedback_information

Identify the required feedback Measurements.

5.4.2.44.7.1.8 Constraint

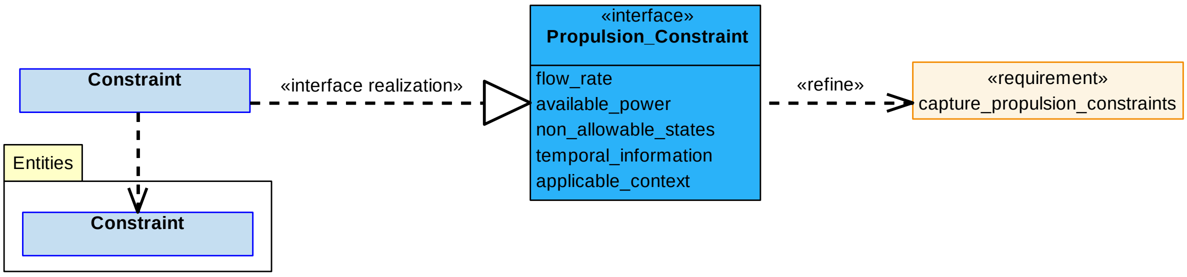


Figure 767: Constraint Service Definition

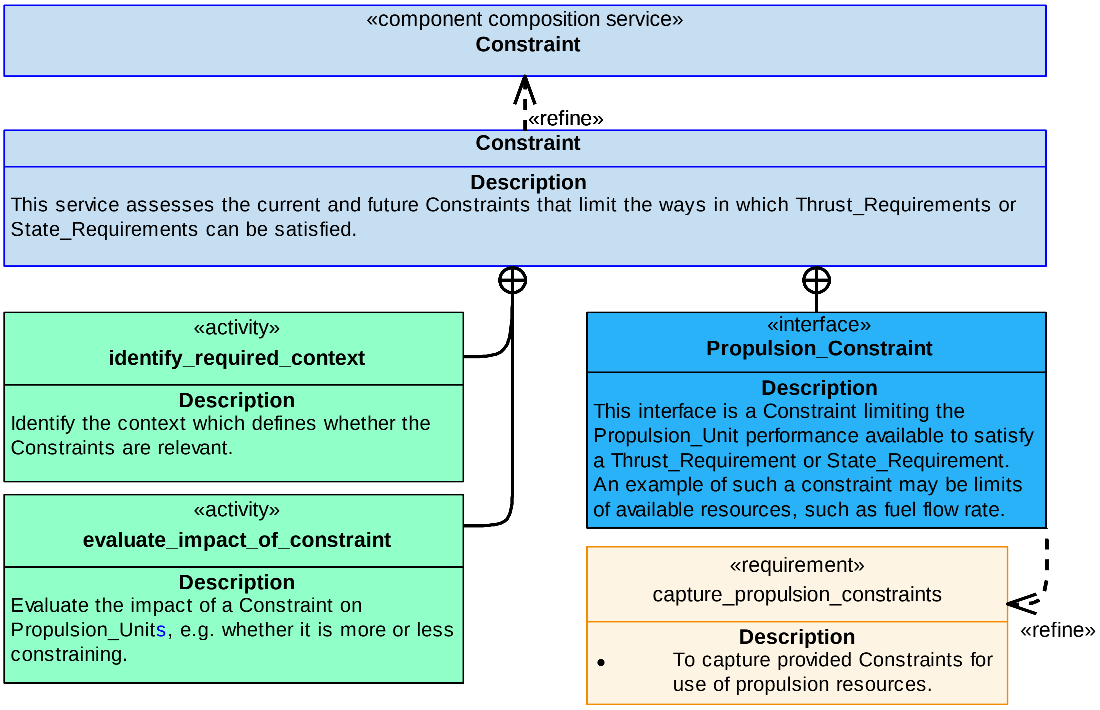

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the current and future Constraints that limit the ways in which Thrust_Requirements or State_Requirements can be satisfied.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of a Constraint on Propulsion_Units, e.g. whether it is more or less constraining.
«interface»
Propulsion_Constraint
Description
This interface is a Constraint limiting the Propulsion_Unit performance available to satisfy a Thrust_Requirement or State_Requirement. An example of such a constraint may be limits of available resources, such as fuel flow rate.
«requirement»
capture_propulsion_constraints
Description
To capture provided Constraints for use of propulsion resources.
«refine»

Figure 768: Constraint Service Policy

Constraint

This service assesses the current and future Constraints that limit the ways in which Thrust_Requirements or State_Requirements can be satisfied.

Interface

Propulsion_Constraint

This interface is a Constraint limiting the Propulsion_Unit performance available to satisfy a Thrust_Requirement or State_Requirement. An example of such a constraint may be limits of available resources, such as fuel flow rate.

Attributes

flow_rate	The maximum flow rate of fuel that is allowed to be used by a Propulsion_Unit.
available_power	The maximum power allowed to be used by a Propulsion_Unit to provide thrust.
non_allowable_states	The non-permitted Propulsion_Unit_Settings and other operational states of the Propulsion_Unit (e.g. a restriction on engine activation).
temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the Constraint is applicable.
propulsion_breach	A statement that the propulsion Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of a Constraint on Propulsion_Units, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.44.7.1.9 Capability

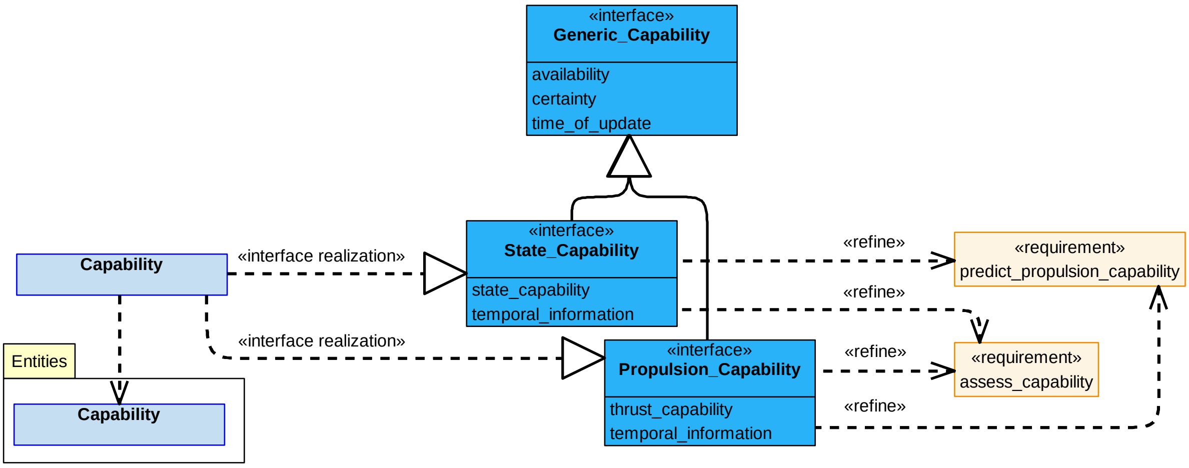


Figure 769: Capability Service Definition

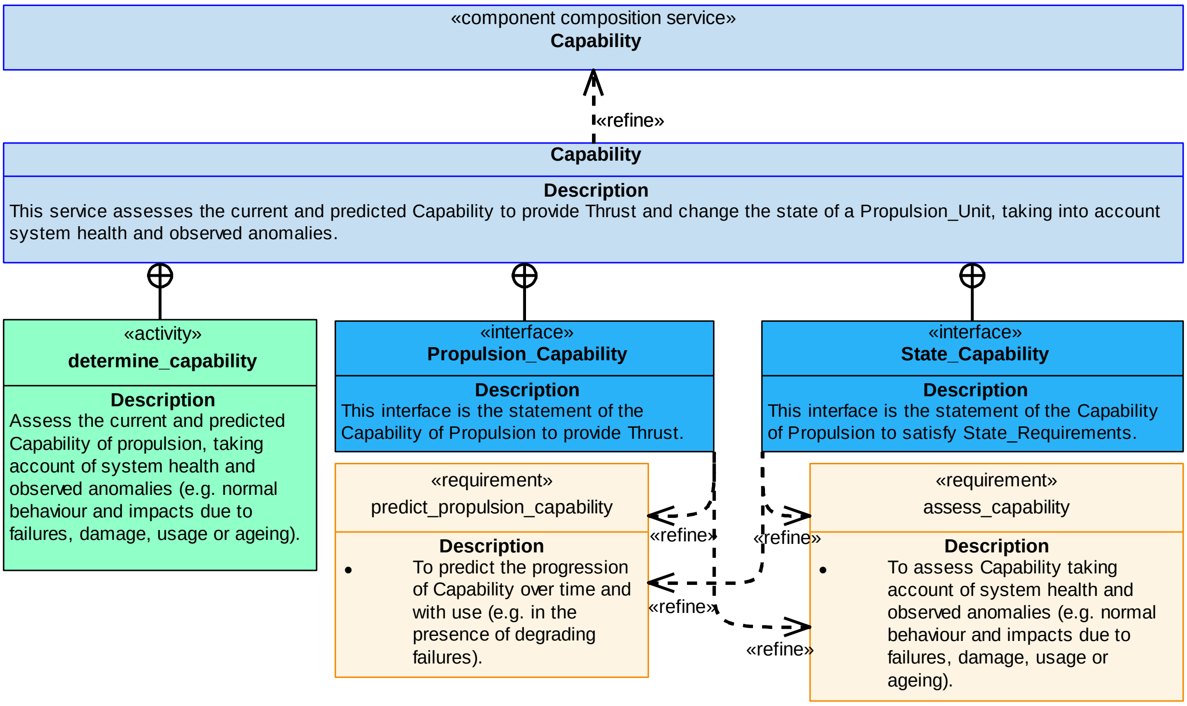

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to provide Thrust and change the state of a Propulsion_Unit, taking into account system health and observed anomalies.
«activity»
determine_capability
Description
Assess the current and predicted Capability of propulsion, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Propulsion_Capability
Description
This interface is the statement of the Capability of Propulsion to provide Thrust.
«interface»
State_Capability
Description
This interface is the statement of the Capability of Propulsion to satisfy State_Requirements.
«requirement»
predict_propulsion_capability
Description
• To predict the progression of Capability over time and with use (e.g. in the presence of degrading failures).
«refine»
«refine»
«refine»
«refine»
«requirement»
assess_capability
Description
• To assess Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Figure 770: Capability Service Policy

Capability

This service assesses the current and predicted Capability to provide Thrust and change the state of a Propulsion_Unit, taking into account system health and observed anomalies.

Interfaces

Propulsion_Capability

This interface is the statement of the Capability of Propulsion to provide Thrust.

Attributes

thrust_capability	The range of Thrust magnitude and direction that capability is being stated against.
temporal_information	Information covering timing of the capability, such as for how long the capability is likely to exist.

State_Capability

This interface is the statement of the Capability of Propulsion to satisfy State_Requirements.

Attributes

state_capability	The Propulsion_Unit state that capability is being stated against.
temporal_information	Information covering timing of the capability, such as for how long the capability is likely to exist.

Activity

determine_capability

Assess the current and predicted Capability of propulsion, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.44.7.1.10 Capability_Evidence

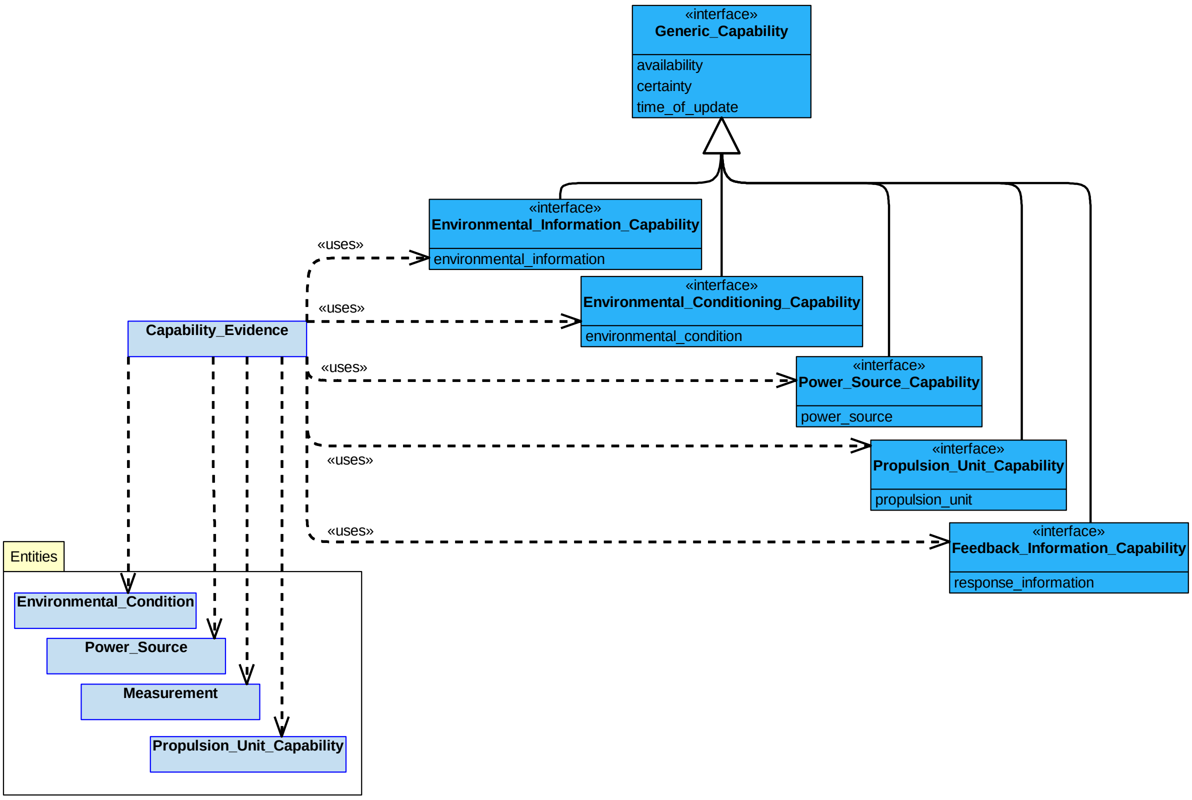


Figure 771: Capability_Evidence Service Definition

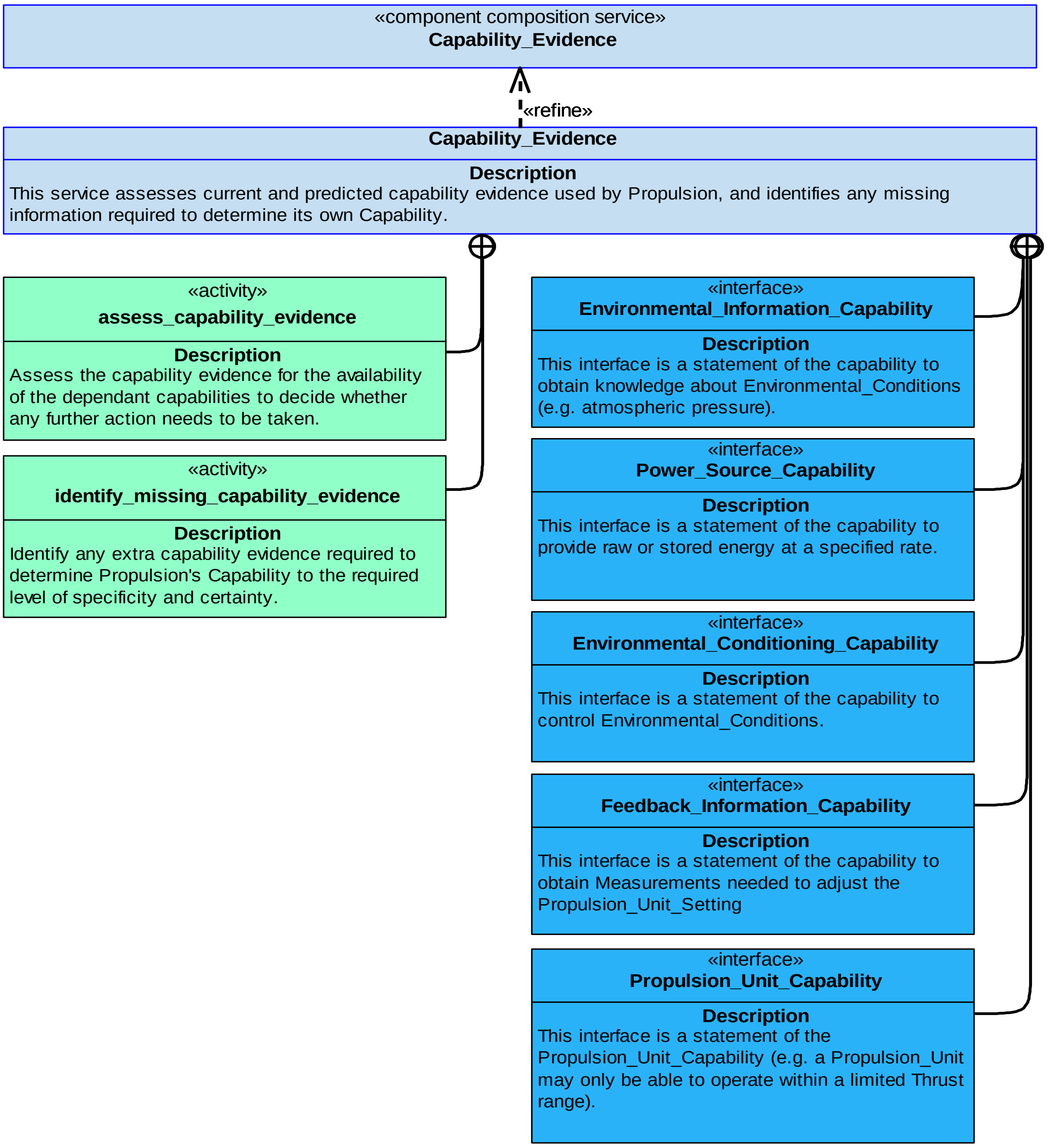

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service assesses current and predicted capability evidence used by Propulsion, and identifies any missing information required to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence for the availability of the dependant capabilities to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine Propulsion's Capability to the required level of specificity and certainty.
«interface»
Environmental_Information_Capability
Description
This interface is a statement of the capability to obtain knowledge about Environmental_Conditions (e.g. atmospheric pressure).
«interface»
Power_Source_Capability
Description
This interface is a statement of the capability to provide raw or stored energy at a specified rate.
«interface»
Environmental_Conditioning_Capability
Description
This interface is a statement of the capability to control Environmental_Conditions.
«interface»
Feedback_Information_Capability
Description
This interface is a statement of the capability to obtain Measurements needed to adjust the Propulsion_Unit_Setting
«interface»
Propulsion_Unit_Capability
Description
This interface is a statement of the Propulsion_Unit_Capability (e.g. a Propulsion_Unit may only be able to operate within a limited Thrust range).

Figure 772: Capability_Evidence Service Policy

Capability_Evidence

This service assesses current and predicted capability evidence used by Propulsion, and identifies any missing information required to determine its own Capability.

Interfaces

Environmental_Conditioning_Capability

This interface is a statement of the capability to control Environmental_Conditions.

Attribute

environmental_condition	The specific Environmental_Condition.

Environmental_Information_Capability

This interface is a statement of the capability to obtain knowledge about Environmental_Conditions (e.g. atmospheric pressure).

Attribute

environmental_information	The specific information about an Environmental_Condition.

Power_Source_Capability

This interface is a statement of the capability to provide raw or stored energy at a specified rate.

Attribute

power_source	The specific Power_Source.

Propulsion_Unit_Capability

This interface is a statement of the Propulsion_Unit_Capability (e.g. a Propulsion_Unit may only be able to operate within a limited Thrust range).

Attribute

propulsion_unit	The specific Propulsion_Unit.

Feedback_Information_Capability

This interface is a statement of the capability to obtain Measurements needed to adjust the Propulsion_Unit_Setting

Attribute

response_information	The type of information relating to the availability of Measurements.

Activities

identify_missing_capability_evidence

Identify any extra capability evidence required to determine Propulsion's Capability to the required level of specificity and certainty.

assess_capability_evidence

Assess the capability evidence for the availability of the dependant capabilities to decide whether any further action needs to be taken.

5.4.2.44.7.2 Service Dependencies

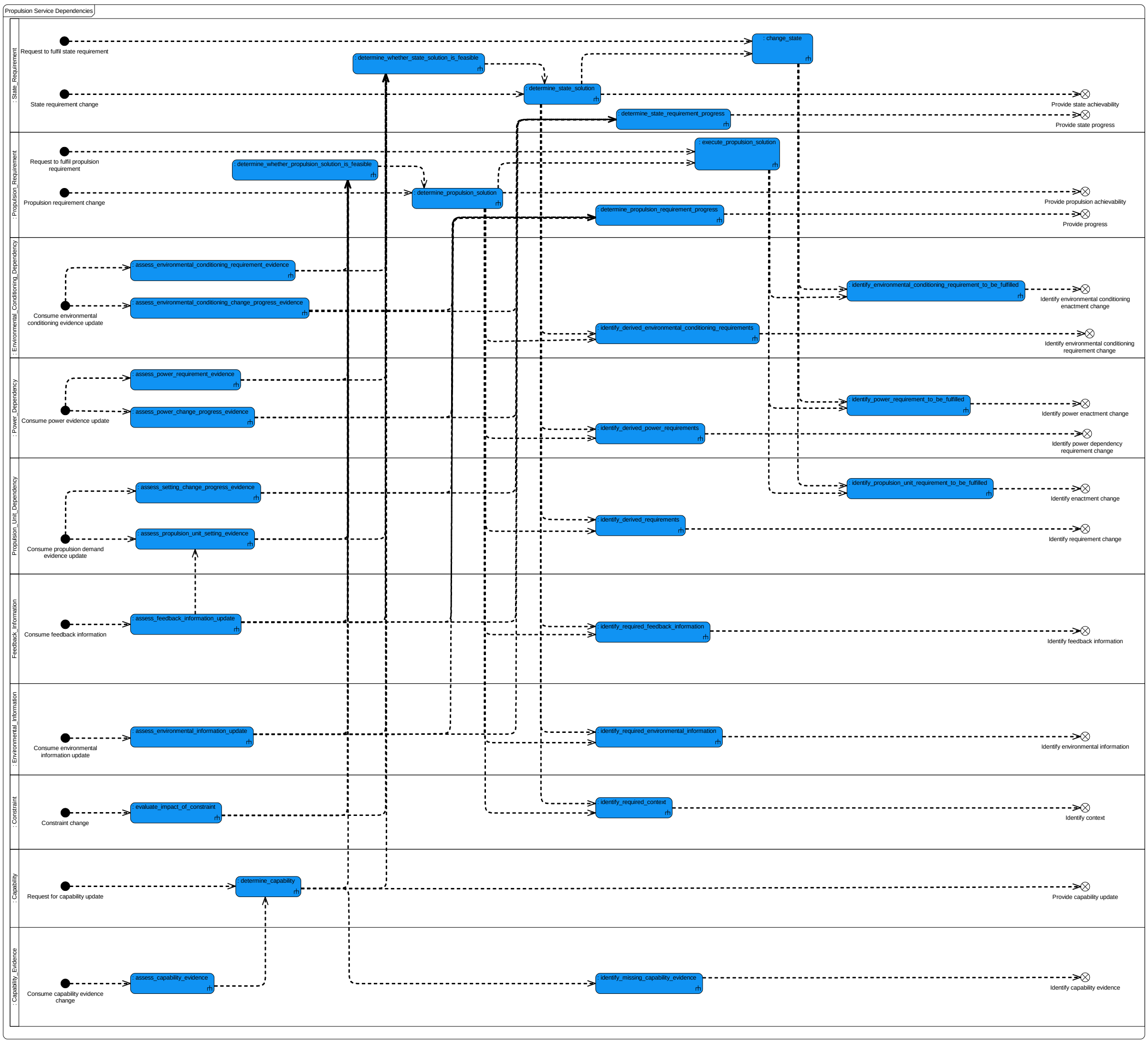

Propulsion Service Dependencies
: State_Requirement
Request to fulfil state requirement
State requirement change
: determine_whether_state_solution_is_feasible
: determine_state_solution
: change_state
: determine_state_requirement_progress
Provide state achievability
Provide state progress
: Propulsion_Requirement
Request to fulfil propulsion requirement
Propulsion requirement change
: determine_whether_propulsion_solution_is_feasible
: determine_propulsion_solution
: execute_propulsion_solution
: determine_propulsion_requirement_progress
Provide propulsion achievability
Provide progress
: Environmental_Conditioning_Dependency
: assess_environmental_conditioning_requirement_evidence
: assess_environmental_conditioning_change_progress_evidence
Consume environmental conditioning evidence update
: identify_environmental_conditioning_requirement_to_be_fulfilled
Identify environmental conditioning enactment change
: identify_derived_environmental_conditioning_requirements
Identify environmental conditioning requirement change
: Power_Dependency
: assess_power_requirement_evidence
: assess_power_change_progress_evidence
Consume power evidence update
: identify_power_requirement_to_be_fulfilled
Identify power enactment change
: identify_derived_power_requirements
Identify power dependency requirement change
: Propulsion_Unit_Dependency
: assess_setting_change_progress_evidence
: assess_propulsion_unit_setting_evidence
Consume propulsion demand evidence update
: identify_propulsion_unit_requirement_to_be_fulfilled
Identify enactment change
: identify_derived_requirements
Identify requirement change
: Feedback_Information
Consume feedback information
: assess_feedback_information_update
: identify_required_feedback_information
Identify feedback information
: Environmental_Information
Consume environmental information update
: assess_environmental_information_update
: identify_required_environmental_information
Identify environmental information
: Constraint
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
: Capability
Request for capability update
: determine_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 773: Propulsion Service Dependencies

5.4.2.45 Reference Times

5.4.2.45.1 Role

The role of Reference Times is to represent the references used to express time and their relationships.

5.4.2.45.2 Overview

Control Architecture

Reference Times is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In order to meet a request placed upon it, the component recommends a Reference_Time for a specific Use taking into account Quality and Confidence. The component also assesses Reference_Times to provide information about the Quality, Confidence or the Difference between two or more Reference_Times.

Examples of Use

Reference Times will be used where:

- Synchronisation of time is required (e.g. with other vehicles and units, external networks, or third party systems).
- Understanding of a time and its relationships to another time is required.
- Understanding of a time and its scope of use is required.
- The quality of a time is required or measured.

5.4.2.45.3 Service Summary

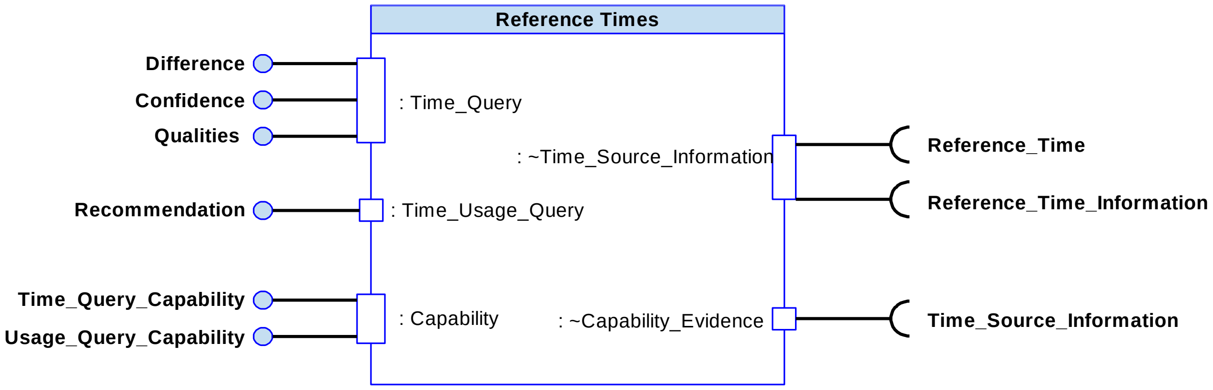


Figure 774: Reference Times Service Summary

5.4.2.45.4 Responsibilities

determine_difference_between_reference_times

- To determine the Difference between Reference_Times.

determine_accuracy_of_a_time

- To determine the Accuracy of a Reference_Time.

determine_confidence_in_a_time

- To determine the reported Confidence in a Reference_Time, relative to one or more of the declared Accuracy or the declared Precision.

determine_choice_of_time

- To determine the Reference_Time for a specific Use.

capture_the_precision_of_time

- To capture the Precision of a Reference_Time.

assess_capability

- To assess the capability to represent Reference_Times, their Quality, and their Use.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of Reference Times Capability over time.

5.4.2.45.5 Subject Matter Semantics

The subject matter of Reference Times is the Use of and Differences between Reference_Times.

Exclusions

The subject matter of Reference Times does not include:

- The measurement of the passing of time or the provision of time, e.g. clocks, pulses, ticks.

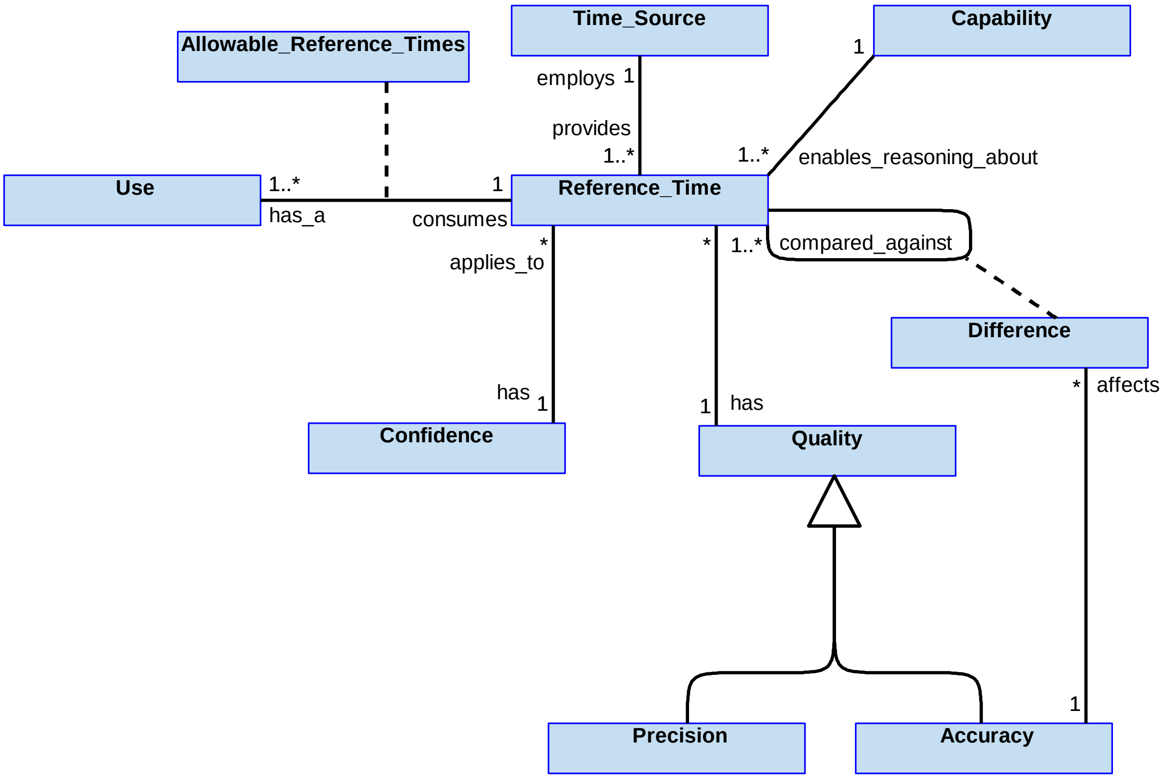


Figure 775: Reference Times Semantics

5.4.2.45.5.1 Entities

Accuracy

The degree to which a time measurement aligns to a Reference_Time. This may be stated or determined.

Allowable_Reference_Times

Rules for allowable Use of a Reference_Time, e.g. GPS derived time being acceptably used as a system wide Reference_Time.

Capability

The capability to represent Reference_Times and their associated details including source, quality, confidence, use and differences.

Confidence

The trust in a Reference_Time, e.g. a level of reliability such as the stratum value in Network Time Protocol or a designated master source.

Difference

The difference in time between two Reference_Times. This may be pre-defined (e.g. number of hours between time zones) or determined as a measured deviation from a specified reference time (clock source).

Precision

The stated granularity of the reference time, e.g. the period of the clock frequency.

Quality

The measureable characteristics of a time, e.g. accuracy and precision.

Reference_Time

A time reference, including the standards and rules of that time, e.g. a time zone, GPS time, system time, simulator time, log time, GMT, or UTC.

Time_Source

Where a time is generated or supplied, for example an atomic clock, satellite time code, GPS clock, or local node clock.

Use

What a time is used for or applied to (e.g. in a GPS, system, node or log).

5.4.2.45.6 Design Rationale

5.4.2.45.6.1 Assumptions

- This component supports the synchronisation of clock times, by providing the Differences (e.g. time offsets), it does not provide specific time values (clock times).
- Clocks are provided by the infrastructure (e.g. an on-board clock) or external devices (e.g. a Global Navigation Satellite System (GNSS) or radio clock).

5.4.2.45.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Reference Times:

- Data Driving - Possible sources of time, confidence, precision, and rules may be data-driven.

Extensions

- It is not considered likely that extension components will be required.

Other Factors that were Taken into Account

- Any particular Reference_Time may not be continuously updated or maintained. The last Reference_Time update could remain available for use but may have a decreasing Quality or Confidence.

Exploitation Considerations

- As time may need to be managed at each computing element or node, an instance of Reference Times may be required at each node of a system.

- An instance of this component may need to support synchronisation with a different instance of Reference Times in another node.
- Individual Reference_Times may be inaccurate or include unexpected discontinuities.
- Multiple Reference_Times may be used to determine a consolidated Reference_Time using weightings (e.g. based on Quality and Confidence).
- This component will determine the Differences between the clocks of two time references, but it does not provide any clocks.

5.4.2.45.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

- Failure of this component to provide the correct time Differences in support of synchronisation may result in hazards where time is used to de-conflict aircraft. For mid-air collision, the safety analysis can take credit for ACAS and so is not the driver. However, where time is used for ensuring an air vehicle does not fly into the fragmentation zone of a weapon released by another air vehicle during a coordinated attack, corruption of time synchronisation may result in fatalities of the air vehicle's occupants and an uncontrolled crash of the air vehicle (with possible third party fatalities). As per the Safety Analysis PYRAMID concept guidance, this drives an IDAL of B.

5.4.2.45.6.4 Security Considerations

The indicative security classification is O.

This component identifies Reference_Times (e.g. system, absolute or local) for use by the Exploiting Platform; using time zones and general area information instead of precise location will allow the component to remain O. There may need to be instances of this component in each node, these may include different security domains; these instances may need to be synchronised. The integrity of this component is key to the audit processes through its use for synchronisation, time stamping and sequencing, etc. Time spoofing can have a major impact on mission effectiveness.

The component is expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** used for determining time data as being from allowable sources.
- **Logging of Security Data** relating to the possible tampering or interruptions in time reporting.
- **Maintaining Audit Records** of changes to time zones or offsets due to mission activities.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** relating to the accuracy of time sources, an unexpected shift in reported time or loss of accuracy may indicate time sources have been compromised.

The component is considered to at least partially satisfy Security Enforcing Functions by:

- **Verifying Integrity of Data**; it is expected this component will detect Differences in Quality and be able to reconfigure to mitigate spoofed time source data etc.

5.4.2.45.7 Services

5.4.2.45.7.1 Service Definitions

5.4.2.45.7.1.1 Time_Query

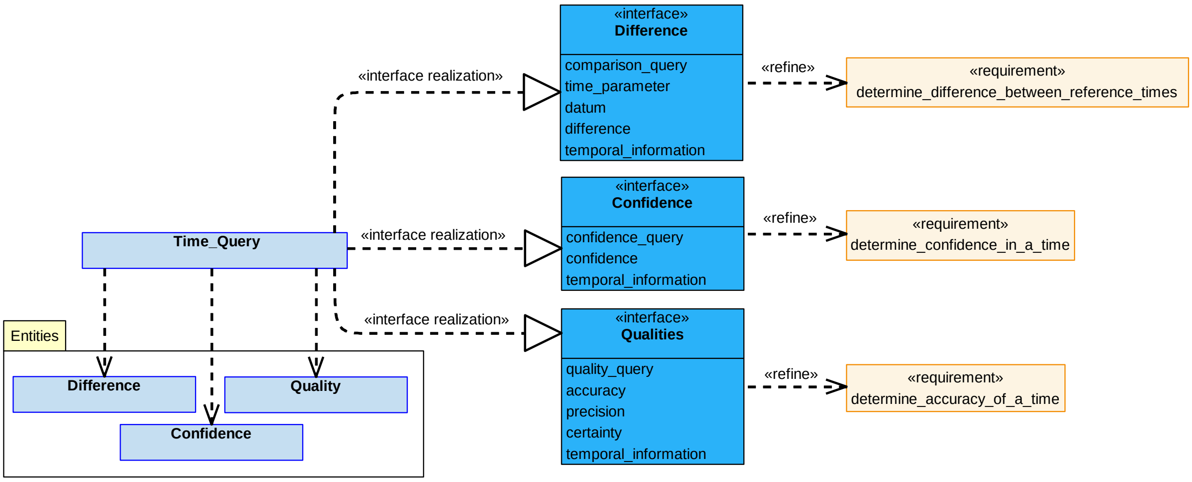
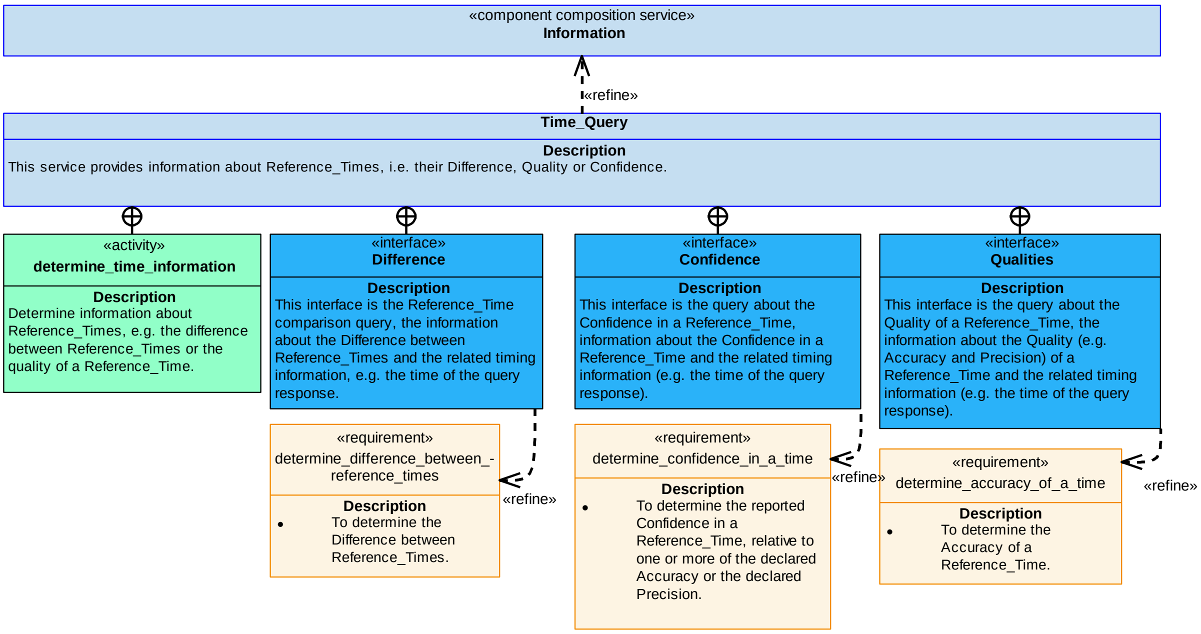


Figure 776: Time_Query Service Definition



This service provides information about Reference_Times, i.e. their Difference, Quality or Confidence.

Determine information about Reference_Times, e.g. the difference between Reference_Times or the quality of a Reference_Time.

This interface is the Reference_Time comparison query, the information about the Difference between Reference_Times and the related timing information, e.g. the time of the query response.

This interface is the query about the Confidence in a Reference_Time, information about the Confidence in a Reference_Time and the related timing information (e.g. the time of the query response).

This interface is the query about the Quality of a Reference_Time, the information about the Quality (e.g. Accuracy and Precision) of a Reference_Time and the related timing information (e.g. the time of the query response).

- To determine the Difference between Reference_Times.

- To determine the reported Confidence in a Reference_Time, relative to one or more of the declared Accuracy or the declared Precision.

- To determine the Accuracy of a Reference_Time.

Figure 777: Time_Query Service Policy

Time_Query

This service provides information about Reference_Times, i.e. their Difference, Quality or Confidence.

Interfaces

Difference

This interface is the Reference_Time comparison query, the information about the Difference between Reference_Times and the related timing information, e.g. the time of the query response.

Attributes

comparison_query	The definition of the query to compare Reference_Times.
time_parameter	The Reference_Time being compared, e.g. the time instance.
datum	Information about a specific Reference_Time used as a datum for the comparison.
difference	The identified difference between the datum and time_parameter.
temporal_information	Information relating to the timing of the query and the response, e.g. the time of the query response or duration for which the response is valid.

Confidence

This interface is the query about the Confidence in a Reference_Time, information about the Confidence in a Reference_Time and the related timing information (e.g. the time of the query response).

Attributes

confidence_query	The definition of the query to determine the Confidence in a Reference_Time.
confidence	The determined Confidence in a Reference_Time.
temporal_information	Information relating to the timing of the query and the response, e.g. the time of the query response or duration for which the response is valid.

Qualities

This interface is the query about the Quality of a Reference_Time, the information about the Quality (e.g. Accuracy and Precision) of a Reference_Time and the related timing information (e.g. the time of the query response).

Attributes

quality_query	The definition of the query to determine the Quality (e.g. Accuracy and Precision) of a Reference_Time.
accuracy	The determined Accuracy of a Reference_Time.
precision	The determined Precision of a Reference_Time.
certainty	The degree of certainty in the determined Quality.
temporal_information	Information relating to the timing of the query and the response, e.g. the time of the query response or duration for which the response is valid.

Activity

determine_time_information

Determine information about Reference_Times, e.g. the difference between Reference_Times or the quality of a Reference_Time.

5.4.2.45.7.1.2 Time_Usage_Query

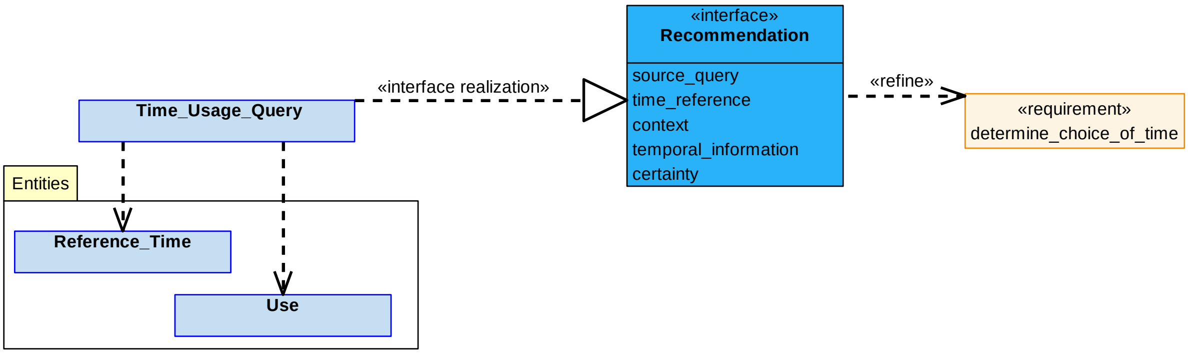


Figure 778: Time_Usage_Query Service Definition

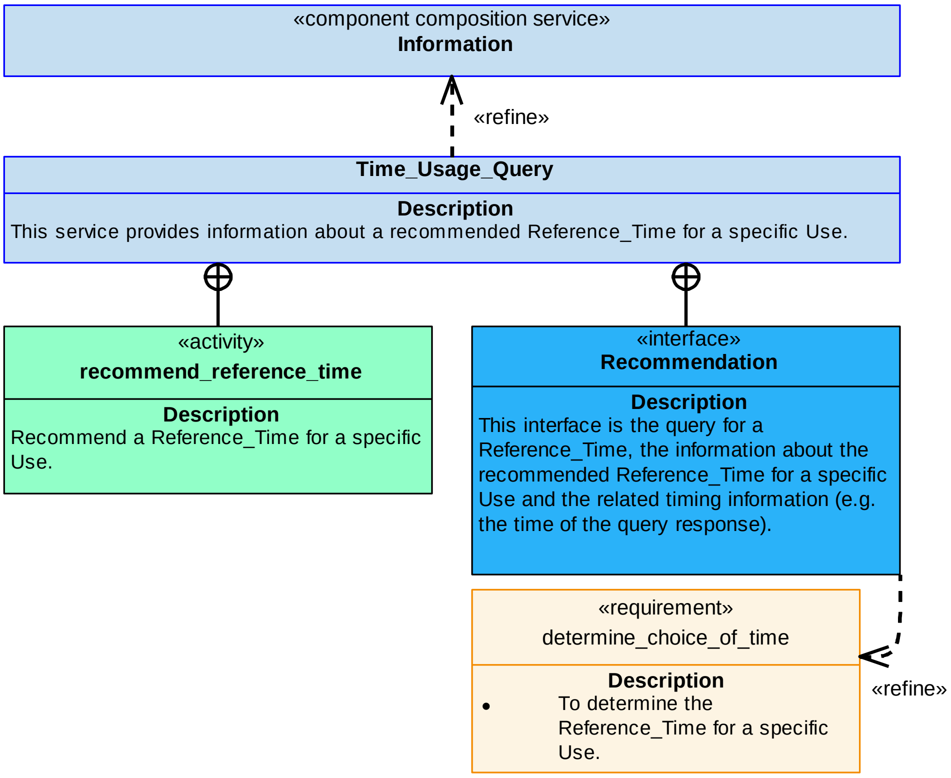

«component composition service»
Information
«refine»
Time_Usage_Query
Description
This service provides information about a recommended Reference_Time for a specific Use.
«activity»
recommend_reference_time
Description
Recommend a Reference_Time for a specific Use.
«interface»
Recommendation
Description
This interface is the query for a Reference_Time, the information about the recommended Reference_Time for a specific Use and the related timing information (e.g. the time of the query response).
«requirement»
determine_choice_of_time
Description
• To determine the Reference_Time for a specific Use.
«refine»

Figure 779: Time_Usage_Query Service Policy

Time_Usage_Query

This service provides information about a recommended Reference_Time for a specific Use.

Interface

Recommendation

This interface is the query for a Reference_Time, the information about the recommended Reference_Time for a specific Use and the related timing information (e.g. the time of the query response).

Attributes

source_query	The definition of the query for the most appropriate Reference_Time.
time_reference	The recommended Reference_Time.
context	The context in which the recommendation applies.
temporal_information	Information relating to the timing of the query and/or recommendation, e.g. the time of the query response or duration for which the response is valid.
certainty	The degree of certainty in the recommended time.

Activity

recommend_reference_time

Recommend a Reference_Time for a specific Use.

5.4.2.45.7.1.3 Time_Source_Information

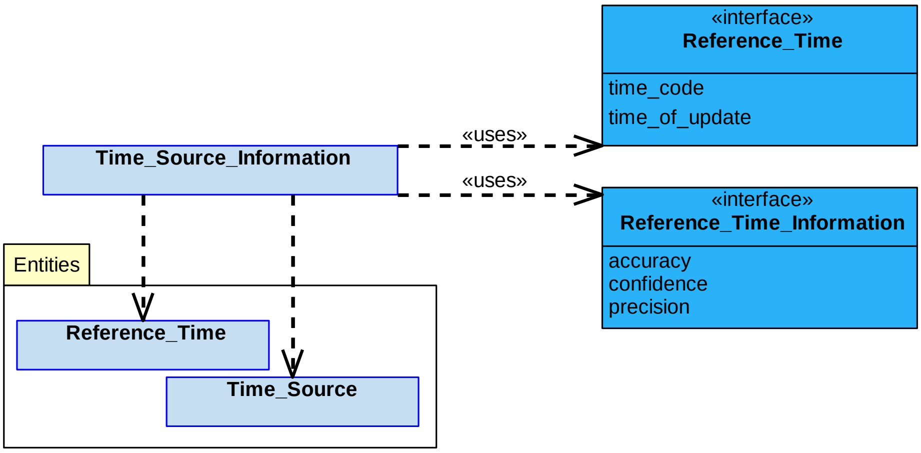


Figure 780: Time Source Information Service Definition

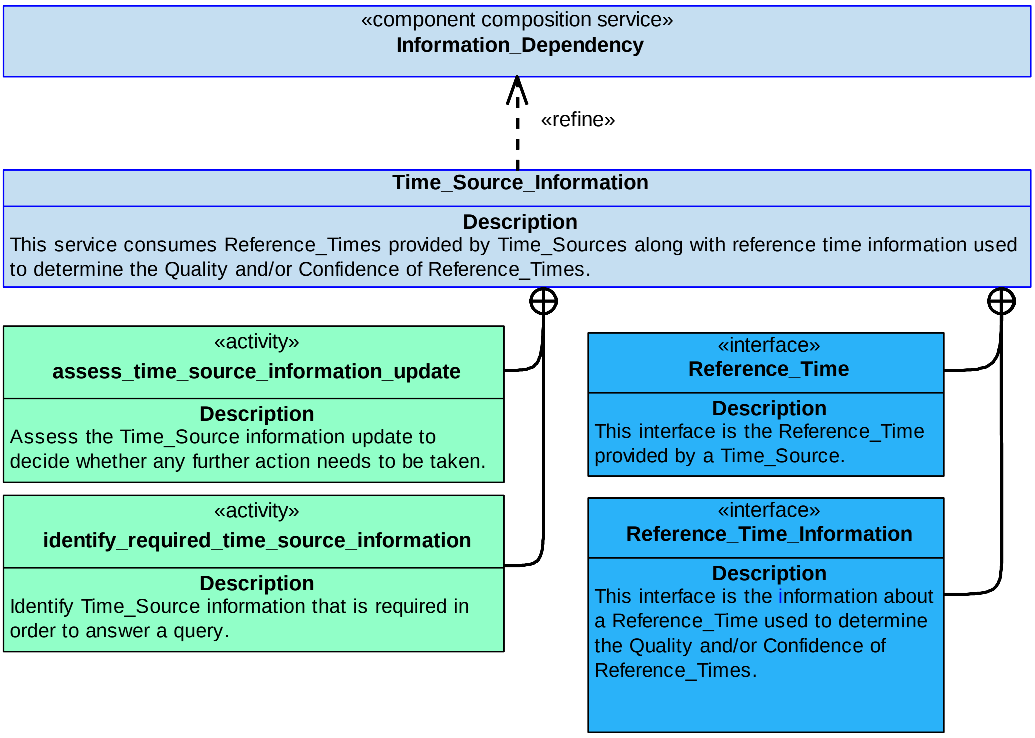

«component composition service»
Information_Dependency
«refine»
Time_Source_Information
Description
This service consumes Reference_Times provided by Time_Sources along with reference time information used to determine the Quality and/or Confidence of Reference_Times.
«activity»
assess_time_source_information_update
Description
Assess the Time_Source information update to decide whether any further action needs to be taken.
«activity»
identify_required_time_source_information
Description
Identify Time_Source information that is required in order to answer a query.
«interface»
Reference_Time
Description
This interface is the Reference_Time provided by a Time_Source.
«interface»
Reference_Time_Information
Description
This interface is the information about a Reference_Time used to determine the Quality and/or Confidence of Reference_Times.

Figure 781: Time Source Information Service Policy

Time_Source_Information

This service consumes Reference_Times provided by Time_Sources along with reference time information used to determine the Quality and/or Confidence of Reference_Times.

Interfaces

Reference_Time

This interface is the Reference_Time provided by a Time_Source.

Attributes

time_code	The value of a particular instance in time.
time_of_update	When the time code was received or taken.

Reference_Time_Information

This interface is the information about a Reference_Time used to determine the Quality and/or Confidence of Reference_Times.

Attributes

accuracy	The degree to which a time measurement aligns to the passing of time.
confidence	The degree of certainty in the Reference_Time.
precision	The granularity of the time code.

Activities

assess_time_source_information_update

Assess the Time_Source information update to decide whether any further action needs to be taken.

identify_required_time_source_information

Identify Time_Source information that is required in order to answer a query.

5.4.2.45.7.1.4 Capability

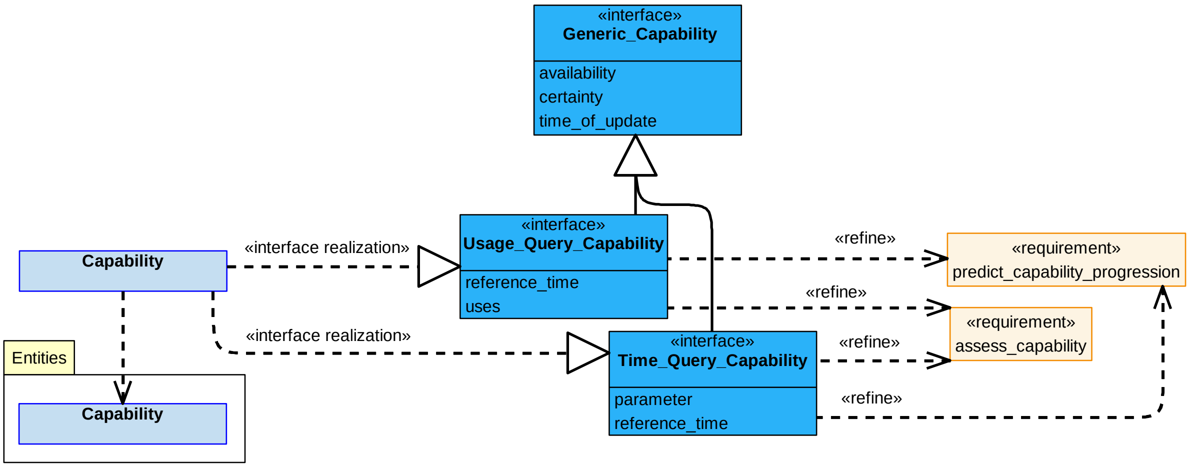


Figure 782: Capability Service Definition

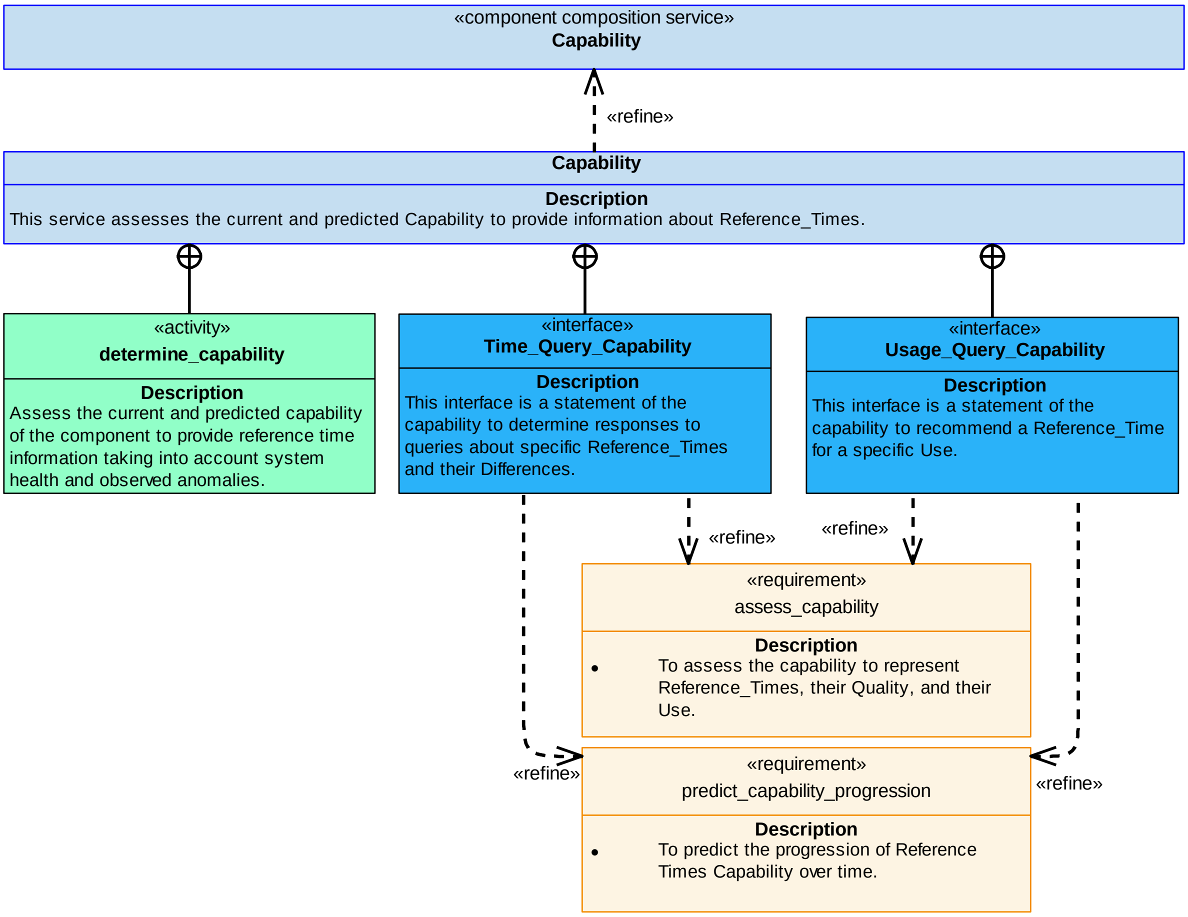

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to provide information about Reference_Times.
«activity»
determine_capability
Description
Assess the current and predicted capability of the component to provide reference time information taking into account system health and observed anomalies.
«interface»
Time_Query_Capability
Description
This interface is a statement of the capability to determine responses to queries about specific Reference_Times and their Differences.
«interface»
Usage_Query_Capability
Description
This interface is a statement of the capability to recommend a Reference_Time for a specific Use.
«refine»
«refine»
«requirement»
assess_capability
Description
To assess the capability to represent Reference_Times, their Quality, and their Use.
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of Reference Times Capability over time.
«refine»

Figure 783: Capability Service Policy

Capability

This service assesses the current and predicted Capability to provide information about Reference_Times.

Interfaces

Time_Query_Capability

This interface is a statement of the capability to determine responses to queries about specific Reference_Times and their Differences.

Attributes

parameter	The property of Reference_Time that can be determined and provided in response to a query, e.g. Accuracy, Difference or Confidence.
reference_time	The Reference_Times the component is able to compare with, e.g. the component is able to compare a given time with GPS time.

Usage_Query_Capability

This interface is a statement of the capability to recommend a Reference_Time for a specific Use.

Attributes

reference_time	The Reference_Times the component is able to recommend, e.g. GPS time.
uses	The Uses that can be supported, e.g. local time or mission time.

Activity

determine_capability

Assess the current and predicted capability of the component to provide reference time information taking into account system health and observed anomalies.

5.4.2.45.7.1.5 Capability_Evidence

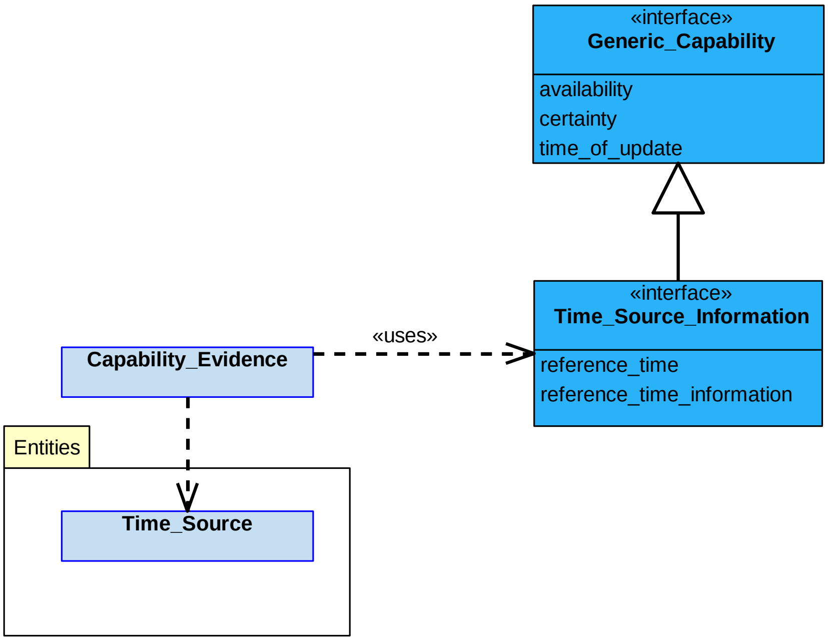


Figure 784: Capability_Evidence Service Definition

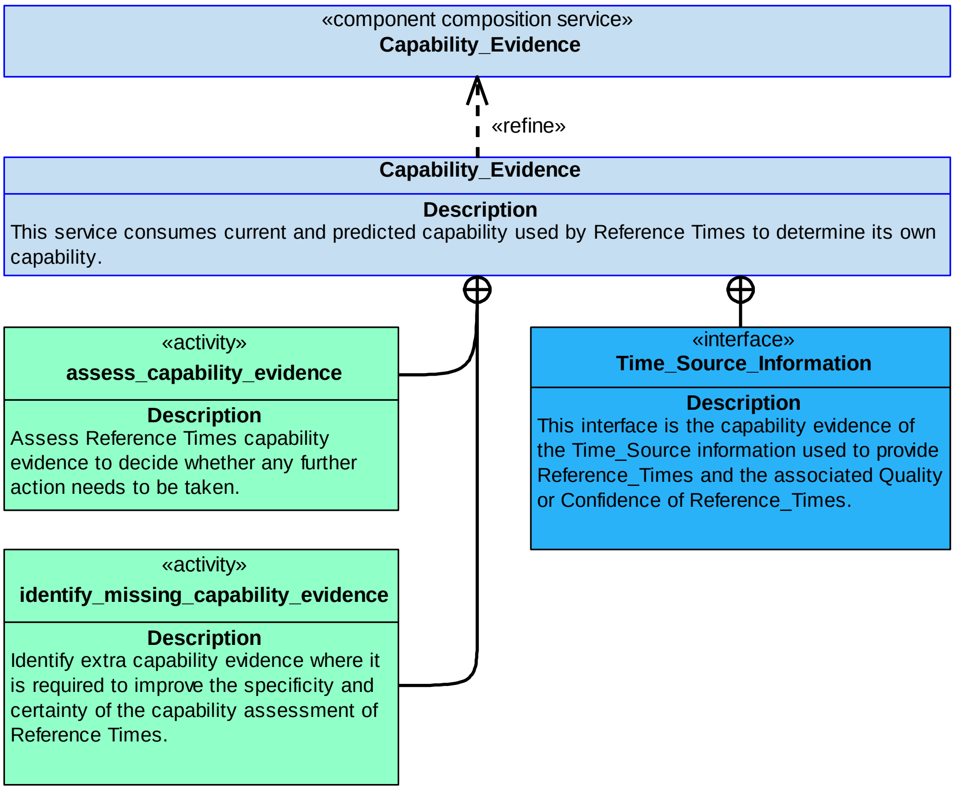

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability used by Reference Times to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess Reference Times capability evidence to decide whether any further action needs to be taken.
«interface»
Time_Source_Information
Description
This interface is the capability evidence of the Time_Source information used to provide Reference_Times and the associated Quality or Confidence of Reference_Times.
«activity»
identify_missing_capability_evidence
Description
Identify extra capability evidence where it is required to improve the specificity and certainty of the capability assessment of Reference Times.

Figure 785: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability used by Reference Times to determine its own capability.

Interface

Time_Source_Information

This interface is the capability evidence of the Time_Source information used to provide Reference_Times and the associated Quality or Confidence of Reference_Times.

Attributes

reference_time	The availability of a Reference_Time provided by a Time_Source.
reference_time_information	The availability and freshness of Quality and/or Confidence information.

Activities

assess_capability_evidence

Assess Reference Times capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify extra capability evidence where it is required to improve the specificity and certainty of the capability assessment of Reference Times.

5.4.2.45.7.2 Service Dependencies

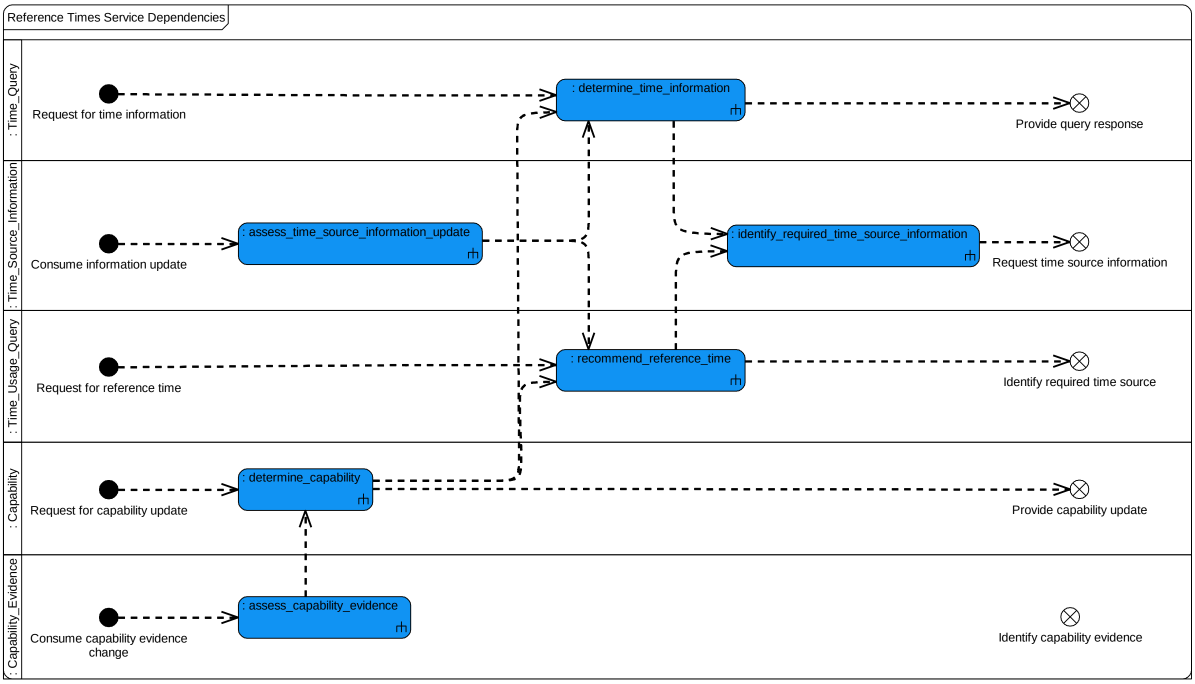

Reference Times Service Dependencies
: Time_Query
: Time_Source_Information
: Time_Usage_Query
: Capability
: Capability_Evidence
Request for time information
: determine_time_information
Provide query response
Consume information update
: assess_time_source_information_update
: identify_required_time_source_information
Request time source information
Request for reference time
: recommend_reference_time
Identify required time source
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
Identify capability evidence

Figure 786: Reference Times Service Dependencies

5.4.2.46 Release Aiming

5.4.2.46.1 Role

The role of Release Aiming is to determine a targeting solution for a given store (e.g. a missile, sonobuoy, or cargo store) for a specific aiming scenario.

5.4.2.46.2 Overview

Control Architecture

Release Aiming is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In response to a set of Requirements and Constraints, Release Aiming determines the Aiming_Solution for a Store between a Release_Point and an Aim_Point, taking account of the Vehicle_Condition and the Environmental_Conditions. Release Aiming can determine an Aim_Point from a provided Release_Point or vice-versa.

Examples of Use

Release Aiming will be used where determination of the store release point/area/zone or the expected store arrival point/area/zone needs to be determined. This can include different types of air to air and air to surface weapons, as well as other deployable assets, such as deployable sensors (e.g. sonobuoys). For example, it can be used to:

- Determine the launch region or ranges within which an air to air missile is capable of successfully hitting a specified target based on the current launch aircraft conditions, the current target conditions and the performance capabilities of the weapon. Different launch regions or ranges may be determined based on different criteria for the likelihood of a successful engagement.
- Determine the impact point of a free fall (ballistic) bomb if released ‘now’ based on the current launch aircraft conditions.
- Determine the viable release region within which a guided bomb can be released, whilst being capable of hitting a specified static target, based on the current launch aircraft conditions and the performance capabilities of the weapon.

5.4.2.46.3 Service Summary

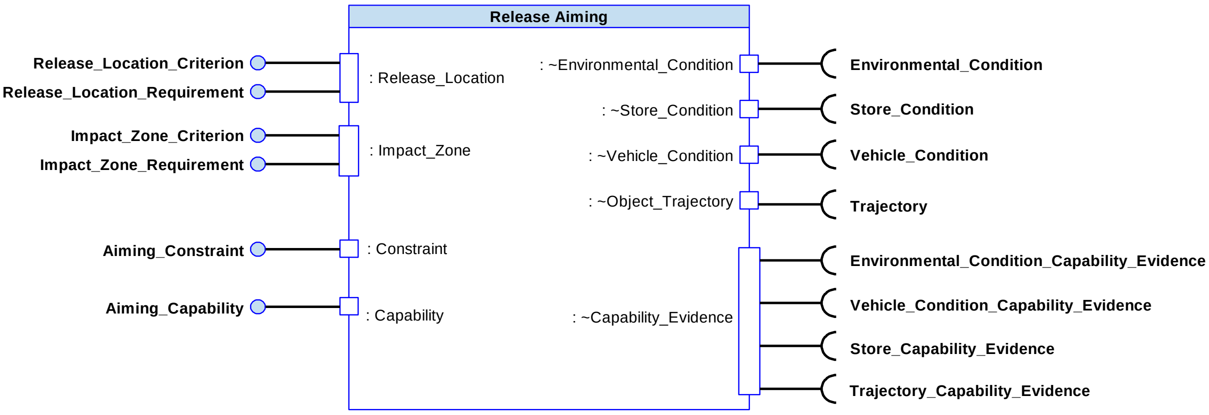


Figure 787: Release Aiming Service Summary

5.4.2.46.4 Responsibilities

capture_release_requirements

- To capture the Requirements to be fulfilled by the Aiming_Solution.

capture_measurement_criteria

- To capture provided Measurement_Criterion/criteria for Aiming_Solutions.

capture_release_constraints

- To capture the externally imposed Constraints that limit where or how the store can be released.

determine_aiming_solutions

- To determine Aiming_Solutions.

determine_predicted_quality_of_aiming_solution

- To determine the predicted quality of the Aiming_Solution against provided Measurement_Criterion/criteria.

identify_aiming_solution_in_progress_remains_feasible

- To identify whether an Aiming_Solution in progress remains feasible given current resources.

assess_capability

- To assess the Capability to provide Release Aiming's services taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_aiming_capability_progression

- To predict the progression of Release Aiming's aiming Capability over time and with use.

5.4.2.46.5 Subject Matter Semantics

The subject matter of Release Aiming is the Release_Point, and the Aim_Point of a Store, and the Aiming_Solution that connects the two.

Exclusions

The subject matter of Release Aiming does not include:

- The control or pointing of the Store.
- The selection between different viable release options that are calculated.
- The release of the Store.
- The prediction of the in-flight position of a previously released Store.
- The calculation of the exact geometric shape of a Store's flight path.

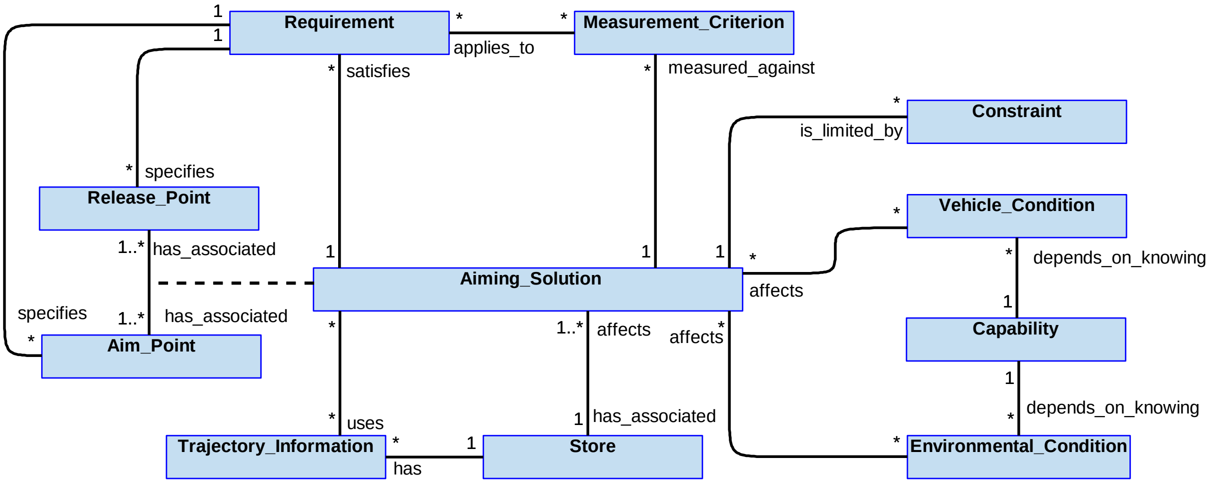


Figure 788: Release Aiming Semantics

5.4.2.46.5.1 Entities

Aiming_Solution

The proposed release solution. This may include the predicted accuracy of the strike and the level of certainty of the modelling.

Capability

A measurement of the capability of the component to provide the release aiming for a store. This will be influenced by system stability and the availability/accuracy of input data.

Constraint

An externally imposed restriction that limits the aiming solution, e.g. a no-impact zone or no-fly zone.

Environmental_Condition

Environmental conditions that are relevant to the release, e.g. wind speed or wind direction.

Aim_Point

The point or region at which the store has been targeted (e.g. a ground target, an enemy air contact, a required store splash point in the water or an area a weapon is capable of impacting). This may not be the terminal impact point; it might instead be the point at which a weapon's terminal guidance takes over.

Measurement_Criterion

A criterion to measure the solution against.

Release_Point

The point or region in which the release is to be performed (e.g. the current position of the vehicle, a pre-defined release point, or the calculated release point to strike a target).

Requirement

The set of requirements to be met when determining the aiming solution. For example, this set may include parameters such as loft release, dive toss, required impact angle, or target to engage.

Store

The object to be aimed, e.g. a bomb, missile, cargo container or sonobuoy.

Vehicle_Condition

A property of the launch vehicle that is relevant to the release, e.g. flight path angle, airspeed, altitude or a degradation state of a subsystem.

Trajectory_Information

Information relating to the trajectory that an object is expected to follow (e.g. the point a target will be in range).

5.4.2.46.6 Design Rationale

5.4.2.46.6.1 Assumptions

- Different algorithms will be required to calculate the Aiming_Solution for different store types, e.g. for A/A missiles or sonobuoys.
- Different varieties of a type of store (e.g. AMRAAM or Meteor A/A missiles) will have different performance data.
- This component will often require data about the intended target, such as position and speed, to generate an Aiming_Solution.

5.4.2.46.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Release Aiming:

- Data Driving - Release Aiming follows this PYRAMID concept as a method of accommodating the different Stores used by an Exploiting Platform.
- Recording and Logging - retention of data relating to store aiming in accordance with this PYRAMID concept.

Extensions

- The aiming responsibilities may be developed as an extension to support the different algorithms required for the release aiming calculations for each of the different Stores (see Component Extensions PYRAMID concept).

Exploitation Considerations

- Release Aiming provides the viable release solutions that meet the specified input requirements and constraints. It should make no judgement of whether or not to perform the release.
- Release Aiming is expected to support both real time scenarios (where the Aiming_Solution is continually updated as the current Environmental_Conditions and Vehicle_Conditions change) and planned scenarios.

5.4.2.46.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component could result in weapons impacting locations not intended by the crew and so result in unintended harm to third parties. This drives a DAL B indicative IDAL.

5.4.2.46.6.4 Security Considerations

The indicative security classification is SNEO.

This component generates and maintains weapon Aiming_Solutions and therefore requires the characteristics of the Store(s) and information on the launch platform; these details are considered SNEO. Where aiming algorithms are data-driven, the associated configuration data will also carry appropriate confidentiality requirements.

The component is one of a group of components involved in the release of stores from the Exploiting Platform, and whilst not responsible for the actual release of the weapon, it does calculate the point at which it should occur in order for it to reach its target as intended. Provision of incorrect or no weapon aiming would affect the combat effectiveness of the Exploiting Platform, and where weapons miss their target can result in harm to third parties and significant reputational damage. To avoid this, the integrity and availability of this component should be appropriately protected.

The component is expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** as being authorised to provide target information.

- **Maintaining Audit Records** of the designated impact zone and the aiming solutions offered and selected to get a store to the target.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.

The component is considered unlikely to directly implement security enforcing functions, but is reliant on the Integrity of its input.

5.4.2.46.7 Services

5.4.2.46.7.1 Service Definitions

5.4.2.46.7.1.1 Release_Location

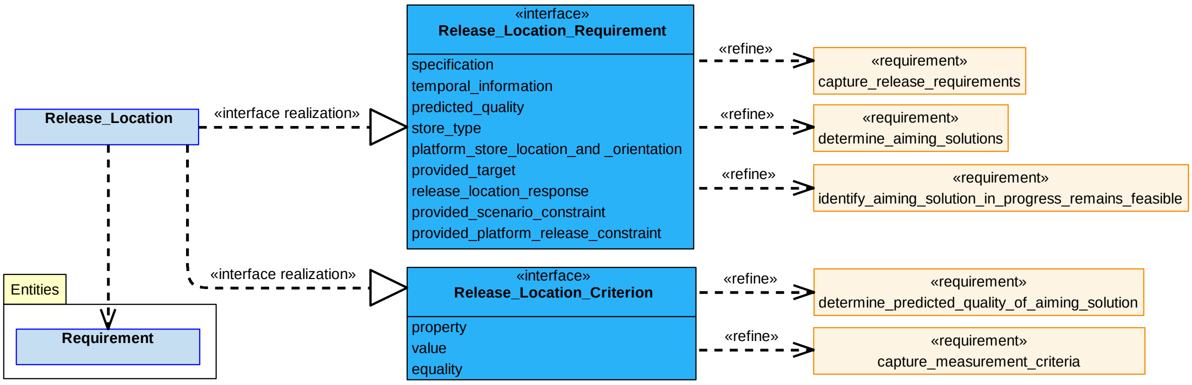

«interface»
Release_Location_Requirement
specification
temporal_information
predicted_quality
store_type
platform_store_location_and _orientation
provided_target
release_location_response
provided_scenario_constraint
provided_platform_release_constraint
«refine»
«requirement»
capture_release_requirements
«refine»
«requirement»
determine_aiming_solutions
«refine»
«requirement»
identify_aiming_solution_in_progress_remains_feasible
Release_Location
«interface realization»
«interface realization»
Entities
Requirement
«interface»
Release_Location_Criterion
property
value
equality
«refine»
«requirement»
determine_predicted_quality_of_aiming_solution
«refine»
«requirement»
capture_measurement_criteria

Figure 789: Release_Location Service Definition

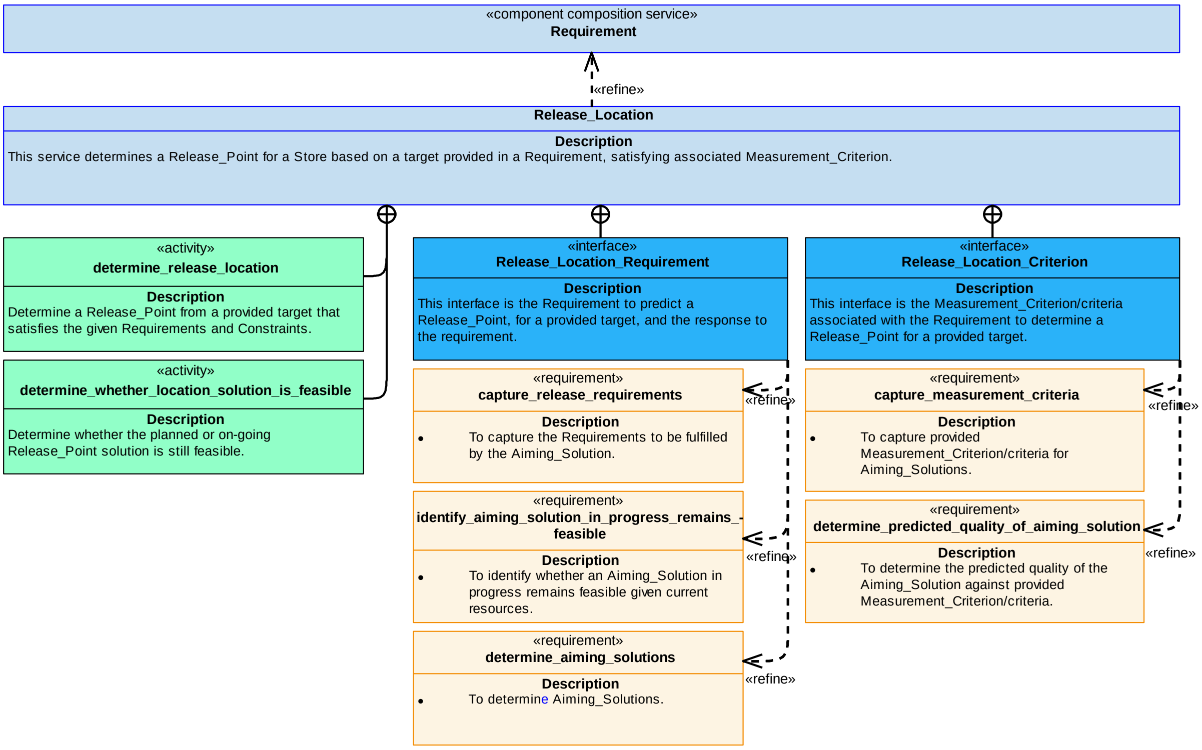

«component composition service»
Requirement
«refine»
Release_Location
Description
This service determines a Release_Point for a Store based on a target provided in a Requirement, satisfying associated Measurement_Criterion.
«activity»
determine_release_location
Description
Determine a Release_Point from a provided target that satisfies the given Requirements and Constraints.
«activity»
determine_whether_location_solution_is_feasible
Description
Determine whether the planned or on-going Release_Point solution is still feasible.
«interface»
Release_Location_Requirement
Description
This interface is the Requirement to predict a Release_Point, for a provided target, and the response to the requirement.
«interface»
Release_Location_Criterion
Description
This interface is the Measurement_Criterion/criteria associated with the Requirement to determine a Release_Point for a provided target.
«requirement»
capture_release_requirements
Description
To capture the Requirements to be fulfilled by the Aiming_Solution.
«refine»
«requirement»
identify_aiming_solution_in_progress_remains_feasible
Description
To identify whether an Aiming_Solution in progress remains feasible given current resources.
«refine»
«requirement»
determine_aiming_solutions
Description
To determine Aiming_Solutions.
«refine»
«requirement»
capture_measurement_criteria
Description
To capture provided Measurement_Criterion/criteria for Aiming_Solutions.
«refine»
«requirement»
determine_predicted_quality_of_aiming_solution
Description
To determine the predicted quality of the Aiming_Solution against provided Measurement_Criterion/criteria.
«refine»

Figure 790: Release_Location Service Policy

Release_Location

This service determines a Release_Point for a Store based on a target provided in a Requirement, satisfying associated Measurement_Criterion.

Interfaces

Release_Location_Criterion

This interface is the Measurement_Criterion/criteria associated with the Requirement to determine a Release_Point for a provided target.

Attributes

property	The property to be measured, e.g. Circular Error Probability (CEP).
value	The measured value of the property, e.g. CEP of 10 metres.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Release_Location_Requirement

This interface is the Requirement to predict a Release_Point, for a provided target, and the response to the requirement.

Attributes

specification	The high level definition of the release location requirement.
temporal_information	Information covering timing, such as start and end times.

predicted_quality	How well the planned solution is predicted to satisfy the requirement.
store_type	The provided type of Store for which an Aiming_Solution is required.
platform_store_location_and _orientation	The provided location on the launch platform of the Store to be aimed and the stores orientation.
provided_target	Provided target details (e.g. target velocity, target position or target type) from which the Aiming_Solution should be determined.
release_location_response	Release location (point, area or volume) returned in response to the requirement.
provided_scenario_constraint	A provided constraint on Release Aiming's behaviour with respect to determining an Aiming_Solution applicable to a particular scenario. This could include no launch/fly/impact/abort zones; launch profile (e.g. loft release); attack orientation (e.g. to minimise collateral damage); store modes or controls (e.g. fly out altitude, pop-up terminal manoeuvre or target impact angle); and method of release (e.g. gravity drop, downward eject, or engine start before launch).
provided_platform_release_constraint	A platform release constraint provided for the Aiming_Solution (e.g. velocity, altitude, or orientation).

Activities

determine_release_location

Determine a Release_Point from a provided target that satisfies the given Requirements and Constraints.

determine_whether_location_solution_is_feasible

Determine whether the planned or on-going Release_Point solution is still feasible.

5.4.2.46.7.1.2 Impact_Zone

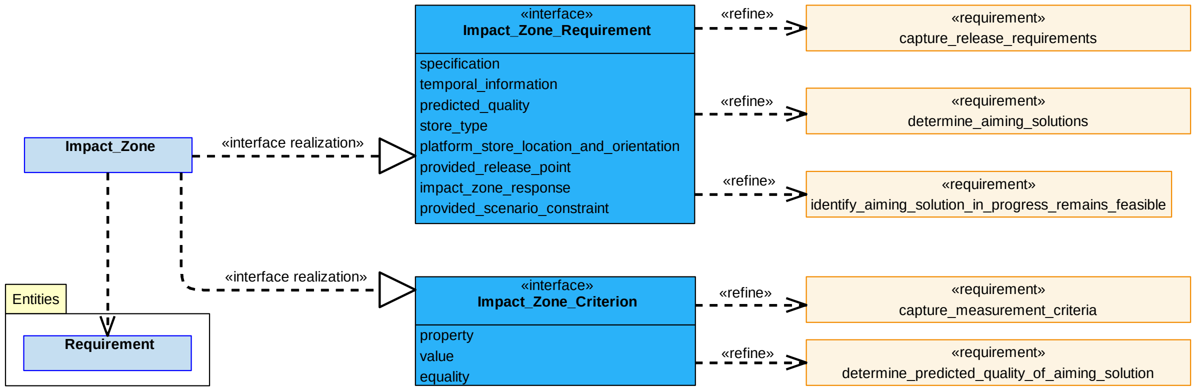


Figure 791: Impact_Zone Service Definition

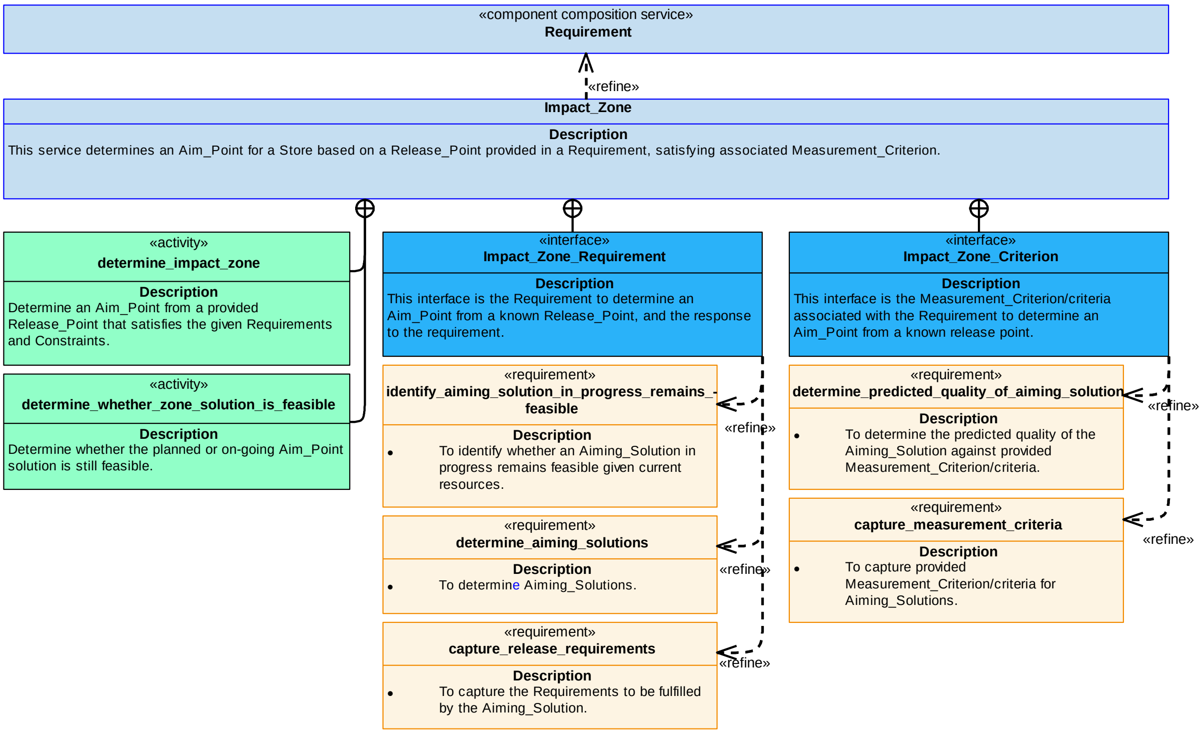

«component composition service»
Requirement
«refine»
Impact_Zone
Description
This service determines an Aim_Point for a Store based on a Release_Point provided in a Requirement, satisfying associated Measurement_Criterion.
«activity»
determine_impact_zone
Description
Determine an Aim_Point from a provided Release_Point that satisfies the given Requirements and Constraints.
«activity»
determine_whether_zone_solution_is_feasible
Description
Determine whether the planned or on-going Aim_Point solution is still feasible.
«interface»
Impact_Zone_Requirement
Description
This interface is the Requirement to determine an Aim_Point from a known Release_Point, and the response to the requirement.
«interface»
Impact_Zone_Criterion
Description
This interface is the Measurement_Criterion/criteria associated with the Requirement to determine an Aim_Point from a known release point.
«requirement»
identify_aiming_solution_in_progress_remains_-feasible
Description
To identify whether an Aiming_Solution in progress remains feasible given current resources.
«refine»
«requirement»
determine_aiming_solutions
Description
To determine Aiming_Solutions.
«refine»
«requirement»
capture_release_requirements
Description
To capture the Requirements to be fulfilled by the Aiming_Solution.
«refine»
«requirement»
determine_predicted_quality_of_aiming_solution
Description
To determine the predicted quality of the Aiming_Solution against provided Measurement_Criterion/criteria.
«refine»
«requirement»
capture_measurement_criteria
Description
To capture provided Measurement_Criterion/criteria for Aiming_Solutions.
«refine»

Figure 792: Impact_Zone Service Policy

Impact_Zone

This service determines an Aim_Point for a Store based on a Release_Point provided in a Requirement, satisfying associated Measurement_Criterion.

Interfaces

Impact_Zone_Criterion

This interface is the Measurement_Criterion/criteria associated with the Requirement to determine an Aim_Point from a known release point.

Attributes

property	The property to be measured, e.g. Circular Error Probability (CEP).
value	The measured value of the property, e.g. CEP of 10 metres.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Impact_Zone_Requirement

This interface is the Requirement to determine an Aim_Point from a known Release_Point, and the response to the requirement.

Attributes

specification	The high level definition of the impact zone requirement.
temporal_information	Information covering timing, such as start and end times.

predicted_quality	How well the planned solution is predicted to satisfy the requirement.
store_type	The provided type of Store for which an Aiming_Solution is required.
platform_store_location_and_orientation	The provided location on the launch platform of the Store to be aimed and the stores orientation.
provided_release_point	The provided Release_Point, from which the Aiming_Solution should be determined.
impact_zone_response	The Aim_Point returned in response to the Requirement.
provided_scenario_constraint	A provided constraint on Release Aiming's behaviour with respect to determining an Aiming_Solution applicable to a particular scenario. This could include no launch/fly/impact/abort zones; launch profile (e.g. loft release); attack orientation (e.g. to minimise collateral damage); store modes or controls (e.g. fly out altitude, pop-up terminal manoeuvre, or target impact angle); and method of release (e.g. gravity drop, downward eject, or engine start before launch).

Activities

determine_impact_zone

Determine an Aim_Point from a provided Release_Point that satisfies the given Requirements and Constraints.

determine_whether_zone_solution_is_feasible

Determine whether the planned or on-going Aim_Point solution is still feasible.

5.4.2.46.7.1.3 Environmental_Condition

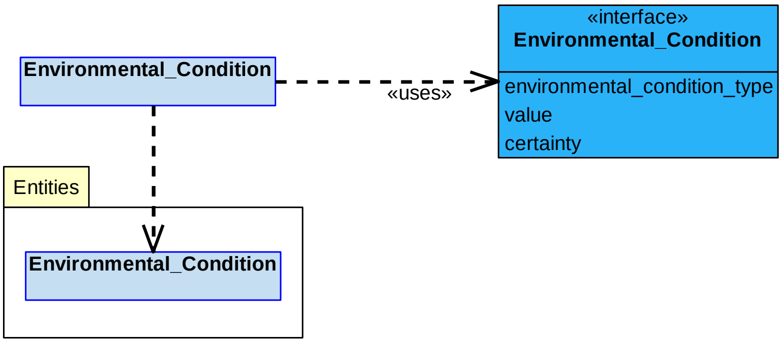


Figure 793: Environmental_Condition Service Definition

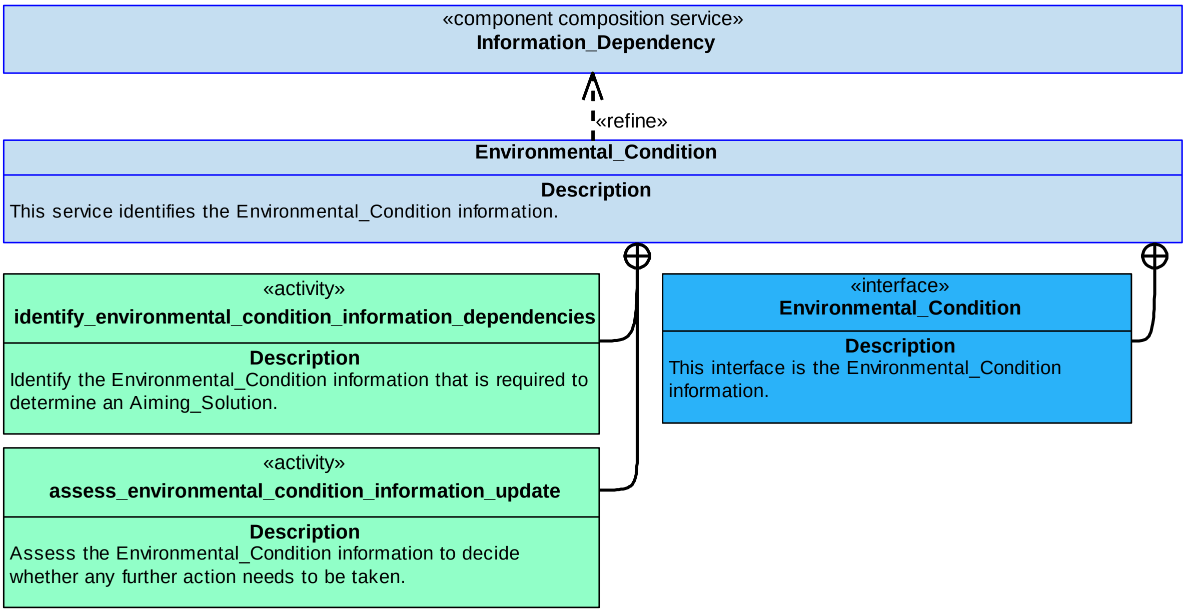

«component composition service»
Information_Dependency
«refine»
Environmental_Condition
Description
This service identifies the Environmental_Condition information.
«activity»
identify_environmental_condition_information_dependencies
Description
Identify the Environmental_Condition information that is required to determine an Aiming_Solution.
«activity»
assess_environmental_condition_information_update
Description
Assess the Environmental_Condition information to decide whether any further action needs to be taken.
«interface»
Environmental_Condition
Description
This interface is the Environmental_Condition information.

Figure 794: Environment_Condition Service Policy

Environmental_Condition

This service identifies the Environmental_Condition information.

Interface

Environmental_Condition

This interface is the Environmental_Condition information.

Attributes

environmental_condition_type	The type of Environmental_Condition, e.g. wind velocity.
value	A value or state of the related condition.
certainty	The certainty or accuracy that the value or state of the related condition is known to.

Activities

assess_environmental_condition_information_update

Assess the Environmental_Condition information to decide whether any further action needs to be taken.

identify_environmental_condition_information_dependencies

Identify the Environmental_Condition information that is required to determine an Aiming_Solution.

5.4.2.46.7.1.4 Store_Condition

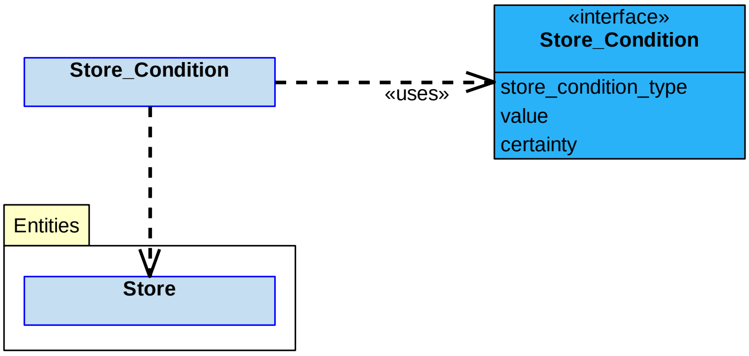


Figure 795: Store_Condition Service Definition

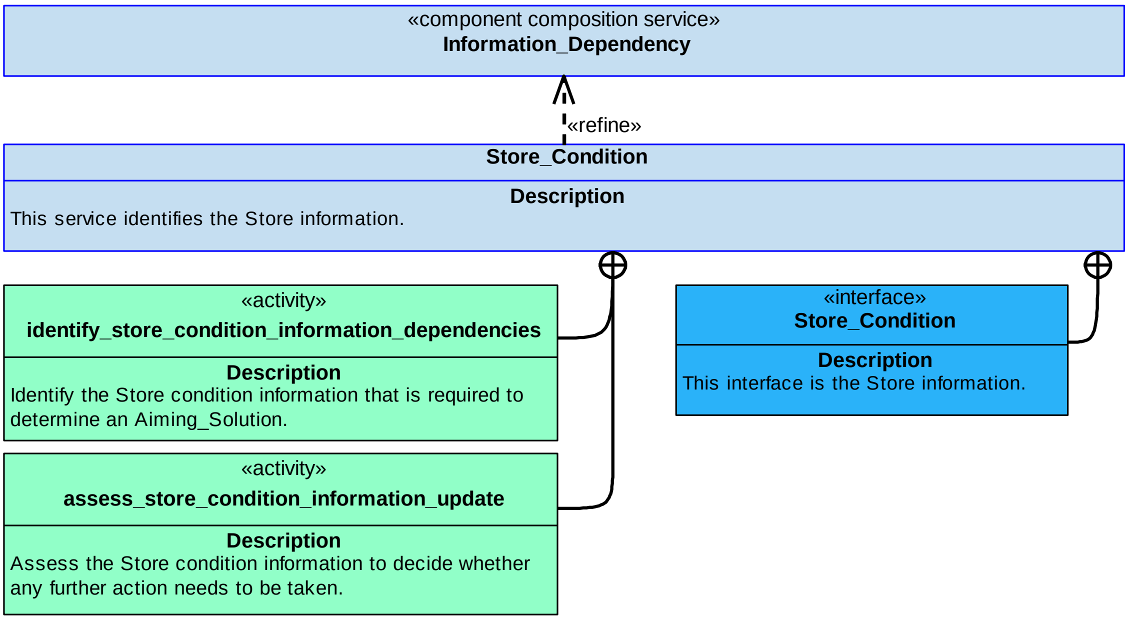

«component composition service»
Information_Dependency
«refine»
Store_Condition
Description
This service identifies the Store information.
«activity»
identify_store_condition_information_dependencies
Description
Identify the Store condition information that is required to determine an Aiming_Solution.
«activity»
assess_store_condition_information_update
Description
Assess the Store condition information to decide whether any further action needs to be taken.
«interface»
Store_Condition
Description
This interface is the Store information.

Figure 796: Store_Condition Service Policy

Store_Condition

This service identifies the Store information.

Interface

Store_Condition

This interface is the Store information.

Attributes

store_condition_type	The type of information about a Store, e.g. fuel type or available fuel quantity.
value	A value or state of the related condition.
certainty	The certainty or accuracy that the value or state of the related condition is known to.

Activities

identify_store_condition_information_dependencies

Identify the Store condition information that is required to determine an Aiming_Solution.

assess_store_condition_information_update

Assess the Store condition information to decide whether any further action needs to be taken.

5.4.2.46.7.1.5 Vehicle_Condition

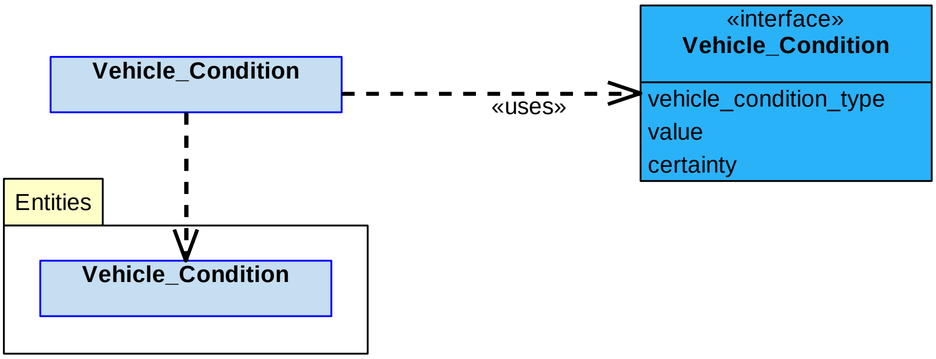


Figure 797: Vehicle_Condition Service Definition

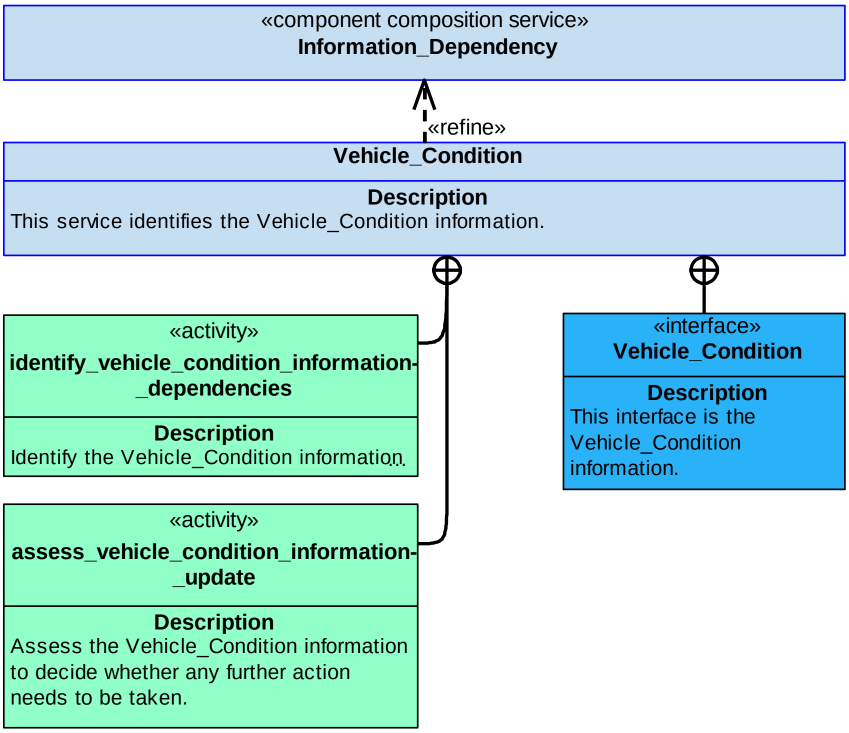

«component composition service»
Information_Dependency
«refine»
Vehicle_Condition
Description
This service identifies the Vehicle_Condition information.
«activity»
identify_vehicle_condition_information-_dependencies
Description
Identify the Vehicle_Condition information.
«activity»
assess_vehicle_condition_information-_update
Description
Assess the Vehicle_Condition information to decide whether any further action needs to be taken.
«interface»
Vehicle_Condition
Description
This interface is the Vehicle_Condition information.

Figure 798: Vehicle_Condition Service Policy

Vehicle_Condition

This service identifies the Vehicle_Condition information.

Interface

Vehicle_Condition

This interface is the Vehicle_Condition information.

Attributes

vehicle_condition_type	The type of Vehicle_Condition (e.g. position, velocity, or orientation).
value	A value or state of the related condition.
certainty	The certainty or accuracy that the value or state of the related condition is known to.

Activities

identify_vehicle_condition_information_dependencies

Identify the Vehicle_Condition information that is required to determine an Aiming_Solution.

assess_vehicle_condition_information_update

Assess the Vehicle_Condition information to decide whether any further action needs to be taken.

5.4.2.46.7.1.6 Object_Trajectory

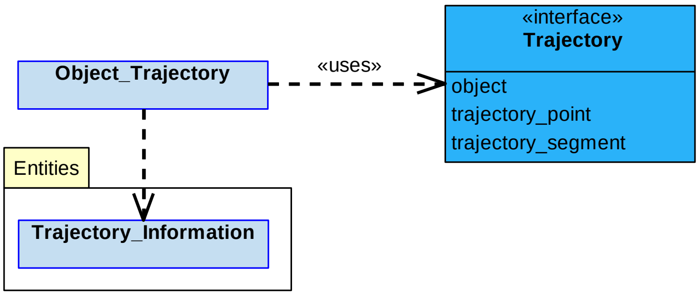


Figure 799: Object_Trajectory Service Definition

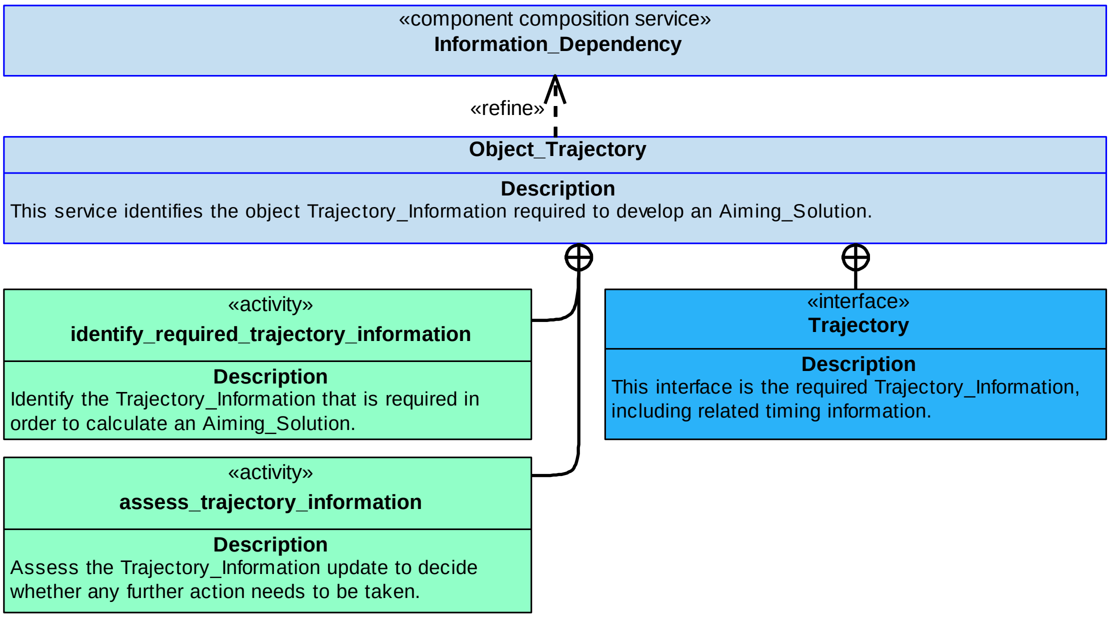

«component composition service»
Information_Dependency
«refine»
Object_Trajectory
Description
This service identifies the object Trajectory_Information required to develop an Aiming_Solution.
«activity»
identify_required_trajectory_information
Description
Identify the Trajectory_Information that is required in order to calculate an Aiming_Solution.
«interface»
Trajectory
Description
This interface is the required Trajectory_Information, including related timing information.
«activity»
assess_trajectory_information
Description
Assess the Trajectory_Information update to decide whether any further action needs to be taken.

Figure 800: Object_Trajectory Service Policy

Object_Trajectory

This service identifies the object Trajectory_Information required to develop an Aiming_Solution.

Interface

Trajectory

This interface is the required Trajectory_Information, including related timing information.

Attributes

object	The object for which a Trajectory_Information is required.
trajectory_point	Trajectory_Information that describes a point along a trajectory, e.g. the Aim_Point for a ground target.
trajectory_segment	Trajectory_Information that describes a path segment of a trajectory.

Activities

identify_required_trajectory_information

Identify the Trajectory_Information that is required in order to calculate an Aiming_Solution.

assess_trajectory_information

Assess the Trajectory_Information update to decide whether any further action needs to be taken.

5.4.2.46.7.1.7 Constraint

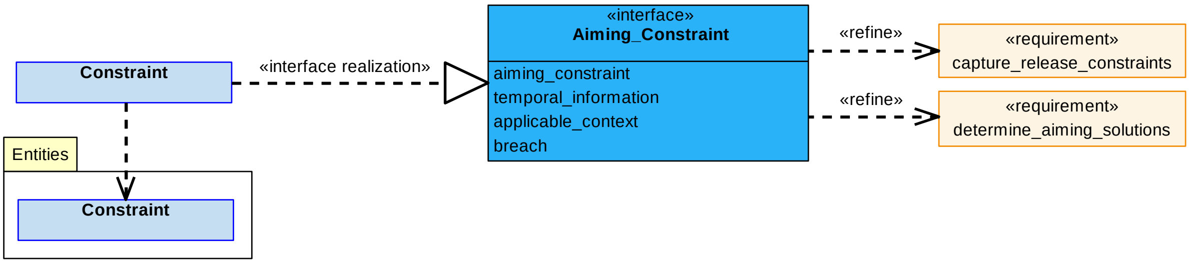


Figure 801: Constraint Service Definition

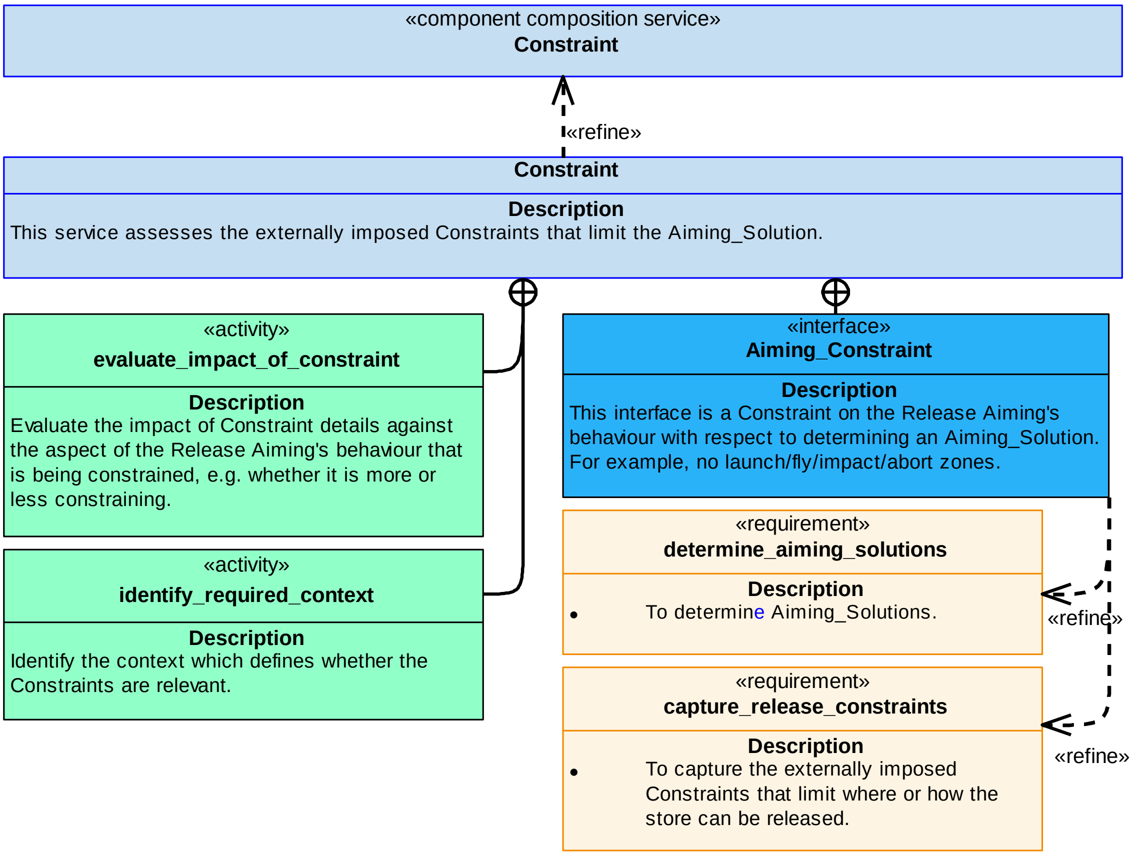

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the externally imposed Constraints that limit the Aiming_Solution.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of the Release Aiming's behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Aiming_Constraint
Description
This interface is a Constraint on the Release Aiming's behaviour with respect to determining an Aiming_Solution. For example, no launch/fly/impact/abort zones.
«requirement»
determine_aiming_solutions
Description
To determine Aiming_Solutions.
«refine»
«requirement»
capture_release_constraints
Description
To capture the externally imposed Constraints that limit where or how the store can be released.
«refine»

Figure 802: Constraint Service Policy

Constraint

This service assesses the externally imposed Constraints that limit the Aiming_Solution.

Interface

Aiming_Constraint

This interface is a Constraint on the Release Aiming's behaviour with respect to determining an Aiming_Solution. For example, no launch/fly/impact/abort zones.

Attributes

aiming_constraint	Aiming Constraints that have been provided, e.g. no fly zones or no impact zones.
temporal_information	Timing information pertaining to the periods of time when the Constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the Constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of the Release Aiming's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.46.7.1.8 Capability

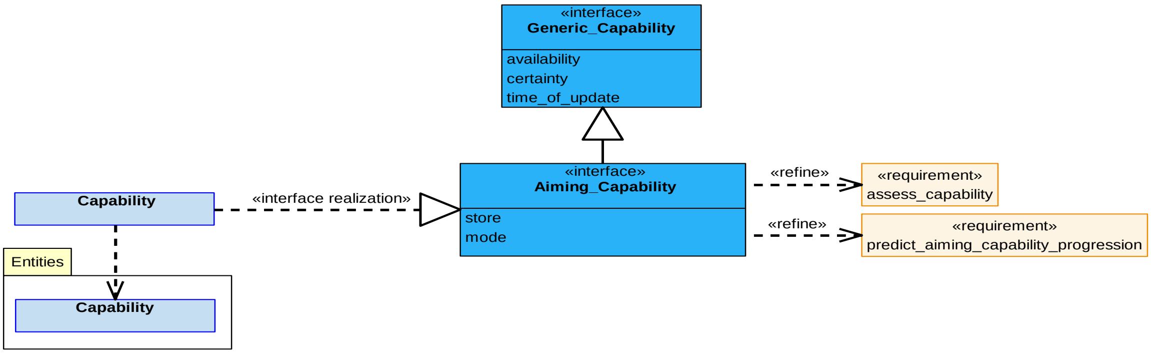


Figure 803: Capability Service Definition

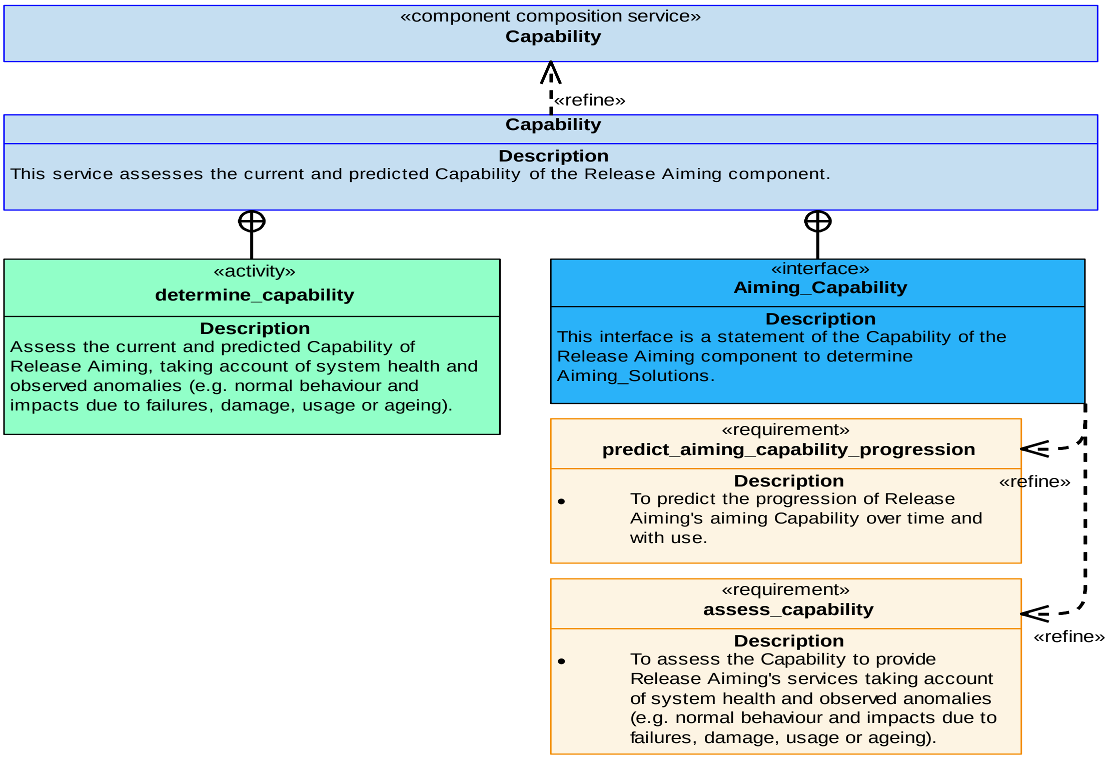

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability of the Release Aiming component.
«activity»
determine_capability
Description
Assess the current and predicted Capability of Release Aiming, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Aiming_Capability
Description
This interface is a statement of the Capability of the Release Aiming component to determine Aiming_Solutions.
«requirement»
predict_aiming_capability_progression
Description
To predict the progression of Release Aiming's aiming Capability over time and with use.
«refine»
«requirement»
assess_capability
Description
To assess the Capability to provide Release Aiming's services taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 804: Capability Service Policy

Capability

This service assesses the current and predicted Capability of the Release Aiming component.

Interface

Aiming_Capability

This interface is a statement of the Capability of the Release Aiming component to determine Aiming_Solutions.

Attributes

store	Supported Store type (e.g. ballistic or guided bomb).
mode	Supported mode of release (e.g. type of aiming modes).

Activity

determine_capability

Assess the current and predicted Capability of Release Aiming, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.46.7.1.9 Capability_Evidence

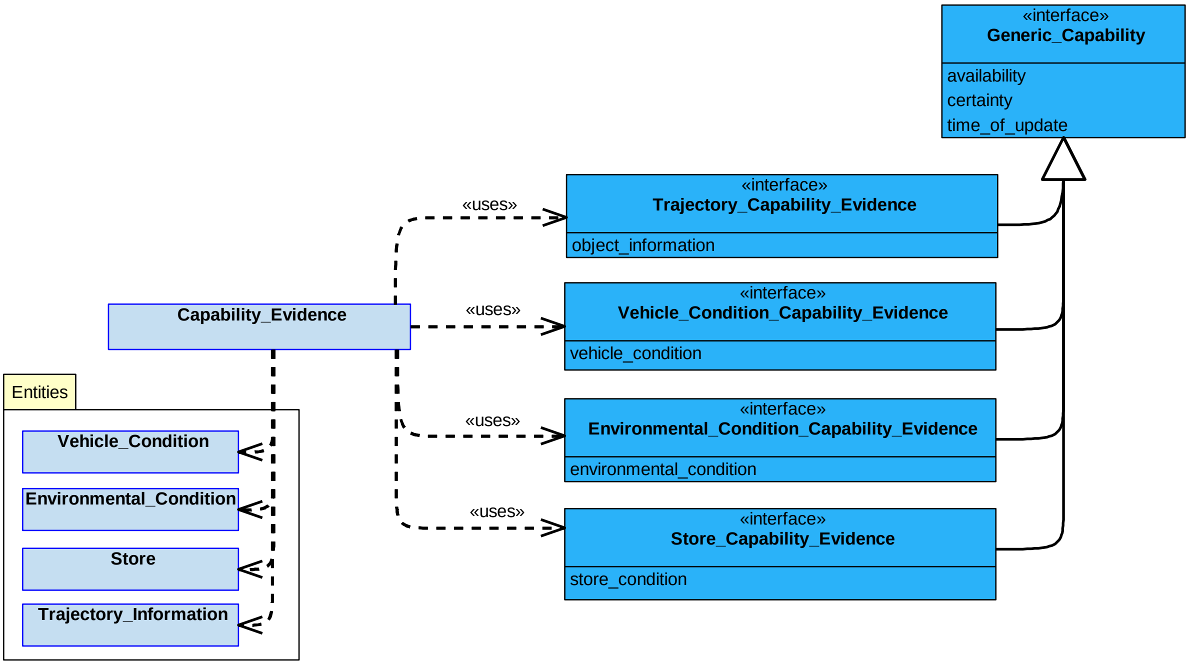


Figure 805: Capability_Evidence Service Definition

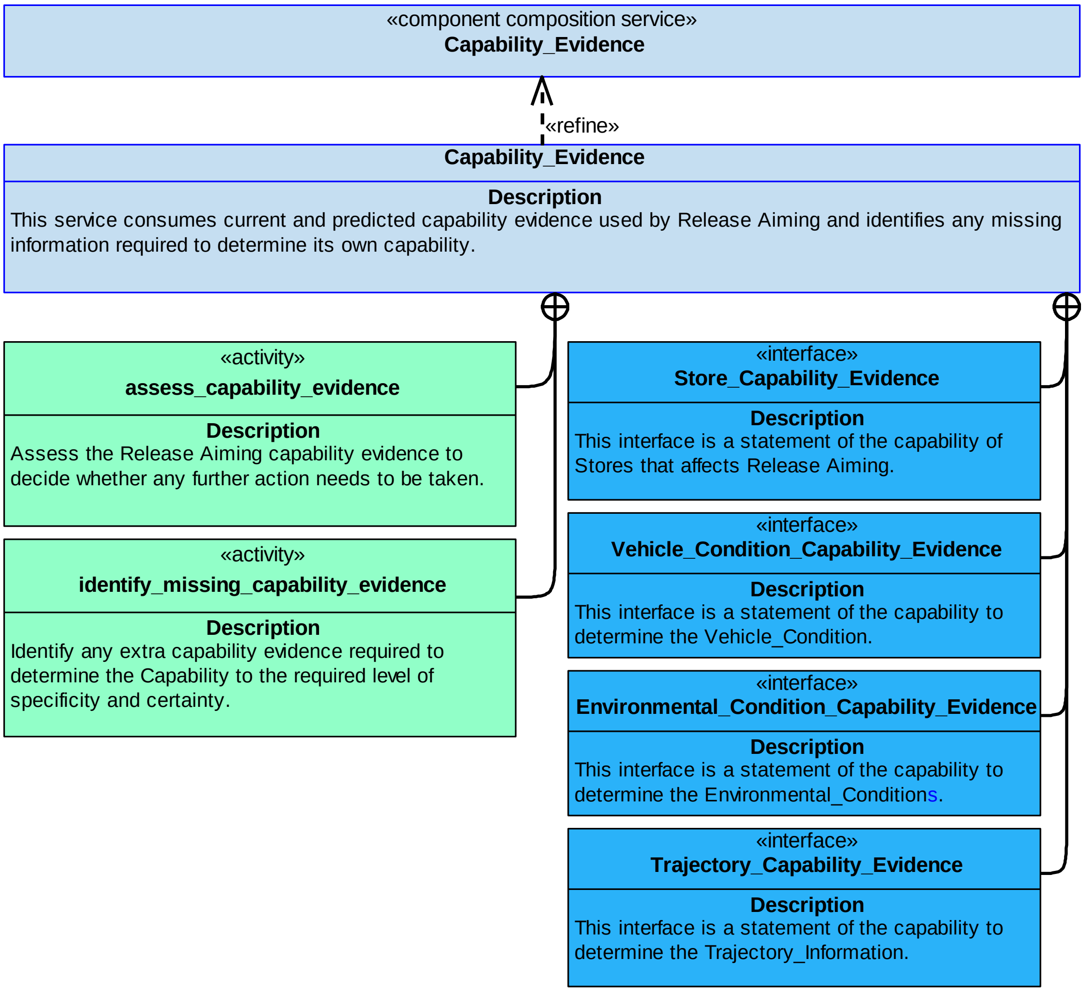

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability evidence used by Release Aiming and identifies any missing information required to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the Release Aiming capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Store_Capability_Evidence
Description
This interface is a statement of the capability of Stores that affects Release Aiming.
«interface»
Vehicle_Condition_Capability_Evidence
Description
This interface is a statement of the capability to determine the Vehicle_Condition.
«interface»
Environmental_Condition_Capability_Evidence
Description
This interface is a statement of the capability to determine the Environmental_Conditions.
«interface»
Trajectory_Capability_Evidence
Description
This interface is a statement of the capability to determine the Trajectory_Information.

Figure 806: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability evidence used by Release Aiming and identifies any missing information required to determine its own capability.

Interfaces

Vehicle_Condition_Capability_Evidence

This interface is a statement of the capability to determine the Vehicle_Condition.

Attribute

vehicle_condition	An aspect of the Vehicle_Condition.

Environmental_Condition_Capability_Evidence

This interface is a statement of the capability to determine the Environmental_Conditions.

Attribute

environmental_condition	An aspect of Environmental_Conditions.

Store_Capability_Evidence

This interface is a statement of the capability of Stores that affects Release Aiming.

Attribute

store_condition	An aspect of a Store.

Trajectory_Capability_Evidence

This interface is a statement of the capability to determine the Trajectory_Information.

Attribute

object_information	An aspect of Trajectory_Information for an object.

Activities

assess_capability_evidence

Assess the Release Aiming capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.46.7.2 Service Dependencies

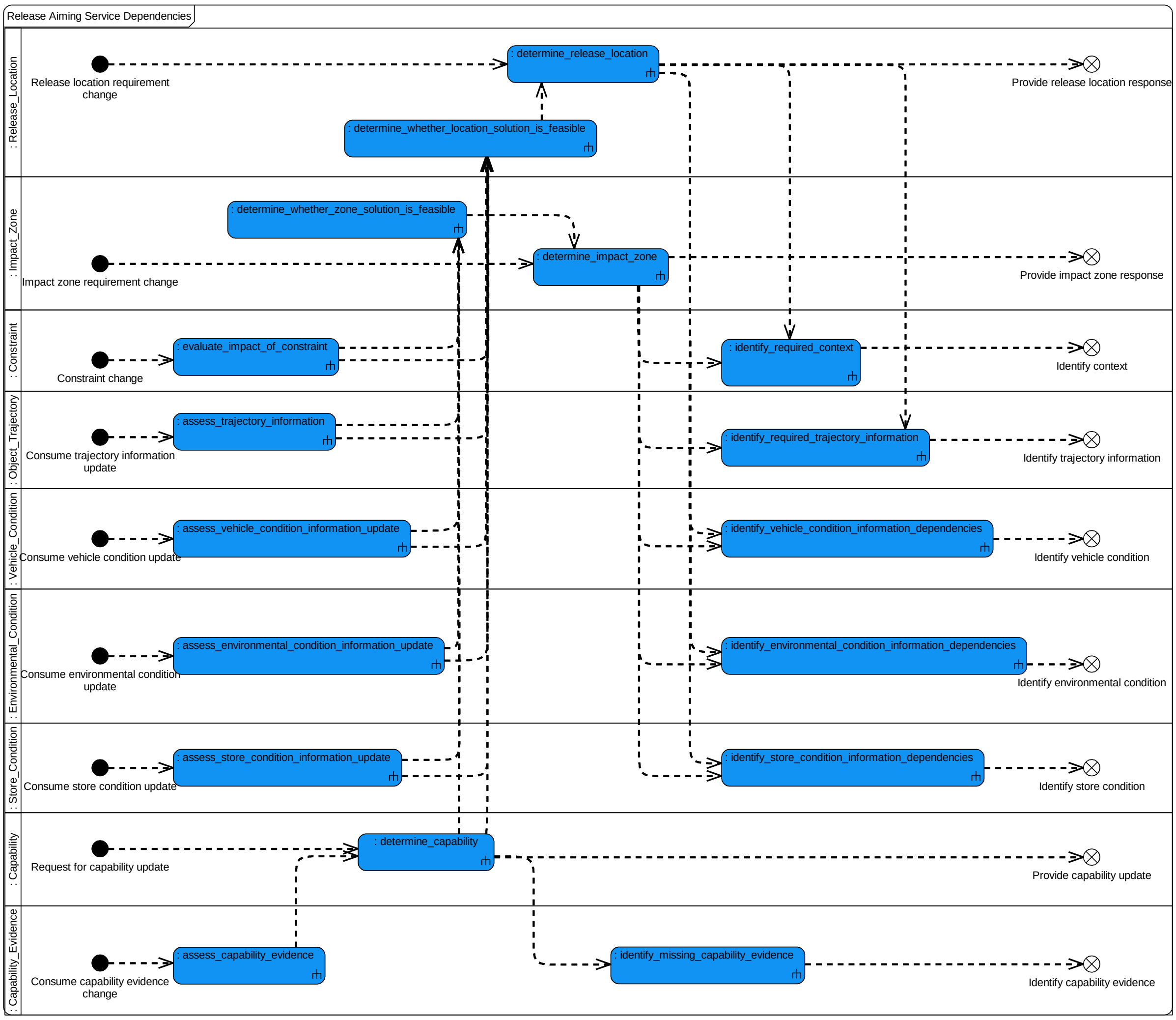

Release Aiming Service Dependencies
: Release_Location
: Impact_Zone
: Constraint
: Object_Trajectory
: Vehicle_Condition
: Environmental_Condition
: Store_Condition
: Capability
: Capability_Evidence
Release location requirement change
: determine_release_location
Provide release location response
: determine_whether_location_solution_is_feasible
: determine_whether_zone_solution_is_feasible
Impact zone requirement change
: determine_impact_zone
Provide impact zone response
: evaluate_impact_of_constraint
Constraint change
: identify_required_context
Identify context
: assess_trajectory_information
Consume trajectory information update
: identify_required_trajectory_information
Identify trajectory information
: assess_vehicle_condition_information_update
Consume vehicle condition update
: identify_vehicle_condition_information_dependencies
Identify vehicle condition
: assess_environmental_condition_information_update
Consume environmental condition update
: identify_environmental_condition_information_dependencies
Identify environmental condition
: assess_store_condition_information_update
Consume store condition update
: identify_store_condition_information_dependencies
Identify store condition
Request for capability update
: determine_capability
Provide capability update
: assess_capability_evidence
Consume capability evidence change
: identify_missing_capability_evidence
Identify capability evidence

Figure 807: Release Aiming Service Dependencies

5.4.2.47 Release Effecting

5.4.2.47.1 Role

The role of Release Effecting is to effect the release of a store.

5.4.2.47.2 Overview

Control Architecture

Release Effecting is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In order to satisfy a Requirement for an operational release or jettison action, Release Effecting will release a Store and determine an associated set of Release_Effecting_Steps according to an applicable Release_Timeline. The Release_Timeline triggers the Release_Effecting_Steps required to perform a release. The release will be monitored throughout to ensure it remains feasible.

Examples of Use

This component can be used where:

- There is a requirement for an operational release of a Store, e.g. a deployable sensor or a bomb.
- There is a requirement for a jettison of a Store, e.g. an external fuel tank.
- There is a requirement for a precautionary release of an on-board defensive Store, e.g. chaff or flares.

5.4.2.47.3 Service Summary

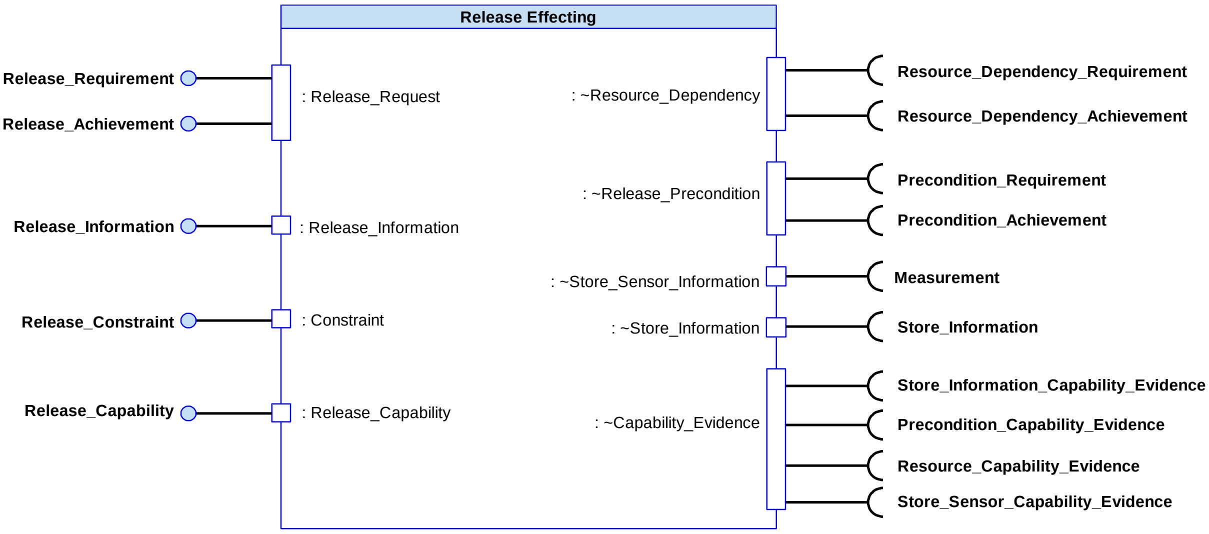


Figure 808: Release Effecting Service Summary

5.4.2.47.4 Responsibilities

capture_store_release_requirements

- To capture provided Requirements for a Store release (e.g. release Store 'X' from Location 'Y' in an armed state).

capture_release_effecting_constraints

- To capture provided Constraints, e.g. Stores are not to be released for training activities.

identify_whether_release_requirement_is_achievable

- To identify whether a Store release Requirement is achievable given current Release_Effecting_Capability.

determine_release_effecting_steps

- To determine the Release_Timeline and the associated Release_Effecting_Steps that meet the given Requirements and Constraints for the release of a Store using available Release_Effecting_Resources.

determine_release_element_states

- To determine the current state of release information on Stores and Release_Effecting_Resources.

identify_pre-conditions

- To identify Pre-conditions required to support Release_Effecting_Steps that fulfil the Release_Timeline.

coordinate_use_of_release_effecting_resources

- To coordinate Release_Effecting_Resources to effect an operational release or jettison by implementing Release_Effecting_Steps in accordance with a Release_Timeline.

determine_store_release_progress

- To determine the progress of the Store release (e.g. report Store release in progress, Store released, Store jettisoned, Store misfired, or Store hung).

assess_release_effecting_capability

- To assess the Release_Effecting_Capability to effect an operational release or jettison, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Release_Effecting_Capability assessment.

predict_capability_progression

- To predict the progression of the Release_Effecting_Capability over time and with use.

5.4.2.47.5 Subject Matter Semantics

The subject matter of Release Effecting is resources used to operationally release or jettison a Store from a platform.

Exclusions

The subject matter of Release Effecting does not include:

- The determination of which Stores should be released or the determination of the required relative timings between Store release events. Except where catering for multi Store devices (e.g. multi Store carriers, dispensers, or guns) that self determine the specific Stores, number of Stores, or relative timings for releases from the device (e.g. for a salvo from a carrier), Release Effecting handles the release of each Store independently.
- The adjustment of a Store for release (e.g. setting fusing options or priming with targeting data), only the Release_Timeline.

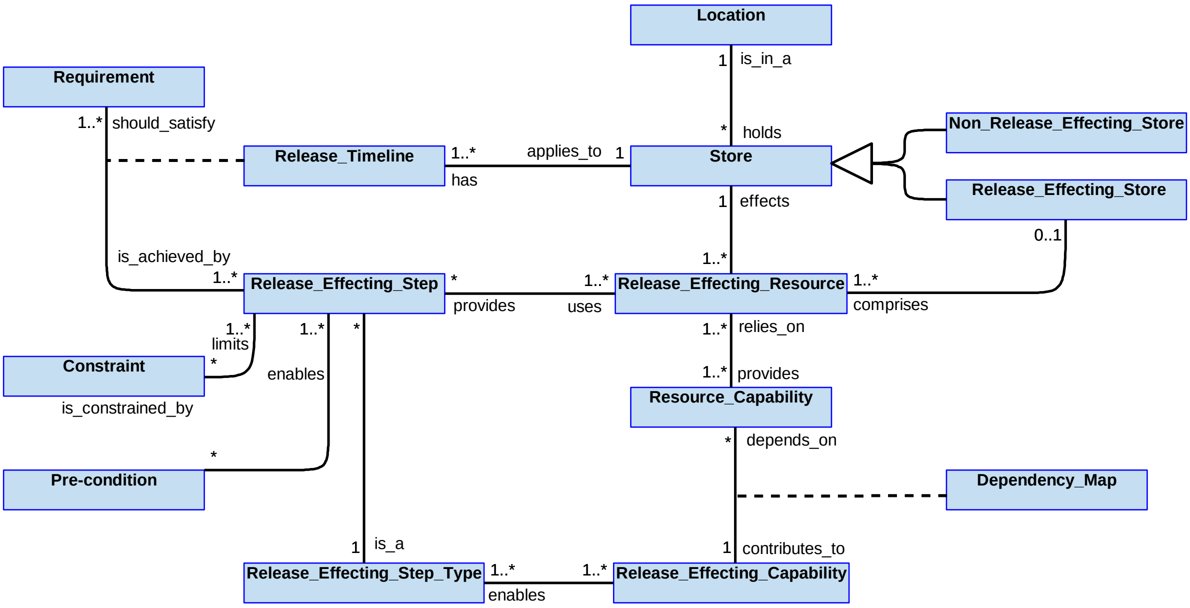


Figure 809: Release Effecting Semantics

5.4.2.47.5.1 Entities

Constraint

An externally imposed restriction that limits when or how Release_Effecting_Steps are performed. For example, no fire supplies are used when the Exploiting Platform is in a training mode.

Dependency_Map

Mapping of how the Release_Effecting_Capability is dependent on the Resource_Capability.

Location

A physical location on the Exploiting Platform, e.g. a hard point.

Non_Release_Effecting_Store

A Store that does not affect the release of other Stores nor effect the release of itself, e.g. a bomb.

Pre-condition

A condition that must be true before an activity can take place (e.g. undercarriage is raised, authorisation is granted or interlocks are enabled).

Release_Effecting_Capability

The capability to release a Store from the Exploiting Platform.

Release_Effecting_Resource

A resource that can be used by Release Effecting (e.g. a release unit, missile rocket motor or arming solenoid).

Release_Effecting_Step

An activity Release Effecting will carry out that, when performed, achieves (or partially achieves) the release of a Store (e.g. a request to enable or disable interlocks, or a request to enable or disable store arming).

Release_Effecting_Step_Type

The kinds of activity Release Effecting knows how to coordinate (e.g. activate arming unit, unlock launcher, start store motor, or open hooks of release unit).

Release_Effecting_Store

A Store that effects the release of other Stores or effects the release of itself (e.g. a weapons launcher or a rail launched missile with a rocket motor).

Release_Timeline

The order and timing in which Release_Effecting_Steps must be performed to meet the Requirement.

Requirement

A requirement placed in order to effect the release of a Store (e.g. to release a number of missiles from a launcher, the identification of the missiles to be released and the order they should be released in).

Resource_Capability

The capability of the underlying resources to release a Store.

Store

An item that can be operationally released or jettisoned from the Exploiting Platform. This can include carriage stores intended to carry other stores and that can be jettisoned (e.g. a multi weapons launcher that carries multiple missiles) or those that give a mission effect (e.g. a bomb or missile, extended range fuel tanks or a sensor pod).

5.4.2.47.6 Design Rationale

5.4.2.47.6.1 Assumptions

None.

5.4.2.47.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Release Effecting:

- Data Driving - For classes of Store and release hardware, Release_Timeline and types/classes of Release_Effecting_Steps.
- Recording and Logging - This PYRAMID concept is applicable to cover logging of data relating to authorisations and release actions including for audit and non-repudiation purposes.

Extensions

- An extension may be developed where the releasing equipment on an Exploiting Platform is radically different or where different classes of Stores are used.

Exploitation Considerations

- The component does not coordinate the release of multiple Stores, Stores Release will place a Requirement on Release Effecting (e.g. a Requirement may be to fire a gun, or to release Store 'X' from Location 'Y').
- The Release_Timeline of a Store would include those actions required at or very close to the point of release which would not be expected to be interrupted or halted. An Exploiting Programme is responsible for determining whether particular commands to Stores are covered by the Release_Timeline or as part of Store preparation.
- As part of executing the Release_Timeline this component may request interlocks are enabled (e.g. safety critical power for release).
- As part of executing the Release_Timeline this component is expected to control the enabling of weapon arming. As well as controlling arming solenoids, arming may also be achieved by electrical discrete or data commands, depending upon the weapon.
- As part of executing the Release_Timeline this component will control the release of Stores. Depending upon the Store type, this could include control of a Stores release unit (e.g. opening the hooks so a Store falls away under gravity), control of launcher equipment, unlocking a rail launched missile or commanding a rail launched missile's rocket motor to fire. This may be achieved by electrical discrete or data commands, depending upon the Store type.
- To execute the Release_Timeline this component may need to monitor the state of the resources being used (e.g. monitor that relays have been activated or lock mechanisms released).

- Release Effecting may choose to determine for itself that a required level of confidence has been attained in the signals and or events it receives, e.g. by cross monitoring of dual channel signals from a mechanical switch. Alternatively, it may require that it is provided with assurances on the signals or events it receives, e.g. validity information is provided on received signals and or events.
- It is likely that this component would not necessarily respond to just the requests from external service Requirements (i.e. other components) to enable or disable high criticality functionality (e.g. Stores Release request for an armed Store release), but also to internal requirements to Release Effecting (e.g. to inhibit arming if the arm interlock signal from Interlocks is not set to enable).

5.4.2.47.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Failure of this component could cause Stores to be released at the wrong time (resulting in Store to Store collision or an out of balance condition) or enable weapon arming when it is not required. These could result in catastrophic consequences.
- Whilst the Interlocks component may be used to prevent release / weapon arming when not appropriate independently of this component, protection is not assumed, as this protection may not be practicable for all Exploiting Programmes or failure cases, e.g. Interlocks is not expected to protect against breaches of minimum release intervals between stations.

5.4.2.47.6.4 Security Considerations

The indicative security classification is SNEO.

This component executes the release of a Store from its Location on the Exploiting Platform following an authorised release request using a Release_Timeline appropriate for that Store. Details of the Release_Timeline, including arming and other conditions that apply, may be operationally significant and therefore considered SNEO. The component is one of a group of components involved in the release of Stores from the Exploiting Platform, performing the final stages covering arming and release. This component is dependent on the integrity of the release request in order that Stores release is not performed when not required. Loss of availability may inhibit the ability to release Stores at the moment required, affecting both operational effectiveness and safety.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to weapon arming, requested releases and whether enacted or not in support of subsequent forensic examination.
- **Maintaining Audit Records** of the release actions performed during the mission, supporting non-repudiation for the release of Stores, whether operational or for jettison purposes.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.

The component is expected to at least partially satisfy security enforcing functions by:

- **Verifying Integrity of Data** for the release command, ensuring it has come from an authorised source.

5.4.2.47.7 Services

5.4.2.47.7.1 Service Definitions

5.4.2.47.7.1.1 Release_Request

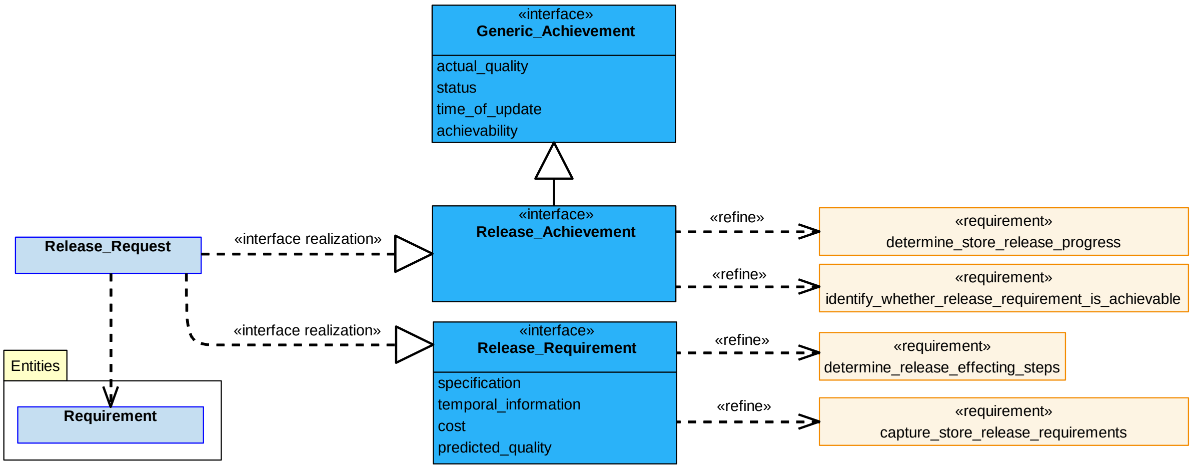
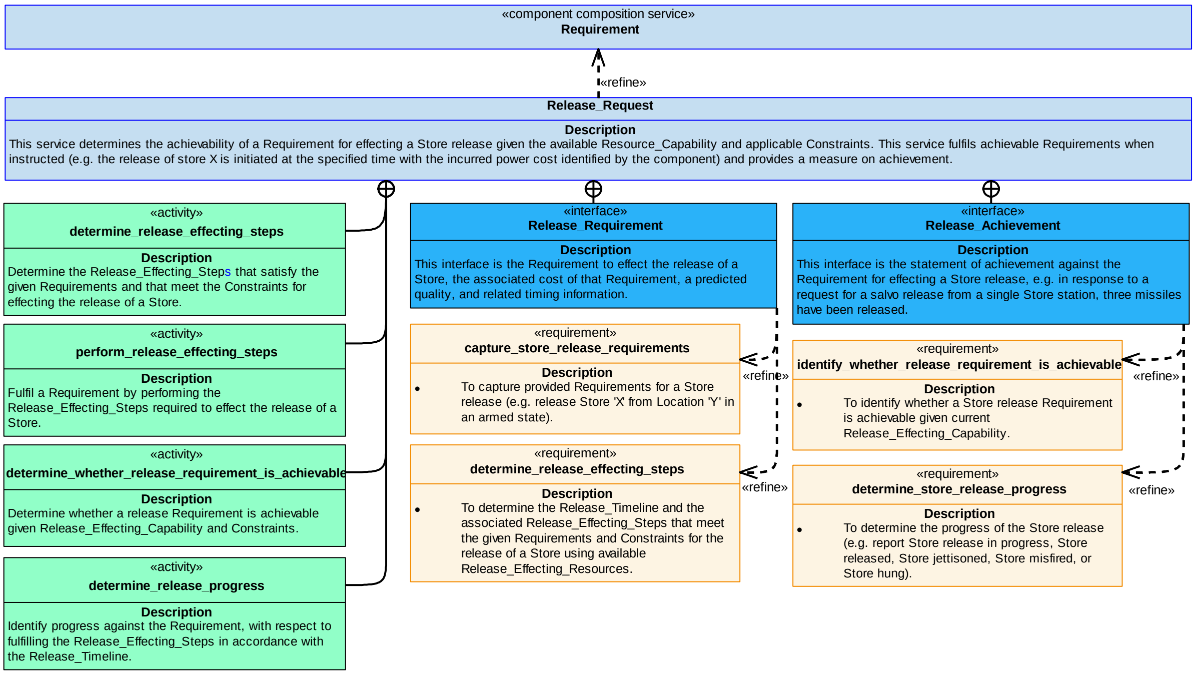


Figure 810: Release_Request Service Definition



Release_Request

Description

This service determines the achievability of a Requirement for effecting a Store release given the available Resource_Capability and applicable Constraints. This service fulfils achievable Requirements when instructed (e.g. the release of store X is initiated at the specified time with the incurred power cost identified by the component) and provides a measure on achievement.

«activity»
determine_release_effecting_steps

Description

Determine the Release_Effecting_Steps that satisfy the given Requirements and that meet the Constraints for effecting the release of a Store.

«activity»
perform_release_effecting_steps

Description

Fulfil a Requirement by performing the Release_Effecting_Steps required to effect the release of a Store.

«activity»
determine_whether_release_requirement_is_achievable

Description

Determine whether a release Requirement is achievable given Release_Effecting_Capability and Constraints.

«activity»
determine_release_progress

Description

Identify progress against the Requirement, with respect to fulfilling the Release_Effecting_Steps in accordance with the Release_Timeline.

«interface»
Release_Requirement

Description

This interface is the Requirement to effect the release of a Store, the associated cost of that Requirement, a predicted quality, and related timing information.

«requirement»
capture_store_release_requirements

Description

- To capture provided Requirements for a Store release (e.g. release Store 'X' from Location 'Y' in an armed state).

«requirement»
determine_release_effecting_steps

Description

- To determine the Release_Timeline and the associated Release_Effecting_Steps that meet the given Requirements and Constraints for the release of a Store using available Release_Effecting_Resources.

«interface»
Release_Achievement

Description

This interface is the statement of achievement against the Requirement for effecting a Store release, e.g. in response to a request for a salvo release from a single Store station, three missiles have been released.

«requirement»
identify_whether_release_requirement_is_achievable

Description

- To identify whether a Store release Requirement is achievable given current Release_Effecting_Capability.

«requirement»
determine_store_release_progress

Description

- To determine the progress of the Store release (e.g. report Store release in progress, Store released, Store jettisoned, Store misfired, or Store hung).

Figure 811: Release_Request Service Policy

Release_Request

This service determines the achievability of a Requirement for effecting a Store release given the available Resource_Capability and applicable Constraints. This service fulfils achievable Requirements when instructed (e.g. the release of store X is initiated at the specified time with the incurred power cost identified by the component) and provides a measure on achievement.

Interfaces

Release_Requirement

This interface is the Requirement to effect the release of a Store, the associated cost of that Requirement, a predicted quality, and related timing information.

Attributes

specification	The definition of a Requirement to effect the release of a Store (e.g. release an armed Sting Ray torpedo from weapons bay station 1 or jettison an extended range fuel pod from the left wing).
temporal_information	Information covering release timing, such as start and end times.
cost	The cost of effecting a release, e.g. the resources used and time taken.
predicted_quality	How well the release solution is predicted to satisfy the Requirement.

Release_Achievement

This interface is the statement of achievement against the Requirement for effecting a Store release, e.g. in response to a request for a salvo release from a single Store station, three missiles have been released.

Activities

determine_release_effecting_steps

Determine the Release_Effecting_Steps that satisfy the given Requirements and that meet the Constraints for effecting the release of a Store.

determine_release_progress

Identify progress against the Requirement, with respect to fulfilling the Release_Effecting_Steps in accordance with the Release_Timeline.

perform_release_effecting_steps

Fulfil a Requirement by performing the Release_Effecting_Steps required to effect the release of a Store.

determine_whether_release_requirement_is_achievable

Determine whether a release Requirement is achievable given Release_Effecting_Capability and Constraints.

5.4.2.47.7.1.2 Resource_Dependency

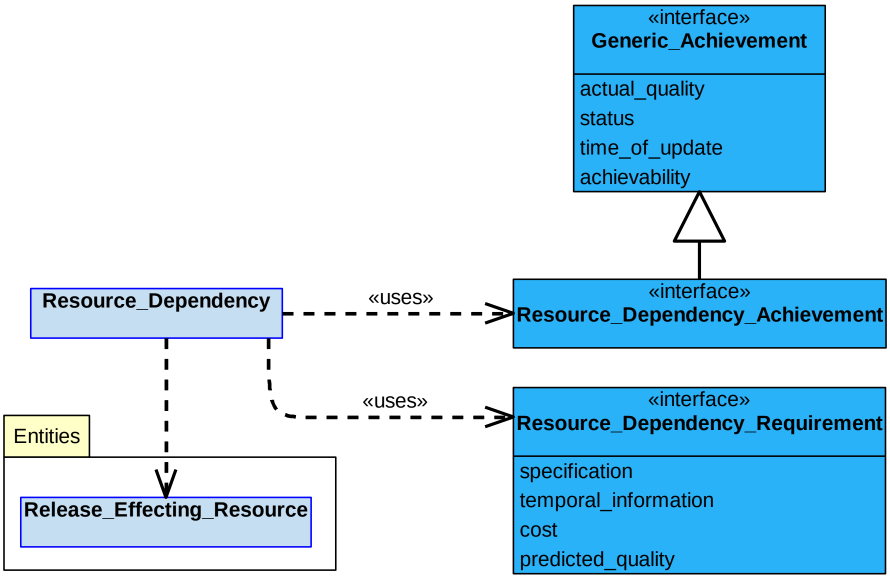
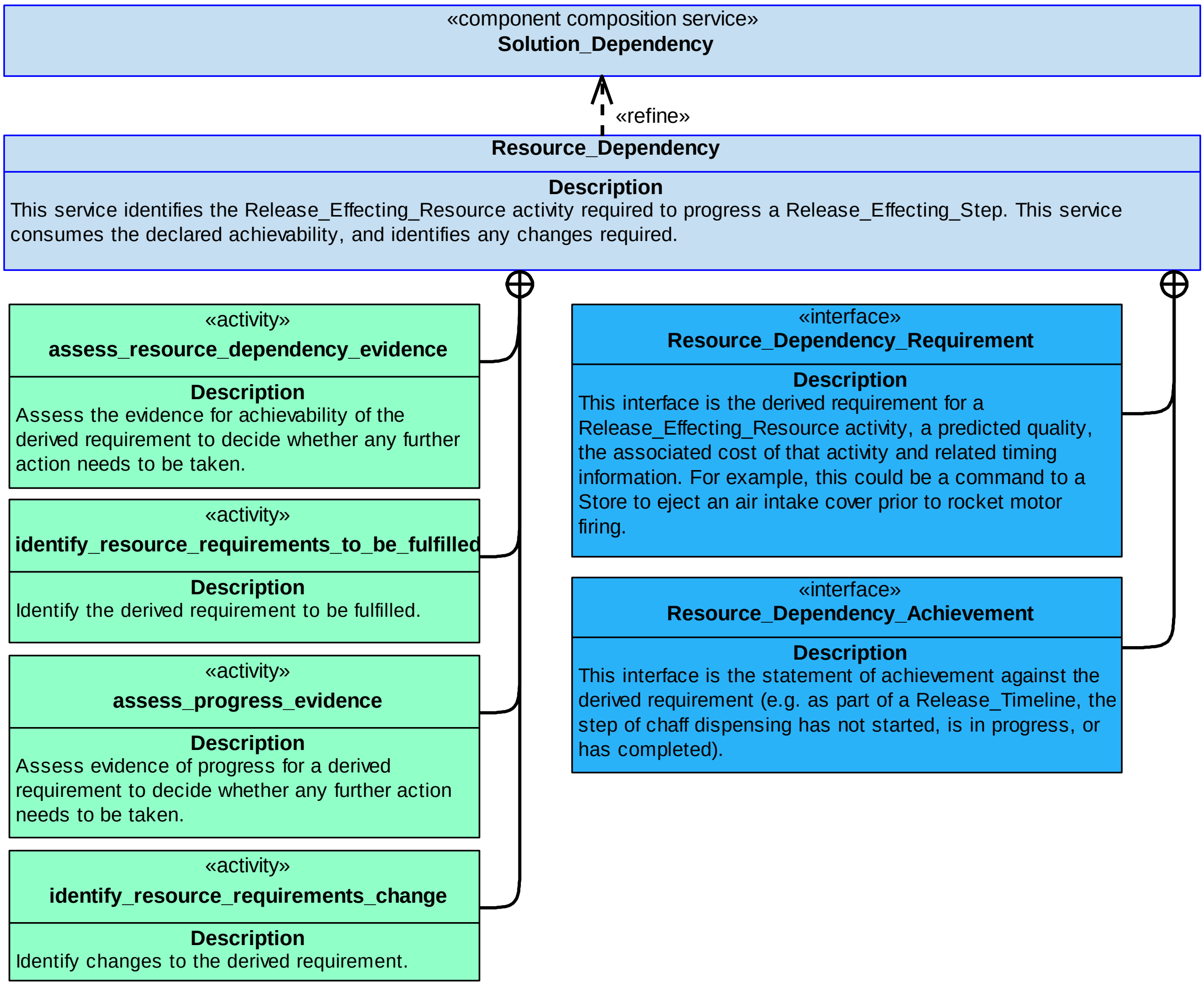


Figure 812: Resource_Dependency Service Definition



Description

This service identifies the Release_Effecting_Resource activity required to progress a Release_Effecting_Step. This service consumes the declared achievability, and identifies any changes required.

«activity»
assess_resource_dependency_evidence

Description

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

«activity»
identify_resource_requirements_to_be_fulfilled

Description

Identify the derived requirement to be fulfilled.

«activity»
assess_progress_evidence

Description

Assess evidence of progress for a derived requirement to decide whether any further action needs to be taken.

«activity»
identify_resource_requirements_change

Description

Identify changes to the derived requirement.

«interface»
Resource_Dependency_Requirement

Description

This interface is the derived requirement for a Release_Effecting_Resource activity, a predicted quality, the associated cost of that activity and related timing information. For example, this could be a command to a Store to eject an air intake cover prior to rocket motor firing.

«interface»
Resource_Dependency_Achievement

Description

This interface is the statement of achievement against the derived requirement (e.g. as part of a Release_Timeline, the step of chaff dispensing has not started, is in progress, or has completed).

Figure 813: Resource_Dependency Service Policy

Resource_Dependency

This service identifies the Release_Effecting_Resource activity required to progress a Release_Effecting_Step. This service consumes the declared achievability, and identifies any changes required.

Interfaces

Resource_Dependency_Requirement

This interface is the derived requirement for a Release_Effecting_Resource activity, a predicted quality, the associated cost of that activity and related timing information. For example, this could be a command to a Store to eject an air intake cover prior to rocket motor firing.

Attributes

specification	The definition of the activity required to be implemented by the Release_Effecting_Resource. For example, to provide power at a location, or a launcher to fire a missile.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of a Release_Effecting_Resource activity in terms of the resources used and the time taken.
predicted_quality	How well the proposed Release_Effecting_Resource activity is predicted to satisfy the Requirement.

Resource_Dependency_Achievement

This interface is the statement of achievement against the derived requirement (e.g. as part of a Release_Timeline, the step of chaff dispensing has not started, is in progress, or has completed).

Activities

assess_resource_dependency_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

assess_progress_evidence

Assess evidence of progress for a derived requirement to decide whether any further action needs to be taken.

identify_resource_requirements_to_be_fulfilled

Identify the derived requirement to be fulfilled.

identify_resource_requirements_change

Identify changes to the derived requirement.

5.4.2.47.7.1.3 Release_Precondition

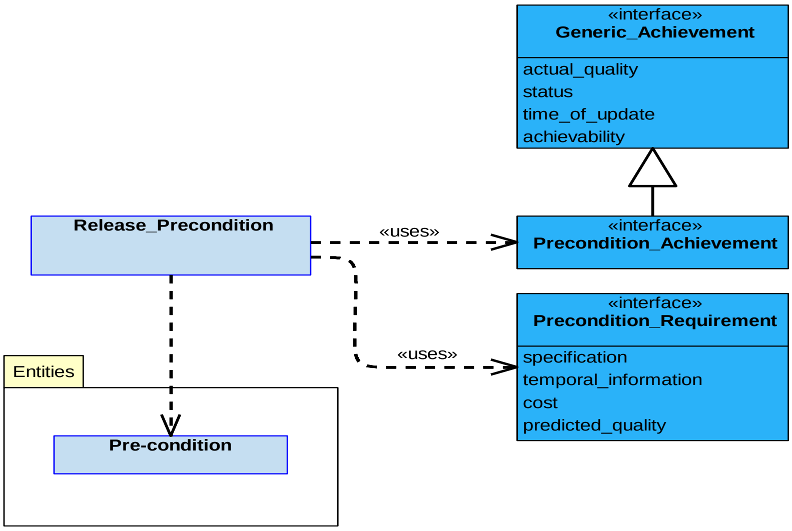
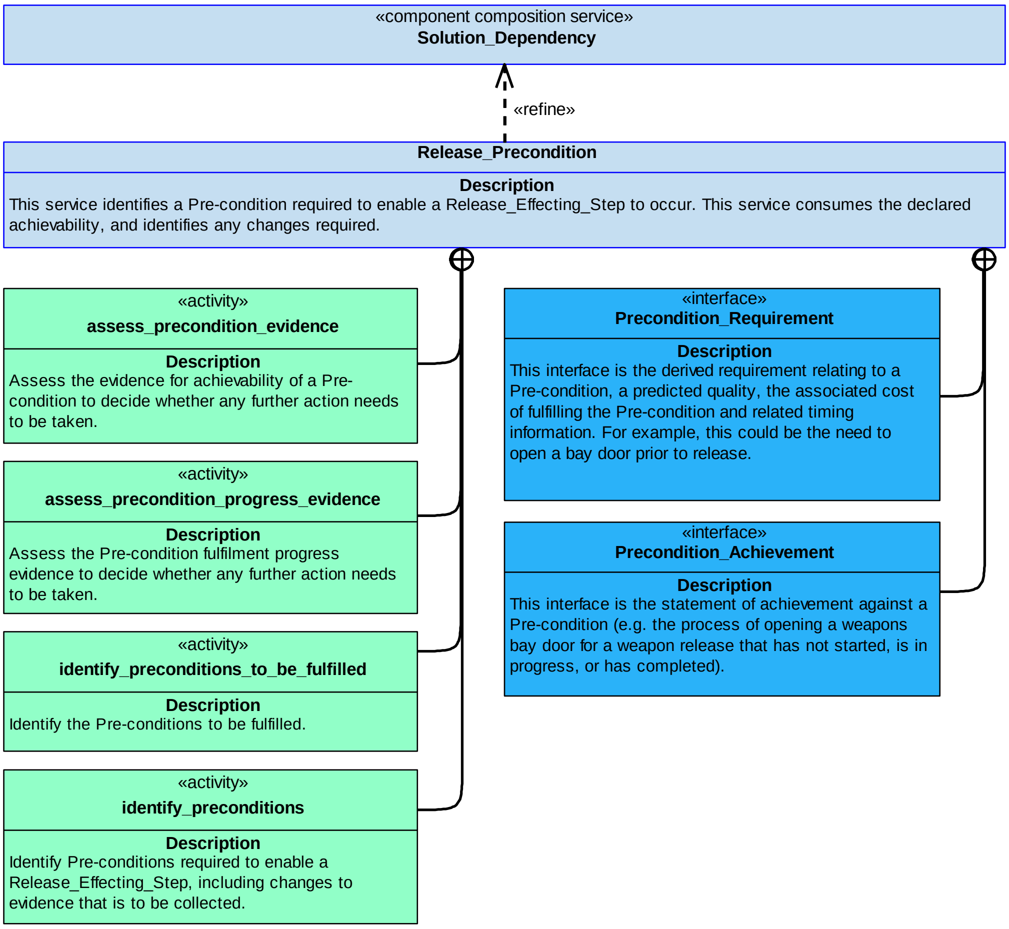


Figure 814: Release_Precondition Service Definition



Release_Precondition

Description
This service identifies a Pre-condition required to enable a Release_Effecting_Step to occur. This service consumes the declared achievability, and identifies any changes required.

«activity»
assess_precondition_evidence

Description
Assess the evidence for achievability of a Pre-condition to decide whether any further action needs to be taken.

«activity»
assess_precondition_progress_evidence

Description
Assess the Pre-condition fulfilment progress evidence to decide whether any further action needs to be taken.

«activity»
identify_preconditions_to_be_fulfilled

Description
Identify the Pre-conditions to be fulfilled.

«activity»
identify_preconditions

Description
Identify Pre-conditions required to enable a Release_Effecting_Step, including changes to evidence that is to be collected.

«interface»
Precondition_Requirement

Description
This interface is the derived requirement relating to a Pre-condition, a predicted quality, the associated cost of fulfilling the Pre-condition and related timing information. For example, this could be the need to open a bay door prior to release.

«interface»
Precondition_Achievement

Description
This interface is the statement of achievement against a Pre-condition (e.g. the process of opening a weapons bay door for a weapon release that has not started, is in progress, or has completed).

Figure 815: Release_Precondition Service Policy

Release_Precondition

This service identifies a Pre-condition required to enable a Release_Effecting_Step to occur. This service consumes the declared achievability, and identifies any changes required.

Interfaces

Precondition_Requirement

This interface is the derived requirement relating to a Pre-condition, a predicted quality, the associated cost of fulfilling the Pre-condition and related timing information. For example, this could be the need to open a bay door prior to release.

Attributes

specification	The definition of a Pre-condition that needs to be fulfilled in order to enable a Release_Effecting_Step to occur, e.g. internal Stores bay doors need to be open.
temporal_information	Information covering the timing of a Pre-condition, such as start and end times.
cost	The cost of fulfilling a Pre-condition in terms of the resources used and the time taken.
predicted_quality	How well a Pre-condition is predicted to be fulfilled.

Precondition_Achievement

This interface is the statement of achievement against a Pre-condition (e.g. the process of opening a weapons bay door for a weapon release that has not started, is in progress, or has completed).

Activities

assess_precondition_evidence

Assess the evidence for achievability of a Pre-condition to decide whether any further action needs to be taken.

assess_precondition_progress_evidence

Assess the Pre-condition fulfilment progress evidence to decide whether any further action needs to be taken.

identify_preconditions_to_be_fulfilled

Identify the Pre-conditions to be fulfilled.

identify_preconditions

Identify Pre-conditions required to enable a Release_Effecting_Step, including changes to evidence that is to be collected.

5.4.2.47.7.1.4 Release_Information

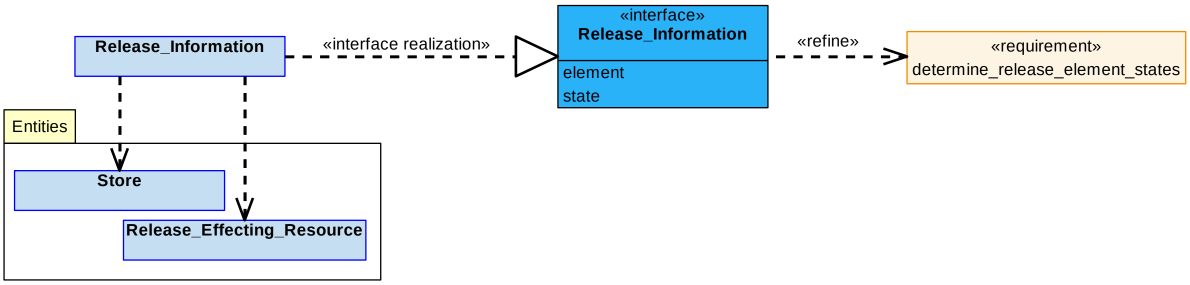
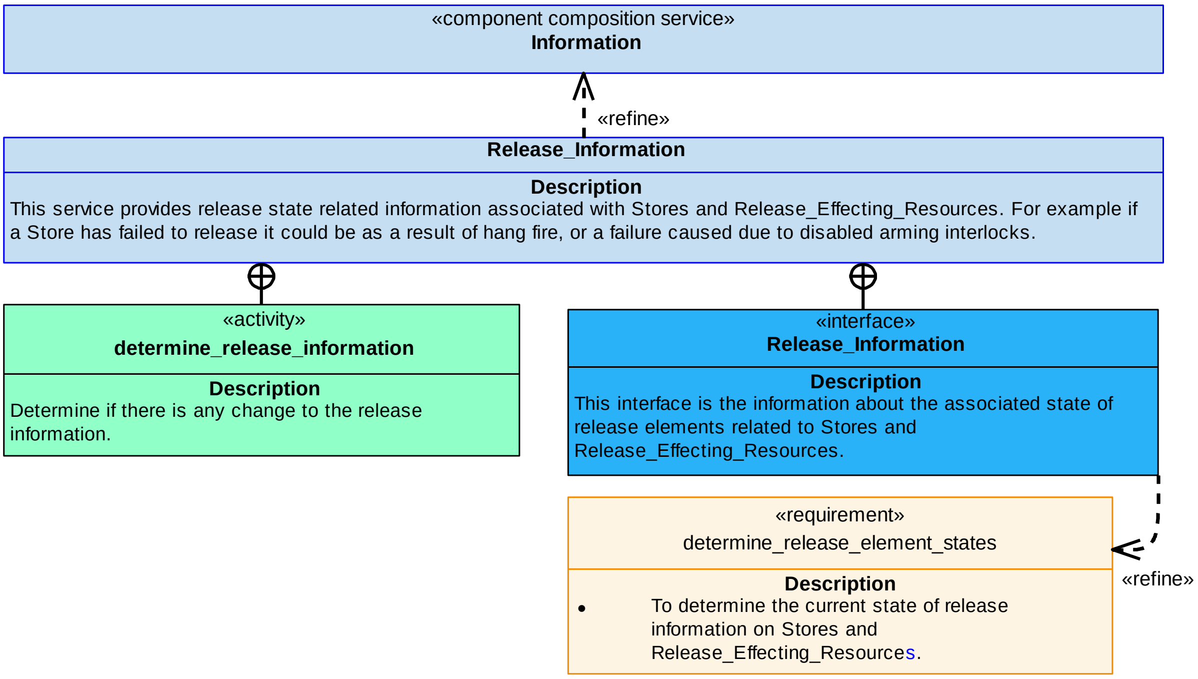


Figure 816: Release_Information Service Definition



Description
This service provides release state related information associated with Stores and Release_Effecting_Resources. For example if a Store has failed to release it could be as a result of hang fire, or a failure caused due to disabled arming interlocks.

Description
Determine if there is any change to the release information.

Description
This interface is the information about the associated state of release elements related to Stores and Release_Effecting_Resources.

Description
- To determine the current state of release information on Stores and Release_Effecting_Resources.

Figure 817: Release_Information Service Policy

Release_Information

This service provides release state related information associated with Stores and Release_Effecting_Resources. For example if a Store has failed to release it could be as a result of hang fire, or a failure caused due to disabled arming interlocks.

Interface

Release_Information

This interface is the information about the associated state of release elements related to Stores and Release_Effecting_Resources.

Attributes

element	The release element the information relates to, e.g. Store X arming.
state	The state of the element, e.g. Store X arming interlock disabled.

Activity

determine_release_information

Determine if there is any change to the release information.

5.4.2.47.7.1.5 Store_Sensor_Information

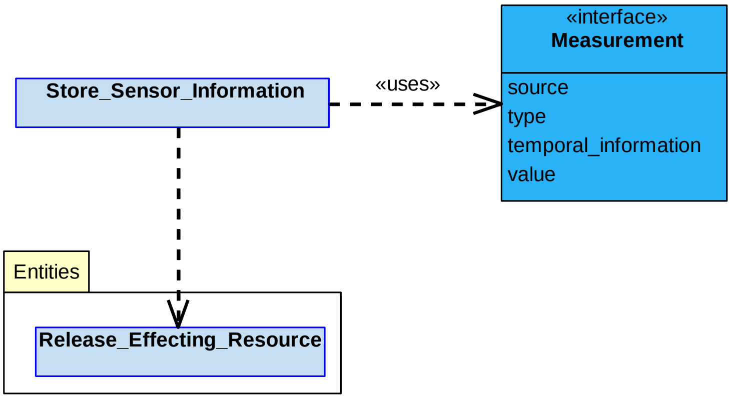


Figure 818: Store_Sensor_Information Service Definition

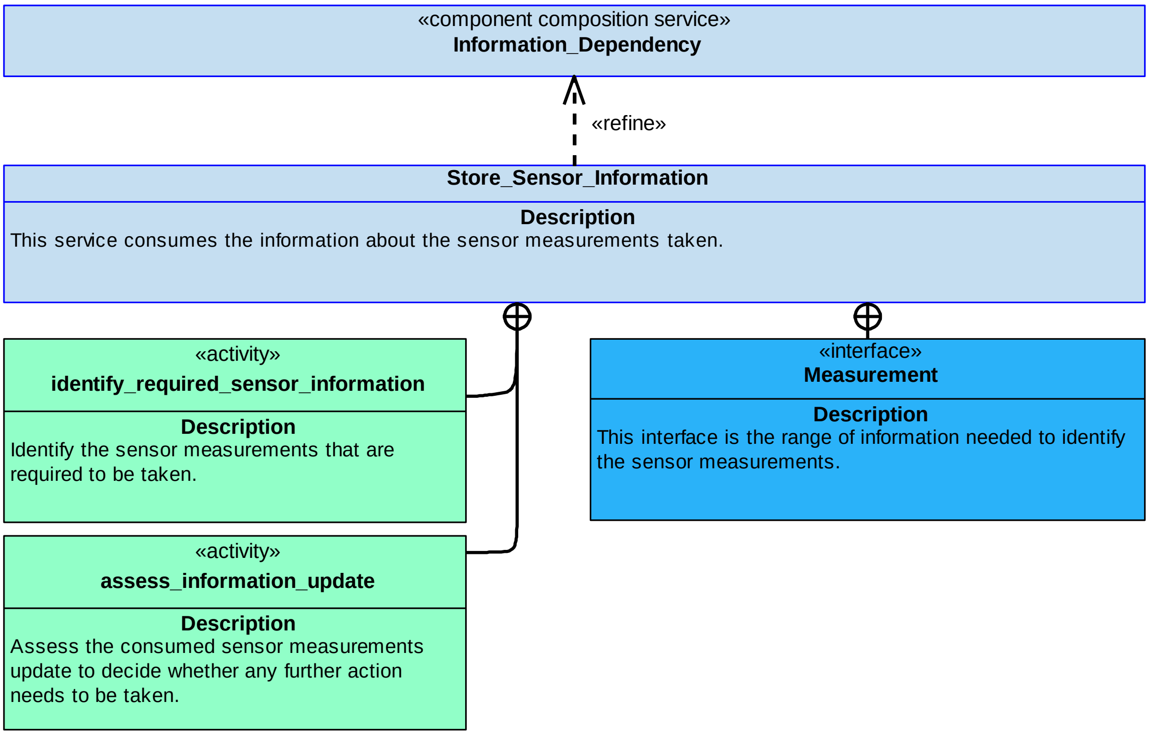

«component composition service»
Information_Dependency
«refine»
Store_Sensor_Information
Description
This service consumes the information about the sensor measurements taken.
«activity»
identify_required_sensor_information
Description
Identify the sensor measurements that are required to be taken.
«interface»
Measurement
Description
This interface is the range of information needed to identify the sensor measurements.
«activity»
assess_information_update
Description
Assess the consumed sensor measurements update to decide whether any further action needs to be taken.

Figure 819: Store_Sensor_Information Service Policy

Store_Sensor_Information

This service consumes the information about the sensor measurements taken.

Interface

Measurement

This interface is the range of information needed to identify the sensor measurements.

Attributes

source	The source of the sensed measurements, e.g. a specific sensor and its location.
type	The type of the sensed measurements, e.g. from the store-on-station sensor.
temporal_information	Information covering the timing of the information being reported.
value	Information describing the measured value of the sensor measurement.

Activities

identify_required_sensor_information

Identify the sensor measurements that are required to be taken.

assess_information_update

Assess the consumed sensor measurements update to decide whether any further action needs to be taken.

5.4.2.47.7.1.6 Store_Information

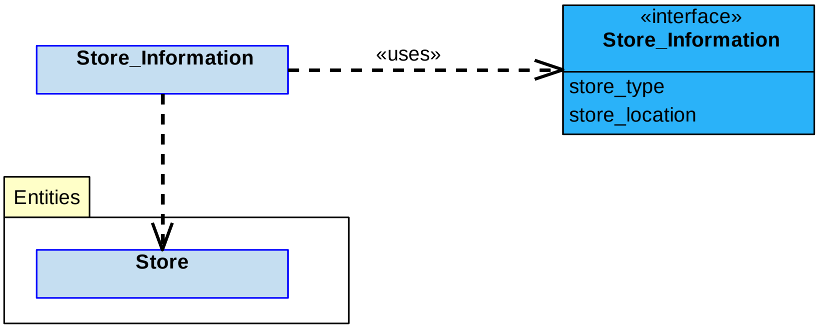


Figure 820: Store_Information Service Definition

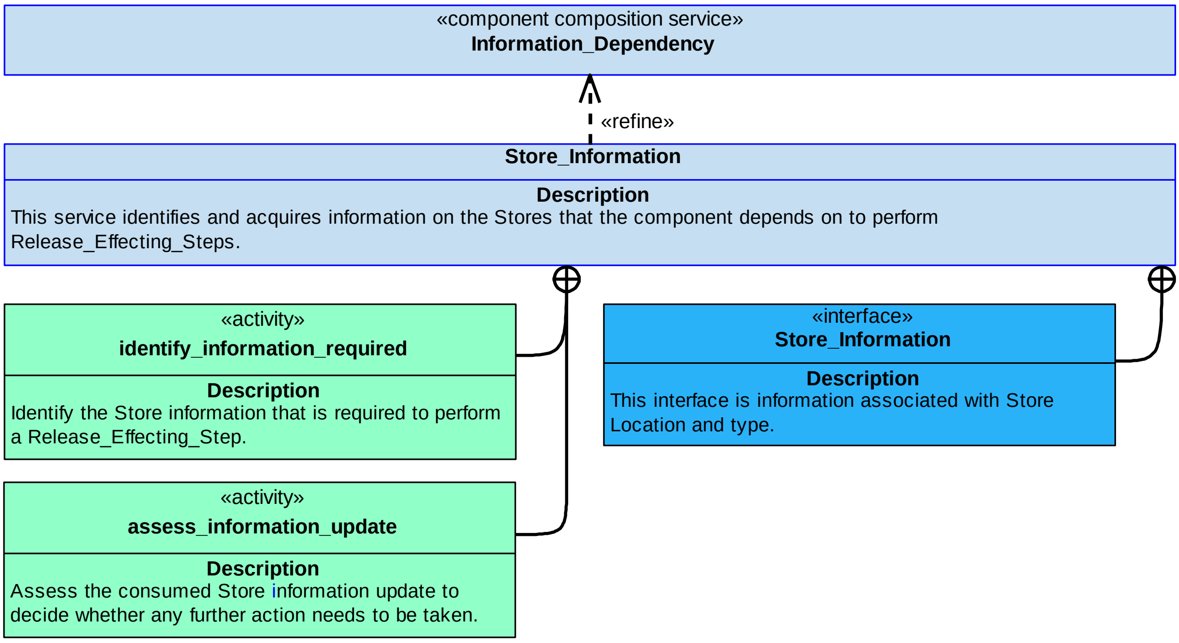

«component composition service»
Information_Dependency
«refine»
Store_Information
Description
This service identifies and acquires information on the Stores that the component depends on to perform Release_Effecting_Steps.
«activity»
identify_information_required
Description
Identify the Store information that is required to perform a Release_Effecting_Step.
«activity»
assess_information_update
Description
Assess the consumed Store information update to decide whether any further action needs to be taken.
«interface»
Store_Information
Description
This interface is information associated with Store Location and type.

Figure 821: Store_Information Service Policy

Store_Information

This service identifies and acquires information on the Stores that the component depends on to perform Release_Effecting_Steps.

Interface

Store_Information

This interface is information associated with Store Location and type.

Attributes

store_type	The definition of a Store type, e.g. GPS bomb.
store_location	The definition of a Store Location, e.g. station Y.

Activities

identify_information_required

Identify the Store information that is required to perform a Release_Effecting_Step.

assess_information_update

Assess the consumed Store information update to decide whether any further action needs to be taken.

5.4.2.47.7.1.7 Constraint

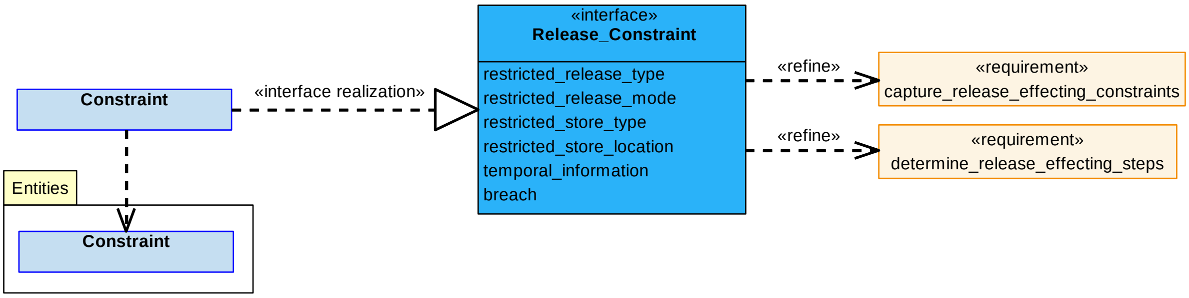


Figure 822: Constraint Service Definition

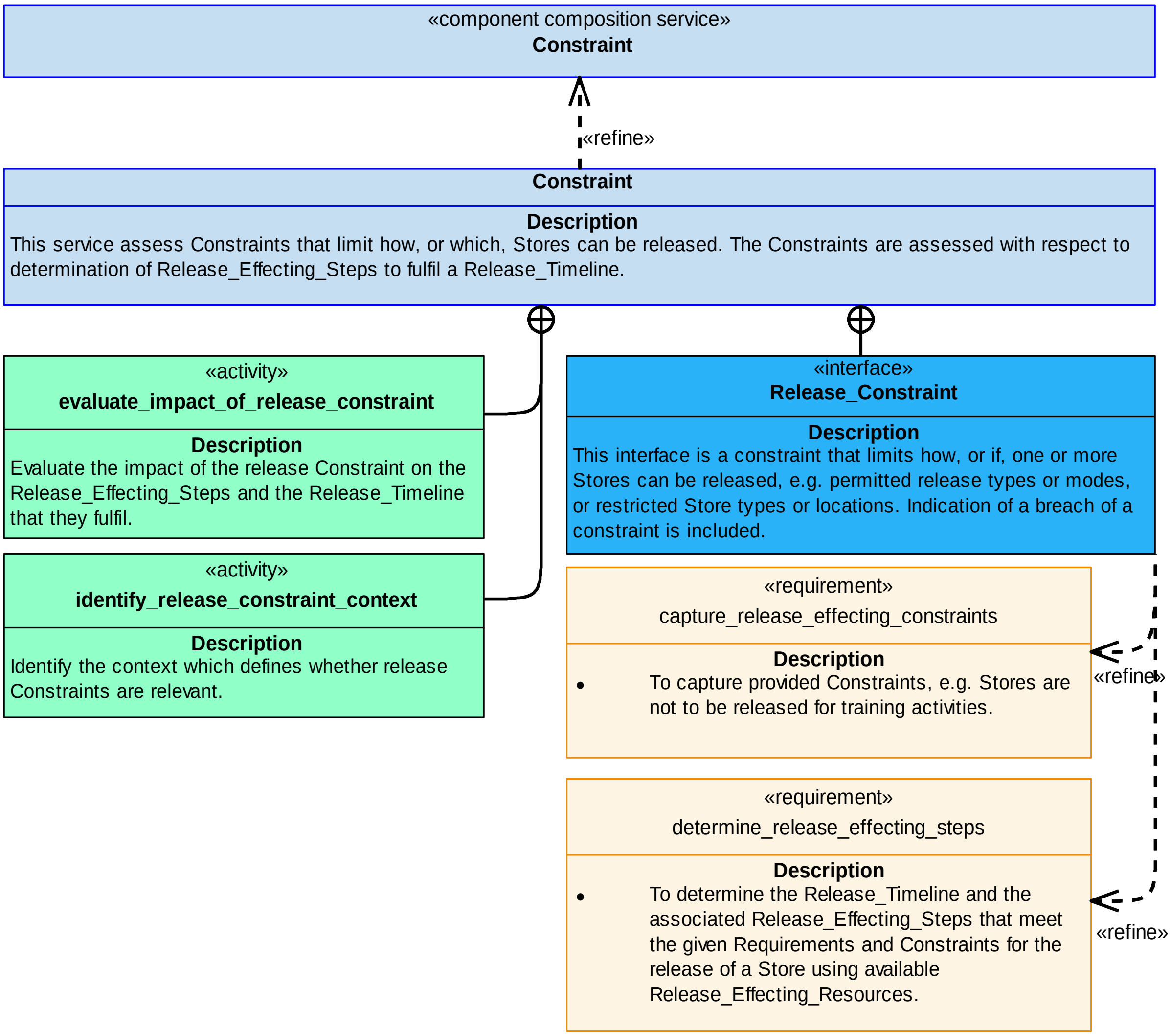

«component composition service»
Constraint
«refine»
Constraint
Description
This service assess Constraints that limit how, or which, Stores can be released. The Constraints are assessed with respect to determination of Release_Effecting_Steps to fulfil a Release_Timeline.
«activity»
evaluate_impact_of_release_constraint
Description
Evaluate the impact of the release Constraint on the Release_Effecting_Steps and the Release_Timeline that they fulfil.
«activity»
identify_release_constraint_context
Description
Identify the context which defines whether release Constraints are relevant.
«interface»
Release_Constraint
Description
This interface is a constraint that limits how, or if, one or more Stores can be released, e.g. permitted release types or modes, or restricted Store types or locations. Indication of a breach of a constraint is included.
«requirement»
capture_release_effecting_constraints
Description
• To capture provided Constraints, e.g. Stores are not to be released for training activities.
«refine»
«requirement»
determine_release_effecting_steps
Description
• To determine the Release_Timeline and the associated Release_Effecting_Steps that meet the given Requirements and Constraints for the release of a Store using available Release_Effecting_Resources.
«refine»

Figure 823: Constraint Service Policy

Constraint

This service assess Constraints that limit how, or which, Stores can be released. The Constraints are assessed with respect to determination of Release_Effecting_Steps to fulfil a Release_Timeline.

Interface

Release_Constraint

This interface is a constraint that limits how, or if, one or more Stores can be released, e.g. permitted release types or modes, or restricted Store types or locations. Indication of a breach of a constraint is included.

Attributes

restricted_release_type	The release types that are restricted for use in a release, e.g. no jettison (which may be due to the vehicle being on the ground).
restricted_release_mode	The release modes that are restricted for use in a release, e.g. no forward firing on the centreline (which may be due to nose wheel landing gear down).
restricted_store_type	The type of Store(s) that are not to be released, e.g. designator pod.
restricted_store_location	The Store Location(s) that are restricted for use in release, e.g. not the outboards (which may be for wing loading).
temporal_information	Information covering timing of the Constraint, such as for how long the constraint will be applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_release_constraint

Evaluate the impact of the release Constraint on the Release_Effecting_Steps and the Release_Timeline that they fulfil.

identify_release_constraint_context

Identify the context which defines whether release Constraints are relevant.

5.4.2.47.7.1.8 Release_Capability

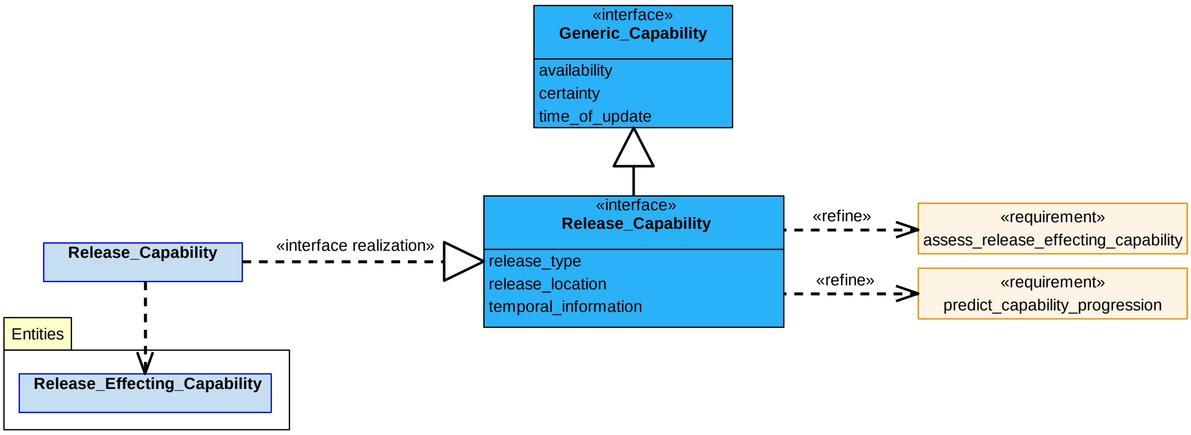


Figure 824: Release_Capability Service Definition

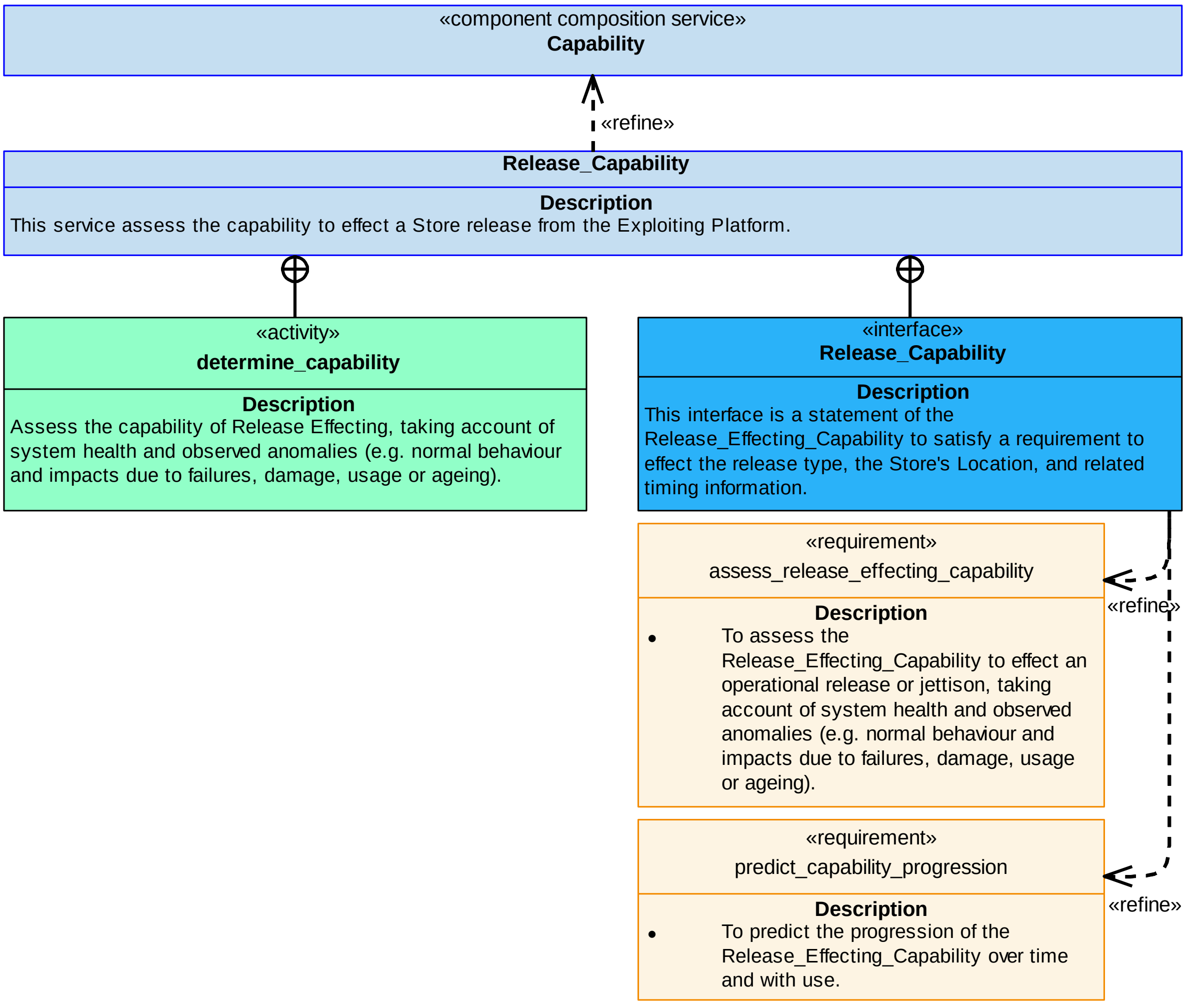

«component composition service»
Capability
«refine»
Release_Capability
Description
This service assess the capability to effect a Store release from the Exploiting Platform.
«activity»
determine_capability
Description
Assess the capability of Release Effecting, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Release_Capability
Description
This interface is a statement of the Release_Effecting_Capability to satisfy a requirement to effect the release type, the Store's Location, and related timing information.
«requirement»
assess_release_effecting_capability
Description
To assess the Release_Effecting_Capability to effect an operational release or jettison, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the Release_Effecting_Capability over time and with use.
«refine»

Figure 825: Release_Capability Service Policy

Release_Capability

This service assess the capability to effect a Store release from the Exploiting Platform.

Interface

Release_Capability

This interface is a statement of the Release_Effecting_Capability to satisfy a requirement to effect the release type, the Store's Location, and related timing information.

Attributes

release_type	The available types of release that can be applied to a Store (e.g. armed release or jettison).
release_location	The Location of the Store (e.g. left inboard inner station or station seven).
temporal_information	Information covering timing of the Release_Effecting_Capability. For example, how much time a missile can operate for on internal power before being committed to a release.

Activity

determine_capability

Assess the capability of Release Effecting, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.47.7.1.9 Capability_Evidence

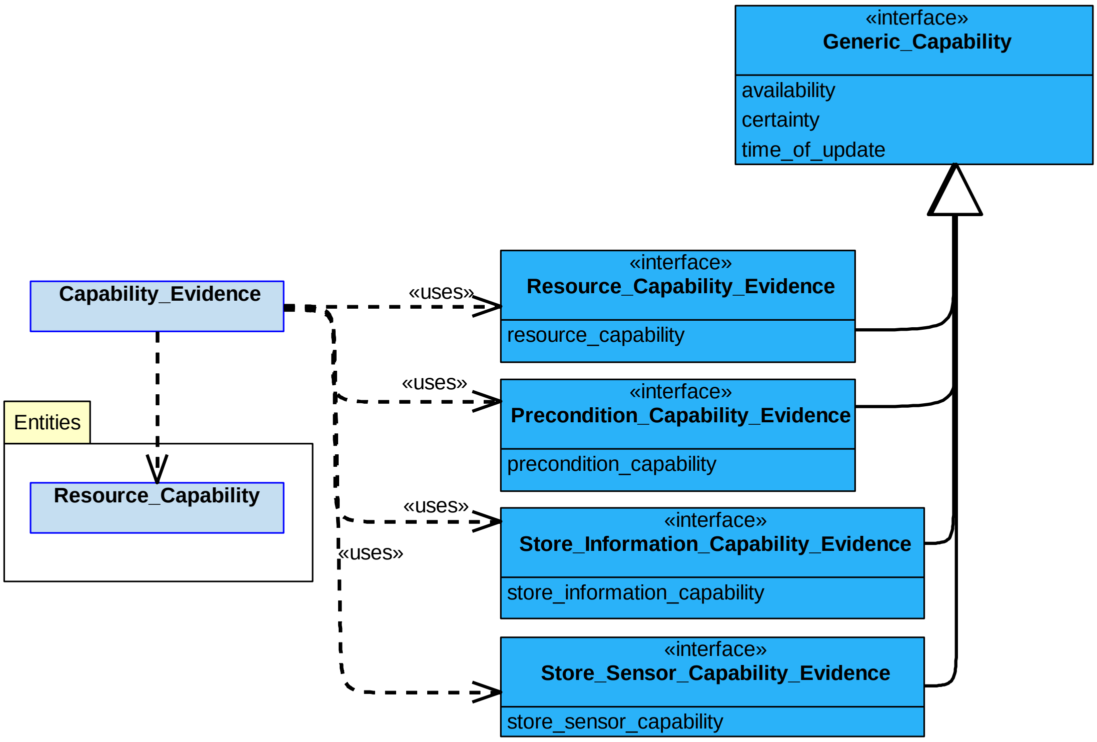


Figure 826: Capability_Evidence Service Definition

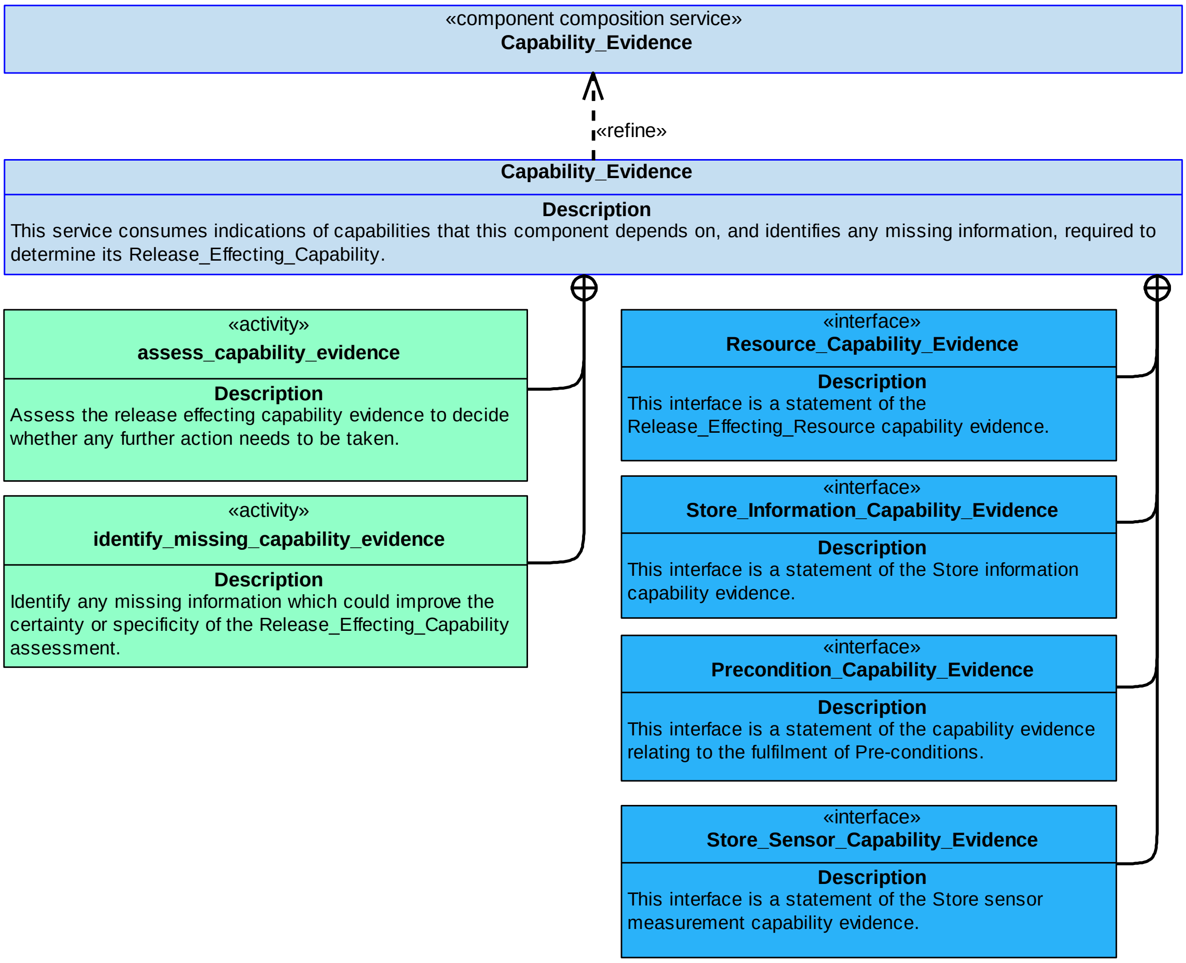

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes indications of capabilities that this component depends on, and identifies any missing information, required to determine its Release_Effecting_Capability.
«activity»
assess_capability_evidence
Description
Assess the release effecting capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any missing information which could improve the certainty or specificity of the Release_Effecting_Capability assessment.
«interface»
Resource_Capability_Evidence
Description
This interface is a statement of the Release_Effecting_Resource capability evidence.
«interface»
Store_Information_Capability_Evidence
Description
This interface is a statement of the Store information capability evidence.
«interface»
Precondition_Capability_Evidence
Description
This interface is a statement of the capability evidence relating to the fulfilment of Pre-conditions.
«interface»
Store_Sensor_Capability_Evidence
Description
This interface is a statement of the Store sensor measurement capability evidence.

Figure 827: Capability_Evidence Service Policy

Capability_Evidence

This service consumes indications of capabilities that this component depends on, and identifies any missing information, required to determine its Release_Effecting_Capability.

Interfaces

Resource_Capability_Evidence

This interface is a statement of the Release_Effecting_Resource capability evidence.

Attribute

resource_capability	The specific capability of a Release_Effecting_Resource for which capability evidence is applicable. For example, the Exploiting Platform's capability to provide power, or a launcher's capability to release missiles.

Store_Information_Capability_Evidence

This interface is a statement of the Store information capability evidence.

Attribute

store_information_capability	The specific Store and its information for which capability evidence is applicable.

Precondition_Capability_Evidence

This interface is a statement of the capability evidence relating to the fulfilment of Pre-conditions.

Attribute

precondition_capability	The specific capability relating to fulfilment of a Pre-condition for which capability evidence is applicable. For example, the Exploiting Platform's ability to open a bomb bay door.

Store_Sensor_Capability_Evidence

This interface is a statement of the Store sensor measurement capability evidence.

Attribute

store_sensor_capability	The specific sensor and measurement type for which capability evidence is applicable.

Activities

assess_capability_evidence

Assess the release effecting capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any missing information which could improve the certainty or specificity of the Release_Effecting_Capability assessment.

5.4.2.47.7.2 Service Dependencies

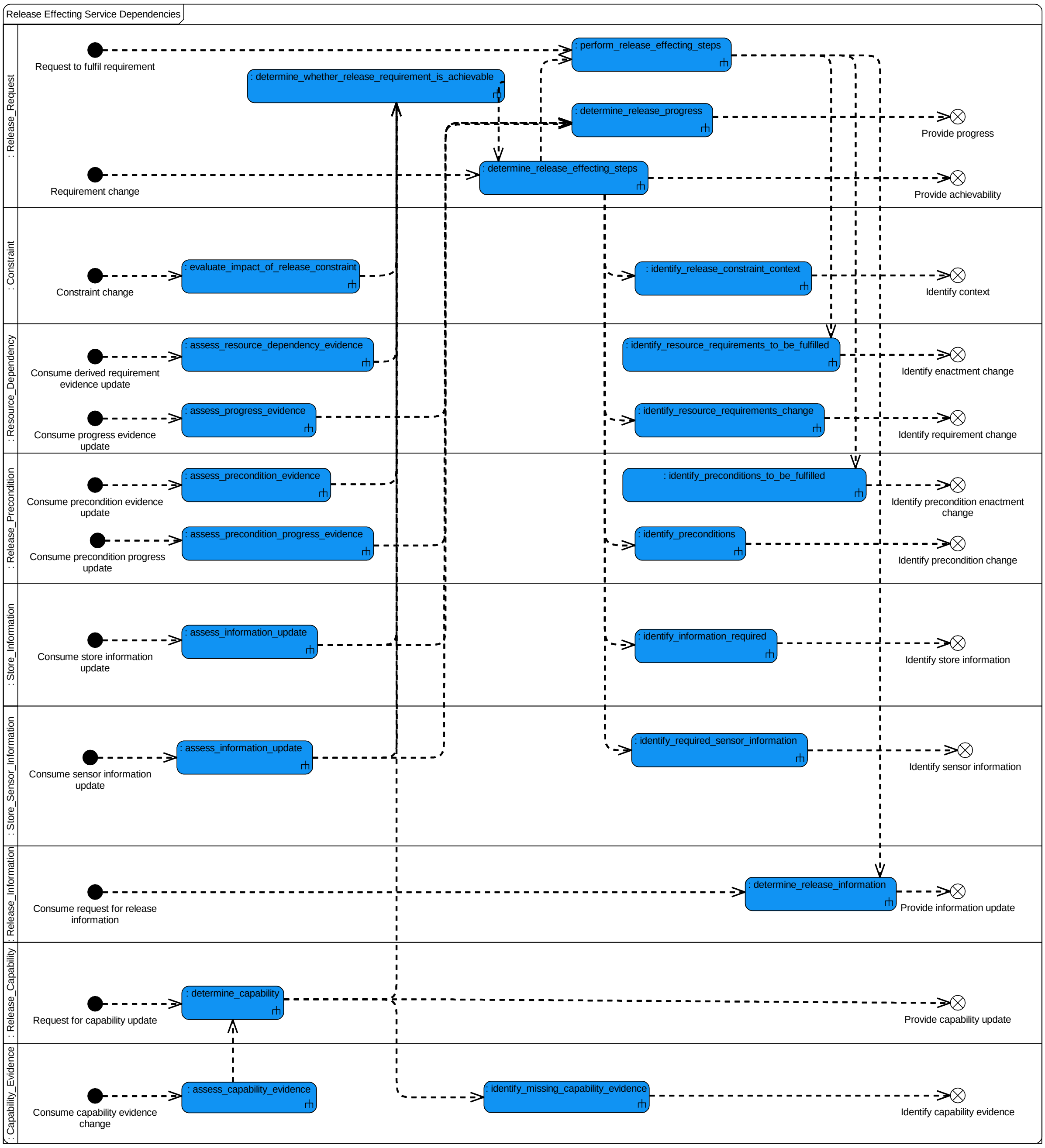

Release Effecting Service Dependencies
: Release_Request
: Constraint
: Resource_Dependency
: Release_Precondition
: Store_Information
: Store_Sensor_Information
: Release_Information
: Release_Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
: perform_release_effecting_steps
: determine_whether_release_requirement_is_achievable
: determine_release_progress
: determine_release_effecting_steps
Provide progress
Provide achievability
Constraint change
: evaluate_impact_of_release_constraint
: identify_release_constraint_context
Identify context
Consume derived requirement evidence update
: assess_resource_dependency_evidence
: identify_resource_requirements_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_progress_evidence
: identify_resource_requirements_change
Identify requirement change
Consume precondition evidence update
: assess_precondition_evidence
: identify_preconditions_to_be_fulfilled
Identify precondition enactment change
Consume precondition progress update
: assess_precondition_progress_evidence
: identify_preconditions
Identify precondition change
Consume store information update
: assess_information_update
: identify_information_required
Identify store information
Consume sensor information update
: assess_information_update
: identify_required_sensor_information
Identify sensor information
Consume request for release information
: determine_release_information
Provide information update
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 828: Release Effecting Service Dependencies

5.4.2.48 Routes

5.4.2.48.1 Role

The role of Routes is to determine and manage the execution of a route to satisfy positioning requirements for a vehicle.

5.4.2.48.2 Overview

Control Architecture

Routes is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a tasker requires a Route to be determined, it will place one or more Positioning_Requirements on Routes. Routes will then determine a Route as a solution which has associated Cost(s) and Pre-condition(s), taking into account any Routing_Constraints. Once authorisation has been given for the Route and other Pre-condition(s) fulfilled, the Route can be selected, so Routes will issue commands to execute the Route. Routes monitors the Route in order to determine if the associated Positioning_Requirement(s) are being fulfilled.

Examples of Use

- Routes will be required for planned movements of a Vehicle between two positions.
- Routes will be required for determining attack and search Routes based on target-related Positioning_Requirements.

5.4.2.48.3 Service Summary

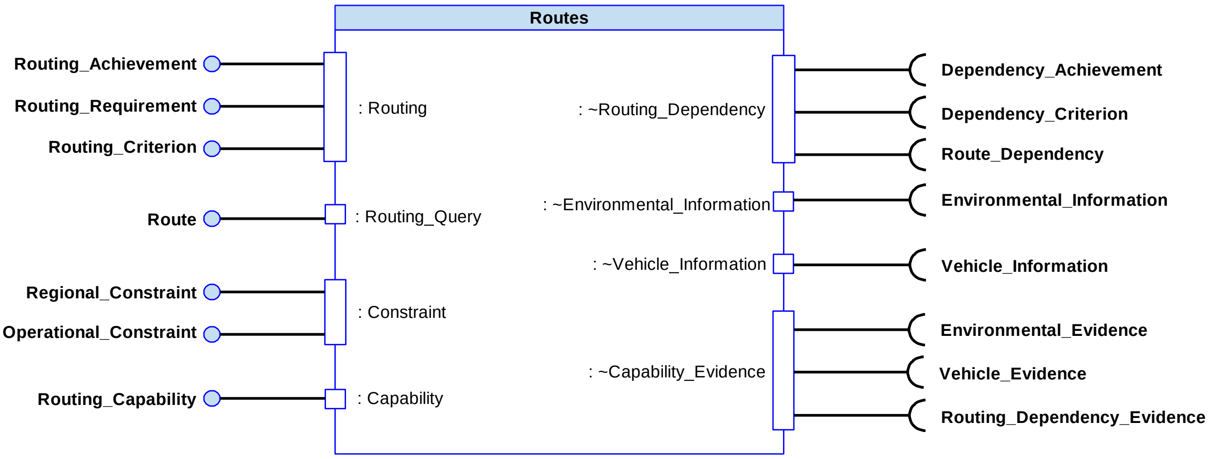


Figure 829: Routes Service Summary

5.4.2.48.4 Responsibilities

capture_positioning_requirements

- To capture given Positioning_Requirements.

capture_measurement_criteria

- To capture provided Measurement_Criterion (e.g. fuel Cost) for Routes.

capture_routing_constraints

- To capture given Routing_Constraints (e.g. weather volumes and threat volumes).

identify_whether_requirement_remains_achievable

- To identify whether a Positioning_Requirement is still achievable given current or predicted Routing_Capability.

determine_route

- To determine a Route that meets the given Positioning_Requirements within the Vehicle_Capability and the given Routing_Constraint (e.g. volumes to remain within or avoid).

identify_pre-conditions

- To identify Pre-conditions required to support a Route or a portion of a Route.

command_route

- To execute a selected Route by commanding the Vehicle to follow a sequence of routepoints.

determine_route_progress

- To determine the progress of a Vehicle against the selected Route.

determine_routing_continuity

- To determine the positional continuity between the adjacent elements of a planned path as well as the overall completeness of the planned path. This could be as part of pre-planning, or ensuring continuity between what is currently being enacted and subsequently planned.

collate_route_cost

- To collate the Cost (e.g. time or fuel use) for any generated Route against the provided Measurement_Criterion.

assess_routing_capability

- To determine the Routing_Capability, taking into account system health and observed anomalies.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Routing_Capability assessment.

predict_routing_capability

- To predict the progression of Routing_Capability over time and with use, taking account of system health and observed anomalies.

5.4.2.48.5 Subject Matter Semantics

The subject matter of Routes is the Route by which a Vehicle can fulfil one or more Positioning_Requirements whilst complying with given Routing_Constraints, as well as the time limits within which the Vehicle is required to arrive at route points.

Exclusions

The subject matter of Routes does not include:

- Obtaining authorisation for a Route.
- The detailed control of the vehicle in order to follow a Route.
- Aspects of vehicle motion beyond the path of a vehicle.
- Defining the speed of the vehicle along the Route.

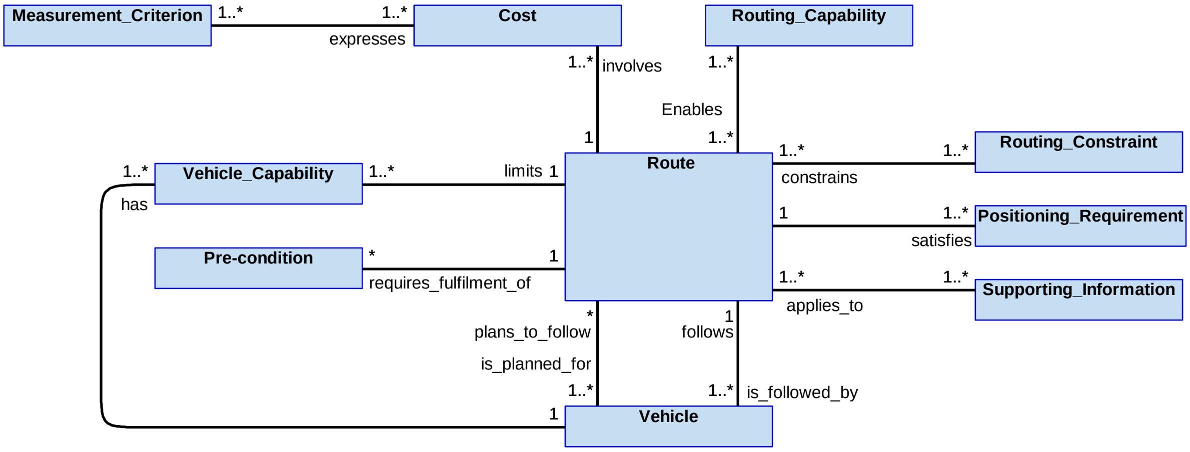


Figure 830: Routes Semantics

5.4.2.48.5.1 Entities

Cost

The predicted amount of a specified entity (e.g. fuel, time or airspace) that will be used during the enactment of the Route.

Measurement_Criterion

Criteria that needs to be costed when determining a Route.

Positioning_Requirement

Requirements to be satisfied by a Route (e.g. a specific point or volume to be passed through within a specified time window).

Pre-condition

Items which need to be fulfilled (e.g. initial position or authorisation) before the Route can be enacted.

Route

The generated path that has to be followed.

Routing_Capability

The ability to determine a Route.

Routing_Constraint

Limitations to be considered when determining a Route (e.g. volumes to remain within or without).

Supporting_Information

Information to be considered when planning a Route. For example, this could be information about the Vehicle or information about the operating environment.

Vehicle

A moveable object that requires a Route.

Vehicle_Capability

The capability of the Vehicle to execute Routes.

5.4.2.48.6 Design Rationale

5.4.2.48.6.1 Assumptions

- Different routing needs will lead to different Positioning_Requirements being placed, e.g. a transit route will have different requirements to a terrain following route.
- Positioning_Requirements placed upon this component could take the form of:
    - Requirements to reach a point at a particular time.
    - Requirements to reach points in a particular sequence.
    - Requirements to perform a particular pattern.
    - Requirements to remain within particular execution limits (e.g. between a minimum and maximum altitude).

5.4.2.48.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Routes:

- Data Driving - The types of Pre-conditions and Costs that can be attributed to a Route could be data-driven if they vary per mission.
- Component Extensions - Different extensions could be developed to achieve different types of Positioning_Requirements.
- Multi-Vehicle Coordination **-** Multiple vehicles may require separate instances of Routes, therefore the Multi-Vehicle Coordination PYRAMID concept is applicable.

Extensions

- New and/or different methods of determining Routes could be developed as extensions.

Exploitation Considerations

- There could be a single or multiple instance(s) of Routes for multiple vehicles (in accordance with Multi-Vehicle Coordination).

5.4.2.48.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

- If positioning requirements from ATS/crew (e.g. airways) are corrupted when being handled by Routes, then the flight path flown by the Vehicle would not meet airspace integration requirements. This could lead to mid-air collision. However, other barriers (e.g. ATS monitoring of air vehicle position and ACAS) mitigate the risk of an actual collision occurring.
- This component collates the cost (fuel required) for the Route. Failure of this calculation could cause fuel to run-out unexpectedly and result in loss of thrust. Loss of thrust could result in fatalities (i.e. catastrophic). However, it is reasonable that Fluids would determine that the available fuel contents are critically low (against a simple fixed threshold) which provides time to take mitigating actions (e.g. increase height to allow sufficient glide range or perform a CTT).

5.4.2.48.6.4 Security Considerations

The indicative security classification is SNEO.

The subject matter of this component means it will contain knowledge of the current and future Route of the vehicle, together with a degree of (in the case of an aircraft, flight) performance data, both of which will have confidentiality requirements. This leads to an indicative component security classification of SNEO.

It is assumed the integrity of the route (destination) request will be assured in order that the correct destination can be reached.

The component is expected to at least partially satisfy security related functions relating to:

- **Maintaining Audit Records** of routing requests and decisions for accountability purposes**.**
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- The generation of **Warnings and Notifications** where the route may be unsafe or infeasible, such a route may indicate the integrity of the route has been compromised (e.g. if a request is for a route through intervening hazards).

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.48.7 Services

5.4.2.48.7.1 Service Definitions

5.4.2.48.7.1.1 Routing

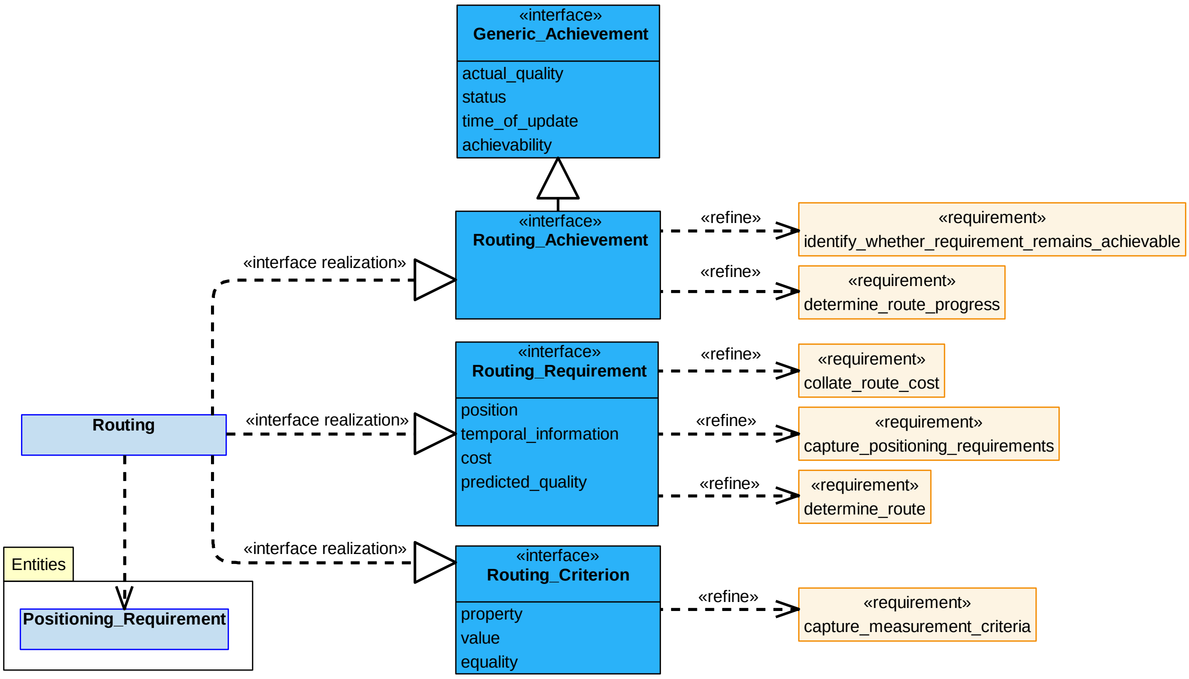
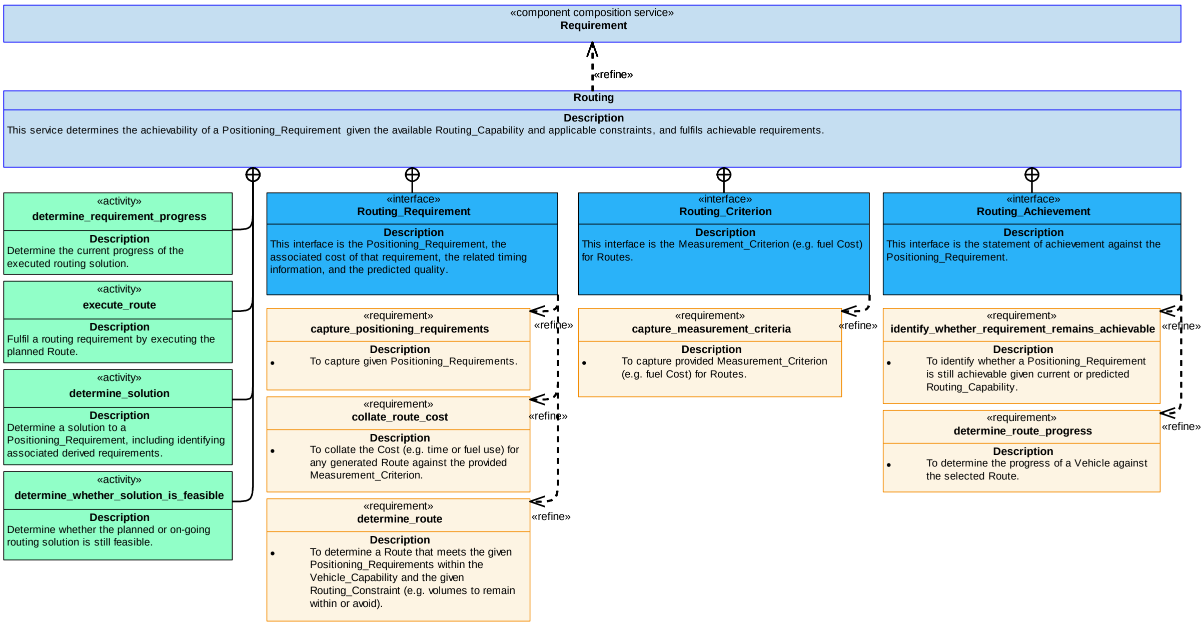


Figure 831: Routing Service Definition



Routing

Description
This service determines the achievability of a Positioning_Requirement given the available Routing_Capability and applicable constraints, and fulfils achievable requirements.

determine_requirement_progress

Description
Determine the current progress of the executed routing solution.

execute_route

Description
Fulfil a routing requirement by executing the planned Route.

determine_solution

Description
Determine a solution to a Positioning_Requirement, including identifying associated derived requirements.

determine_whether_solution_is_feasible

Description
Determine whether the planned or on-going routing solution is still feasible.

Routing_Requirement

Description
This interface is the Positioning_Requirement, the associated cost of that requirement, the related timing information, and the predicted quality.

capture_positioning_requirements

Description
- To capture given Positioning_Requirements.

collate_route_cost

Description
- To collate the Cost (e.g. time or fuel use) for any generated Route against the provided Measurement_Criterion.

determine_route

Description
- To determine a Route that meets the given Positioning_Requirements within the Vehicle_Capability and the given Routing_Constraint (e.g. volumes to remain within or avoid).

Routing_Criterion

Description
This interface is the Measurement_Criterion (e.g. fuel Cost) for Routes.

capture_measurement_criteria

Description
- To capture provided Measurement_Criterion (e.g. fuel Cost) for Routes.

Routing_Achievement

Description
This interface is the statement of achievement against the Positioning_Requirement.

identify_whether_requirement_remains_achievable

Description
- To identify whether a Positioning_Requirement is still achievable given current or predicted Routing_Capability.

determine_route_progress

Description
- To determine the progress of a Vehicle against the selected Route.

Figure 832: Routing Service Policy

Routing

This service determines the achievability of a Positioning_Requirement given the available Routing_Capability and applicable constraints, and fulfils achievable requirements.

Interfaces

Routing_Requirement

This interface is the Positioning_Requirement, the associated cost of that requirement, the related timing information, and the predicted quality.

Attributes

position	The position required to be achieved.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used, time taken.
predicted_quality	How well the proposed routing solution is predicted to satisfy the requirement.

Routing_Criterion

This interface is the Measurement_Criterion (e.g. fuel Cost) for Routes.

Attributes

property	The property to be measured, e.g. number of miles.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Routing_Achievement

This interface is the statement of achievement against the Positioning_Requirement.

Activities

determine_requirement_progress

Determine the current progress of the executed routing solution.

determine_solution

Determine a solution to a Positioning_Requirement, including identifying associated derived requirements.

execute_route

Fulfil a routing requirement by executing the planned Route.

determine_whether_solution_is_feasible

Determine whether the planned or on-going routing solution is still feasible.

5.4.2.48.7.1.2 Routing_Dependency

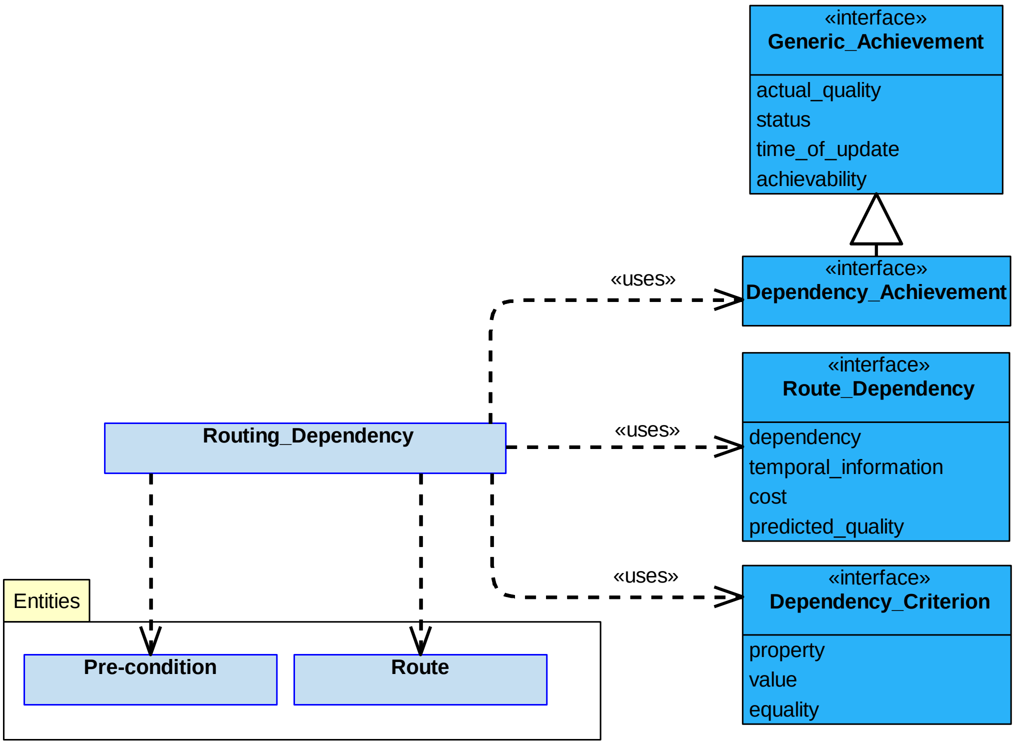


Figure 833: Routing_Dependency Service Definition

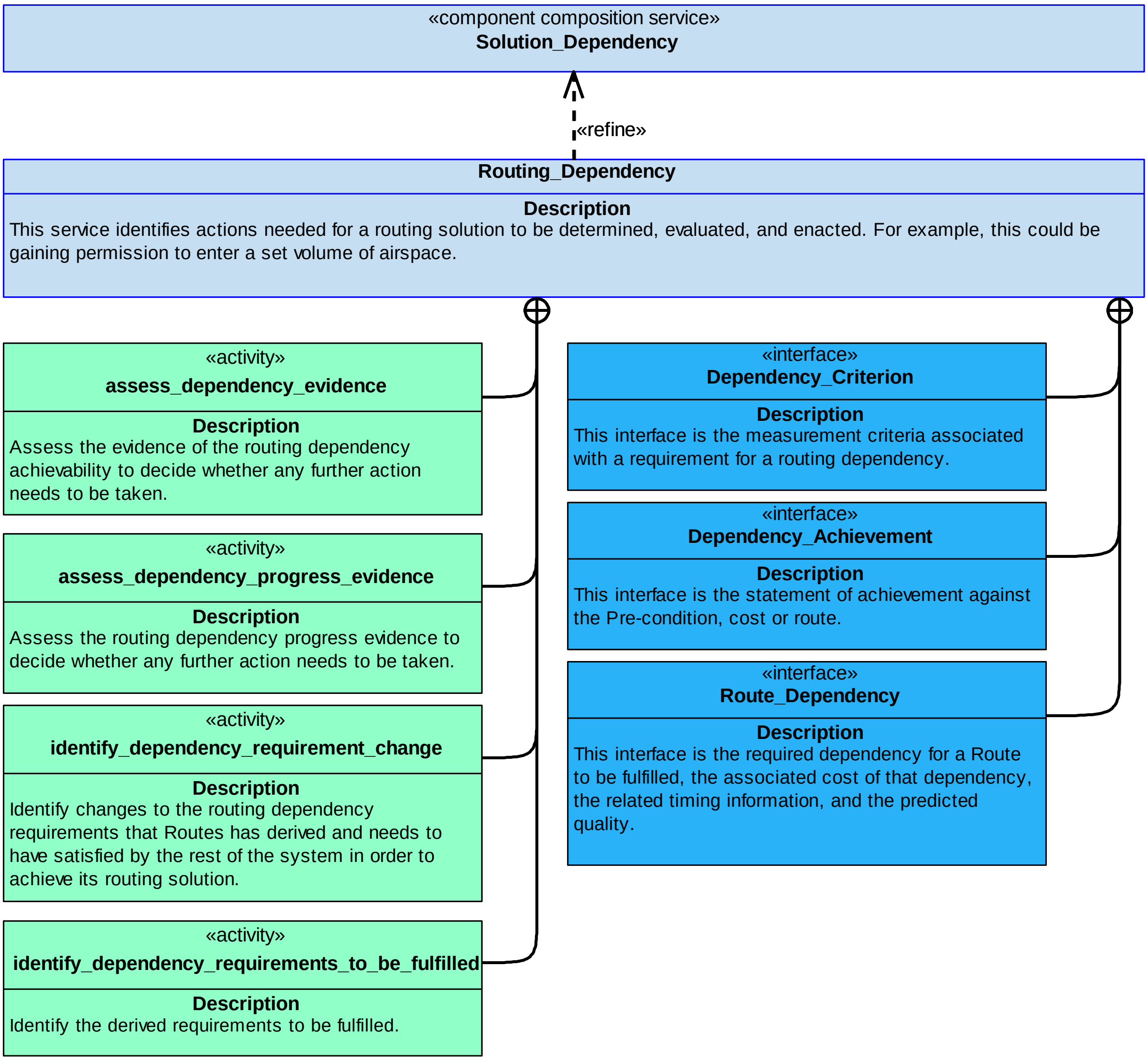

«component composition service»
Solution_Dependency
«refine»
Routing_Dependency
Description
This service identifies actions needed for a routing solution to be determined, evaluated, and enacted. For example, this could be gaining permission to enter a set volume of airspace.
«activity»
assess_dependency_evidence
Description
Assess the evidence of the routing dependency achievability to decide whether any further action needs to be taken.
«activity»
assess_dependency_progress_evidence
Description
Assess the routing dependency progress evidence to decide whether any further action needs to be taken.
«activity»
identify_dependency_requirement_change
Description
Identify changes to the routing dependency requirements that Routes has derived and needs to have satisfied by the rest of the system in order to achieve its routing solution.
«activity»
identify_dependency_requirements_to_be_fulfilled
Description
Identify the derived requirements to be fulfilled.
«interface»
Dependency_Criterion
Description
This interface is the measurement criteria associated with a requirement for a routing dependency.
«interface»
Dependency_Achievement
Description
This interface is the statement of achievement against the Pre-condition, cost or route.
«interface»
Route_Dependency
Description
This interface is the required dependency for a Route to be fulfilled, the associated cost of that dependency, the related timing information, and the predicted quality.

Figure 834: Routing_Dependency Service Policy

Routing_Dependency

This service identifies actions needed for a routing solution to be determined, evaluated, and enacted. For example, this could be gaining permission to enter a set volume of airspace.

Interfaces

Route_Dependency

This interface is the required dependency for a Route to be fulfilled, the associated cost of that dependency, the related timing information, and the predicted quality.

Attributes

dependency	The definition of the dependency required, e.g. the requirement to have certain permissions to use some airspace, the waypoints to be reached, or the fuel required to fly the Route.
temporal_information	Information covering timing, such as start and end times, e.g. how long permission is granted to be in a set airspace.

cost	The cost of executing the dependency solution, e.g. resources used or time taken.
predicted_quality	How well the proposed dependency solution is predicted to satisfy the requirement.

Dependency_Criterion

This interface is the measurement criteria associated with a requirement for a routing dependency.

Attributes

property	The property to be measured, e.g. time allowed for permission to enter airspace.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Dependency_Achievement

This interface is the statement of achievement against the Pre-condition, cost or route.

Activities

assess_dependency_evidence

Assess the evidence of the routing dependency achievability to decide whether any further action needs to be taken.

assess_dependency_progress_evidence

Assess the routing dependency progress evidence to decide whether any further action needs to be taken.

identify_dependency_requirement_change

Identify changes to the routing dependency requirements that Routes has derived and needs to have satisfied by the rest of the system in order to achieve its routing solution.

identify_dependency_requirements_to_be_fulfilled

Identify the derived requirements to be fulfilled.

5.4.2.48.7.1.3 Routing_Query

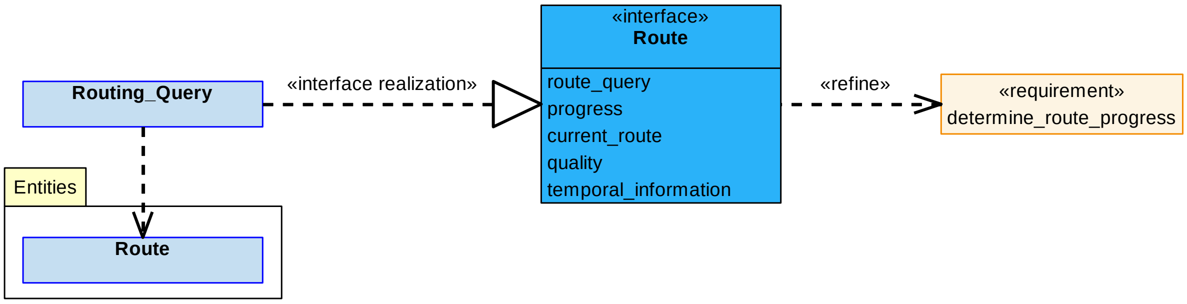


Figure 835: Routing_Query Service Definition

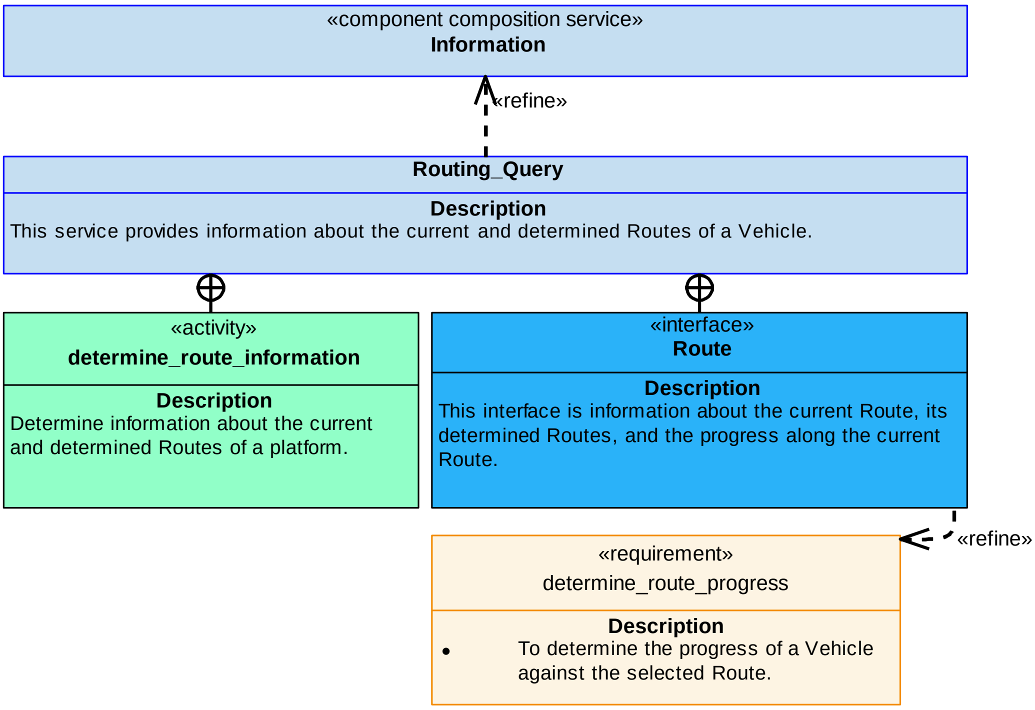

«component composition service»
Information
«refine»
Routing_Query
Description
This service provides information about the current and determined Routes of a Vehicle.
«activity»
determine_route_information
Description
Determine information about the current and determined Routes of a platform.
«interface»
Route
Description
This interface is information about the current Route, its determined Routes, and the progress along the current Route.
«refine»
«requirement»
determine_route_progress
Description
To determine the progress of a Vehicle against the selected Route.

Figure 836: Routing_Query Service Policy

Routing_Query

This service provides information about the current and determined Routes of a Vehicle.

Interface

Route

This interface is information about the current Route, its determined Routes, and the progress along the current Route.

Attributes

route_query	The definition of the query for information about the current Route and the determined Routes of a Vehicle.
progress	A Vehicles progress along a Route.
current_route	The current Route of a Vehicle.
quality	The quality of the routing information.
temporal_information	Timing information, such as when the routing information was provided.

Activity

determine_route_information

Determine information about the current and determined Routes of a platform.

5.4.2.48.7.1.4 Environmental_Information

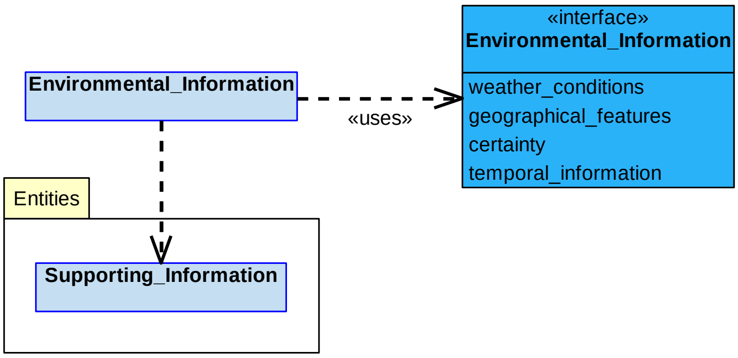


Figure 837: Environmental_Information Service Definition

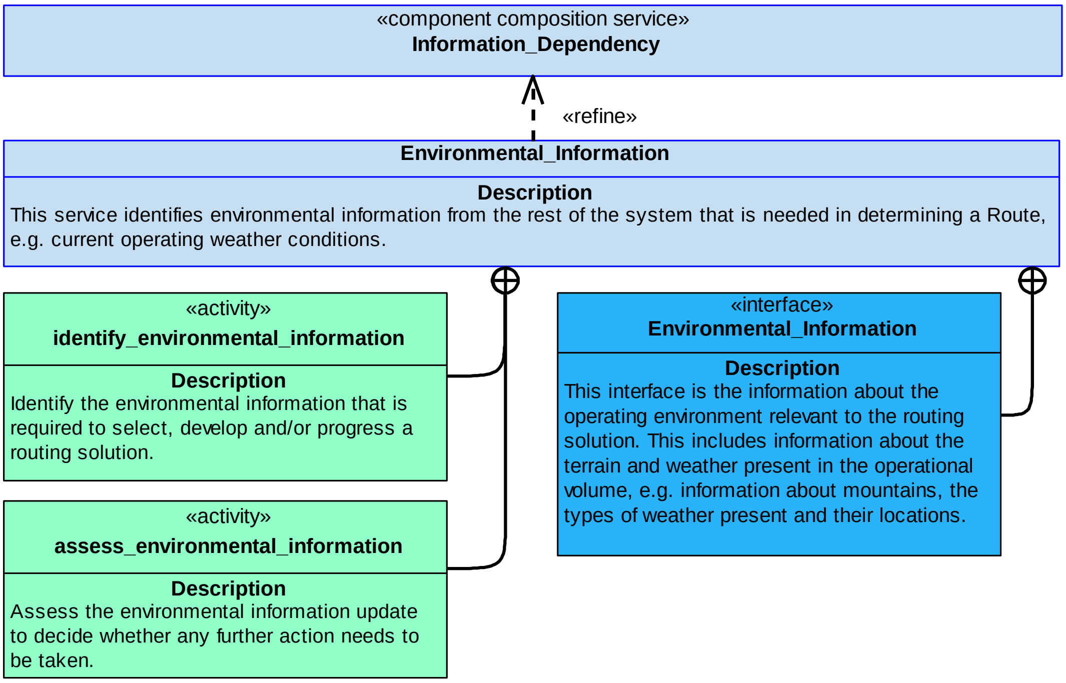

«component composition service»
Information_Dependency
«refine»
Environmental_Information
Description
This service identifies environmental information from the rest of the system that is needed in determining a Route, e.g. current operating weather conditions.
«activity»
identify_environmental_information
Description
Identify the environmental information that is required to select, develop and/or progress a routing solution.
«activity»
assess_environmental_information
Description
Assess the environmental information update to decide whether any further action needs to be taken.
«interface»
Environmental_Information
Description
This interface is the information about the operating environment relevant to the routing solution. This includes information about the terrain and weather present in the operational volume, e.g. information about mountains, the types of weather present and their locations.

Figure 838: Environmental_Information Service Policy

Environmental_Information

This service identifies environmental information from the rest of the system that is needed in determining a Route, e.g. current operating weather conditions.

Interface

Environmental_Information

This interface is the information about the operating environment relevant to the routing solution. This includes information about the terrain and weather present in the operational volume, e.g. information about mountains, the types of weather present and their locations.

Attributes

weather_conditions	The state of a type of meteorological condition at a given time and place, e.g. raining with a strong northerly wind.
geographical_features	The geographical features a Vehicle has to aviate around, e.g. a building or terrain.
certainty	The level of certainty of the environmental information.
temporal_information	Timing information pertaining to the reporting or applicability of the associated environmental information.

Activities

assess_environmental_information

Assess the environmental information update to decide whether any further action needs to be taken.

identify_environmental_information

Identify the environmental information that is required to select, develop and/or progress a routing solution.

5.4.2.48.7.1.5 Vehicle_Information

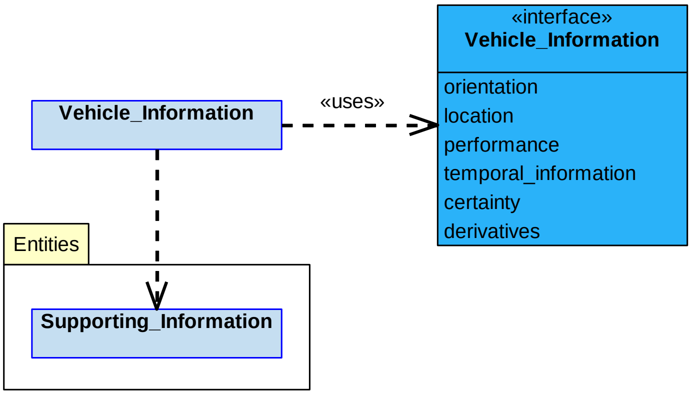


Figure 839: Vehicle_Information Service Definition

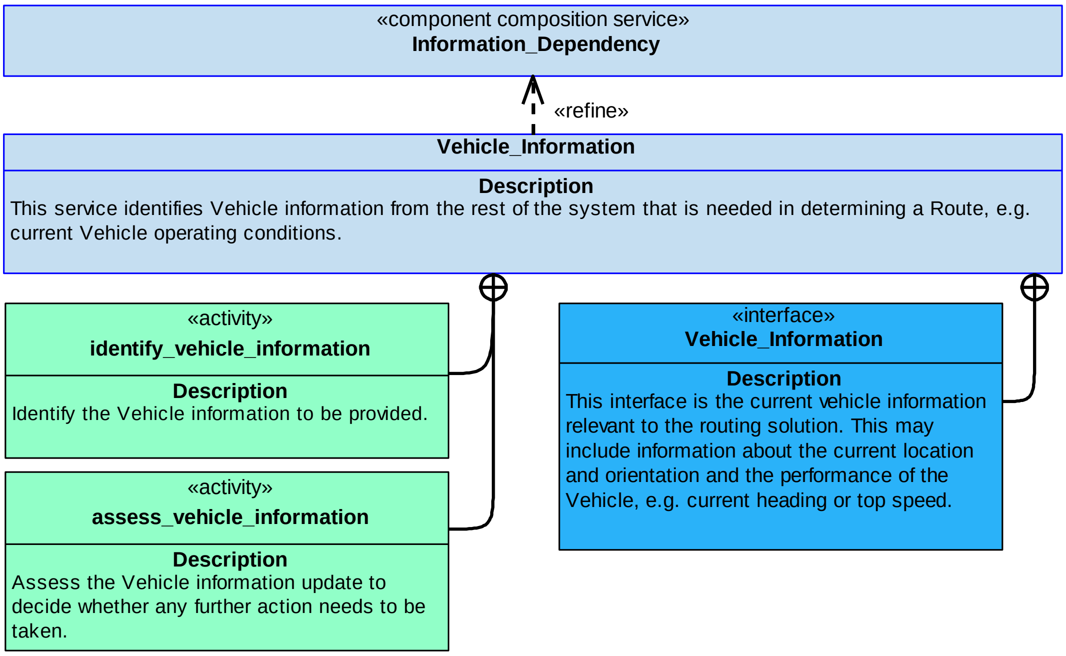

«component composition service»
Information_Dependency
«refine»
Vehicle_Information
Description
This service identifies Vehicle information from the rest of the system that is needed in determining a Route, e.g. current Vehicle operating conditions.
«activity»
identify_vehicle_information
Description
Identify the Vehicle information to be provided.
«activity»
assess_vehicle_information
Description
Assess the Vehicle information update to decide whether any further action needs to be taken.
«interface»
Vehicle_Information
Description
This interface is the current vehicle information relevant to the routing solution. This may include information about the current location and orientation and the performance of the Vehicle, e.g. current heading or top speed.

Figure 840: Vehicle_Information Service Policy

Vehicle_Information

This service identifies Vehicle information from the rest of the system that is needed in determining a Route, e.g. current Vehicle operating conditions.

Interface

Vehicle_Information

This interface is the current vehicle information relevant to the routing solution. This may include information about the current location and orientation and the performance of the Vehicle, e.g. current heading or top speed.

Attributes

orientation	The orientation of a Vehicle.
location	The current location of a Vehicle.
performance	The current operational performance information of a Vehicle.
temporal_information	Timing information pertaining to the reporting or applicability of the associated Vehicle information.
certainty	The level of certainty of the Vehicle information.
derivatives	The derivatives associated with the Vehicle location and orientation, e.g. velocity and acceleration.

Activities

assess_vehicle_information

Assess the Vehicle information update to decide whether any further action needs to be taken.

identify_vehicle_information

Identify the Vehicle information to be provided.

5.4.2.48.7.1.6 Constraint

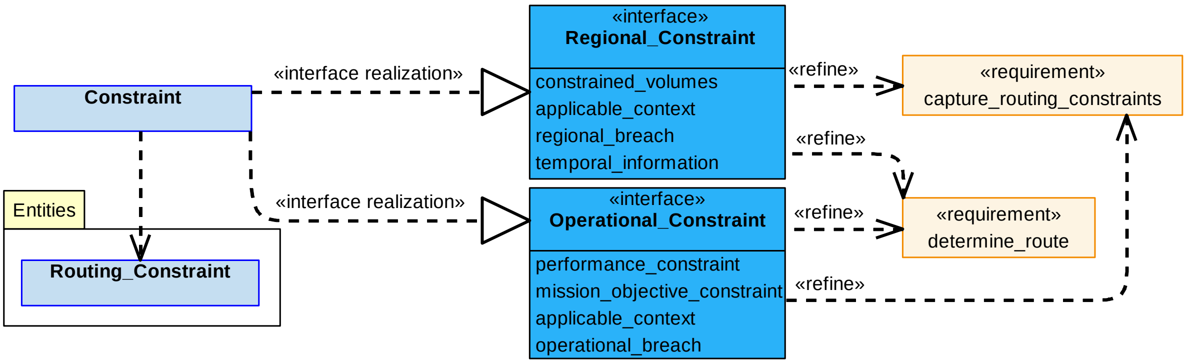
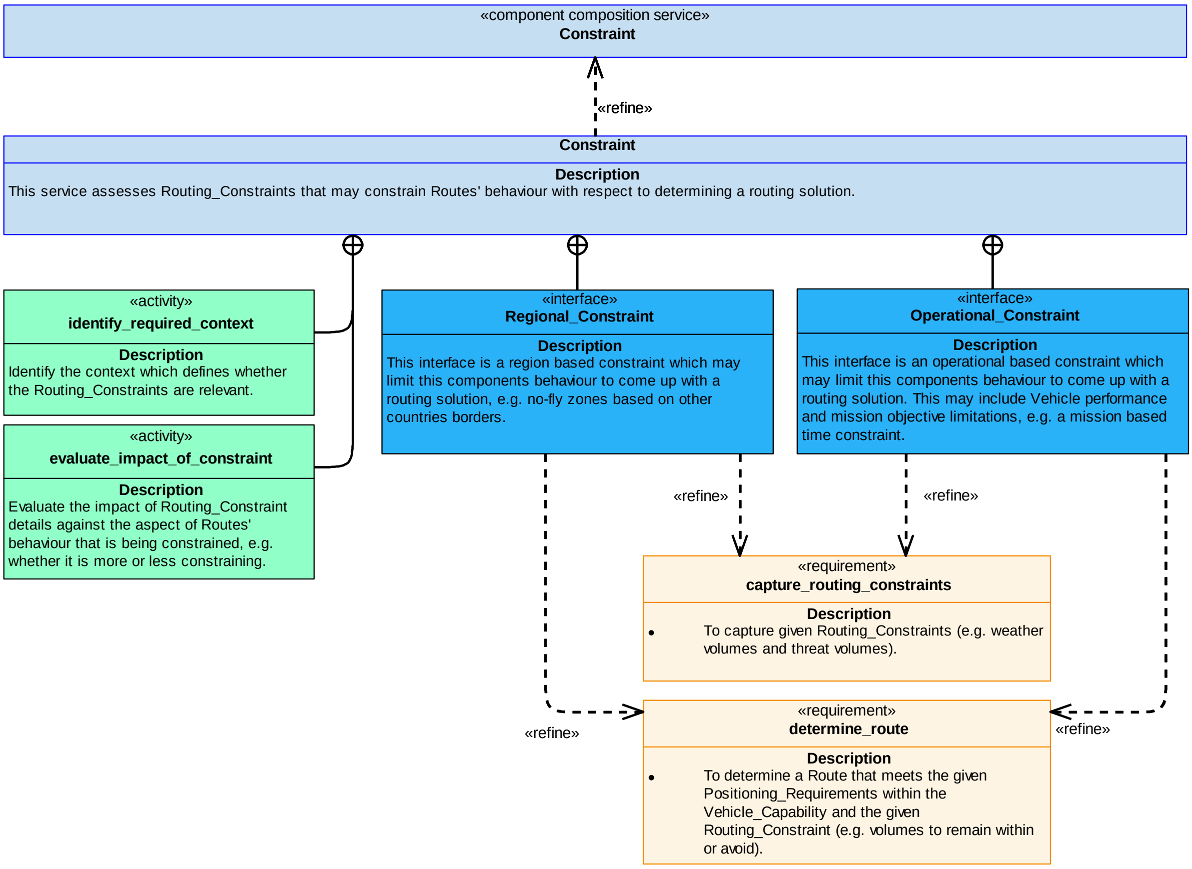


Figure 841: Constraint Service Definition



«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Routing_Constraints that may constrain Routes' behaviour with respect to determining a routing solution.
«activity»
identify_required_context
Description
Identify the context which defines whether the Routing_Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Routing_Constraint details against the aspect of Routes' behaviour that is being constrained, e.g. whether it is more or less constraining.
«interface»
Regional_Constraint
Description
This interface is a region based constraint which may limit this components behaviour to come up with a routing solution, e.g. no-fly zones based on other countries borders.
«interface»
Operational_Constraint
Description
This interface is an operational based constraint which may limit this components behaviour to come up with a routing solution. This may include Vehicle performance and mission objective limitations, e.g. a mission based time constraint.
«refine»
«refine»
«requirement»
capture_routing_constraints
Description
To capture given Routing_Constraints (e.g. weather volumes and threat volumes).
«refine»
«refine»
«requirement»
determine_route
Description
To determine a Route that meets the given Positioning_Requirements within the Vehicle_Capability and the given Routing_Constraint (e.g. volumes to remain within or avoid).

Figure 842: Constraint Service Policy

Constraint

This service assesses Routing_Constraints that may constrain Routes' behaviour with respect to determining a routing solution.

Interfaces

Regional_Constraint

This interface is a region based constraint which may limit this components behaviour to come up with a routing solution, e.g. no-fly zones based on other countries borders.

Attributes

constrained_volumes	The volumes which the Vehicle is not allowed to use. For example, areas of busy airspace with aviation limitations in place or no-fly zones.
applicable_context	The context in which the constraint is applicable.
regional_breach	A statement that the regional constraint has been breached.
temporal_information	Timing information pertaining to the reporting or applicability of the associated constraint information.

Operational_Constraint

This interface is an operational based constraint which may limit this components behaviour to come up with a routing solution. This may include Vehicle performance and mission objective limitations, e.g. a mission based time constraint.

Attributes

performance_constraint	A constraint on the performance parameters of a routing solution imposed by the Vehicle, e.g. altitude limits, or minimum and maximum speed.
mission_objective_constraint	A constraint on the routing solution imposed to achieve the mission objective.
applicable_context	The context in which the constraint is applicable.
operational_breach	A statement that the operational constraint has been breached.

Activities

identify_required_context

Identify the context which defines whether the Routing_Constraints are relevant.

evaluate_impact_of_constraint

Evaluate the impact of Routing_Constraint details against the aspect of Routes' behaviour that is being constrained, e.g. whether it is more or less constraining.

5.4.2.48.7.1.7 Capability

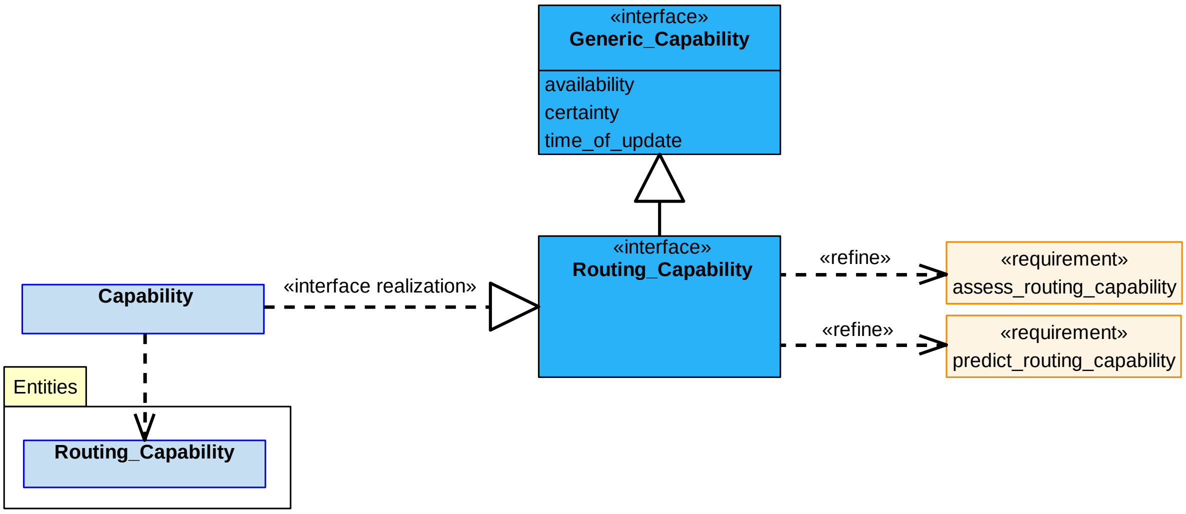


Figure 843: Capability Service Definition

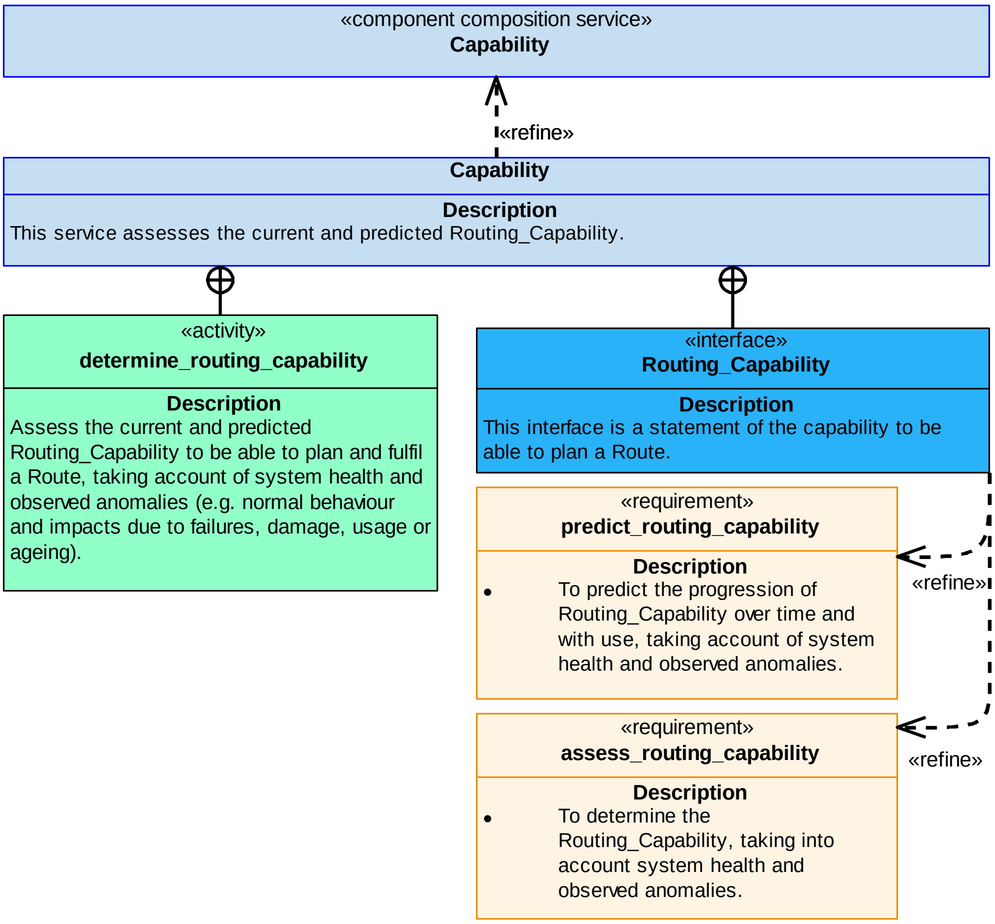

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Routing_Capability.
«activity»
determine_routing_capability
Description
Assess the current and predicted Routing_Capability to be able to plan and fulfil a Route, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Routing_Capability
Description
This interface is a statement of the capability to be able to plan a Route.
«requirement»
predict_routing_capability
Description
To predict the progression of Routing_Capability over time and with use, taking account of system health and observed anomalies.
«refine»
«requirement»
assess_routing_capability
Description
To determine the Routing_Capability, taking into account system health and observed anomalies.
«refine»

Figure 844: Capability Service Policy

Capability

This service assesses the current and predicted Routing_Capability.

Interface

Routing_Capability

This interface is a statement of the capability to be able to plan a Route.

Activity

determine_routing_capability

Assess the current and predicted Routing_Capability to be able to plan and fulfil a Route, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.48.7.1.8 Capability_Evidence

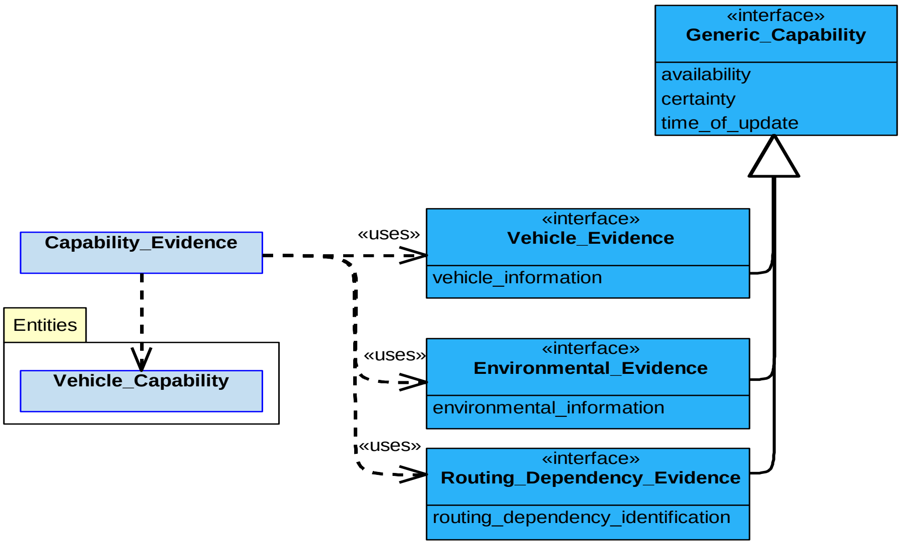


Figure 845: Capability_Evidence Service Definition

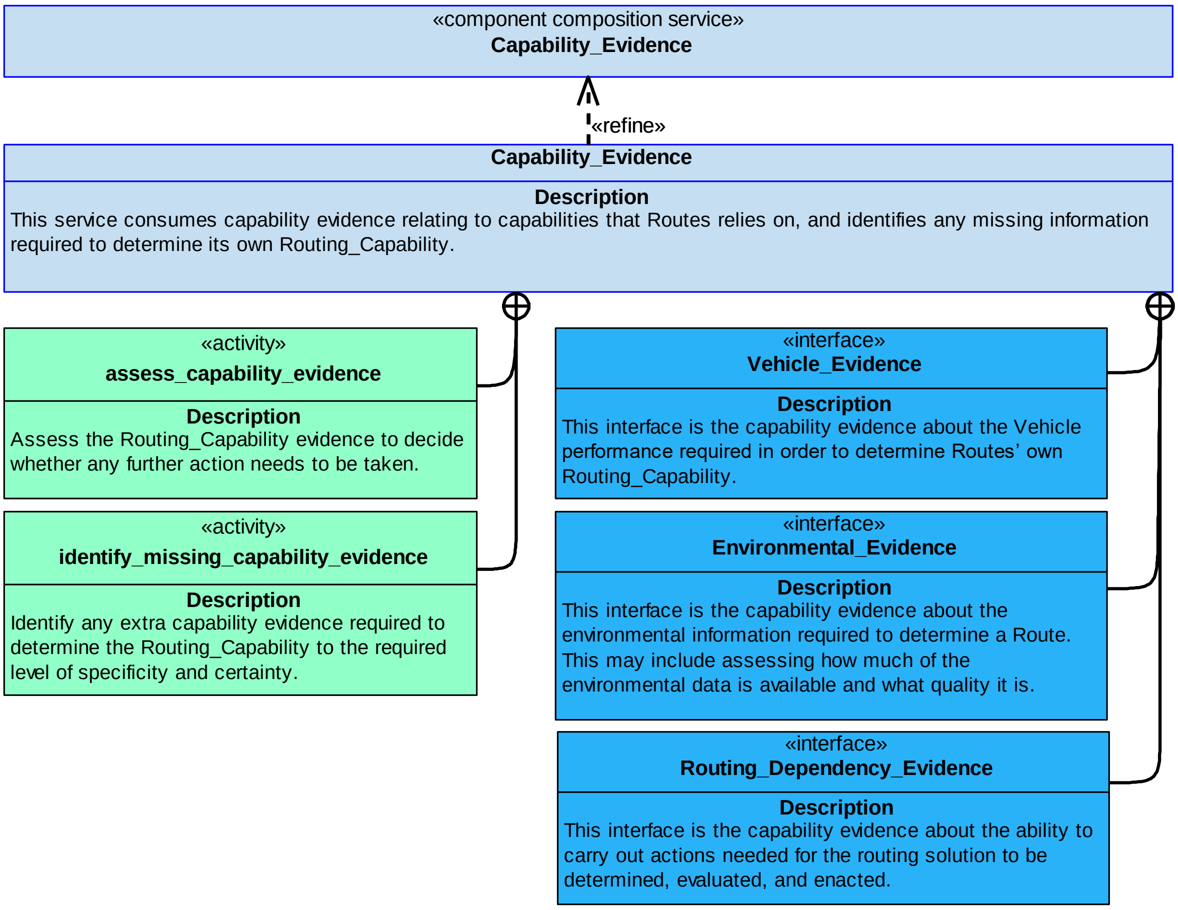

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capability evidence relating to capabilities that Routes relies on, and identifies any missing information required to determine its own Routing_Capability.
«activity»
assess_capability_evidence
Description
Assess the Routing_Capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Routing_Capability to the required level of specificity and certainty.
«interface»
Vehicle_Evidence
Description
This interface is the capability evidence about the Vehicle performance required in order to determine Routes' own Routing_Capability.
«interface»
Environmental_Evidence
Description
This interface is the capability evidence about the environmental information required to determine a Route. This may include assessing how much of the environmental data is available and what quality it is.
«interface»
Routing_Dependency_Evidence
Description
This interface is the capability evidence about the ability to carry out actions needed for the routing solution to be determined, evaluated, and enacted.

Figure 846: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability evidence relating to capabilities that Routes relies on, and identifies any missing information required to determine its own Routing_Capability.

Interfaces

Vehicle_Evidence

This interface is the capability evidence about the Vehicle performance required in order to determine Routes' own Routing_Capability.

Attribute

vehicle_information	The specific parameter to which the statement applies (e.g. location and orientation).

Environmental_Evidence

This interface is the capability evidence about the environmental information required to determine a Route. This may include assessing how much of the environmental data is available and what quality it is.

Attribute

environmental_information	The specific aspect of environmental information to which the statement applies (e.g. wind direction).

Routing_Dependency_Evidence

This interface is the capability evidence about the ability to carry out actions needed for the routing solution to be determined, evaluated, and enacted.

Attribute

routing_dependency_identification	An indication of the specific routing dependency for which capability evidence is being provided.

Activities

assess_capability_evidence

Assess the Routing_Capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Routing_Capability to the required level of specificity and certainty.

5.4.2.48.7.2 Service Dependencies

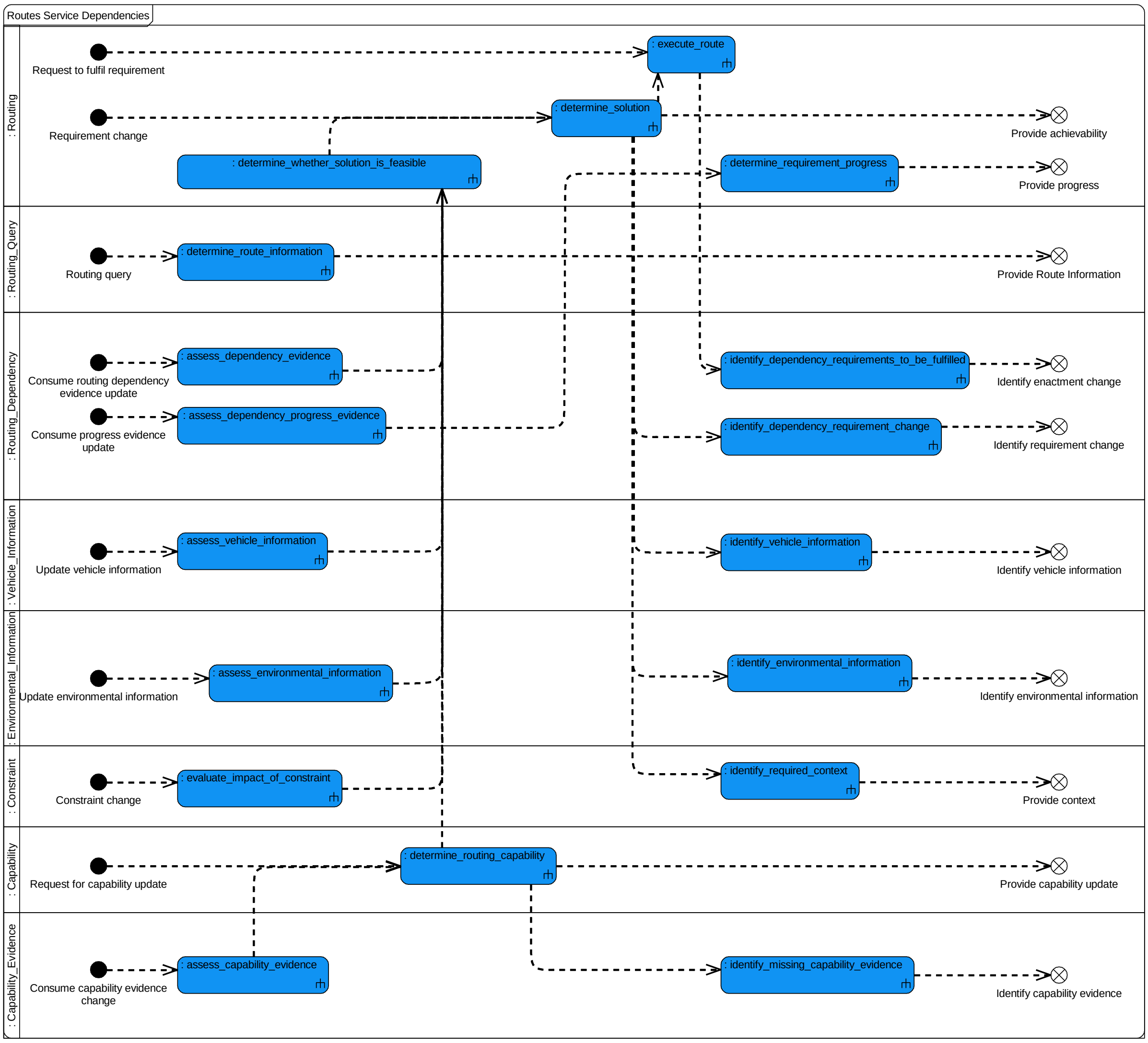

Routes Service Dependencies
: Routing
: Routing_Query
: Routing_Dependency
: Vehicle_Information
: Environmental_Information
: Constraint
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_route
Requirement change
: determine_solution
Provide achievability
: determine_whether_solution_is_feasible
: determine_requirement_progress
Provide progress
Routing query
: determine_route_information
Provide Route Information
Consume routing dependency evidence update
: assess_dependency_evidence
: identify_dependency_requirements_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_dependency_progress_evidence
: identify_dependency_requirement_change
Identify requirement change
Update vehicle information
: assess_vehicle_information
: identify_vehicle_information
Identify vehicle information
Update environmental information
: assess_environmental_information
: identify_environmental_information
Identify environmental information
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Provide context
Request for capability update
: determine_routing_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 847: Routes Service Dependencies

5.4.2.49 Semantic Translation

5.4.2.49.1 Role

The role of Semantic Translation is to translate between data semantics of systems.

5.4.2.49.2 Overview

Control Architecture

Semantic Translation is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In response to a requirement to enable information from Systems with different semantics to be shared, Semantic Translation applies Semantic_Rules to perform an Information_Translation of the information, determine any necessary Transitions, acquire the Information to which the Transitions are to be applied, performs them and provides the translated Information.

Examples of Use

Semantic Translation is used when the relationship between the semantics of Systems is not simple, and cannot be closed using a bridge. For example:

- The local System uses a different communication paradigm to the remote System (such as a remote procedure call mapping to a publish-subscribe message).
- When an interpretation of information (not just the type of information), based on the semantics of a remote system, determines the communication type of the remote system. For example, a local track is only to be passed externally to the TDL system if the local track is more accurate than the track the TDL system already has.
- The high level concepts between the two systems are different (such as where one System uses a commander role to determine sensor control handover in a five-way handover, whereas another System uses a STANAG 4586 based three-way handover with no commander role).

5.4.2.49.3 Service Summary

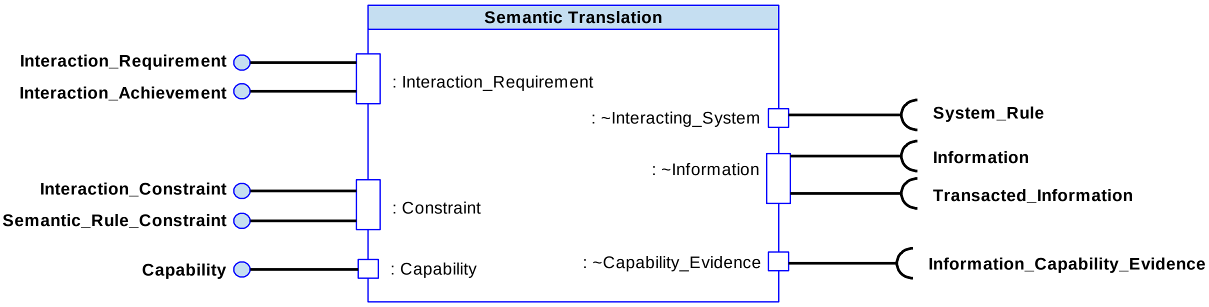


Figure 848: Semantic Translation Service Summary

5.4.2.49.4 Responsibilities

capture_interaction_requirements

- To capture provided requirements for Interactions between Systems.

capture_interaction_constraints

- To capture provided constraints on Interactions and application of Semantic_Rules.

determine_if_interaction_remains_achievable

- To determine if an Interaction requirement remains achievable given current Capability and Constraints.

determine_transaction

- To determine how to meet the given requirements for an Interaction between Systems.

deliver_system_interactions

- To apply the Semantic_Rule provided by an external System in order to translate between internal and external understandings.

determine_quality_of_interaction

- To determine the quality of the Interaction provided by Semantic Translation during execution, measured against given requirements.

assess_capability

- To assess the Capability to provide Semantic Translation's services taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

predict_capability_progression

- To predict the progression of Semantic Translation Capability over time and with use.

5.4.2.49.5 Subject Matter Semantics

The subject matter of Semantic Translation is Transactions resulting in the understanding of information in Systems with different semantics.

Exclusions

The subject matter of Semantic Translation does not include:

- Communications across security domains.
- Low level communication protocols used in data transfer.

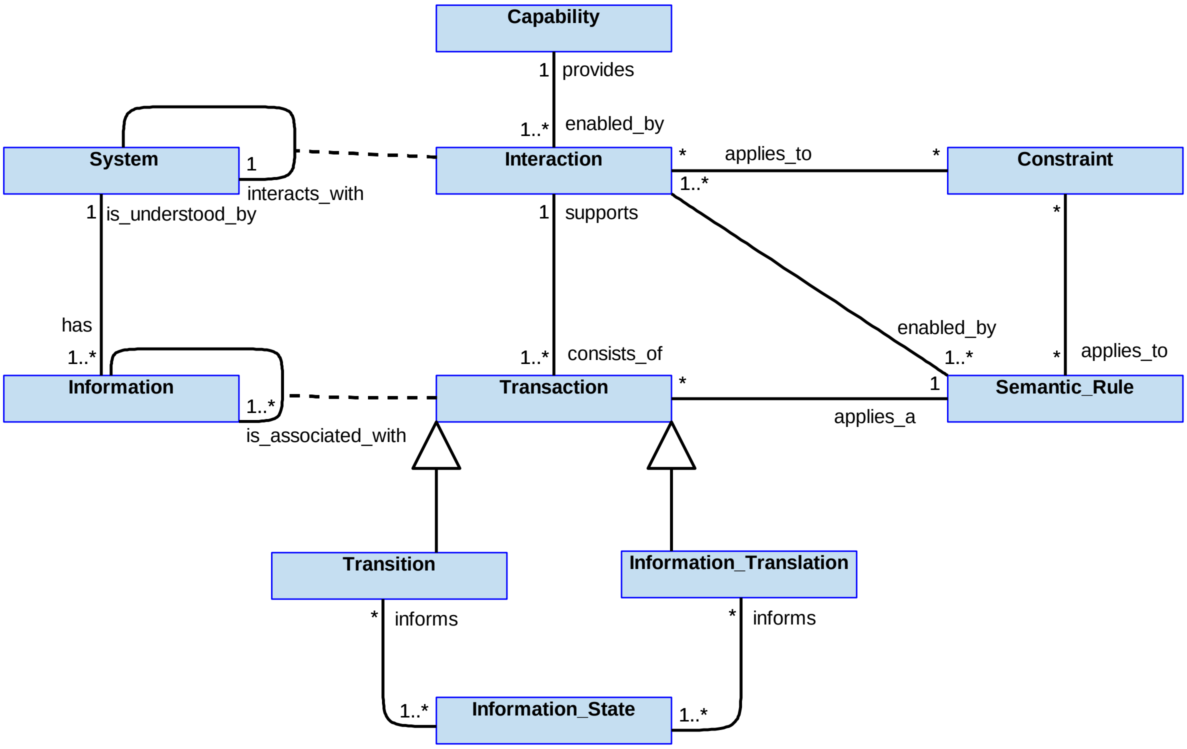


Figure 849: Semantic Translation Semantics

5.4.2.49.5.1 Entities

Capability

The capability of the component to translate between semantics of Systems.

Constraint

An externally placed limit on an allowable Interaction or the application of a Semantic_Rule.

Information

Data that is understood.

Information_State

The state of Information at a point in the lifecycle of an interaction. For example, the quality of a track that is received via a TDL, or the status of a control handover.

Information_Translation

The conversion of information between semantics.

Interaction

A synergetic relationship between Systems.

Semantic_Rule

The rules of the semantics of the Systems, and how the semantics interact.

System

A discrete entity with its own semantics of information, e.g. a PYRAMID air platform, a non-PYRAMID weapon system, or a communication system.

Transaction

An action that results in an understanding of received Information.

Transition

A change in information state in a System. This may not necessarily be reflected in the other System. For example, a potential target position is updated in one system, but the update is not shared as the quality of the information is less than that already held in the other system.

5.4.2.49.6 Design Rationale

5.4.2.49.6.1 Assumptions

- The encapsulation of data into communication protocols is the domain of Data Distribution.
- Semantic Translation is aware it is exchanging information with a System, has some understanding of which System it is currently communicating with, and has an understanding of the Information_State.

5.4.2.49.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Semantic Translation:

- Data Exchange - This component is highly involved in the exchange of information.
- Use of Communications - This PYRAMID concept specifies how communications are managed by components such as this one.
- Recording and Logging - The component will perform data logging.

Exploitation Considerations

- Semantic Translation has memory and awareness of the transactional state of a System, the status of information it has previously processed, etc.
- Semantic Translation would only be required in one of the Systems, between which an Interaction is required.

5.4.2.49.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Although this component can provide semantic translation between different systems, that may give rise to a lower indicative IDAL, this safety analysis is specifically related to the interface to a deployable asset.
- Failure of this component could cause loss of information, erroneous information, or corruption of information being provided to a deployed asset (e.g. a weapon or drone) resulting in inappropriate or uncontrolled operation of the asset, potentially leading to a catastrophic event.

Where instances of this component contribute to hazards that are less severe, then the Exploiting Programme may require a less onerous DAL.

5.4.2.49.6.4 Security Considerations

The indicative security classification is O but will vary according to the data representations.

This component is positioned at the interface between the Exploiting Platform and non-PYRAMID external entities (e.g. coalition forces) translating data according to the applicable data representation. It will be deployed within each security domain that will exchange data with external parties, taking the classification of that domain; it will not communicate across security domains, however it may be used to support data preparation for cross-domain communications. The incorrect functioning of this component may compromise confidentiality, integrity or availability of data exchanged and will need a high degree of protection.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to changes made in high-value data through the translation process or in the Semantic_Rules applied to achieve translation, etc.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.
- **Supporting Secure Remote Operation** through its involvement in the structuring of handover control messages and validation of authorisations, etc.

The component is expected to at least partially satisfy security enforcing functions by:

- **Restricting Access to Data** based on application of the rules of semantic translation between internal and external systems (e.g. only allowable data types for the external system will be translated).
- **Verifying Integrity of Data** for translated data.

5.4.2.49.7 Services

5.4.2.49.7.1 Service Definitions

5.4.2.49.7.1.1 Interaction_Requirement

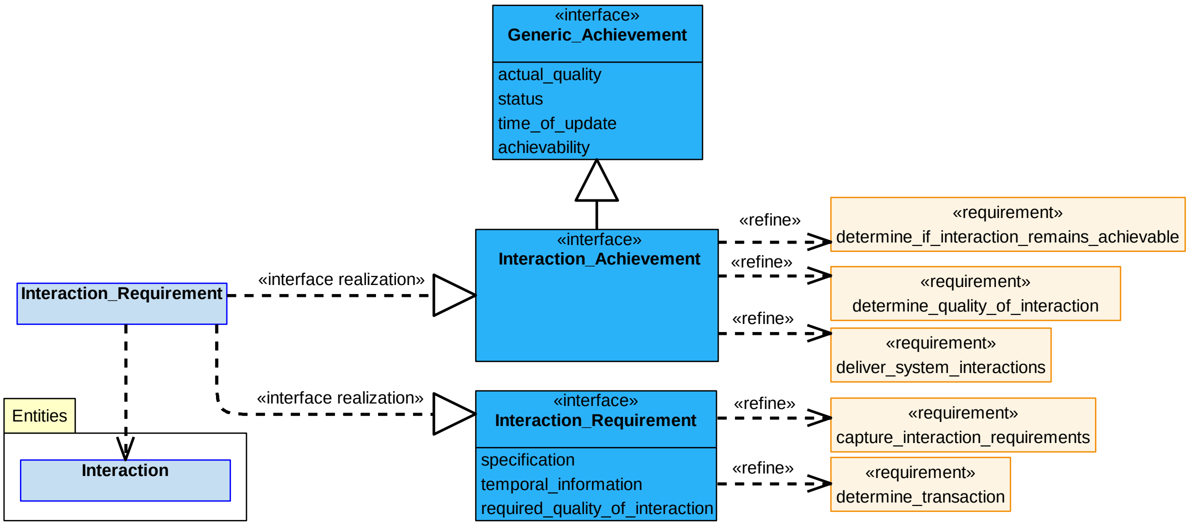


Figure 850: Interaction_Requirement Service Definition

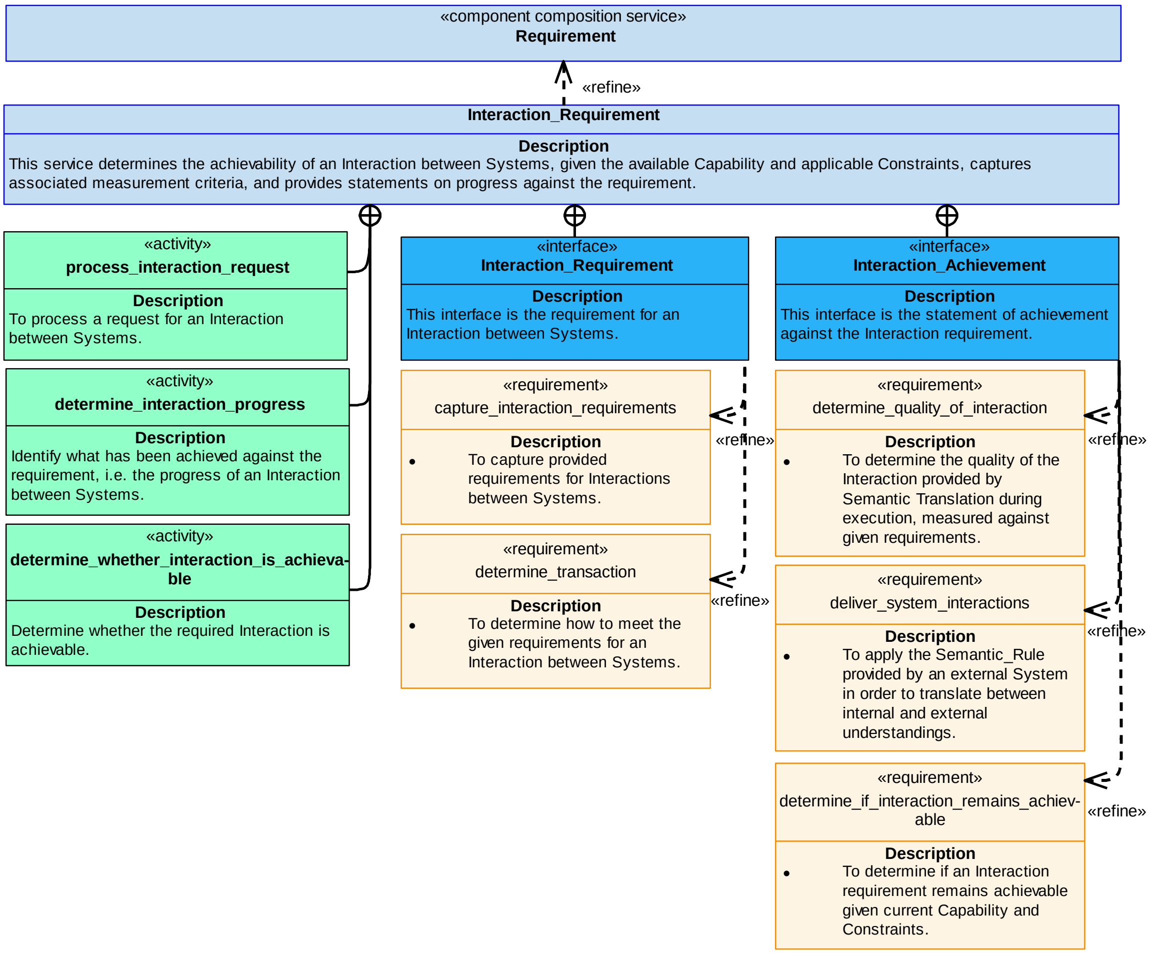

«component composition service»
Requirement
«refine»
Interaction_Requirement
Description
This service determines the achievability of an Interaction between Systems, given the available Capability and applicable Constraints, captures associated measurement criteria, and provides statements on progress against the requirement.
«activity»
process_interaction_request
Description
To process a request for an Interaction between Systems.
«activity»
determine_interaction_progress
Description
Identify what has been achieved against the requirement, i.e. the progress of an Interaction between Systems.
«activity»
determine_whether_interaction_is_achievable
Description
Determine whether the required Interaction is achievable.
«interface»
Interaction_Requirement
Description
This interface is the requirement for an Interaction between Systems.
«requirement»
capture_interaction_requirements
Description
To capture provided requirements for Interactions between Systems.
«refine»
«requirement»
determine_transaction
Description
To determine how to meet the given requirements for an Interaction between Systems.
«refine»
«interface»
Interaction_Achievement
Description
This interface is the statement of achievement against the Interaction requirement.
«requirement»
determine_quality_of_interaction
Description
To determine the quality of the Interaction provided by Semantic Translation during execution, measured against given requirements.
«refine»
«requirement»
deliver_system_interactions
Description
To apply the Semantic_Rule provided by an external System in order to translate between internal and external understandings.
«refine»
«requirement»
determine_if_interaction_remains_achievable
Description
To determine if an Interaction requirement remains achievable given current Capability and Constraints.
«refine»

Figure 851: Interaction_Requirement Service Policy

Interaction_Requirement

This service determines the achievability of an Interaction between Systems, given the available Capability and applicable Constraints, captures associated measurement criteria, and provides statements on progress against the requirement.

Interfaces

Interaction_Requirement

This interface is the requirement for an Interaction between Systems.

Attributes

specification	The definition of the semantic translation requirement. For example, enable an Interaction to occur with a specific external System.
temporal_information	Information covering timing, such as start and end times.
required_quality_of_interaction	A quality that the Interaction must meet.

Interaction_Achievement

This interface is the statement of achievement against the Interaction requirement.

Activities

process_interaction_request

To process a request for an Interaction between Systems.

determine_interaction_progress

Identify what has been achieved against the requirement, i.e. the progress of an Interaction between Systems.

determine_whether_interaction_is_achievable

Determine whether the required Interaction is achievable.

5.4.2.49.7.1.2 Interacting_System

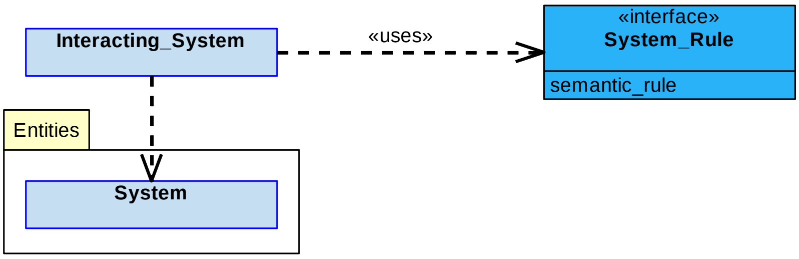


Figure 852: Interacting_System Service Definition

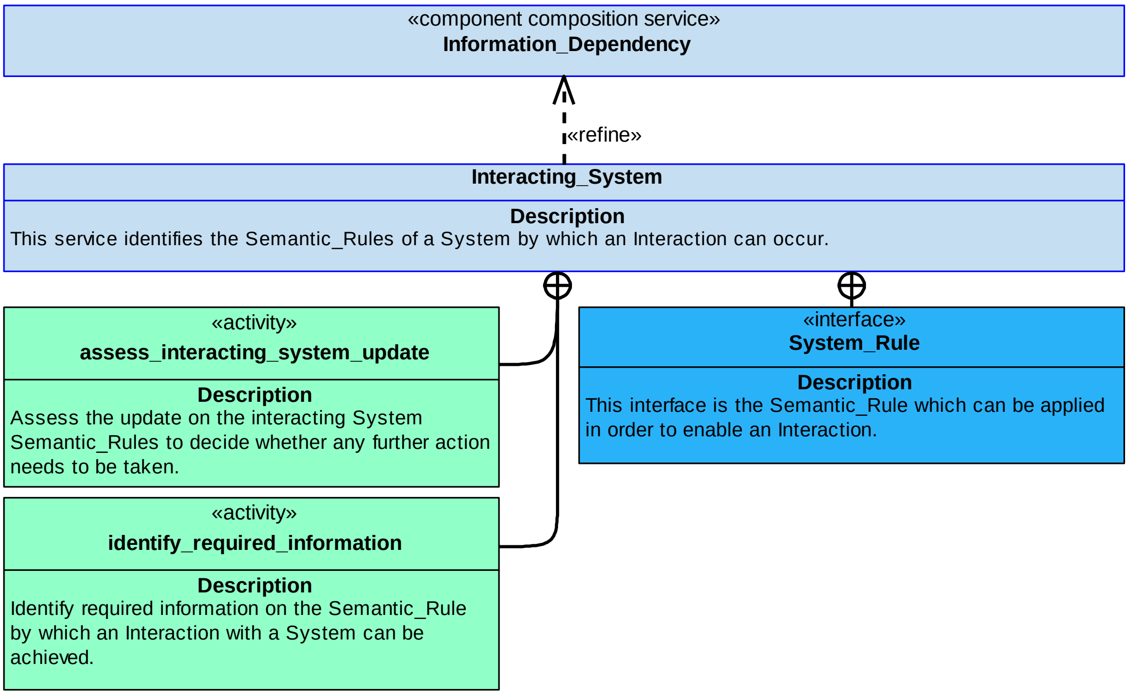

«component composition service»
Information_Dependency
«refine»
Interacting_System
Description
This service identifies the Semantic_Rules of a System by which an Interaction can occur.
«activity»
assess_interacting_system_update
Description
Assess the update on the interacting System Semantic_Rules to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify required information on the Semantic_Rule by which an Interaction with a System can be achieved.
«interface»
System_Rule
Description
This interface is the Semantic_Rule which can be applied in order to enable an Interaction.

Figure 853: Interacting_System Service Policy

Interacting_System

This service identifies the Semantic_Rules of a System by which an Interaction can occur.

Interface

System_Rule

This interface is the Semantic_Rule which can be applied in order to enable an Interaction.

Attribute

semantic_rule	The Semantic_Rules by which an Interaction with a System can be achieved.

Activities

assess_interacting_system_update

Assess the update on the interacting System Semantic_Rules to decide whether any further action needs to be taken.

identify_required_information

Identify required information on the Semantic_Rule by which an Interaction with a System can be achieved.

5.4.2.49.7.1.3 Information

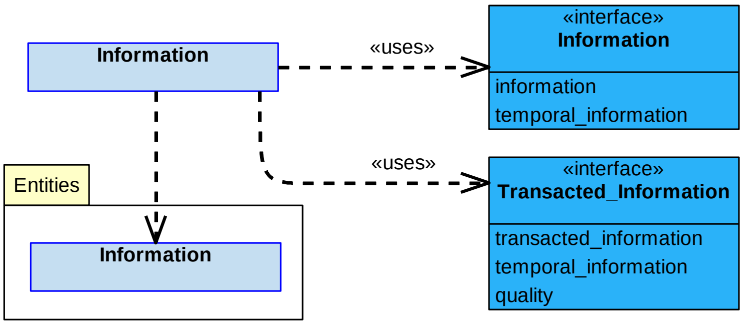


Figure 854: Information Service Definition

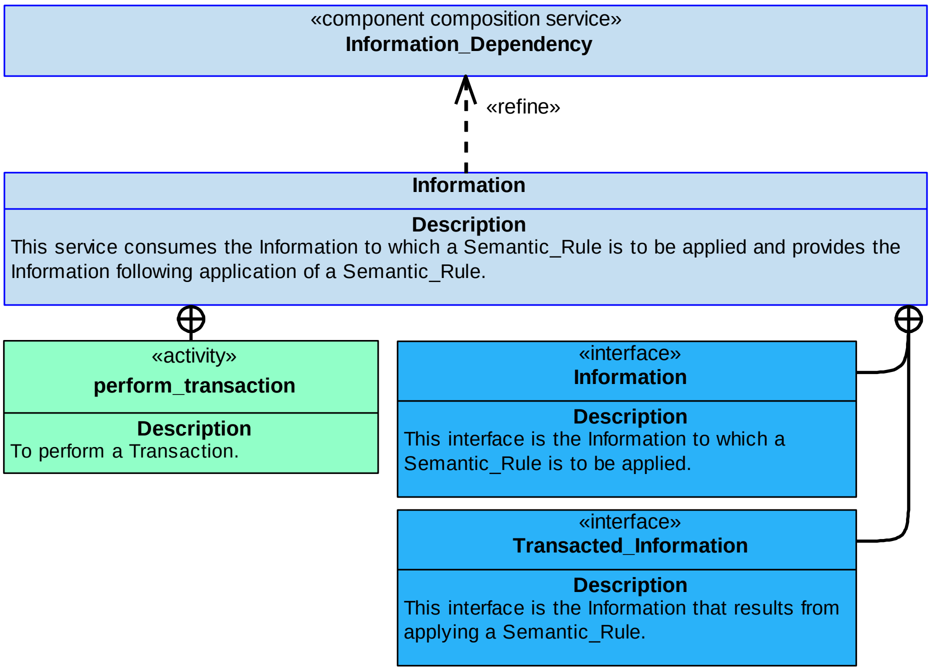

«component composition service»
Information_Dependency
«refine»
Information
Description
This service consumes the Information to which a Semantic_Rule is to be applied and provides the Information following application of a Semantic_Rule.
«activity»
perform_transaction
Description
To perform a Transaction.
«interface»
Information
Description
This interface is the Information to which a Semantic_Rule is to be applied.
«interface»
Transacted_Information
Description
This interface is the Information that results from applying a Semantic_Rule.

Figure 855: Information Service Policy

Information

This service consumes the Information to which a Semantic_Rule is to be applied and provides the Information following application of a Semantic_Rule.

Interfaces

Information

This interface is the Information to which a Semantic_Rule is to be applied.

Attributes

information	Information to which a Semantic_Rule is to be applied.
temporal_information	Information covering timing, such as when the Information was obtained.

Transacted_Information

This interface is the Information that results from applying a Semantic_Rule.

Attributes

transacted_information	Information that results from applying a Semantic_Rule.
temporal_information	Information covering timing, such as when the Information was translated between semantics.
quality	The quality of the translated Information.

Activity

perform_transaction

To perform a Transaction.

5.4.2.49.7.1.4 Constraint

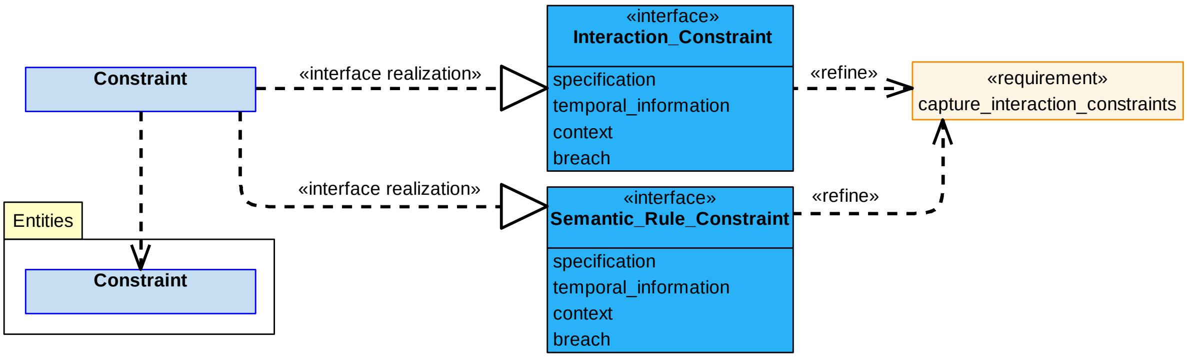


Figure 856: Constraint Service Definition

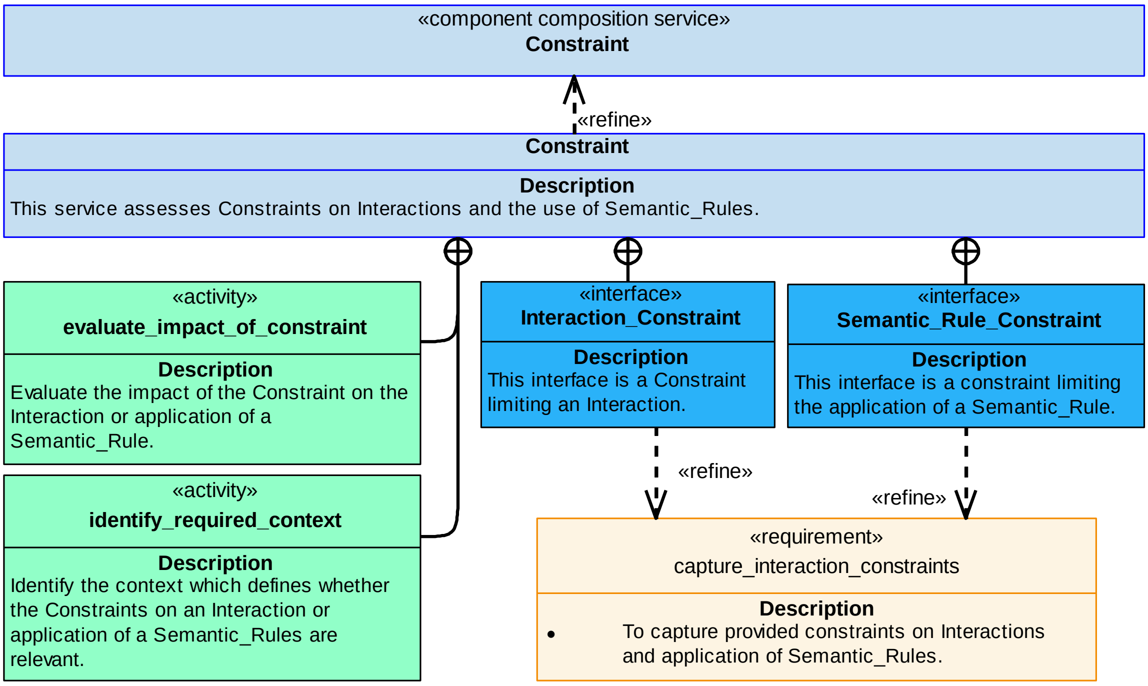

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints on Interactions and the use of Semantic_Rules.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of the Constraint on the Interaction or application of a Semantic_Rule.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints on an Interaction or application of a Semantic_Rules are relevant.
«interface»
Interaction_Constraint
Description
This interface is a Constraint limiting an Interaction.
«interface»
Semantic_Rule_Constraint
Description
This interface is a constraint limiting the application of a Semantic_Rule.
«refine»
«refine»
«requirement»
capture_interaction_constraints
Description
To capture provided constraints on Interactions and application of Semantic_Rules.

Figure 857: Constraint Service Policy

Constraint

This service assesses Constraints on Interactions and the use of Semantic_Rules.

Interfaces

Interaction_Constraint

This interface is a Constraint limiting an Interaction.

Attributes

specification	Specification of the Constraint restricting an Interaction.
temporal_information	Information covering timing of a Constraint, such as start time and duration, or end time.

context	The context in which the Interaction Constraint is applicable.
breach	A statement that the Constraint has been breached.

Semantic_Rule_Constraint

This interface is a constraint limiting the application of a Semantic_Rule.

Attributes

specification	Specification of the Constraint restricting application of a Semantic_Rule.
temporal_information	Information covering timing of a Constraint, such as start time and duration, or end time.
context	The context in which the Semantic_Rule Constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of the Constraint on the Interaction or application of a Semantic_Rule.

identify_required_context

Identify the context which defines whether the Constraints on an Interaction or application of a Semantic_Rules are relevant.

5.4.2.49.7.1.5 Capability

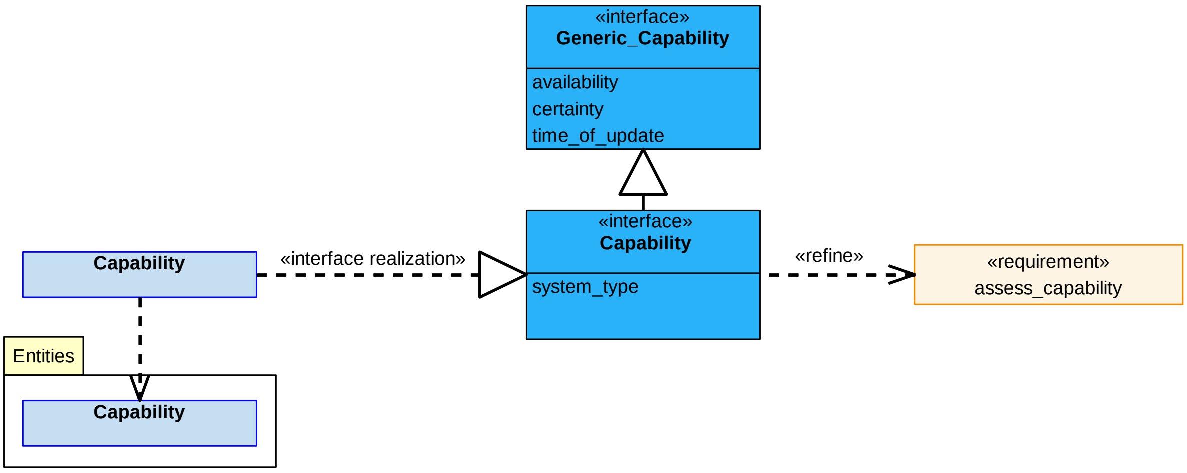


Figure 858: Capability Service Definition

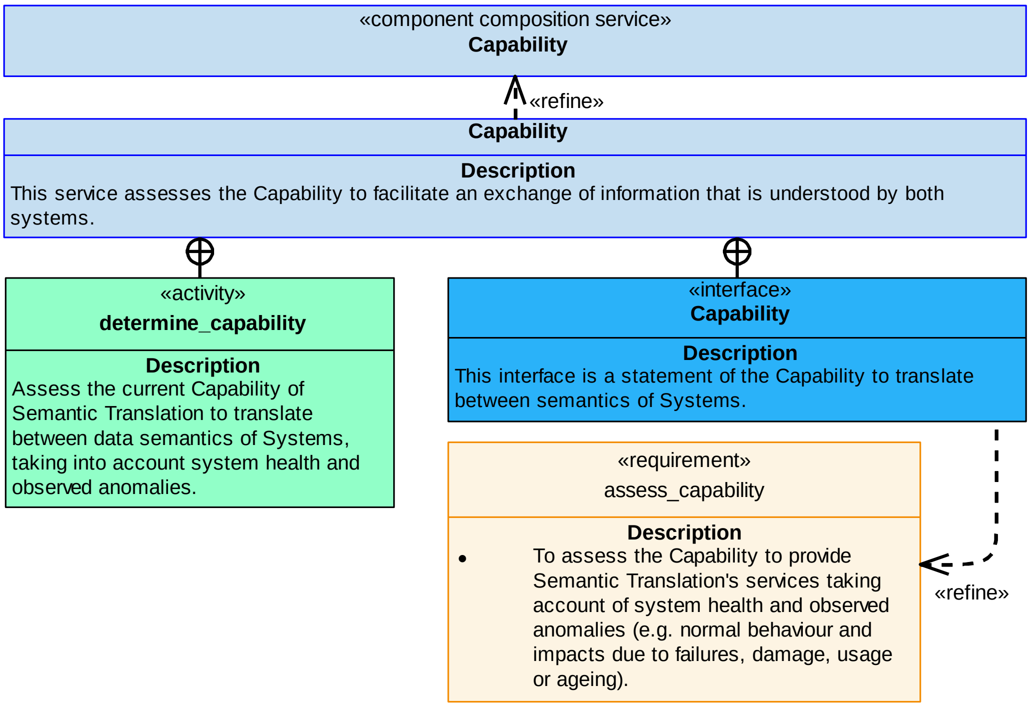

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the Capability to facilitate an exchange of information that is understood by both systems.
«activity»
determine_capability
Description
Assess the current Capability of Semantic Translation to translate between data semantics of Systems, taking into account system health and observed anomalies.
«interface»
Capability
Description
This interface is a statement of the Capability to translate between semantics of Systems.
«requirement»
assess_capability
Description
To assess the Capability to provide Semantic Translation's services taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 859: Capability Service Policy

Capability

This service assesses the Capability to facilitate an exchange of information that is understood by both systems.

Interface

Capability

This interface is a statement of the Capability to translate between semantics of Systems.

Attribute

system_type	The type of System with which an Interaction can occur.

Activity

determine_capability

Assess the current Capability of Semantic Translation to translate between data semantics of Systems, taking into account system health and observed anomalies.

5.4.2.49.7.1.6 Capability_Evidence

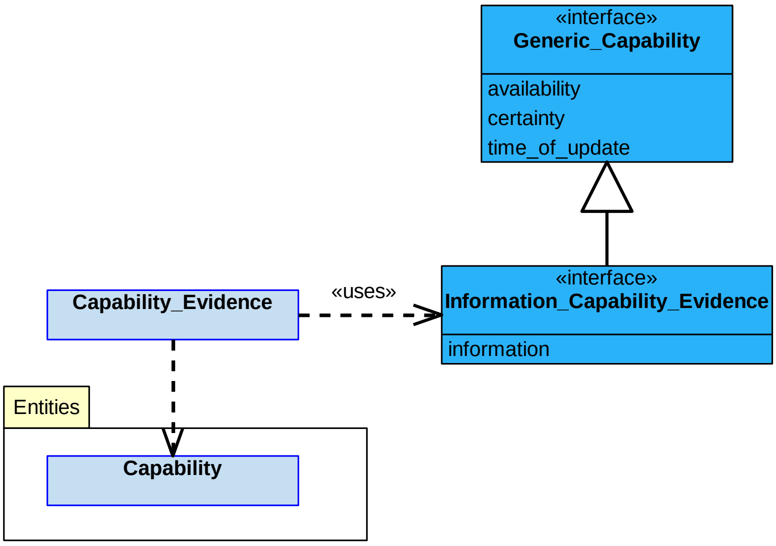


Figure 860: Capability_Evidence Service Definition

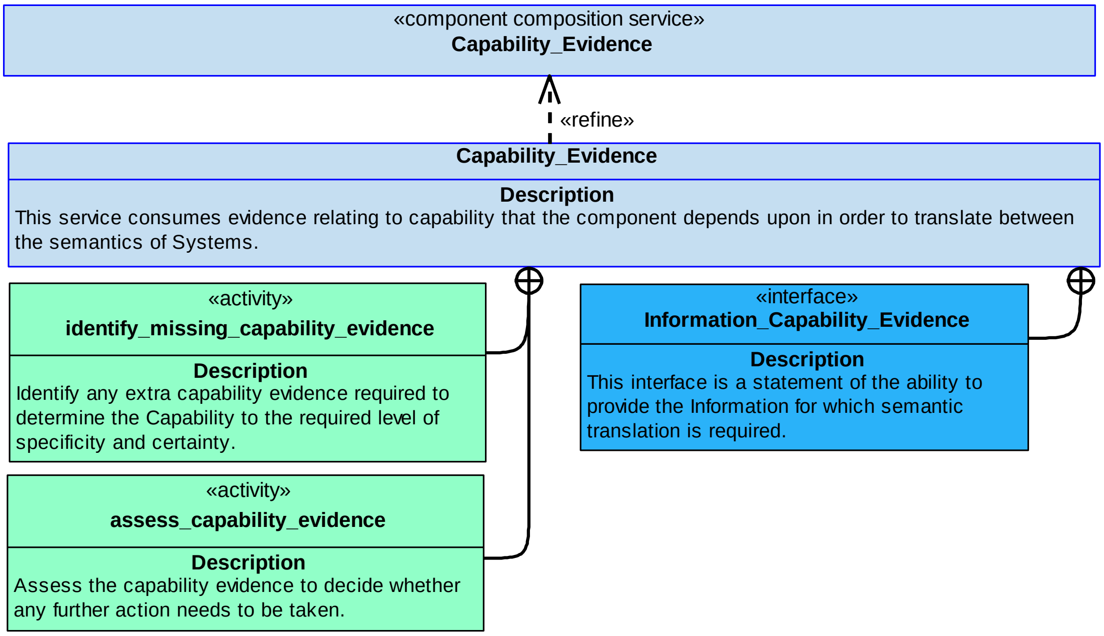

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes evidence relating to capability that the component depends upon in order to translate between the semantics of Systems.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«interface»
Information_Capability_Evidence
Description
This interface is a statement of the ability to provide the Information for which semantic translation is required.

Figure 861: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence relating to capability that the component depends upon in order to translate between the semantics of Systems.

Interface

Information_Capability_Evidence

This interface is a statement of the ability to provide the Information for which semantic translation is required.

Attribute

information	Identification of the type and source of Information that can be provided to the component for semantic translation.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.49.7.2 Service Dependencies

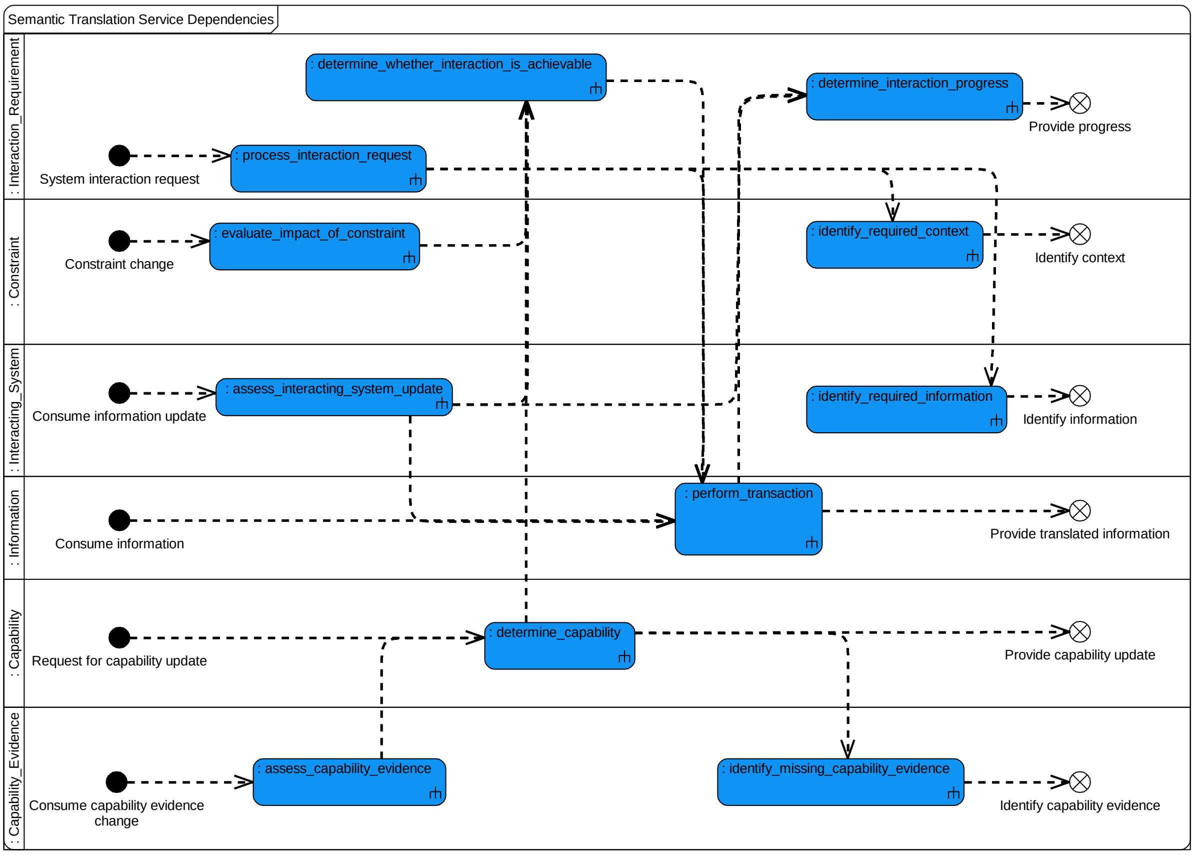

Semantic Translation Service Dependencies
: Interaction_Requirement
: Constraint
: Interacting_System
: Information
: Capability
: Capability_Evidence
: determine_whether_interaction_is_achievable
: determine_interaction_progress
Provide progress
System interaction request
: process_interaction_request
: evaluate_impact_of_constraint
Constraint change
: identify_required_context
Identify context
Consume information update
: assess_interacting_system_update
: identify_required_information
Identify information
: perform_transaction
Consume information
Provide translated information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 862: Semantic Translation Service Dependencies

5.4.2.50 Sensing

5.4.2.50.1 Role

The role of Sensing is to perform sensing actions by using resources.

5.4.2.50.2 Overview

Control Architecture

Sensing is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a Sensing_Requirement is received, Sensing will determine a Sensing_Solution consisting of Sensing_Steps, each with associated Pre-conditions (e.g. a measurement activity with a pre-condition concerning distance to target). The Sensing_Solution is then enacted, which will involve coordinated control of Sensing_Resources. The Sensing_Solution will deliver data acquired from Sensing_Resources.

Examples of Use

Sensing is required where generation and coordination of a sequence of Sensing_Steps may be necessary. For example:

- To provide data that can be used to detect, recognise, and identify objects.
- To provide data that can be used to target objects with ordnance.

5.4.2.50.3 Service Summary

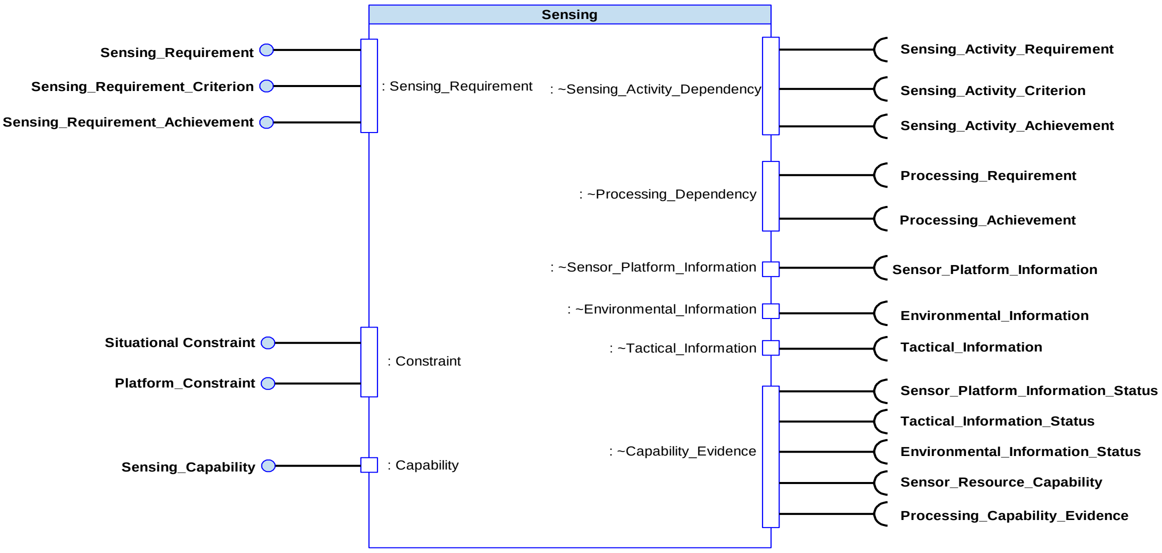

Sensing
Sensing_Requirement
Sensing_Requirement_Criterion
Sensing_Requirement_Achievement
: Sensing_Requirement
: ~Sensing_Activity_Dependency
Sensing_Activity_Requirement
Sensing_Activity_Criterion
Sensing_Activity_Achievement
: ~Processing_Dependency
Processing_Requirement
Processing_Achievement
: ~Sensor_Platform_Information
Sensor_Platform_Information
: ~Environmental_Information
Environmental_Information
: ~Tactical_Information
Tactical_Information
Situational Constraint
Platform_Constraint
: Constraint
: ~Capability_Evidence
Sensor_Platform_Information_Status
Tactical_Information_Status
Environmental_Information_Status
Sensor_Resource_Capability
Processing_Capability_Evidence
Sensing_Capability
: Capability

Figure 863: Sensing Service Summary

5.4.2.50.4 Responsibilities

capture_sensing_action_requirements

- To capture given Sensing_Requirements (e.g. target criteria, location and time).

capture_sensing_action_measurement_criteria

- To capture given Measurement_Criterion/criteria for sensing output (e.g. recon image quality).

capture_sensing_action_constraints

- To capture given sensing Constraints (e.g. spatial restrictions or EMCON restrictions on active sensing).

identify_whether_requirement_remains_achievable

- To identify if a Sensing_Solution in progress remains achievable given current resources.

determine_sensing_solution

- To determine a Sensing_Solution (i.e. a sequence of sensing activities) that meets the given Sensing_Requirements using available Sensing_Resources, within the provided Constraints and prevailing external factors (e.g. weather).

determine_predicted_quality_of_sensing_solution

- To determine the predicted quality of a Sensing_Solution against given Measurement_Criterion/Criteria.

identify_sensing_solution_pre-conditions

- To identify Pre-conditions to support a Sensing_Solution.

coordinate_sensing_solution

- To execute a Sensing_Solution by commanding Sensing_Resources.

identify_progress_of_sensing_solution

- To identify the progress of a Sensing_Solution against the Sensing_Requirement.

capture_actual_quality_of_deliverables

- To capture the actual quality of the deliverables provided by a Sensing_Solution, measured against given Sensing_Requirements and Measurement_Criterion/Criteria.

assess_sensing_action_capability

- To determine the available Sensing_Capability provided by installed sensing resources, taking into account anomalies and sensor health.

identify_required_capability_information

- To identify missing resource information that is required for assessing Sensing_Capability.

predict_sensing_action_capability_progression

- To predict the progression of Sensing_Capability over time and with use.

5.4.2.50.5 Subject Matter Semantics

The subject matter of Sensing is the Sensing_Resources that can be used to measure properties of the environment.

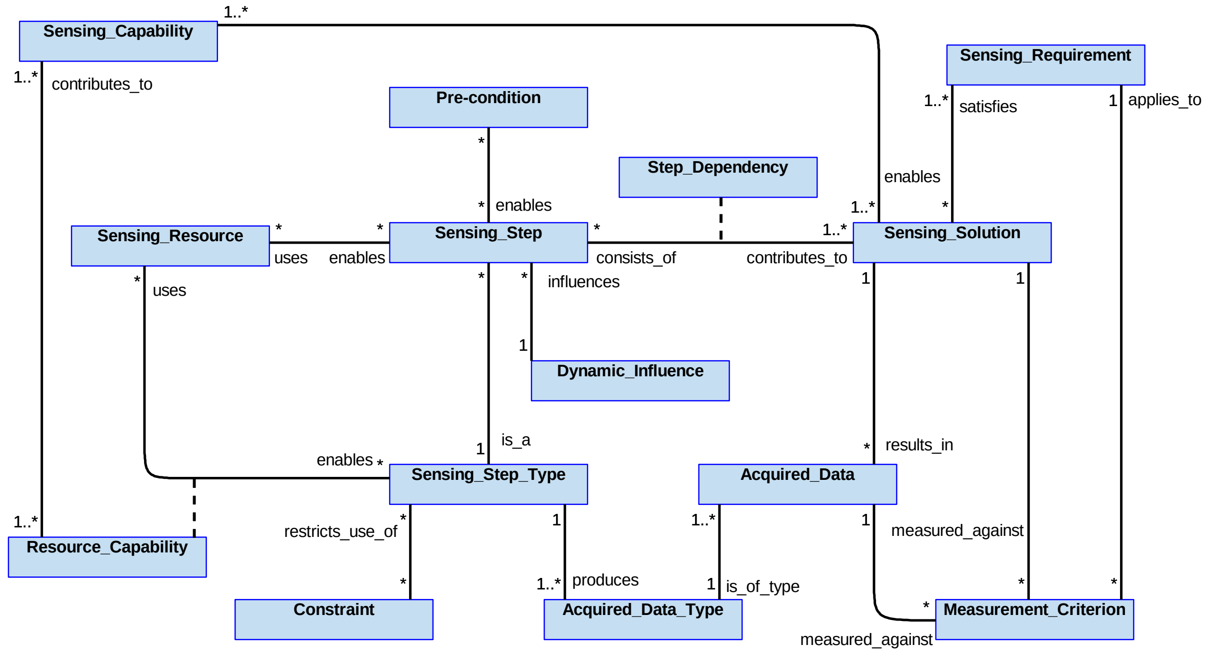


Figure 864: Sensing Semantics

5.4.2.50.5.1 Entities

Acquired_Data

Data acquired through measuring properties of an environment. This component does not handle this data directly but coordinates its acquisition and will handle metadata describing it.

Acquired_Data_Type

The type of the data produced by a Sensing_Step (e.g. bitmap image, SAR image, radar bearing and range information or ES parametric data).

Resource_Capability

The range of Sensing_Step_Types that can be performed with a specific Sensing_Resource.

Constraint

A restriction on when or how a Sensing_Step_Type can be used (e.g. spatial restrictions or EMCON restrictions on active sensing).

Measurement_Criterion

A criterion that the quality of a Sensing_Solution and its Acquired_Data will be measured against (e.g. speed or efficiency of the solution can be measured, and timeliness or completeness of the data can be measured).

Pre-condition

A condition that must be true before a Sensing_Step can take place (e.g. the availability of Sensing_Resources or processing resources).

Sensing_Requirement

A requirement placed on Sensing for the acquisition of data from an environment that will fulfil an information demand.

Sensing_Resource

A resource that can be instructed to execute a Sensing_Step (e.g. sensor equipment).

Sensing_Solution

A combination of Sensing_Steps which will fulfil a Sensing_Requirement.

Sensing_Step

An operational demand that Sensing places on a Sensing_Resource.

Sensing_Step_Type

The type of a Sensing_Step (e.g. wide area search, cued search or priority tracking).

Step_Dependency

A dependency on Sensing_Steps that affects their type, quality, timing or order of execution, and the degree to which steps can be executed in parallel or in series (e.g. one step provides input to another step, a time gap is needed for processing data between steps, or a step providing input must reach a minimum quality level or be of a specific type of data).

Sensing_Capability

The range of Sensing_Requirement types that can be fulfilled by Sensing with current resources and constraints.

Dynamic_Influence

Platform, environmental or tactical information that influences the planning or enactment of Sensing_Solutions. For example atmospheric conditions affecting EW wave propagation, weather features occluding targets of sensing activity, or vehicle speed and position that determine sensor field of view, rate of change and direction of movement.

5.4.2.50.6 Design Rationale

5.4.2.50.6.1 Assumptions

- This component does not handle any crypto that may be required by sensors.
- The component will contain knowledge of tactical sensing capabilities in order to be able to command the appropriate sensing actions.
- In some Exploiting Platforms, it may be possible for sensor control to be granted to an external party.

- The types and configurations of installed resources used to perform tactical sensing on a given host Exploiting Platform, or within a group of Exploiting Platforms, may frequently vary, e.g. different mission scenarios will require different build sets that will use different combinations of the appropriate sensor and other resource components. Therefore the capabilities, commands and data associated with tactical sensing resources may also vary frequently.
- The total set of possible tactical sensing types that could be used on host Exploiting Platforms, or within a group of Exploiting Platforms, will vary infrequently (i.e. new methods of exploiting the electromagnetic spectrum or other detectable phenomena will not often be invented).

5.4.2.50.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Sensing:

- Constraint Management - It is important for Sensing to identify solution-based constraints on other components so that Sensing_Step Pre-conditions are met (e.g. the Exploiting Platform must be a certain distance from a target).
- Control Architecture - As an action component, Sensing interacts with other components as described in the Control Architecture PYRAMID concept.
- Dependency Management - Dynamic dependencies are especially important, e.g. direct interaction with Sensor Data Interpretation, so that the component can base its choice of sensing techniques on feedback about the sensor data interpretation (see section Carrying out a Sensing Task).
- Data Driving - Different combinations of installed sensing capability will be used in different Exploiting Programmes, and additional new capabilities or configurations of resource types may be installed on an Exploiting Platform. This will result in a variety of sensing solutions and how they are coordinated (see also Multi-Vehicle Coordination, below). This variation is expected to be accommodated by configuring the Sensing component through data driving. This facilitates reusability, exploitability, and maintainability.
- Recording and Logging - Logging operations and record retention will be performed in accordance with this PYRAMID concept.
- Multi-Vehicle Coordination - In multi-vehicle arrangements Sensing_Capability is expected to be determined and provided on a UAS wide basis.

Extensions

- The means of calculating which combination of sensing capabilities will achieve a tactical solution could itself be specialised through an extension to Sensing. This contributes towards the component and its extensions being configurable, resilient against obsolescence and exploitable.

Exploitation Considerations

- The effective deployment of tactical Sensing_Solutions requires their planning and execution to take into account dynamic external factors (e.g. weather or target observability).
- Sensing is responsible for coordinating and controlling use of tactical sensor resources, but not for processing their outputs (i.e. it commands sensors to acquire data but it does not interpret the Acquired_Data). However, Sensing can take into account current and predicted quality and availability of the Acquired_Data, together with feedback from other components such as Sensor Data Interpretation, to adapt Sensing_Solutions during their planning and execution to optimise effectiveness.
- While data driving allows for variation in sensing solutions and their coordination, the stability of the interfaces that both require and implement a particular sensing solution should remain as stable as possible in order to facilitate usability, adaptability, supportability and exploitability.

5.4.2.50.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

There are a number of hazards this component could cause, including:

- Where harm may be caused by transmissions by active sensors, this component is a contributor to inadvertent firing. However, it is expected, unless transmission would cause no more than minor injury to third parties, that the Interlocks component would interact directly with the active sensor resources to prevent any harmful transmission. Therefore, for this failure mode, DAL C would be appropriate.
- Whilst this component coordinates the use of sensing resources, the products of the sensors are expected to be used directly by other components - i.e. not via the Sensing component. Therefore, it is not expected that this component would result in erroneous geolocation of targets. However, as sensors are used to support the designation of targets, the failure of this component could result in erroneous designation of a target. This would result in weapons impacting locations not intended by the crew and so result in unintended harm to third parties. This drives a DAL B indicative IDAL.

5.4.2.50.6.4 Security Considerations

The indicative security classification is SNEO but will vary according to the deployment.

This component plans and executes complex sensing activities through the use of tactical sensors based on knowledge of their capabilities, the details of which are generally expected to be SNEO, although this may vary depending on the sensor's capabilities. There may need to be multiple instances or variants of this component in different security domains. The integrity and availability of this component can have an impact on the combat effectiveness of the Exploiting Platform (e.g. unauthorised emissions may increase the observability of the Exploiting Platform, and the loss of sensing will prevent the accumulation of the required information). The integrity and availability will need to be protected according to the equipment being directed by the component.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to sensing performed during the mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.
- **Supporting Secure Remote Operation** through planning sensing for accumulation of sensed information for autonomous decision-making and operation.
- Performing **System Status and Monitoring** of demanded versus provided sensing operation, unexpected sensing or loss of sensing might indicate a possible cyber attack.

The component is considered unlikely to directly implement security enforcing functions, but will be subject to EMCON rules.

5.4.2.50.7 Services

5.4.2.50.7.1 Service Definitions

5.4.2.50.7.1.1 Sensing_Requirement

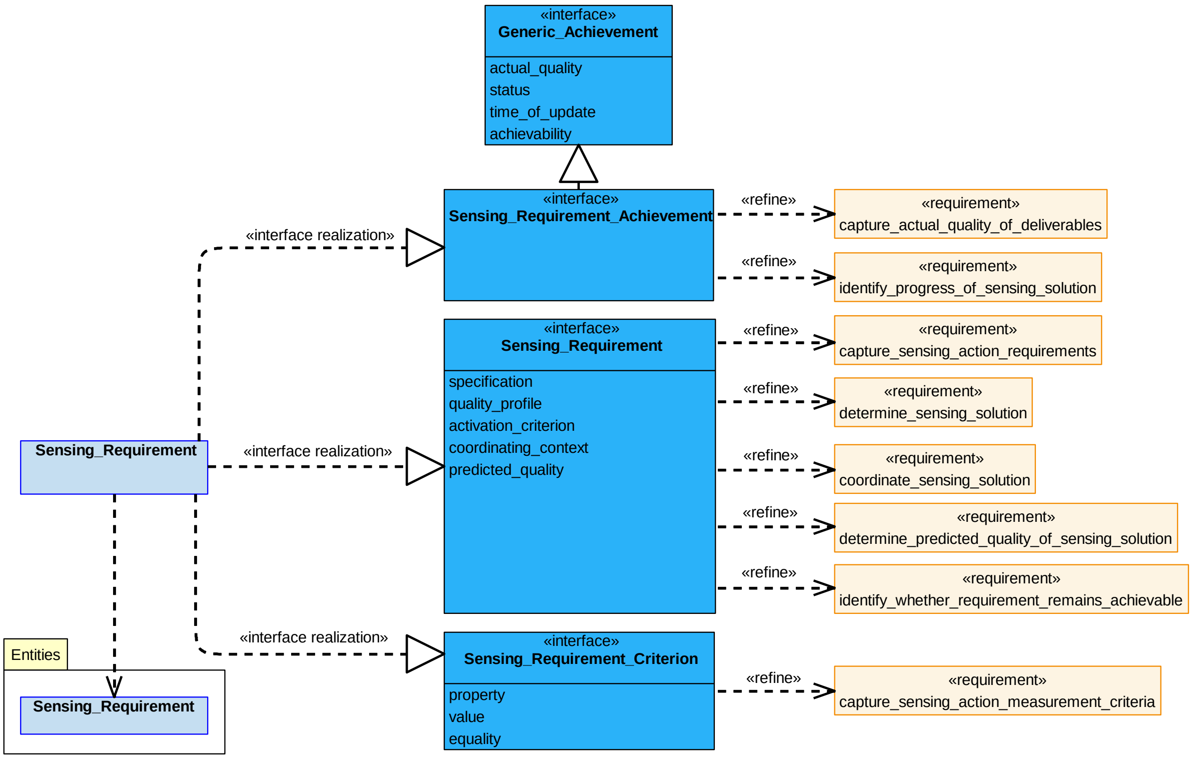


Figure 865: Sensing_Requirement Service Definition

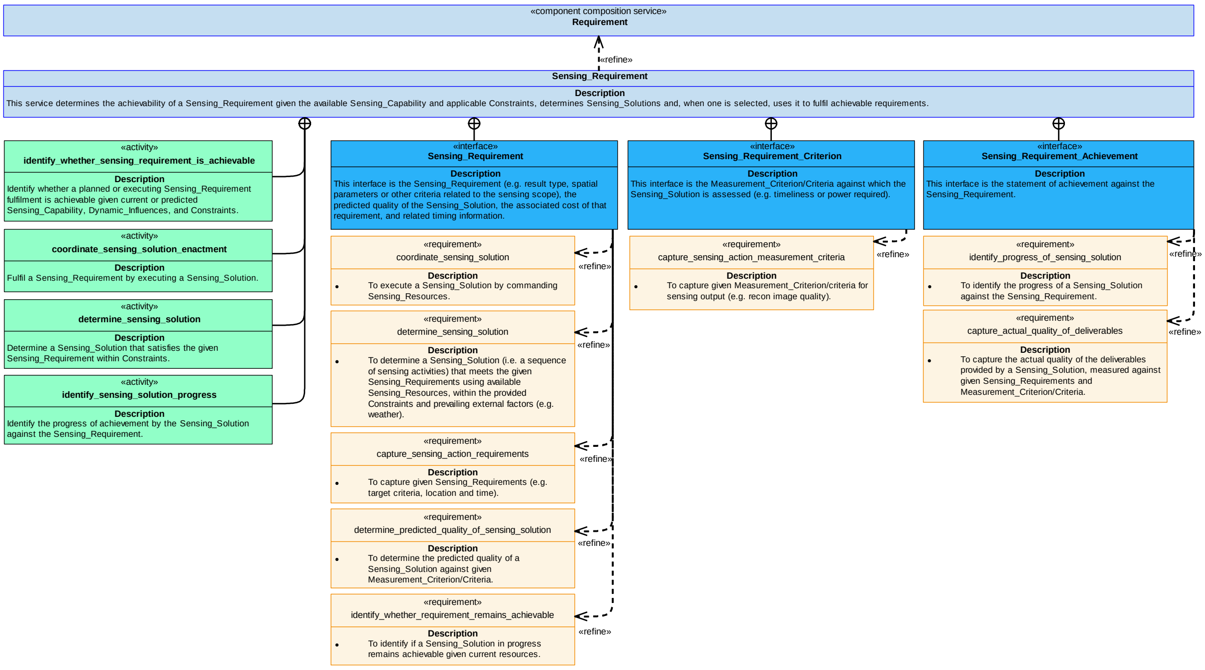

«component composition service»
Requirement
«refine»
Sensing_Requirement
Description
This service determines the achievability of a Sensing_Requirement given the available Sensing_Capability and applicable Constraints, determines Sensing_Solutions and, when one is selected, uses it to fulfil achievable requirements.
«activity»
identify_whether_sensing_requirement_is_achievable
Description
Identify whether a planned or executing Sensing_Requirement fulfilment is achievable given current or predicted Sensing_Capability, Dynamic_Influences, and Constraints.
«activity»
coordinate_sensing_solution_enactment
Description
Fulfil a Sensing_Requirement by executing a Sensing_Solution.
«activity»
determine_sensing_solution
Description
Determine a Sensing_Solution that satisfies the given Sensing_Requirement within Constraints.
«activity»
identify_sensing_solution_progress
Description
Identify the progress of achievement by the Sensing_Solution against the Sensing_Requirement.
«interface»
Sensing_Requirement
Description
This interface is the Sensing_Requirement (e.g. result type, spatial parameters or other criteria related to the sensing scope), the predicted quality of the Sensing_Solution, the associated cost of that requirement, and related timing information.
«requirement»
coordinate_sensing_solution
Description
To execute a Sensing_Solution by commanding Sensing_Resources.
«refine»
«requirement»
determine_sensing_solution
Description
To determine a Sensing_Solution (i.e. a sequence of sensing activities) that meets the given Sensing_Requirements using available Sensing_Resources, within the provided Constraints and prevailing external factors (e.g. weather).
«refine»
«requirement»
capture_sensing_action_requirements
Description
To capture given Sensing_Requirements (e.g. target criteria, location and time).
«refine»
«requirement»
determine_predicted_quality_of_sensing_solution
Description
To determine the predicted quality of a Sensing_Solution against given Measurement_Criterion/Criteria.
«refine»
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify if a Sensing_Solution in progress remains achievable given current resources.
«refine»
«interface»
Sensing_Requirement_Criterion
Description
This interface is the Measurement_Criterion/Criteria against which the Sensing_Solution is assessed (e.g. timeliness or power required).
«requirement»
capture_sensing_action_measurement_criteria
Description
To capture given Measurement_Criterion/criteria for sensing output (e.g. recon image quality).
«refine»
«interface»
Sensing_Requirement_Achievement
Description
This interface is the statement of achievement against the Sensing_Requirement.
«requirement»
identify_progress_of_sensing_solution
Description
To identify the progress of a Sensing_Solution against the Sensing_Requirement.
«refine»
«requirement»
capture_actual_quality_of_deliverables
Description
To capture the actual quality of the deliverables provided by a Sensing_Solution, measured against given Sensing_Requirements and Measurement_Criterion/Criteria.
«refine»

Figure 866: Sensing_Requirement Service Policy

Sensing_Requirement

This service determines the achievability of a Sensing_Requirement given the available Sensing_Capability and applicable Constraints, determines Sensing_Solutions and, when one is selected, uses it to fulfil achievable requirements.

Interfaces

Sensing_Requirement

This interface is the Sensing_Requirement (e.g. result type, spatial parameters or other criteria related to the sensing scope), the predicted quality of the Sensing_Solution, the associated cost of that requirement, and related timing information.

Attributes

specification	The scope or target to be sensed (e.g. point, area, volume or object of interest criteria).
quality_profile	Acceptable quality thresholds and gradients (i.e. minimum vs ideal level) to be obtained by the Sensing_Solution, specified appropriately for the type of Sensing_Requirement.
activation_criterion	How and when the Sensing_Requirement fulfilment should be triggered once selected.
coordinating_context	Identification and character of coordination required by other actions. For example, where Sensing is capturing data to be used as part of a larger task, the identification of other actions involved along with the nature of their interaction as it pertains to Sensing.
predicted_quality	How well the proposed Sensing_Solution is predicted to satisfy the Sensing_Requirement.

Sensing_Requirement_Achievement

This interface is the statement of achievement against the Sensing_Requirement.

Sensing_Requirement_Criterion

This interface is the Measurement_Criterion/Criteria against which the Sensing_Solution is assessed (e.g. timeliness or power required).

Attributes

property	The criterion property to be measured (e.g. a specific physical quantity such as area in square miles, a cost or quality factor such as electrical power usage, or an expression of the importance attaching to the Sensing_Requirement).
value	The amount related to the property to be measured, e.g. 50 square miles.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

identify_whether_sensing_requirement_is_achievable

Identify whether a planned or executing Sensing_Requirement fulfilment is achievable given current or predicted Sensing_Capability, Dynamic_Influences, and Constraints.

coordinate_sensing_solution_enactment

Fulfil a Sensing_Requirement by executing a Sensing_Solution.

determine_sensing_solution

Determine a Sensing_Solution that satisfies the given Sensing_Requirement within Constraints.

identify_sensing_solution_progress

Identify the progress of achievement by the Sensing_Solution against the Sensing_Requirement.

5.4.2.50.7.1.2 Sensing_Activity_Dependency

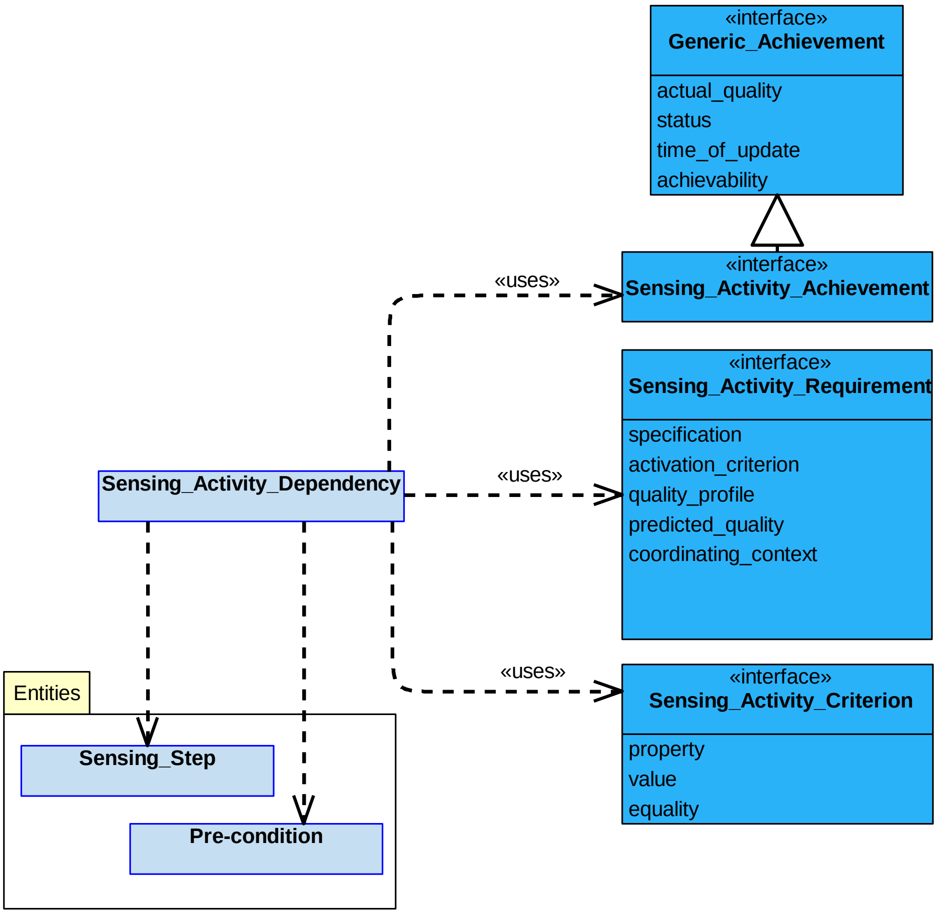


Figure 867: Sensing_Activity_Dependency Service Definition

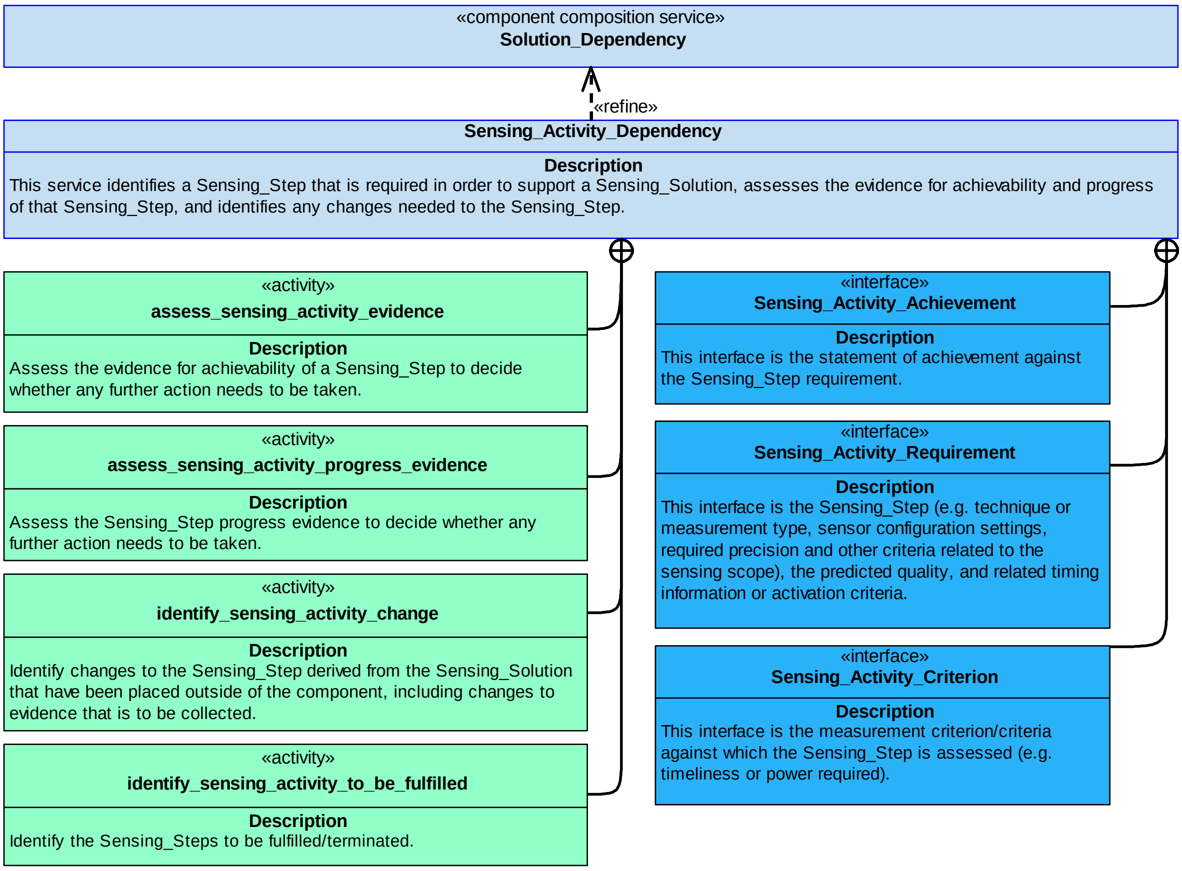

«component composition service»
Solution_Dependency
«refine»
Sensing_Activity_Dependency
Description
This service identifies a Sensing_Step that is required in order to support a Sensing_Solution, assesses the evidence for achievability and progress of that Sensing_Step, and identifies any changes needed to the Sensing_Step.
«activity»
assess_sensing_activity_evidence
Description
Assess the evidence for achievability of a Sensing_Step to decide whether any further action needs to be taken.
«activity»
assess_sensing_activity_progress_evidence
Description
Assess the Sensing_Step progress evidence to decide whether any further action needs to be taken.
«activity»
identify_sensing_activity_change
Description
Identify changes to the Sensing_Step derived from the Sensing_Solution that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
identify_sensing_activity_to_be_fulfilled
Description
Identify the Sensing_Steps to be fulfilled/terminated.
«interface»
Sensing_Activity_Achievement
Description
This interface is the statement of achievement against the Sensing_Step requirement.
«interface»
Sensing_Activity_Requirement
Description
This interface is the Sensing_Step (e.g. technique or measurement type, sensor configuration settings, required precision and other criteria related to the sensing scope), the predicted quality, and related timing information or activation criteria.
«interface»
Sensing_Activity_Criterion
Description
This interface is the measurement criterion/criteria against which the Sensing_Step is assessed (e.g. timeliness or power required).

Figure 868: Sensing_Activity_Dependency Service Policy

Sensing_Activity_Dependency

This service identifies a Sensing_Step that is required in order to support a Sensing_Solution, assesses the evidence for achievability and progress of that Sensing_Step, and identifies any changes needed to the Sensing_Step.

Interfaces

Sensing_Activity_Achievement

This interface is the statement of achievement against the Sensing_Step requirement.

Sensing_Activity_Requirement

This interface is the Sensing_Step (e.g. technique or measurement type, sensor configuration settings, required precision and other criteria related to the sensing scope), the predicted quality, and related timing information or activation criteria.

Attributes

specification	Detailed sensor configuration settings (e.g. frequency bands and dwell time profiles in order to express the technique required).
activation_criterion	Trigger for initiating and ceasing the sensor activity, e.g. how low latency dwelling will be initiated and ceased.

quality_profile	Precision and accuracy of measurements required, for example in the setting of field of regard, resolution levels, etc.
predicted_quality	How well the sensing action dependency is predicted to satisfy the requirement for the Sensing_Step.
coordinating_context	Identification and character of coordination of other actions required by this component. For example, where this component needs data to be captured that requires cooperation between resources, this might need to be conveyed, along with the nature of the interaction as it pertains to a particular Sensing_Step.

Sensing_Activity_Criterion

This interface is the measurement criterion/criteria against which the Sensing_Step is assessed (e.g. timeliness or power required).

Attributes

property	The criterion property to be measured, e.g. a cost or quality factor such as time taken or electrical power usage, or an expression of the importance attaching to the Sensing_Step.
value	The amount related to the property to be measured, e.g. 50 microseconds.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

assess_sensing_activity_evidence

Assess the evidence for achievability of a Sensing_Step to decide whether any further action needs to be taken.

assess_sensing_activity_progress_evidence

Assess the Sensing_Step progress evidence to decide whether any further action needs to be taken.

identify_sensing_activity_change

Identify changes to the Sensing_Step derived from the Sensing_Solution that have been placed outside of the component, including changes to evidence that is to be collected.

identify_sensing_activity_to_be_fulfilled

Identify the Sensing_Steps to be fulfilled/terminated.

5.4.2.50.7.1.3 Processing_Dependency

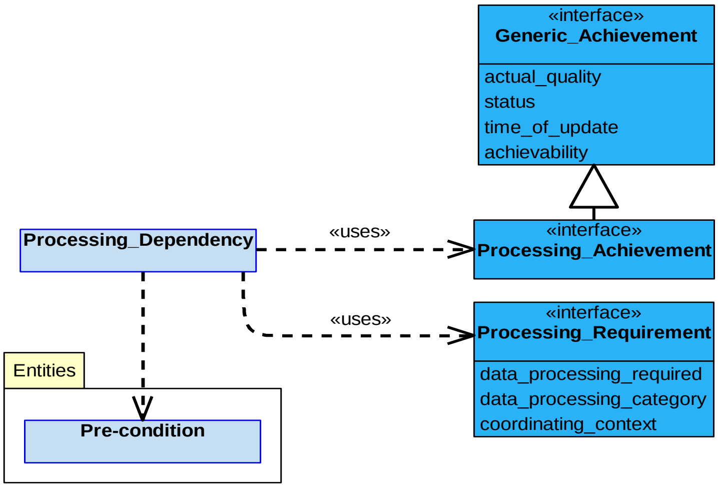


Figure 869: Processing_Dependency Service Definition

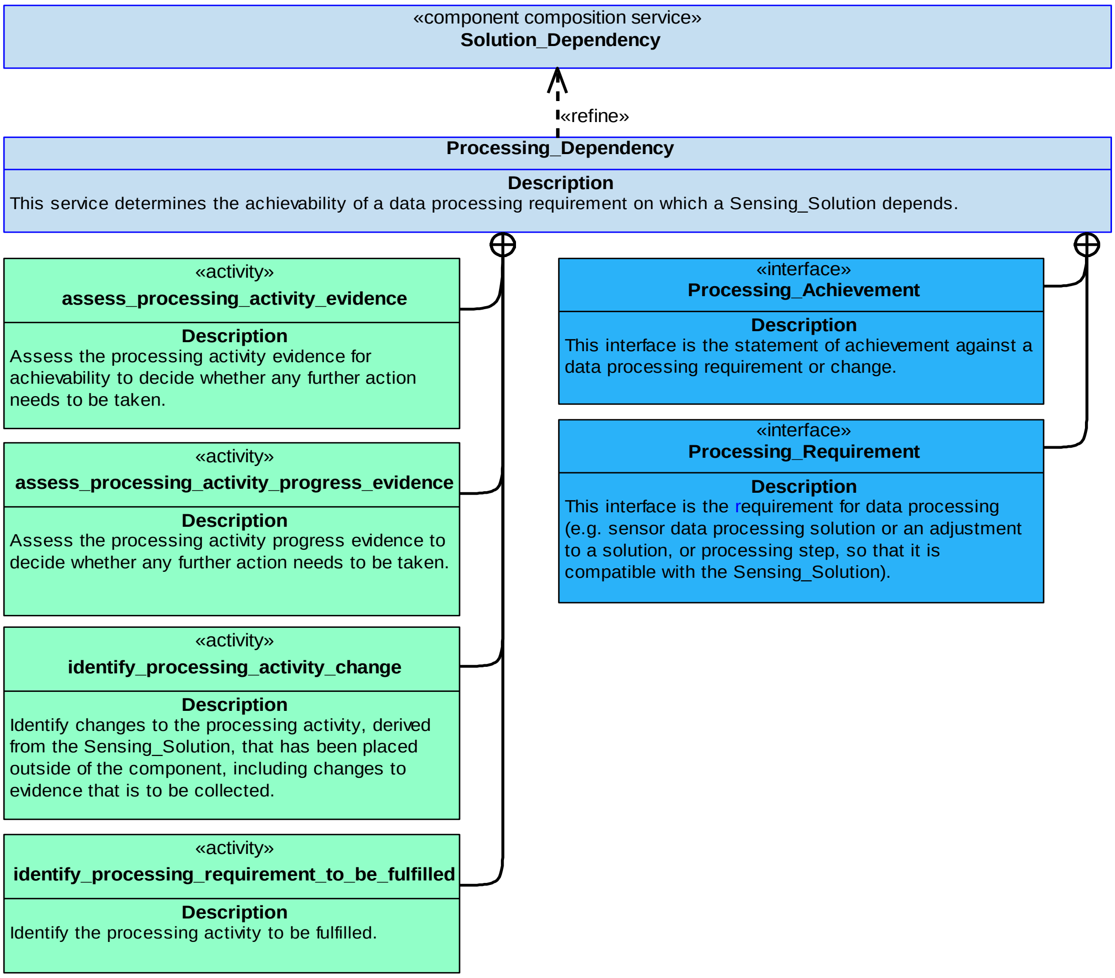

«component composition service»
Solution_Dependency
«refine»
Processing_Dependency
Description
This service determines the achievability of a data processing requirement on which a Sensing_Solution depends.
«activity»
assess_processing_activity_evidence
Description
Assess the processing activity evidence for achievability to decide whether any further action needs to be taken.
«activity»
assess_processing_activity_progress_evidence
Description
Assess the processing activity progress evidence to decide whether any further action needs to be taken.
«activity»
identify_processing_activity_change
Description
Identify changes to the processing activity, derived from the Sensing_Solution, that has been placed outside of the component, including changes to evidence that is to be collected.
«activity»
identify_processing_requirement_to_be_fulfilled
Description
Identify the processing activity to be fulfilled.
«interface»
Processing_Achievement
Description
This interface is the statement of achievement against a data processing requirement or change.
«interface»
Processing_Requirement
Description
This interface is the requirement for data processing (e.g. sensor data processing solution or an adjustment to a solution, or processing step, so that it is compatible with the Sensing_Solution).

Figure 870: Processing_Dependency Service Policy

Processing_Dependency

This service determines the achievability of a data processing requirement on which a Sensing_Solution depends.

Interfaces

Processing_Achievement

This interface is the statement of achievement against a data processing requirement or change.

Processing_Requirement

This interface is the requirement for data processing (e.g. sensor data processing solution or an adjustment to a solution, or processing step, so that it is compatible with the Sensing_Solution).

Attributes

data_processing_required	The data processing of the Acquired_Data that is required as part of the Sensing_Solution.
data_processing_category	The specific data or category of data to which a processing dependency applies.
coordinating_context	Identification of other actions required by this component; for example, where this component is capturing data to be used as part of a larger task that includes co-dependent data interpretation action(s), the identification of the other actions involved might be needed, along with the nature of their interaction as it pertains to a particular Sensing_Step.

Activities

assess_processing_activity_evidence

Assess the processing activity evidence for achievability to decide whether any further action needs to be taken.

assess_processing_activity_progress_evidence

Assess the processing activity progress evidence to decide whether any further action needs to be taken.

identify_processing_activity_change

Identify changes to the processing activity, derived from the Sensing_Solution, that has been placed outside of the component, including changes to evidence that is to be collected.

identify_processing_requirement_to_be_fulfilled

Identify the processing activity to be fulfilled.

5.4.2.50.7.1.4 Sensor_Platform_Information

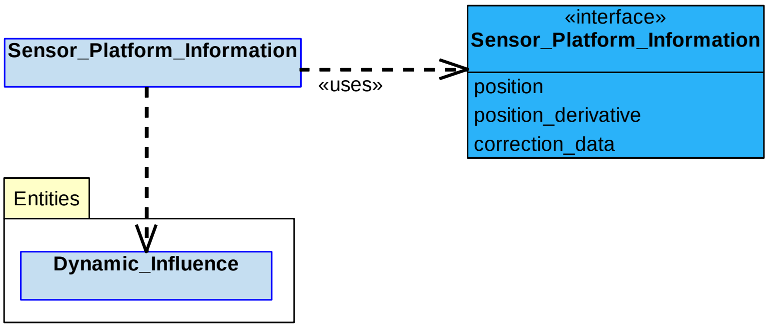


Figure 871: Platform_Information Service Definition

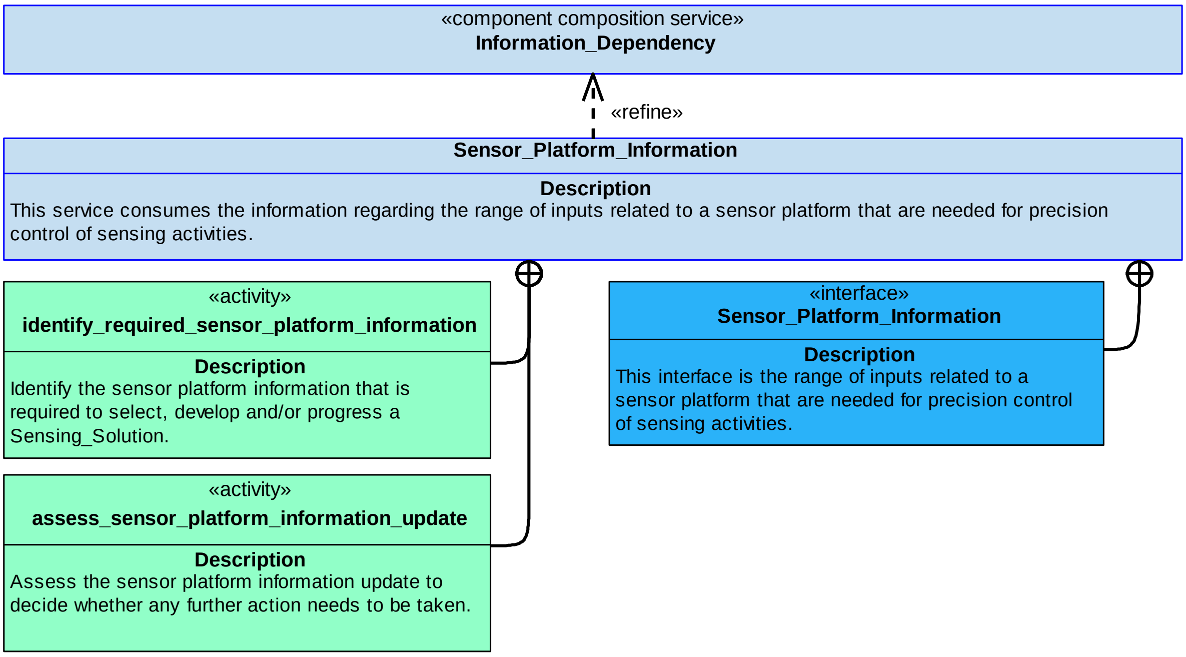

«component composition service»
Information_Dependency
«refine»
Sensor_Platform_Information
Description
This service consumes the information regarding the range of inputs related to a sensor platform that are needed for precision control of sensing activities.
«activity»
identify_required_sensor_platform_information
Description
Identify the sensor platform information that is required to select, develop and/or progress a Sensing_Solution.
«activity»
assess_sensor_platform_information_update
Description
Assess the sensor platform information update to decide whether any further action needs to be taken.
«interface»
Sensor_Platform_Information
Description
This interface is the range of inputs related to a sensor platform that are needed for precision control of sensing activities.

Figure 872: Platform_Information Service Policy

Sensor_Platform_Information

This service consumes the information regarding the range of inputs related to a sensor platform that are needed for precision control of sensing activities.

Interface

Sensor_Platform_Information

This interface is the range of inputs related to a sensor platform that are needed for precision control of sensing activities.

Attributes

position	Current and predicted location and orientation of the sensor platform(s) that will be used to perform Sensing_Steps.

position_derivative	A derivative of the sensor platform’s location or orientation (e.g. velocity or acceleration).
correction_data	Spatial correction data for sensors affected by elastic deformation, caused by dynamic forces acting on parts of the sensor platform's frame. For example, sensors mounted on wing tips or other sensor platform extremities.

Activities

assess_sensor_platform_information_update

Assess the sensor platform information update to decide whether any further action needs to be taken.

identify_required_sensor_platform_information

Identify the sensor platform information that is required to select, develop and/or progress a Sensing_Solution.

5.4.2.50.7.1.5 Environmental_Information

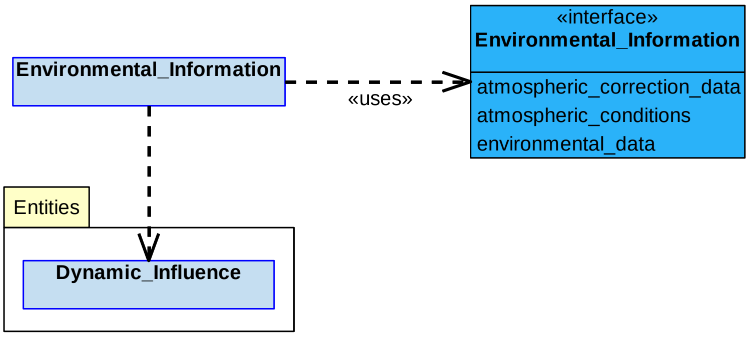


Figure 873: Environmental_Information Service Definition

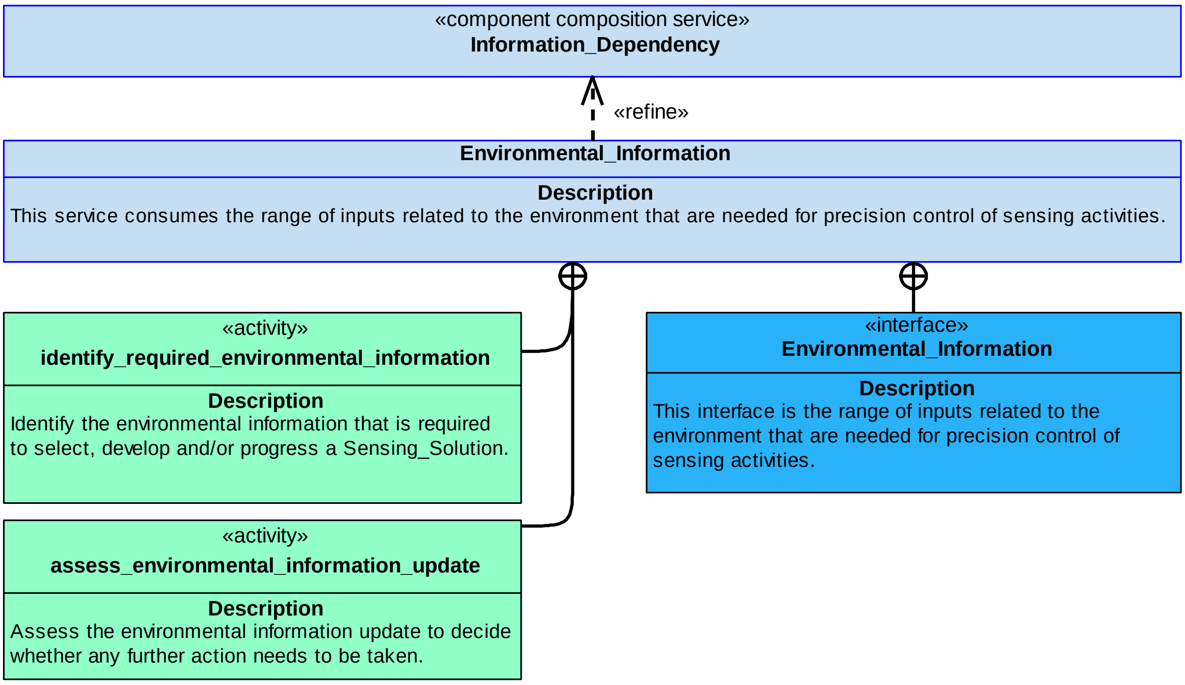

«component composition service»
Information_Dependency
«refine»
Environmental_Information
Description
This service consumes the range of inputs related to the environment that are needed for precision control of sensing activities.
«activity»
identify_required_environmental_information
Description
Identify the environmental information that is required to select, develop and/or progress a Sensing_Solution.
«activity»
assess_environmental_information_update
Description
Assess the environmental information update to decide whether any further action needs to be taken.
«interface»
Environmental_Information
Description
This interface is the range of inputs related to the environment that are needed for precision control of sensing activities.

Figure 874: Environmental_Information Service Policy

Environmental_Information

This service consumes the range of inputs related to the environment that are needed for precision control of sensing activities.

<u>Interface</u>

Environmental_Information

This interface is the range of inputs related to the environment that are needed for precision control of sensing activities.

<u>Attributes</u>

atmospheric_correction_data	Correction data for sensors related to changes in atmospheric conditions that affects sensing processes and performance.
atmospheric_conditions	Current and predicted weather conditions and features.
environmental_data	Information describing surfaces and features in the environment to be sensed. This may include land terrain or other environments.

<u>Activities</u>

assess_environmental_information_update

Assess the environmental information update to decide whether any further action needs to be taken.

identify_required_environmental_information

Identify the environmental information that is required to select, develop and/or progress a Sensing_Solution.

5.4.2.50.7.1.6 Tactical_Information

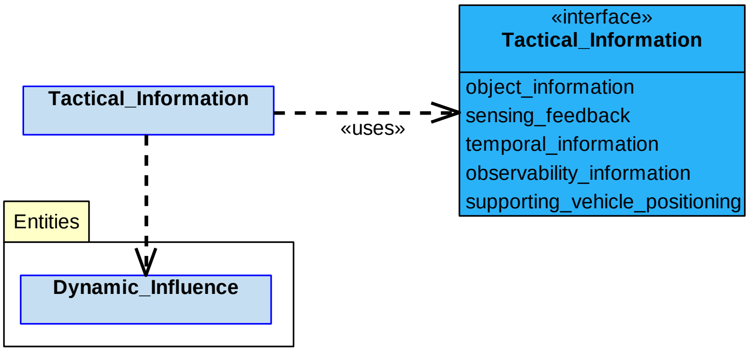


Figure 875: Tactical_Information Service Definition

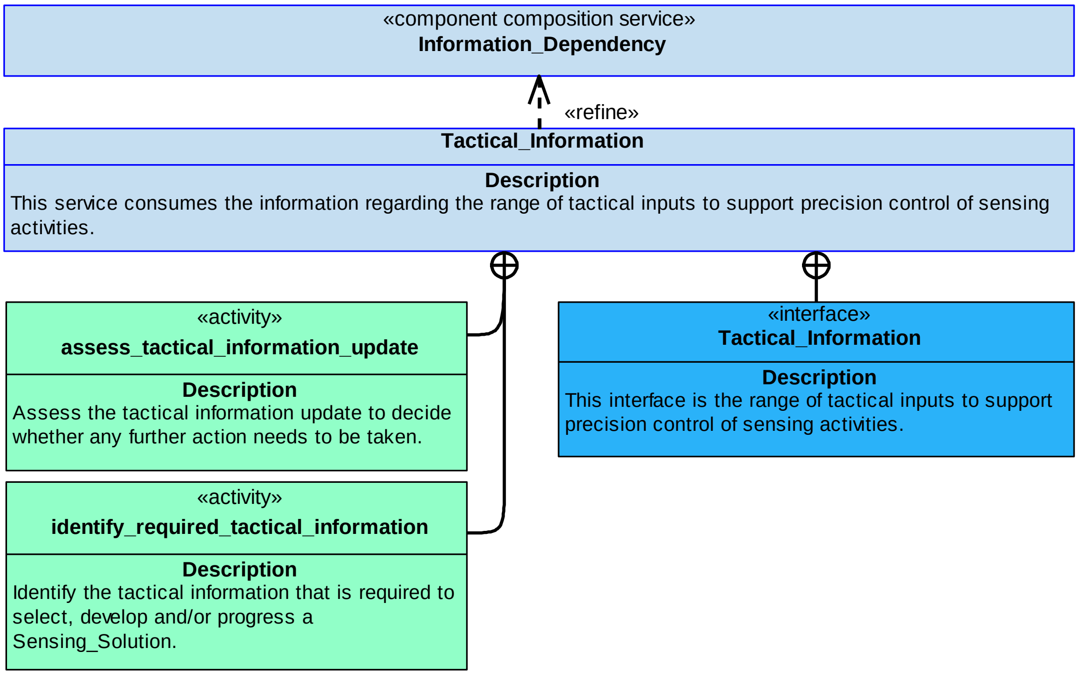

«component composition service»
Information_Dependency
«refine»
Tactical_Information
Description
This service consumes the information regarding the range of tactical inputs to support precision control of sensing activities.
«activity»
assess_tactical_information_update
Description
Assess the tactical information update to decide whether any further action needs to be taken.
«activity»
identify_required_tactical_information
Description
Identify the tactical information that is required to select, develop and/or progress a Sensing_Solution.
«interface»
Tactical_Information
Description
This interface is the range of tactical inputs to support precision control of sensing activities.

Figure 876: Tactical_Information Service Policy

Tactical_Information

This service consumes the information regarding the range of tactical inputs to support precision control of sensing activities.

Interface

Tactical_Information

This interface is the range of tactical inputs to support precision control of sensing activities.

Attributes

object_information	Dynamic information about sensing target criteria, location and movement, or that of surrounding objects.

sensing_feedback	Dynamic feedback to indicate quality of sensor returns for general adjustment of Sensing_Steps or Sensing_Solutions.
temporal_information	Timing information required for low latency synchronisation of sensing techniques.
observability_information	Dynamic and interactive observability factors affecting the use of active sensing techniques.
supporting_vehicle_positioning	Positioning and kinematics of vehicles involved in multi-vehicle sensing techniques.

Activities

assess_tactical_information_update

Assess the tactical information update to decide whether any further action needs to be taken.

identify_required_tactical_information

Identify the tactical information that is required to select, develop and/or progress a Sensing_Solution.

5.4.2.50.7.1.7 Constraint

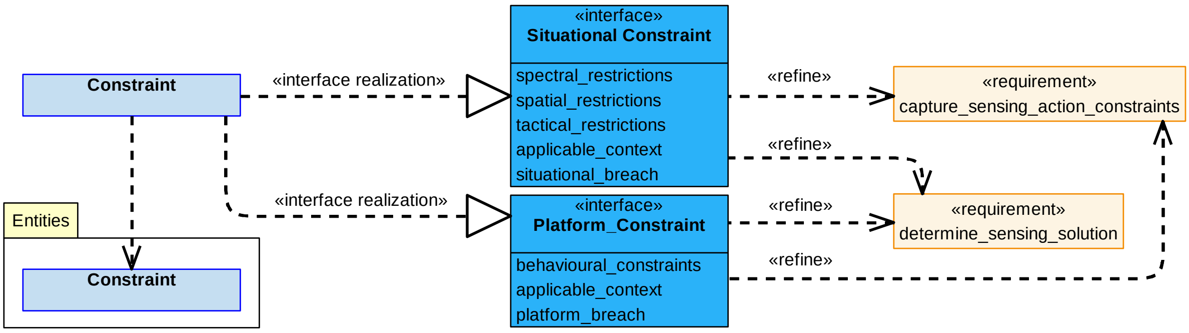


Figure 877: Constraint Service Definition

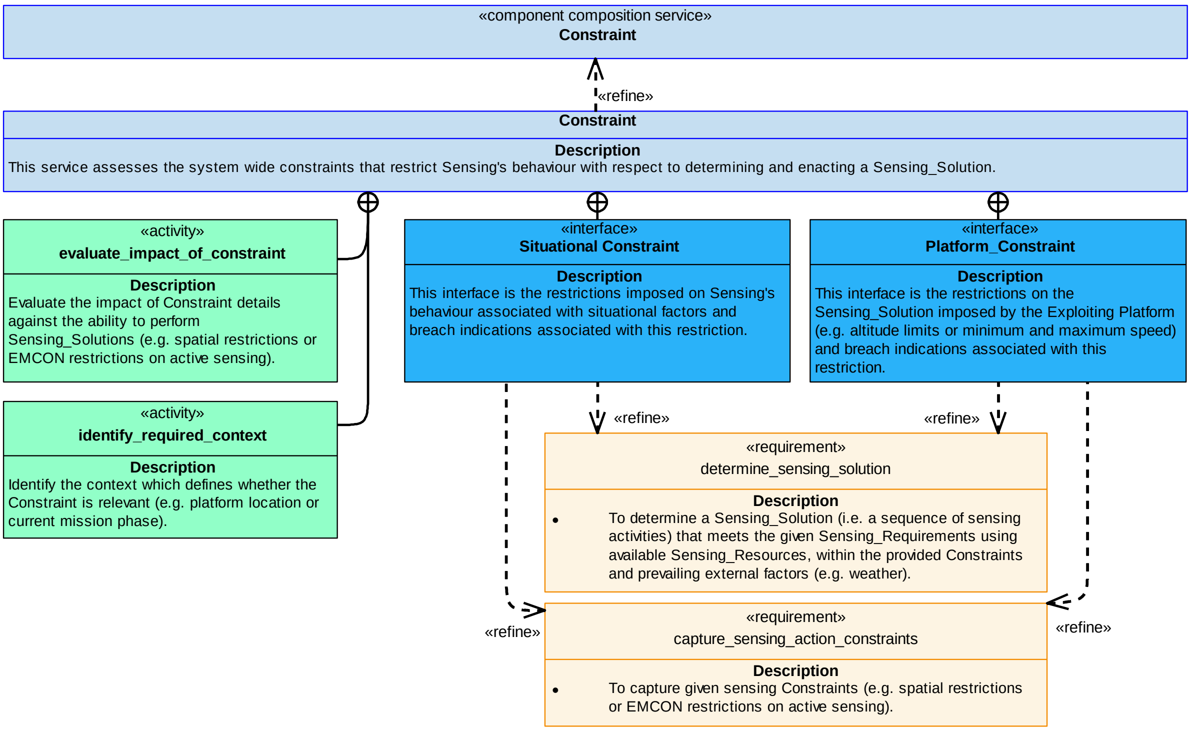

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the system wide constraints that restrict Sensing's behaviour with respect to determining and enacting a Sensing_Solution.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the ability to perform Sensing_Solutions (e.g. spatial restrictions or EMCON restrictions on active sensing).
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraint is relevant (e.g. platform location or current mission phase).
«interface»
Situational Constraint
Description
This interface is the restrictions imposed on Sensing's behaviour associated with situational factors and breach indications associated with this restriction.
«interface»
Platform_Constraint
Description
This interface is the restrictions on the Sensing_Solution imposed by the Exploiting Platform (e.g. altitude limits or minimum and maximum speed) and breach indications associated with this restriction.
«refine»
«refine»
«requirement»
determine_sensing_solution
Description
To determine a Sensing_Solution (i.e. a sequence of sensing activities) that meets the given Sensing_Requirements using available Sensing_Resources, within the provided Constraints and prevailing external factors (e.g. weather).
«refine»
«refine»
«requirement»
capture_sensing_action_constraints
Description
To capture given sensing Constraints (e.g. spatial restrictions or EMCON restrictions on active sensing).

Figure 878: Constraint Service Policy

Constraint

This service assesses the system wide constraints that restrict Sensing's behaviour with respect to determining and enacting a Sensing_Solution.

Interfaces

Situational Constraint

This interface is the restrictions imposed on Sensing's behaviour associated with situational factors and breach indications associated with this restriction.

Attributes

spectral_restrictions	EMCON restrictions applied to the Sensing_Solution. For example, ranges of the EM spectrum that the vehicle is not permitted to use due to regulatory restrictions.
spatial_restrictions	Constraints on the Sensing_Solution due to restrictions on the vehicle's position and orientation, e.g. no fly zones.
tactical_restrictions	Restrictions applied to the Sensing_Solution for tactical reasons, e.g. prevention of EM transmission within 50 miles of a known threat such as a SAM site.
applicable_context	The context in which the Constraint is applicable.
situational_breach	A statement that the situational Constraint has been breached.

Platform_Constraint

This interface is the restrictions on the Sensing_Solution imposed by the Exploiting Platform (e.g. altitude limits or minimum and maximum speed) and breach indications associated with this restriction.

Attributes

behavioural_constraints	Constraints on the Sensing_Solution imposed by the Exploiting Platform (e.g. altitude limits, or minimum and maximum speed).
applicable_context	The context in which the Constraint is applicable.
platform_breach	A statement that the platform Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the ability to perform Sensing_Solutions (e.g. spatial restrictions or EMCON restrictions on active sensing).

identify_required_context

Identify the context which defines whether the Constraint is relevant (e.g. platform location or current mission phase).

5.4.2.50.7.1.8 Capability

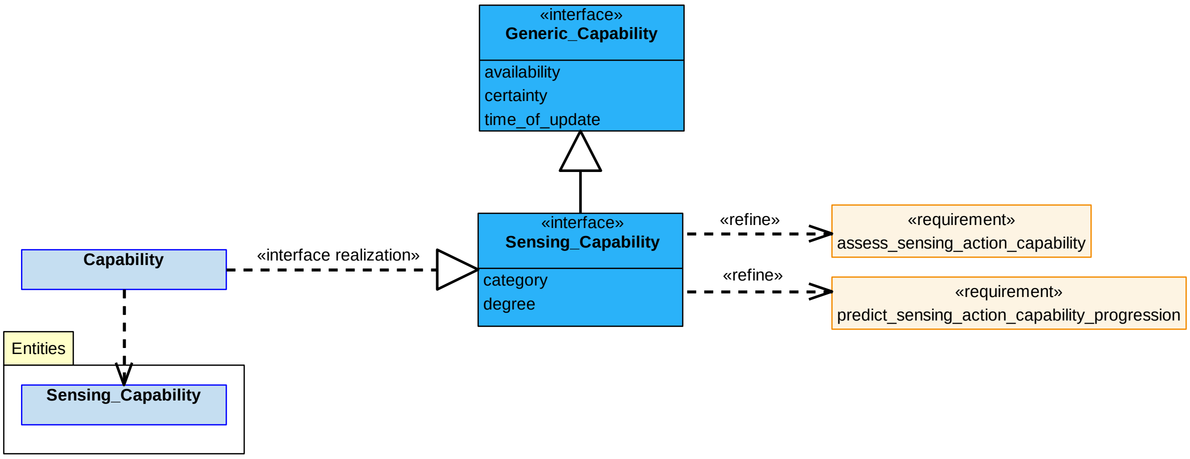


Figure 879: Capability Service Definition

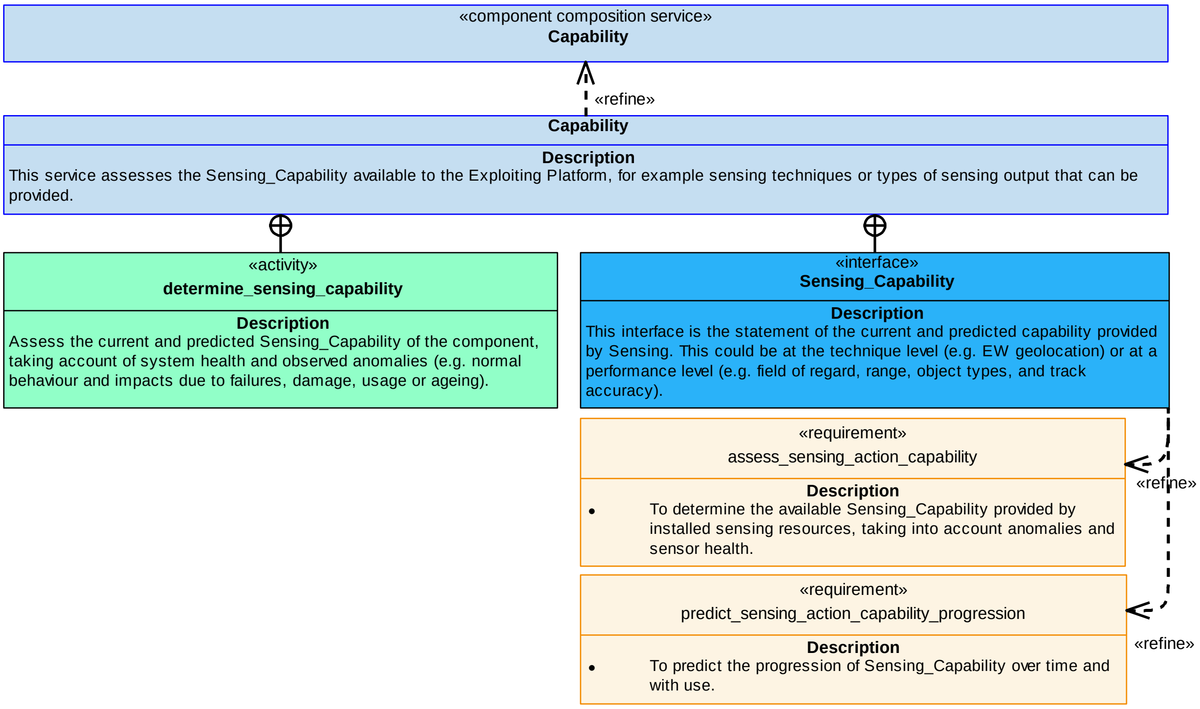

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the Sensing_Capability available to the Exploiting Platform, for example sensing techniques or types of sensing output that can be provided.
«activity»
determine_sensing_capability
Description
Assess the current and predicted Sensing_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Sensing_Capability
Description
This interface is the statement of the current and predicted capability provided by Sensing. This could be at the technique level (e.g. EW geolocation) or at a performance level (e.g. field of regard, range, object types, and track accuracy).
«requirement»
assess_sensing_action_capability
Description
To determine the available Sensing_Capability provided by installed sensing resources, taking into account anomalies and sensor health.
«refine»
«requirement»
predict_sensing_action_capability_progression
Description
To predict the progression of Sensing_Capability over time and with use.
«refine»

Figure 880: Capability Service Policy

Capability

This service assesses the Sensing_Capability available to the Exploiting Platform, for example sensing techniques or types of sensing output that can be provided.

Interface

Sensing_Capability

This interface is the statement of the current and predicted capability provided by Sensing. This could be at the technique level (e.g. EW geolocation) or at a performance level (e.g. field of regard, range, object types, and track accuracy).

Attributes

category	The type or category of Sensing_Capability that is being provided.
degree	The level of performance or effectiveness that can be achieved for the associated Sensing_Capability.

Activity

determine_sensing_capability

Assess the current and predicted Sensing_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.50.7.1.9 Capability_Evidence

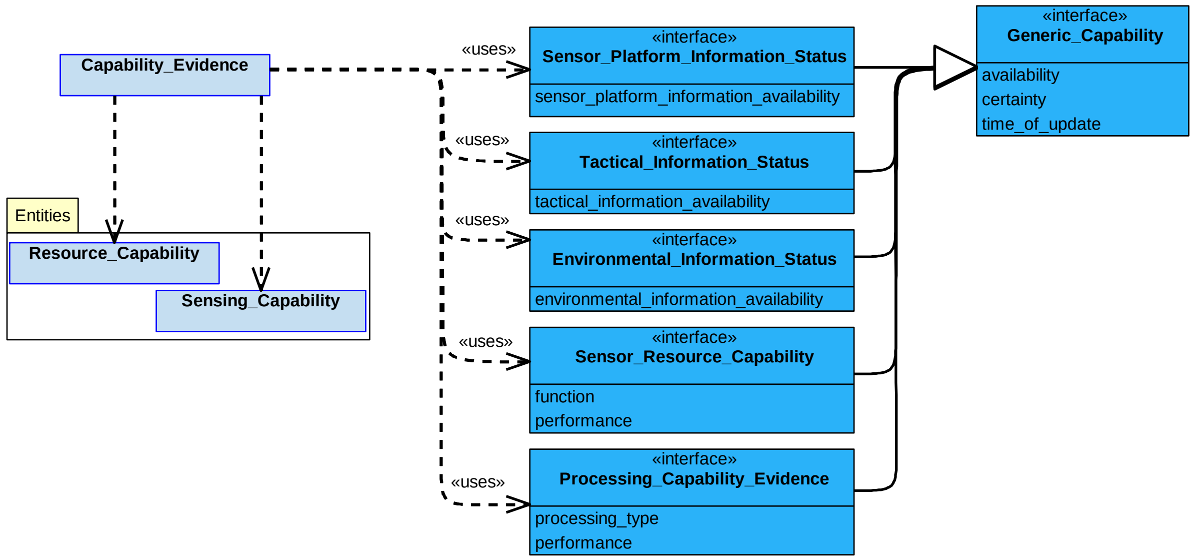

Capability_Evidence
Entities
Resource_Capability
Sensing_Capability
«uses»
«uses»
«uses»
«uses»
«uses»
«interface»
Sensor_Platform_Information_Status
sensor_platform_information_availability
«interface»
Tactical_Information_Status
tactical_information_availability
«interface»
Environmental_Information_Status
environmental_information_availability
«interface»
Sensor_Resource_Capability
function
performance
«interface»
Processing_Capability_Evidence
processing_type
performance
«interface»
Generic_Capability
availability
certainty
time_of_update

Figure 881: Capability_Evidence Service Definition

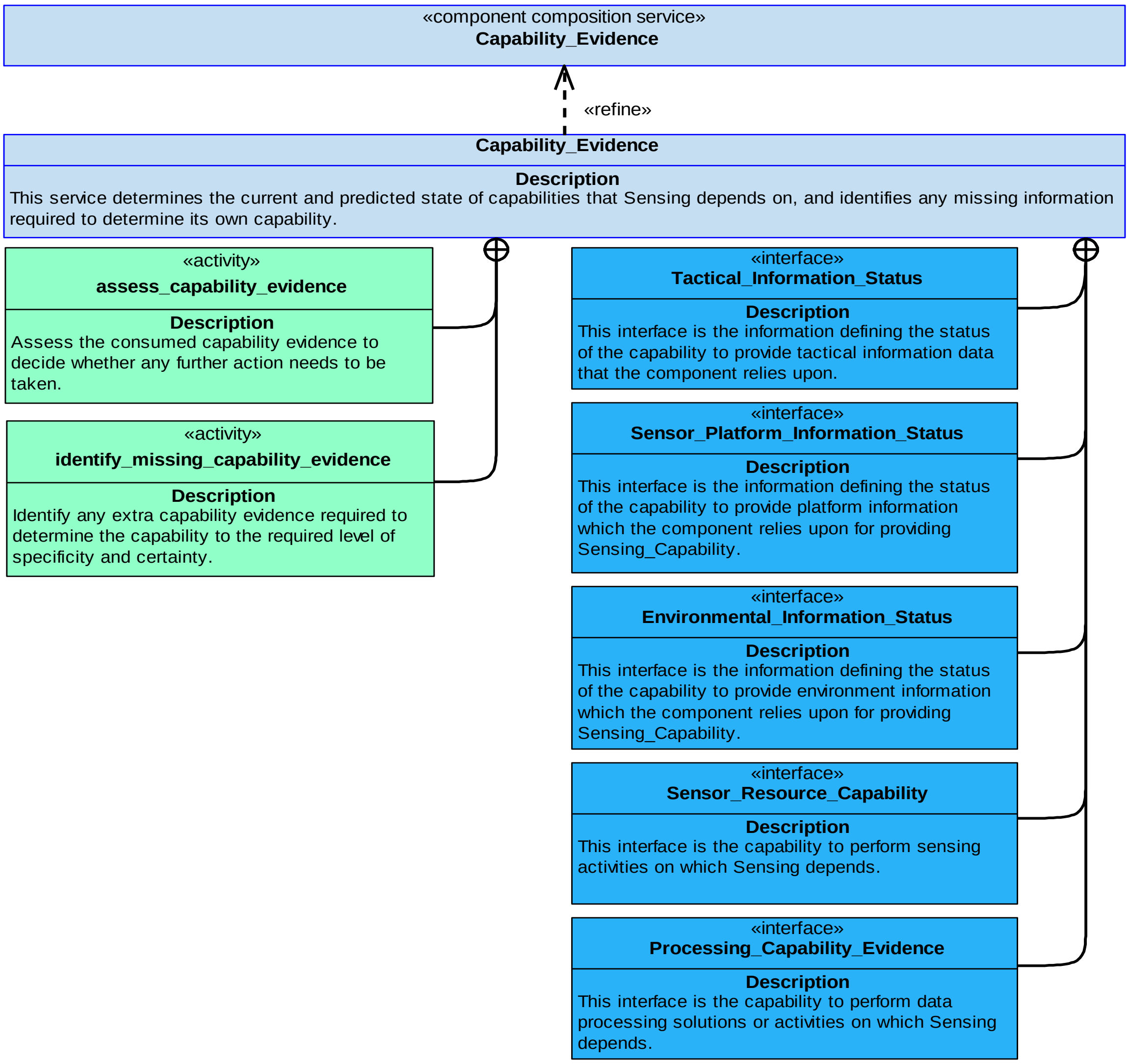

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service determines the current and predicted state of capabilities that Sensing depends on, and identifies any missing information required to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the consumed capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.
«interface»
Tactical_Information_Status
Description
This interface is the information defining the status of the capability to provide tactical information data that the component relies upon.
«interface»
Sensor_Platform_Information_Status
Description
This interface is the information defining the status of the capability to provide platform information which the component relies upon for providing Sensing_Capability.
«interface»
Environmental_Information_Status
Description
This interface is the information defining the status of the capability to provide environment information which the component relies upon for providing Sensing_Capability.
«interface»
Sensor_Resource_Capability
Description
This interface is the capability to perform sensing activities on which Sensing depends.
«interface»
Processing_Capability_Evidence
Description
This interface is the capability to perform data processing solutions or activities on which Sensing depends.

Figure 882: Capability_Evidence Service Policy

Capability_Evidence

This service determines the current and predicted state of capabilities that Sensing depends on, and identifies any missing information required to determine its own capability.

Interfaces

Tactical_Information_Status

This interface is the information defining the status of the capability to provide tactical information data that the component relies upon.

Attribute

tactical_information_availability	The availability of tactical information used in the provision of Sensing_Capability.

Sensor_Platform_Information_Status

This interface is the information defining the status of the capability to provide platform information which the component relies upon for providing Sensing_Capability.

Attribute

sensor_platform_information_availability	The availability of sensor platform information used in the provision of Sensing_Capability.

Environmental_Information_Status

This interface is the information defining the status of the capability to provide environment information which the component relies upon for providing Sensing_Capability.

Attribute

environmental_information_availability	The availability of environment information used in the provision of Sensing_Capability.

Sensor_Resource_Capability

This interface is the capability to perform sensing activities on which Sensing depends.

Attributes

function	The specific function or technique, including the control options for its use. For example, priority search, track, sample, or coordinated search.
performance	The level or degree of capability available for the associated function, taking into account the type, location and fit of the sensor equipment on the vehicle. For example, field of regard, sampling rate, or minimum detectable signal.

Processing_Capability_Evidence

This interface is the capability to perform data processing solutions or activities on which Sensing depends.

Attributes

processing_type	The specific data processing technique or process. For example, capability to process particular kinds of information (e.g. SAR images or specific types of electronic surveillance returns).
performance	The level or degree of capability available for the associated processing type.

Activities

assess_capability_evidence

Assess the consumed capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.50.7.2 Service Dependencies

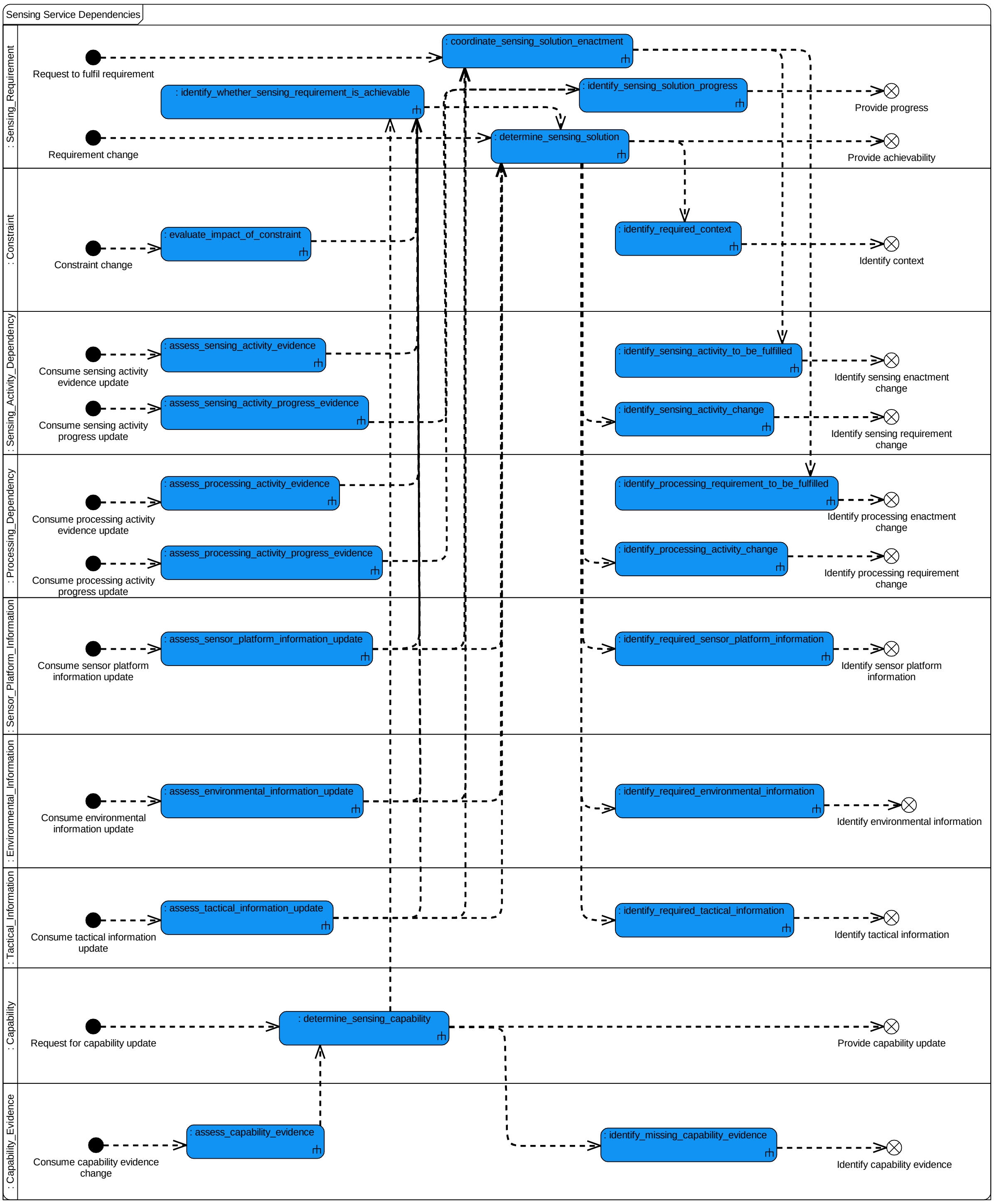

Sensing Service Dependencies
: Sensing_Requirement
: Constraint
: Sensing_Activity_Dependency
: Processing_Dependency
: Sensor_Platform_Information
: Environmental_Information
: Tactical_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
: coordinate_sensing_solution_enactment
: identify_whether_sensing_requirement_is_achievable
: identify_sensing_solution_progress
: determine_sensing_solution
Provide progress
Provide achievability
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume sensing activity evidence update
: assess_sensing_activity_evidence
Consume sensing activity progress update
: assess_sensing_activity_progress_evidence
: identify_sensing_activity_to_be_fulfilled
Identify sensing enactment change
: identify_sensing_activity_change
Identify sensing requirement change
Consume processing activity evidence update
: assess_processing_activity_evidence
Consume processing activity progress update
: assess_processing_activity_progress_evidence
: identify_processing_requirement_to_be_fulfilled
Identify processing enactment change
: identify_processing_activity_change
Identify processing requirement change
Consume sensor platform information update
: assess_sensor_platform_information_update
: identify_required_sensor_platform_information
Identify sensor platform information
Consume environmental information update
: assess_environmental_information_update
: identify_required_environmental_information
Identify environmental information
Consume tactical information update
: assess_tactical_information_update
: identify_required_tactical_information
Identify tactical information
Request for capability update
: determine_sensing_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 883: Sensing Service Dependencies

5.4.2.51 Sensor Data Interpretation

5.4.2.51.1 Role

The role of Sensor Data Interpretation is to coordinate the extraction of meaning from data produced by sensors.

5.4.2.51.2 Overview

Control Architecture

Sensor Data Interpretation is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a Requirement is received, Sensor Data Interpretation will determine a Data_Interpretation_Solution consisting of one or more activities, each with an associated Data_Interpretation_Dependency or dependencies (e.g. the timely availability of Sensor_Data_Products on which a Data_Interpretation_Dependency is based). The Data_Interpretation_Solution is then enacted, which will involve coordinated control of Interpretation_Resources. The Data_Interpretation_Solution will produce an Interpreted_Data_Product from the Interpretation_Resources.

Examples of Use

Sensor Data Interpretation is required where coordinated planning and execution of activities which satisfy a Data_Interpretation_Dependency or dependencies may be necessary. For example:

- To coordinate development of Interpreted_Data_Products that detect, recognise, or identify objects from Sensor_Data_Products for the purpose of reconnaissance.
- To coordinate development of Interpreted_Data_Products that locate objects from Sensor_Data_Products in order that they can be targeted with ordnance.

5.4.2.51.3 Service Summary

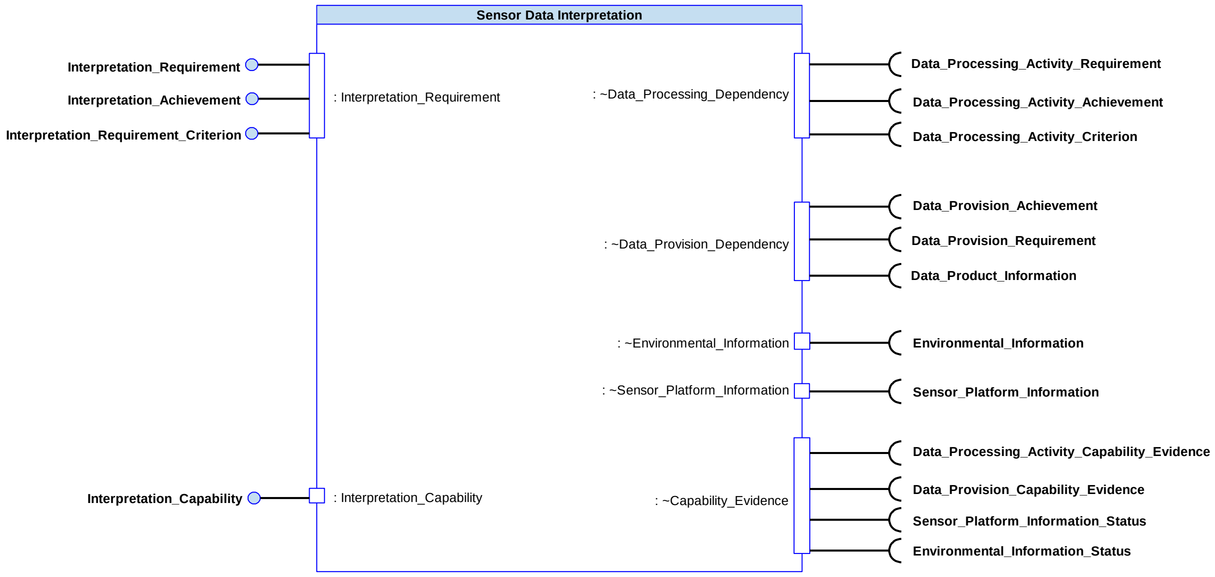

Sensor Data Interpretation
Interpretation_Requirement
Interpretation_Achievement
Interpretation_Requirement_Criterion
: Interpretation_Requirement
: ~Data_Processing_Dependency
Data_Processing_Activity_Requirement
Data_Processing_Activity_Achievement
Data_Processing_Activity_Criterion
: ~Data_Provision_Dependency
Data_Provision_Achievement
Data_Provision_Requirement
Data_Product_Information
: ~Environmental_Information
Environmental_Information
: ~Sensor_Platform_Information
Sensor_Platform_Information
Interpretation_Capability
: Interpretation_Capability
: ~Capability_Evidence
Data_Processing_Activity_Capability_Evidence
Data_Provision_Capability_Evidence
Sensor_Platform_Information_Status
Environmental_Information_Status

Figure 884: Sensor Data Interpretation Service Summary

5.4.2.51.4 Responsibilities

capture_requirements

- To capture provided Requirements for Interpreted_Data_Products (e.g. determine the position of objects of a particular type within a region).

capture_measurement_criteria

- To capture the method for measuring a Data_Interpretation_Solution (e.g. by comparison with a required confidence in the result).

identify_if_requirement_remains_achievable

- To identify if a Requirement remains achievable given current Interpretation_Resources.

determine_solution

- To determine a coordinated and combined sequence of activities to use as the Data_Interpretation_Solution which satisfies an interpretation Requirement.

determine_predicted_quality_of_solution

- To determine the quality of a proposed Data_Interpretation_Solution against given Measurement_Criterion.

determine_solution_dependencies

- To identify any Sensor_Data_Provision_Dependency and Data_Interpretation_Dependency required to support the Data_Interpretation_Solution.

coordinate_solution

- To coordinate the execution of a Data_Interpretation_Solution.

identify_progress_of_solution

- To identify the progress of a Data_Interpretation_Solution against the Requirements.

determine_quality_of_deliverables

- To determine the quality of the Interpreted_Data_Product, measured against given Requirements and Measurement_Criterion.

assess_capability

- To assess the Interpretation_Resource_Capability that can be applied to sensor data, taking account of observed anomalies (e.g. erroneous outputs of activities which satisfy a Data_Interpretation_Dependency or dependencies).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Interpretation_Capability assessment.

predict_capability_progression

- To predict the progression of Interpretation_Capability over time and with use.

5.4.2.51.5 Subject Matter Semantics

The subject matter of Sensor Data Interpretation is requirements for data interpretation, including the solutions required to meet data interpretation requirements, the resources that can be used to provide these solutions and the quality of interpreted data needed to meet the requirements.

Exclusions

The subject matter of Sensor Data Interpretation does not include:

- The actual measurements in the raw Sensor_Data_Product or Interpreted_Data_Product (although information about the availability and quality of this data are within the subject matter of Sensor Data Interpretation).
- Solutions for the capture of additional raw Sensor_Data_Products needed for a Data_Interpretation_Solution.

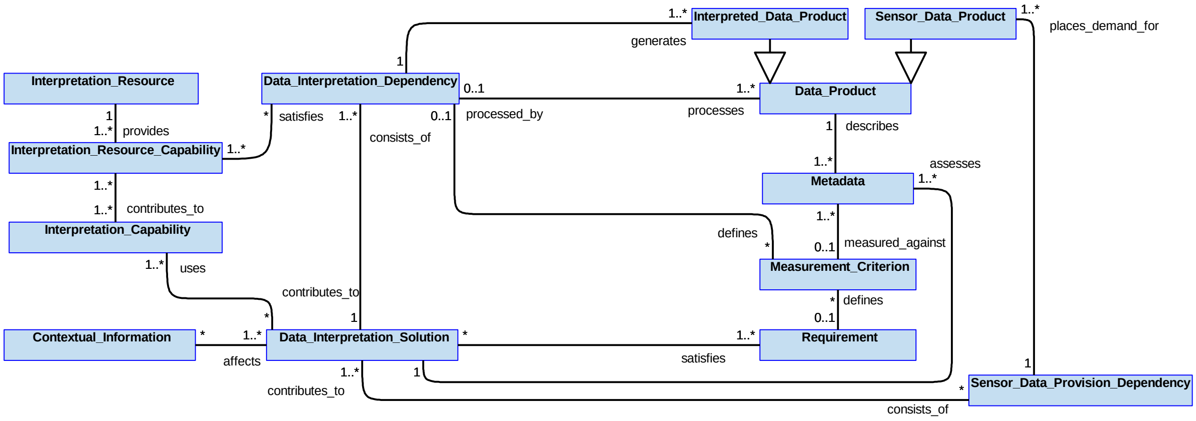


Figure 885: Sensor Data Interpretation Subject Matter Diagram

5.4.2.51.5.1 Entities

Interpretation_Resource_Capability

The capability to satisfy a Data_Interpretation_Dependency provided by an Interpretation_Resource.

Data_Product

Any data that can be subject to interpretation, or data which has already been interpreted and may or may not be subject to further interpretation.

Data_Interpretation_Solution

A planned combination of activities that identifies required sensor data and extracts meaning from it.

Data_Interpretation_Dependency

A dependency for an algorithmic process upon which the interpretation of sensor data depends.

Interpretation_Capability

The ability of Sensor Data Interpretation to coordinate activities to derive a particular type of meaning from a particular type or types of sensor data.

Interpretation_Resource

A resource capable of applying algorithmic processes to data. This represents a combination of processing power with an algorithm that is able to extract or amalgamate information from the provided data, e.g. a pattern recognition algorithm given adequate processing time in order to detected ground based objects in a SAR image.

Interpreted_Data_Product

The product of executed data interpretation activities (e.g. a data product with improved confidence level, which has been filtered to remove clutter, or data about identified objects), which may be subsequently used in further interpretation activities.

Measurement_Criterion

A criterion for measuring the achievement of required quality (e.g. speed, confidence, precision, or accuracy). This may be a final overall quality criterion or define the required quality at an intermediate stage.

Requirement

A requirement to derive meaning from sensor data (e.g. to locate enemy tanks in a region of territory or provide human readable recon images enhanced from SAR data).

Sensor_Data_Product

Data captured by sensor resource(s) that is to be interpreted (e.g. radar returns, images, or sound recordings). This may be from a single sensor resource or data store, or originate from multiple sources.

Contextual_Information

Platform, environmental, or tactical information that can influence the approach to interpreting sensor data (e.g. atmospheric conditions, vehicle speed, and position).

Sensor_Data_Provision_Dependency

A definition of sensor data of a particular type and quality upon which the interpretation of sensor data depends.

Metadata

Statements about aspects of data, such as its type, age, source, or quality, which can be considered independently of the data itself.

5.4.2.51.6 Design Rationale

5.4.2.51.6.1 Assumptions

- The component will need a sufficient understanding of objects of interest (e.g. from mission data) and track sets at the appropriate abstraction level in order to coordinate the generation of Interpreted_Data_Products.
- Additional capabilities of new Interpretation_Resources or configurations installed on an Exploiting Platform can be added easily, which can cater for changes in Interpretation_Resource_Capability between missions.

5.4.2.51.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Sensor Data Interpretation:

- Control Architecture - As an action component, Sensor Data Interpretation will interact with service and resource components. Sensor Data Interpretation may also interact with the Task Layer components. It can be appropriate for Tasks to implement dynamic dependencies between actions that Sensor Data Interpretation is concerned with (e.g. a direct feedback link to Sensing so that the component can coordinate its choice of interpretation techniques with the type and quality of sensor product being generated).
- Data Driving - Different combinations of installed Interpretation_Resource_Capability will be used by different Exploiting Programmes, this is expected to be accommodated by the Sensor Data Interpretation component by configuration through data driving.
- Recording and Logging - Logging operations and record retention will be performed in accordance with this PYRAMID concept.
- Tactical Information - For Exploiting Platforms where variable control of sensor data handling is required, Sensor Products is controlled by the Sensor Data Interpretation component.

Extensions

- The construction of Data_Interpretation_Solutions (i.e. calculating which combination of Data_Interpretation_Dependency should be applied to Sensor_Data_Products to achieve the required Interpreted_Data_Product output) could be separated out using an extension point on the Sensor Data Interpretation component.

Exploitation Considerations

- Sensor Data Interpretation is responsible for coordinating Interpretation_Resources, but not for capturing their inputs (i.e. it sets up Data_Interpretation_Solutions using Interpretation_Resources to act on Sensor_Data_Product, but it does not handle Sensor_Data_Product directly).
- Sensor Data Interpretation can use a Sensor_Data_Product's Metadata to adapt Data_Interpretation_Solutions during their planning and execution to optimise effectiveness. This will require tight coordination with the components providing the Sensor_Data_Product or acting as Interpretation_Resources, and may require the use of feedback loops between those components and Sensor Data Interpretation.
- Sensor Data Interpretation is able to coordinate Interpretation_Resources to extract meaning from stored (as well as 'live') Sensor_Data_Products.
- Sensor Data Interpretation may raise a dependency against Data_Interpretation_Solutions for the system to obtain Sensor_Data_Products of a given type and quality. Either to capture a new Sensor_Data_Product or to improve the quality of Sensor_Data_Products being used by the Data_Interpretation_Solution in progress.

5.4.2.51.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

Failure of this component could result an object being misidentified, potentially resulting in a lethal attack on the object. Functions causing incorrect targeting of stores are required to be DAL B. However, in this case it is expected that either:

- Sensor data unaffected by this component (e.g. video of the target) is analysed by a human prior to authorising an attack.
- A human has pre-authorised that attacks can be prosecuted when objects meeting certain criteria are detected.

Based on the additional checks by a human, it is considered that DAL C is appropriate for this component.

5.4.2.51.6.4 Security Considerations

The indicative security classification is SNEO.

This component is responsible for managing the interpretation of available Sensor_Data_Products, the content of which is considered SNEO during a mission and will likely coordinate the use of SNEO algorithms. The component is one of a group that will maintain the integrity and availability of the sensor data and its use. The integrity and availability needs appropriate protection to prevent interference leading to incorrect identification of objects and therefore incorrect targeting and engagement of friendly or neutral entities.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of extracted information produced during the mission. Such information may be used to make tactical decisions, e.g. selection of a chosen target from a set of possible target options.
- **Supporting Secure Remote Operation** of autonomous decision making based on the extracted meaning of sensed data.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.51.7 Services

5.4.2.51.7.1 Service Definitions

5.4.2.51.7.1.1 Interpretation_Requirement

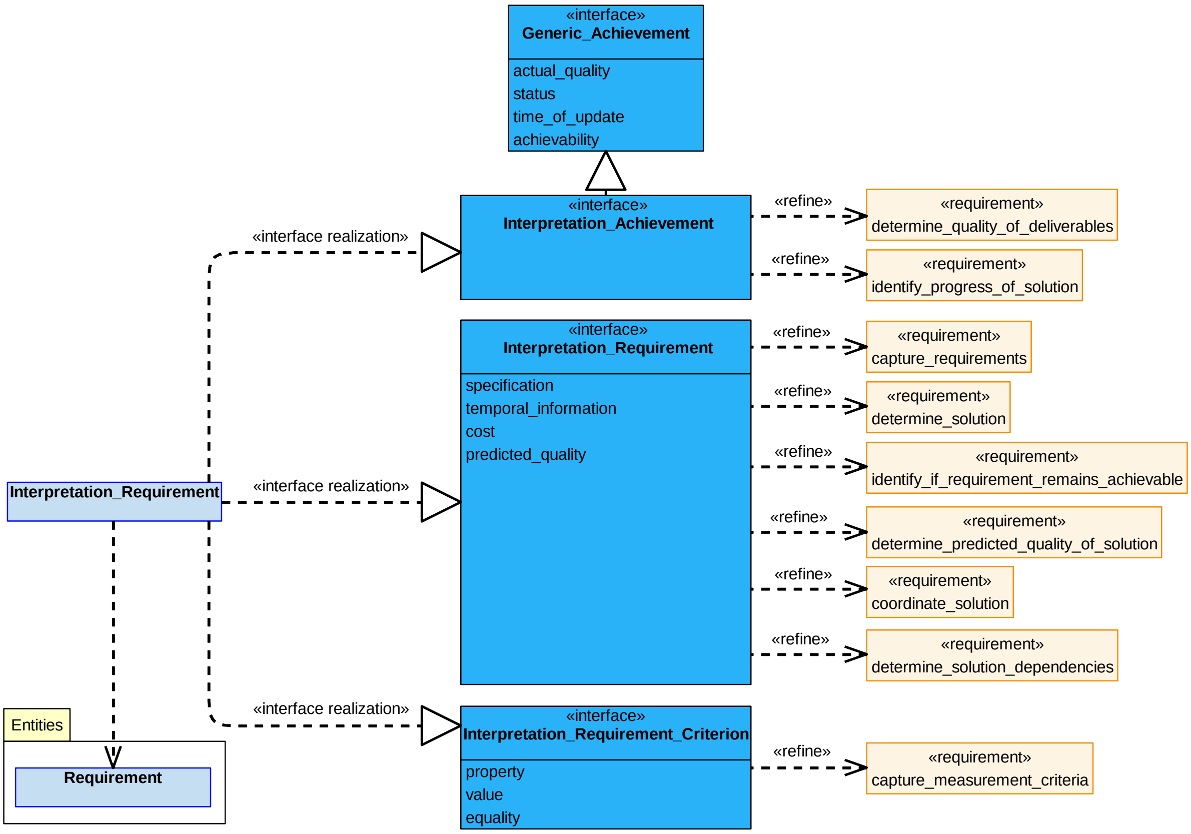


Figure 886: Interpretation_Requirement Service Definition

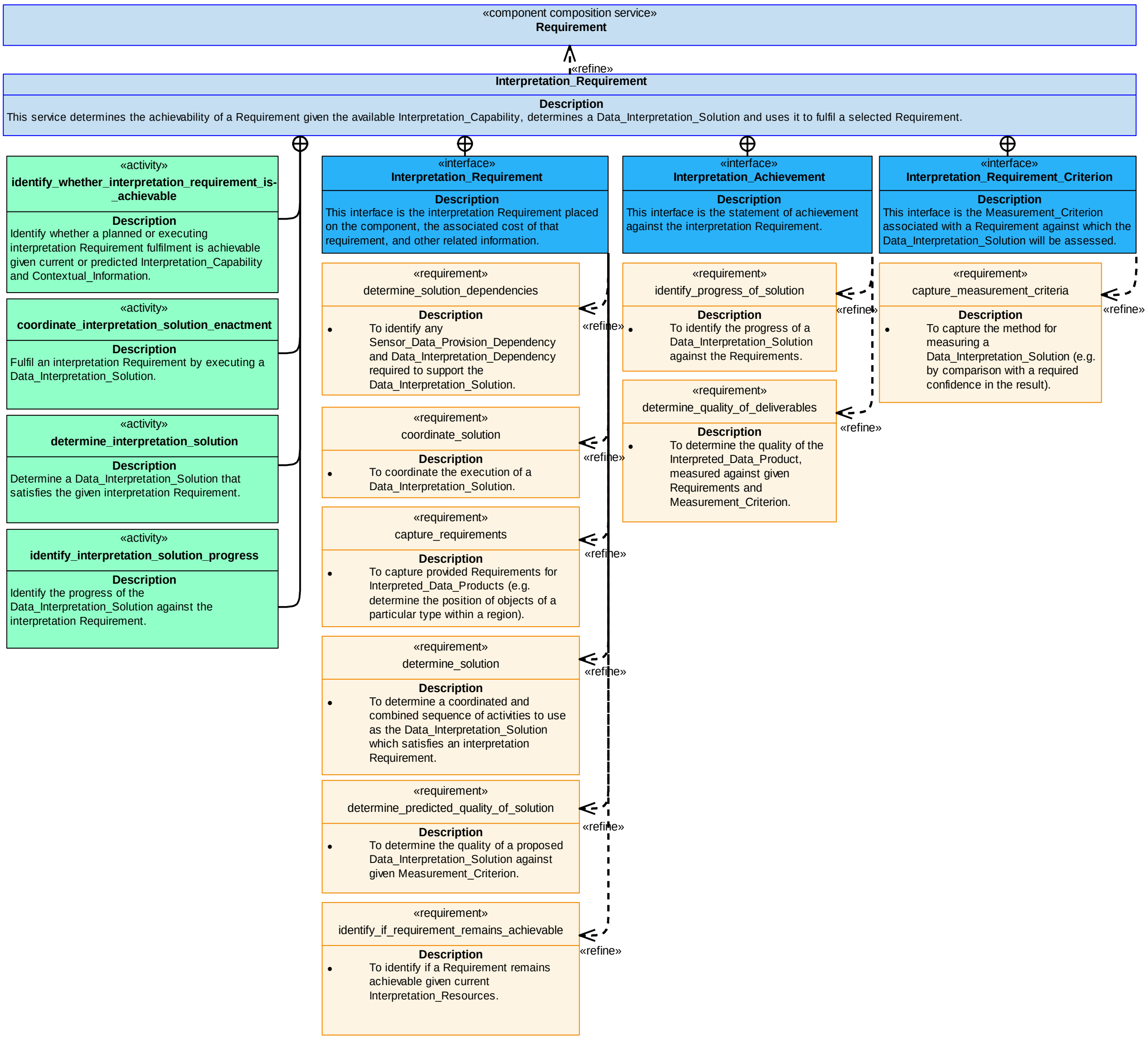

«component composition service»
Requirement
«refine»
Interpretation_Requirement
Description
This service determines the achievability of a Requirement given the available Interpretation_Capability, determines a Data_Interpretation_Solution and uses it to fulfil a selected Requirement.
«activity»
identify_whether_interpretation_requirement_is_achievable
Description
Identify whether a planned or executing interpretation Requirement fulfilment is achievable given current or predicted Interpretation_Capability and Contextual_Information.
«activity»
coordinate_interpretation_solution_enactment
Description
Fulfil an interpretation Requirement by executing a Data_Interpretation_Solution.
«activity»
determine_interpretation_solution
Description
Determine a Data_Interpretation_Solution that satisfies the given interpretation Requirement.
«activity»
identify_interpretation_solution_progress
Description
Identify the progress of the Data_Interpretation_Solution against the interpretation Requirement.
«interface»
Interpretation_Requirement
Description
This interface is the interpretation Requirement placed on the component, the associated cost of that requirement, and other related information.
«interface»
Interpretation_Achievement
Description
This interface is the statement of achievement against the interpretation Requirement.
«interface»
Interpretation_Requirement_Criterion
Description
This interface is the Measurement_Criterion associated with a Requirement against which the Data_Interpretation_Solution will be assessed.
«requirement»
determine_solution_dependencies
Description
To identify any Sensor_Data_Provision_Dependency and Data_Interpretation_Dependency required to support the Data_Interpretation_Solution.
«refine»
«requirement»
coordinate_solution
Description
To coordinate the execution of a Data_Interpretation_Solution.
«refine»
«requirement»
capture_requirements
Description
To capture provided Requirements for Interpreted_Data_Products (e.g. determine the position of objects of a particular type within a region).
«refine»
«requirement»
determine_solution
Description
To determine a coordinated and combined sequence of activities to use as the Data_Interpretation_Solution which satisfies an interpretation Requirement.
«refine»
«requirement»
determine_predicted_quality_of_solution
Description
To determine the quality of a proposed Data_Interpretation_Solution against given Measurement_Criterion.
«refine»
«requirement»
identify_if_requirement_remains_achievable
Description
To identify if a Requirement remains achievable given current Interpretation_Resources.
«refine»
«requirement»
identify_progress_of_solution
Description
To identify the progress of a Data_Interpretation_Solution against the Requirements.
«refine»
«requirement»
determine_quality_of_deliverables
Description
To determine the quality of the Interpreted_Data_Product, measured against given Requirements and Measurement_Criterion.
«refine»
«requirement»
capture_measurement_criteria
Description
To capture the method for measuring a Data_Interpretation_Solution (e.g. by comparison with a required confidence in the result).
«refine»

Figure 887: Interpretation_Requirement Service Policy

Interpretation_Requirement

This service determines the achievability of a Requirement given the available Interpretation_Capability, determines a Data_Interpretation_Solution and uses it to fulfil a selected Requirement.

Interfaces

Interpretation_Requirement_Criterion

This interface is the Measurement_Criterion associated with a Requirement against which the Data_Interpretation_Solution will be assessed.

Attributes

property	The criterion property to be measured (e.g. a specific quantity or cost, such as processing time in milliseconds).
value	The amount related to the property to be measured, e.g. 60 milliseconds.

equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Interpretation_Requirement

This interface is the interpretation Requirement placed on the component, the associated cost of that requirement, and other related information.

Attributes

specification	The scope of the interpretation action required (e.g. locate enemy tanks in a region of territory, or provide human readable recon images enhanced from SAR data).
temporal_information	Information governing when the interpretation Requirement is to be carried out, such as a starting point (when a particular input is available), or end point (to continue until a particular time or other trigger is reached).
cost	The cost of executing the Data_Interpretation_Solution, for example, processing time taken.
predicted_quality	How well a proposed Data_Interpretation_Solution is predicted to satisfy the Requirement.

Interpretation_Achievement

This interface is the statement of achievement against the interpretation Requirement.

Activities

identify_whether_interpretation_requirement_is_achievable

Identify whether a planned or executing interpretation Requirement fulfilment is achievable given current or predicted Interpretation_Capability and Contextual_Information.

coordinate_interpretation_solution_enactment

Fulfil an interpretation Requirement by executing a Data_Interpretation_Solution.

determine_interpretation_solution

Determine a Data_Interpretation_Solution that satisfies the given interpretation Requirement.

identify_interpretation_solution_progress

Identify the progress of the Data_Interpretation_Solution against the interpretation Requirement.

5.4.2.51.7.1.2 Data_Processing_Dependency

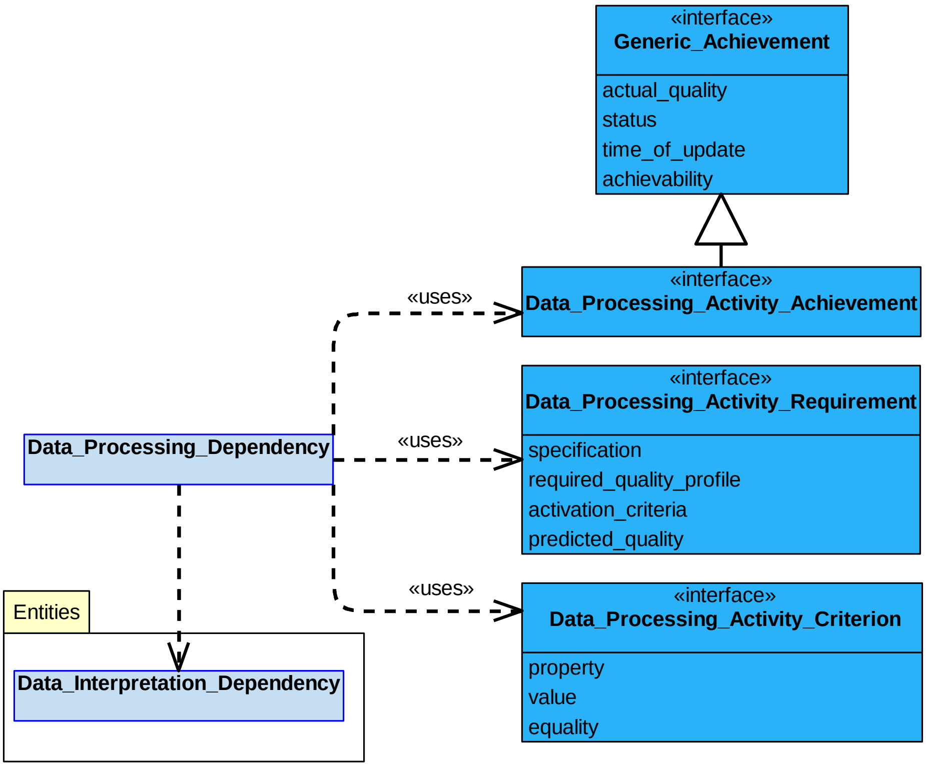


Figure 888: Data_Processing_Dependency Service Definition

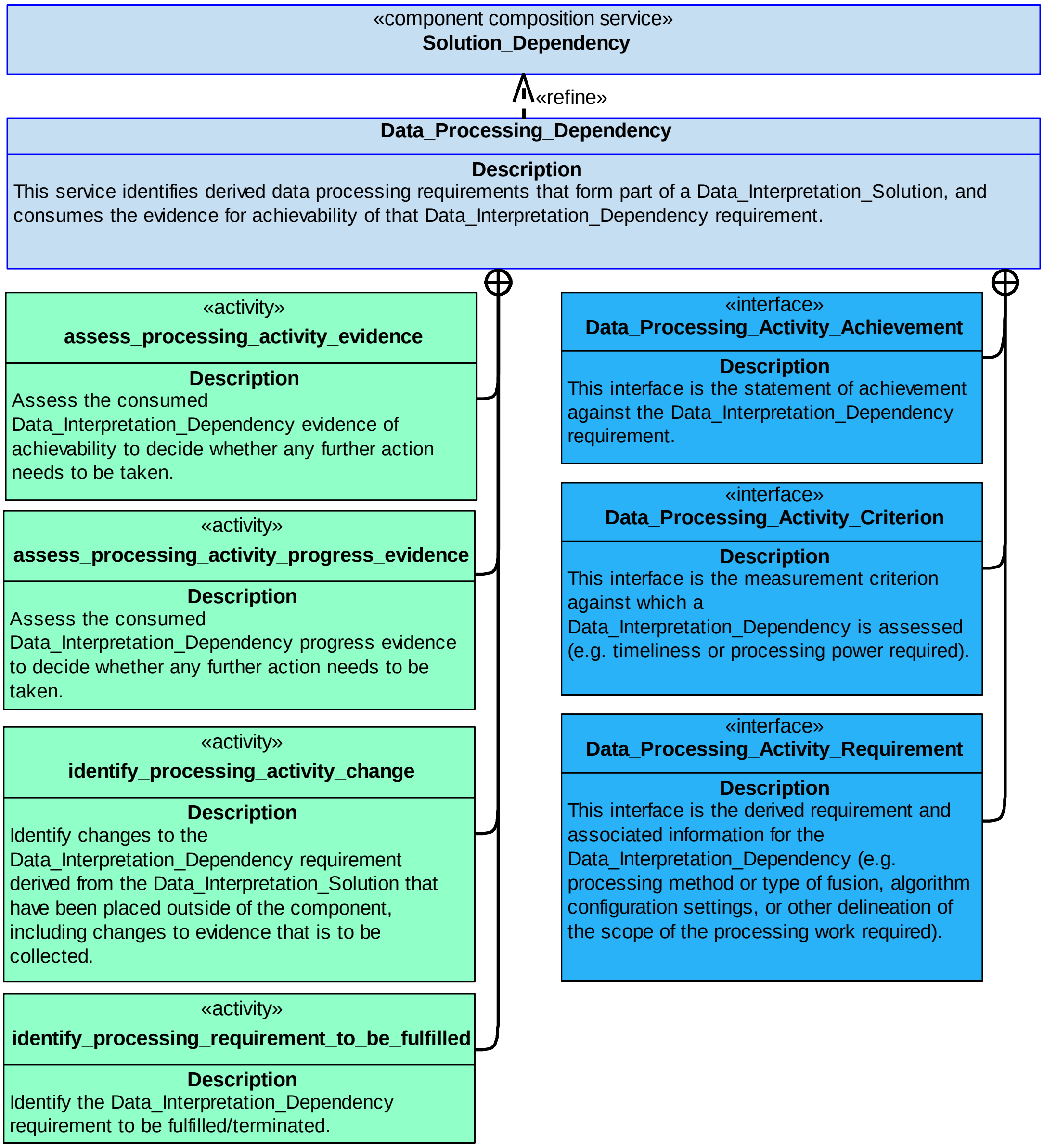

«component composition service»
Solution_Dependency
«refine»
Data_Processing_Dependency
Description
This service identifies derived data processing requirements that form part of a Data_Interpretation_Solution, and consumes the evidence for achievability of that Data_Interpretation_Dependency requirement.
«activity»
assess_processing_activity_evidence
Description
Assess the consumed Data_Interpretation_Dependency evidence of achievability to decide whether any further action needs to be taken.
«activity»
assess_processing_activity_progress_evidence
Description
Assess the consumed Data_Interpretation_Dependency progress evidence to decide whether any further action needs to be taken.
«activity»
identify_processing_activity_change
Description
Identify changes to the Data_Interpretation_Dependency requirement derived from the Data_Interpretation_Solution that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
identify_processing_requirement_to_be_fulfilled
Description
Identify the Data_Interpretation_Dependency requirement to be fulfilled/terminated.
«interface»
Data_Processing_Activity_Achievement
Description
This interface is the statement of achievement against the Data_Interpretation_Dependency requirement.
«interface»
Data_Processing_Activity_Criterion
Description
This interface is the measurement criterion against which a Data_Interpretation_Dependency is assessed (e.g. timeliness or processing power required).
«interface»
Data_Processing_Activity_Requirement
Description
This interface is the derived requirement and associated information for the Data_Interpretation_Dependency (e.g. processing method or type of fusion, algorithm configuration settings, or other delineation of the scope of the processing work required).

Figure 889: Data_Processing_Dependency Service Policy

Data_Processing_Dependency

This service identifies derived data processing requirements that form part of a Data_Interpretation_Solution, and consumes the evidence for achievability of that Data_Interpretation_Dependency requirement.

Interfaces

Data_Processing_Activity_Achievement

This interface is the statement of achievement against the Data_Interpretation_Dependency requirement.

Data_Processing_Activity_Criterion

This interface is the measurement criterion against which a Data_Interpretation_Dependency is assessed (e.g. timeliness or processing power required).

Attributes

property	The criterion property to be measured (e.g. a cost, such as time taken for a data processing activity).
value	The amount related to the property to be measured, e.g. 15 milliseconds.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Data_Processing_Activity_Requirement

This interface is the derived requirement and associated information for the Data_Interpretation_Dependency (e.g. processing method or type of fusion, algorithm configuration settings, or other delineation of the scope of the processing work required).

Attributes

specification	Scope and detail of the Data_Interpretation_Dependency technique needed, for example, object detections, signal processing, image sharpening or fusion algorithm.
required_quality_profile	Precision and accuracy required for the Data_Interpretation_Dependency for example, specific confidence levels, resolution of output, granularity of object detections.
activation_criteria	Trigger for initiating and ceasing the Data_Interpretation_Dependency, e.g. how a pattern matching algorithm will be triggered and the criteria for determining how long it should run for.
predicted_quality	How well the data processing dependency is predicted to satisfy the requirement for the Data_Interpretation_Dependency.

Activities

assess_processing_activity_evidence

Assess the consumed Data_Interpretation_Dependency evidence of achievability to decide whether any further action needs to be taken.

assess_processing_activity_progress_evidence

Assess the consumed Data_Interpretation_Dependency progress evidence to decide whether any further action needs to be taken.

identify_processing_activity_change

Identify changes to the Data_Interpretation_Dependency requirement derived from the Data_Interpretation_Solution that have been placed outside of the component, including changes to evidence that is to be collected.

identify_processing_requirement_to_be_fulfilled

Identify the Data_Interpretation_Dependency requirement to be fulfilled/terminated.

5.4.2.51.7.1.3 Data_Provision_Dependency

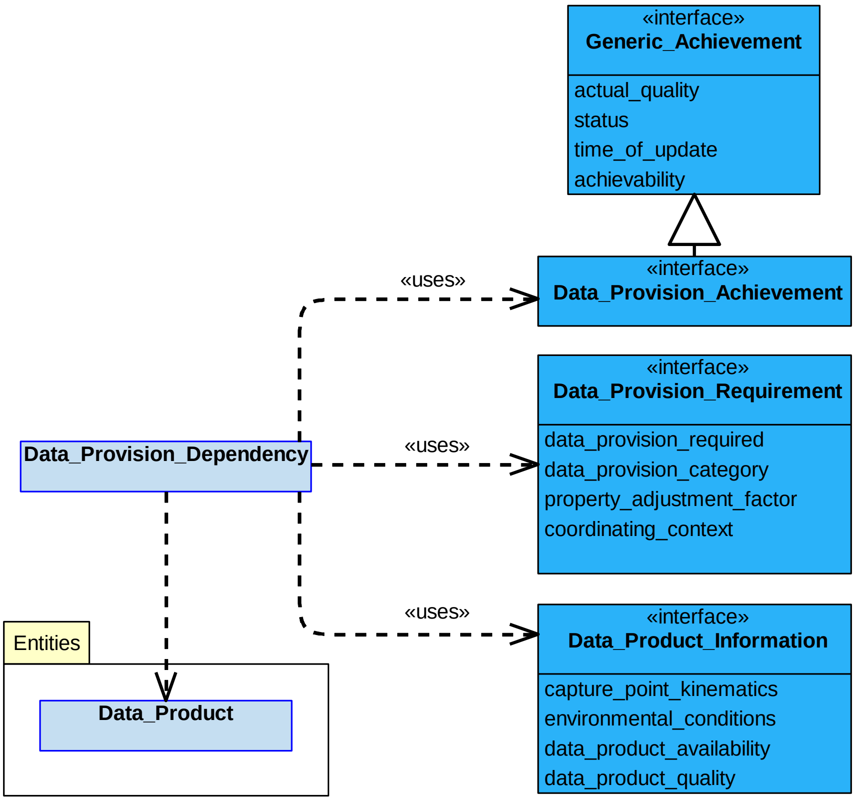


Figure 890: Data_Provision_Dependency Service Definition

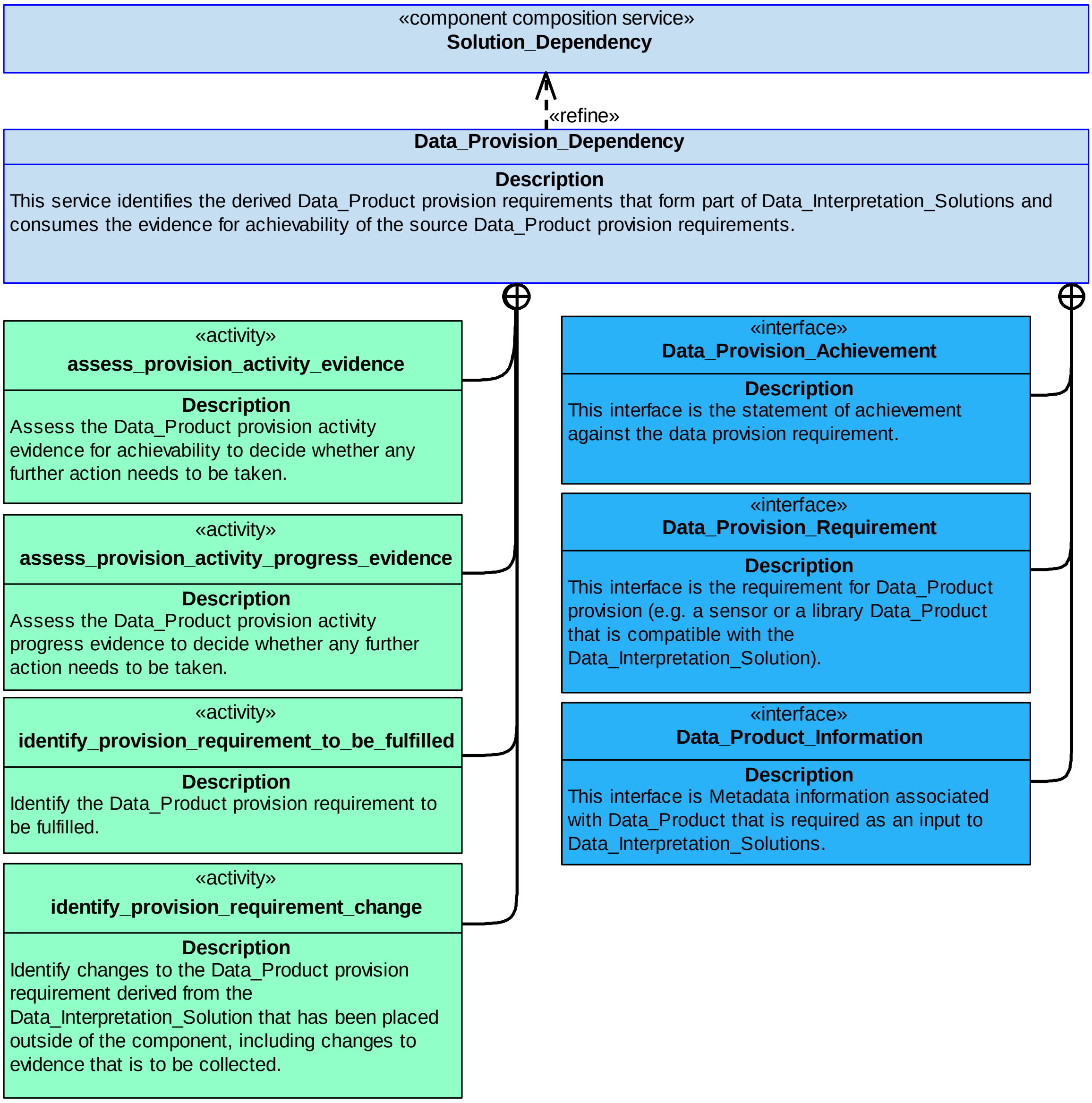

«component composition service»
Solution_Dependency
«refine»
Data_Provision_Dependency
Description
This service identifies the derived Data_Product provision requirements that form part of Data_Interpretation_Solutions and consumes the evidence for achievability of the source Data_Product provision requirements.
«activity»
assess_provision_activity_evidence
Description
Assess the Data_Product provision activity evidence for achievability to decide whether any further action needs to be taken.
«activity»
assess_provision_activity_progress_evidence
Description
Assess the Data_Product provision activity progress evidence to decide whether any further action needs to be taken.
«activity»
identify_provision_requirement_to_be_fulfilled
Description
Identify the Data_Product provision requirement to be fulfilled.
«activity»
identify_provision_requirement_change
Description
Identify changes to the Data_Product provision requirement derived from the Data_Interpretation_Solution that has been placed outside of the component, including changes to evidence that is to be collected.
«interface»
Data_Provision_Achievement
Description
This interface is the statement of achievement against the data provision requirement.
«interface»
Data_Provision_Requirement
Description
This interface is the requirement for Data_Product provision (e.g. a sensor or a library Data_Product that is compatible with the Data_Interpretation_Solution).
«interface»
Data_Product_Information
Description
This interface is Metadata information associated with Data_Product that is required as an input to Data_Interpretation_Solutions.

Figure 891: Data_Provision_Dependency Service Policy

Data_Provision_Dependency

This service identifies the derived Data_Product provision requirements that form part of Data_Interpretation_Solutions and consumes the evidence for achievability of the source Data_Product provision requirements.

Interfaces

Data_Provision_Achievement

This interface is the statement of achievement against the data provision requirement.

Data_Provision_Requirement

This interface is the requirement for Data_Product provision (e.g. a sensor or a library Data_Product that is compatible with the Data_Interpretation_Solution).

Attributes

data_provision_required	A required Data_Product provision of a specified type and quality, e.g. a type of Sensor_Data_Product or data sourced from a library.
data_provision_category	Category of source data provision or technique whose interpretive element will yield better results with improved Sensor_Data_Product capture.
property_adjustment_factor	The adjustment that is required to be achieved for any Data_Product corresponding to the defined data_provision_category. The adjustment would generally be achieved through the acquisition of sensor measurements, using suitable settings.
coordinating_context	Specific Data_Product details that are either required or can be expected through the sensor measurement acquisition process (e.g. details that influence or are influenced by the angles from which measurements are made or the lighting conditions when measurements are made). This can include different details corresponding to different coordination points within the overall sensor measurement acquisition process (e.g. the details will differ based on different route legs and changes to sensor settings).

Data_Product_Information

This interface is Metadata information associated with Data_Product that is required as an input to Data_Interpretation_Solutions.

Attributes

capture_point_kinematics	Spatial location, orientation and velocity of sensor platforms and sensors at the time of the Data_Product capture.
environmental_conditions	Environmental conditions applicable when the Data_Product was captured that may affect the Data_Interpretation_Solution.
data_product_availability	The availability of the Data_Product, e.g. the start and end times of existing library data.
data_product_quality	The quality of the Data_Product, e.g. accuracy, precision and confidence.

Activities

assess_provision_activity_evidence

Assess the Data_Product provision activity evidence for achievability to decide whether any further action needs to be taken.

assess_provision_activity_progress_evidence

Assess the Data_Product provision activity progress evidence to decide whether any further action needs to be taken.

identify_provision_requirement_change

Identify changes to the Data_Product provision requirement derived from the Data_Interpretation_Solution that has been placed outside of the component, including changes to evidence that is to be collected.

identify_provision_requirement_to_be_fulfilled

Identify the Data_Product provision requirement to be fulfilled.

5.4.2.51.7.1.4 Environmental_Information

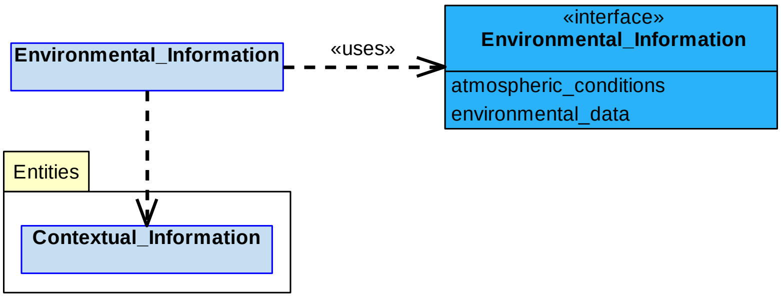


Figure 892: Information_Dependency Service Definition

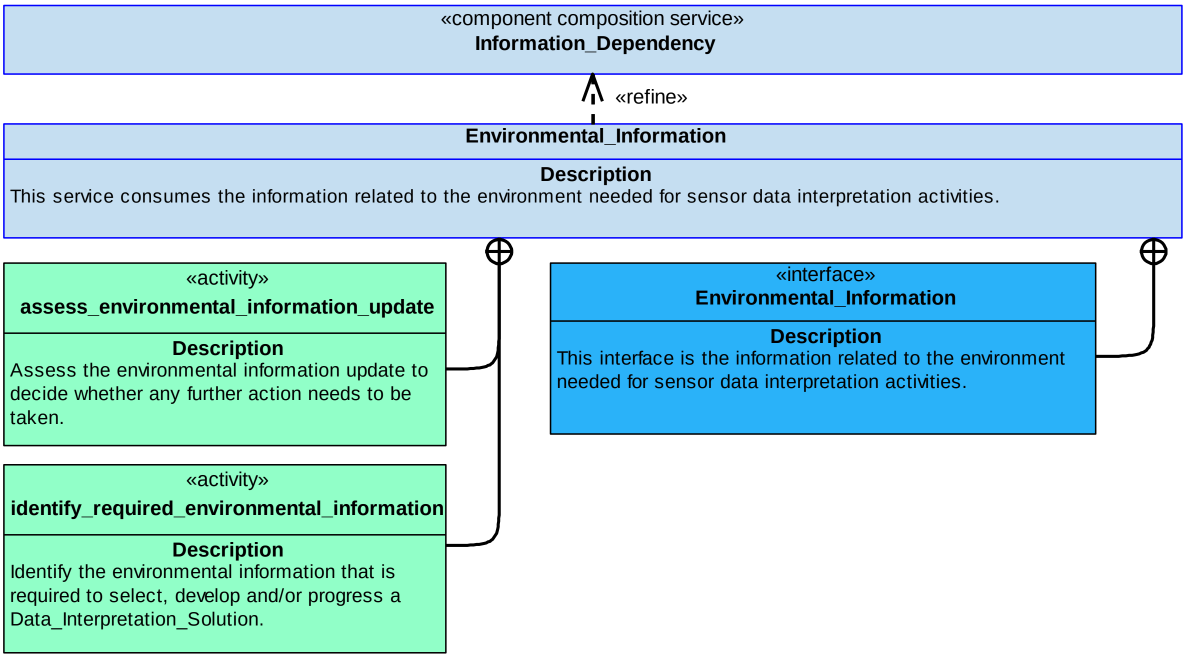

«component composition service»
Information_Dependency
«refine»
Environmental_Information
Description
This service consumes the information related to the environment needed for sensor data interpretation activities.
«activity»
assess_environmental_information_update
Description
Assess the environmental information update to decide whether any further action needs to be taken.
«activity»
identify_required_environmental_information
Description
Identify the environmental information that is required to select, develop and/or progress a Data_Interpretation_Solution.
«interface»
Environmental_Information
Description
This interface is the information related to the environment needed for sensor data interpretation activities.

Figure 893: Information_Dependency Service Policy

Environmental_Information

This service consumes the information related to the environment needed for sensor data interpretation activities.

Interface

Environmental_Information

This interface is the information related to the environment needed for sensor data interpretation activities.

Attributes

atmospheric_conditions	Current and predicted weather conditions and features.
environmental_data	Information describing surfaces and features in the environment.

Activities

assess_environmental_information_update

Assess the environmental information update to decide whether any further action needs to be taken.

identify_required_environmental_information

Identify the environmental information that is required to select, develop and/or progress a Data_Interpretation_Solution.

5.4.2.51.7.1.5 Sensor_Platform_Information

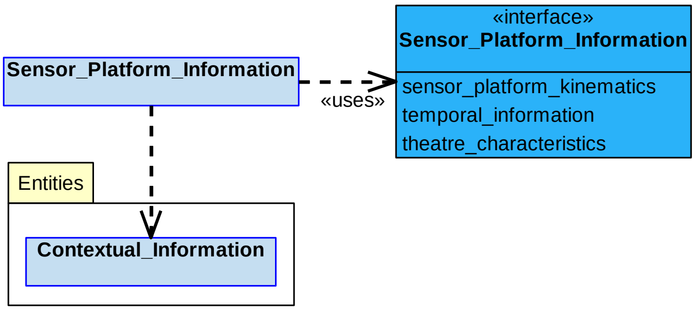


Figure 894: Sensor_Platform_Information Service Definition

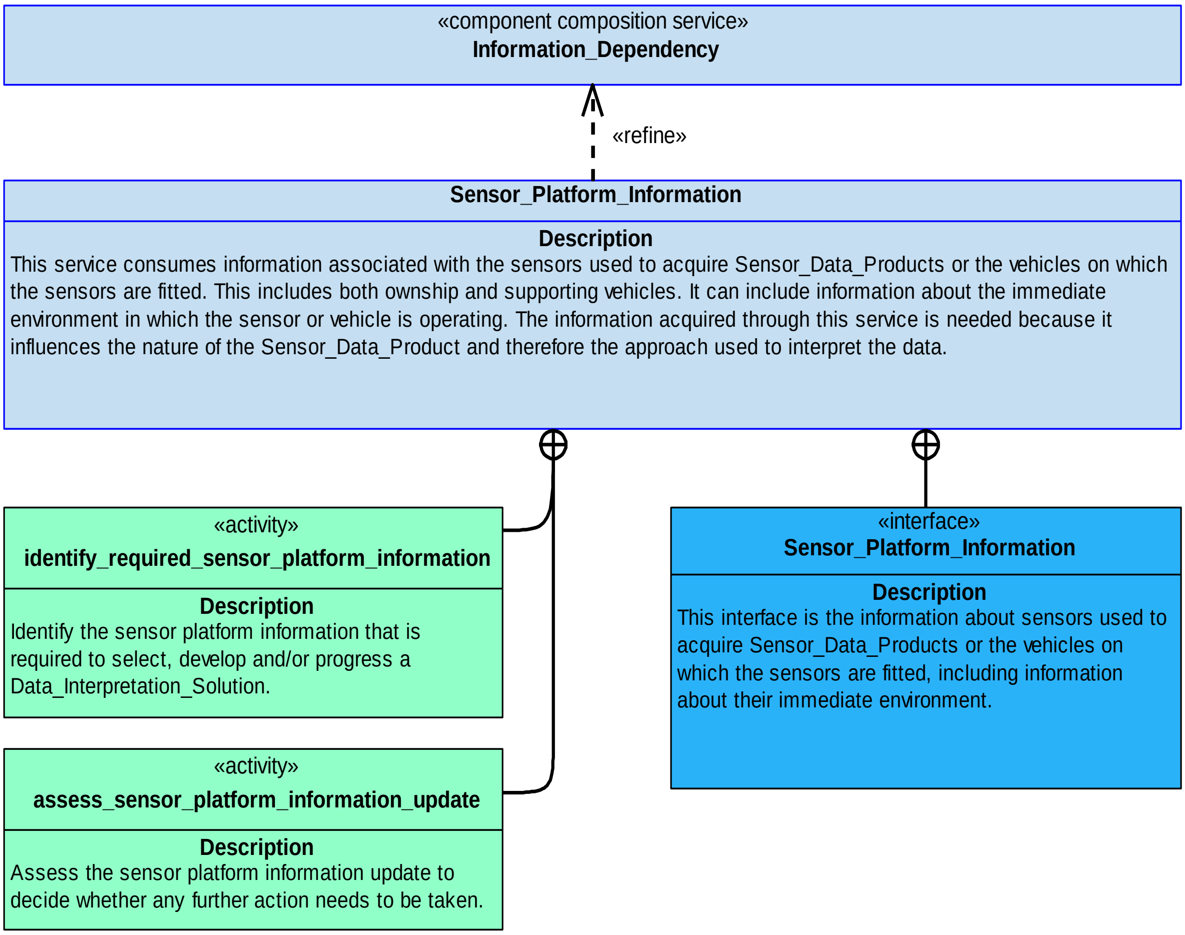

«component composition service»
Information_Dependency
«refine»
Sensor_Platform_Information
Description
This service consumes information associated with the sensors used to acquire Sensor_Data_Products or the vehicles on which the sensors are fitted. This includes both ownship and supporting vehicles. It can include information about the immediate environment in which the sensor or vehicle is operating. The information acquired through this service is needed because it influences the nature of the Sensor_Data_Product and therefore the approach used to interpret the data.
«activity»
identify_required_sensor_platform_information
Description
Identify the sensor platform information that is required to select, develop and/or progress a Data_Interpretation_Solution.
«activity»
assess_sensor_platform_information_update
Description
Assess the sensor platform information update to decide whether any further action needs to be taken.
«interface»
Sensor_Platform_Information
Description
This interface is the information about sensors used to acquire Sensor_Data_Products or the vehicles on which the sensors are fitted, including information about their immediate environment.

Figure 895: Sensor_Platform_Information Service Policy

Sensor_Platform_Information

This service consumes information associated with the sensors used to acquire Sensor_Data_Products or the vehicles on which the sensors are fitted. This includes both ownship and supporting vehicles. It can include information about the immediate environment in which the sensor or vehicle is operating. The information acquired through this service is needed because it influences the nature of the Sensor_Data_Product and therefore the approach used to interpret the data.

Interface

Sensor_Platform_Information

This interface is the information about sensors used to acquire Sensor_Data_Products or the vehicles on which the sensors are fitted, including information about their immediate environment.

Attributes

sensor_platform_kinematics	Location, orientation, velocity and acceleration of sensor platform(s) providing Sensor_Data_Products.

temporal_information	Timing of supporting platform availability for new Sensor_Data_Product capture and low latency synchronisation of sensing product processing techniques.
theatre_characteristics	Known environment characteristics that may be important in Data_Interpretation_Solution planning, such as filtering out a background of non-target objects like civilian vehicles.

Activities

identify_required_sensor_platform_information

Identify the sensor platform information that is required to select, develop and/or progress a Data_Interpretation_Solution.

assess_sensor_platform_information_update

Assess the sensor platform information update to decide whether any further action needs to be taken.

5.4.2.51.7.1.6 Interpretation_Capability

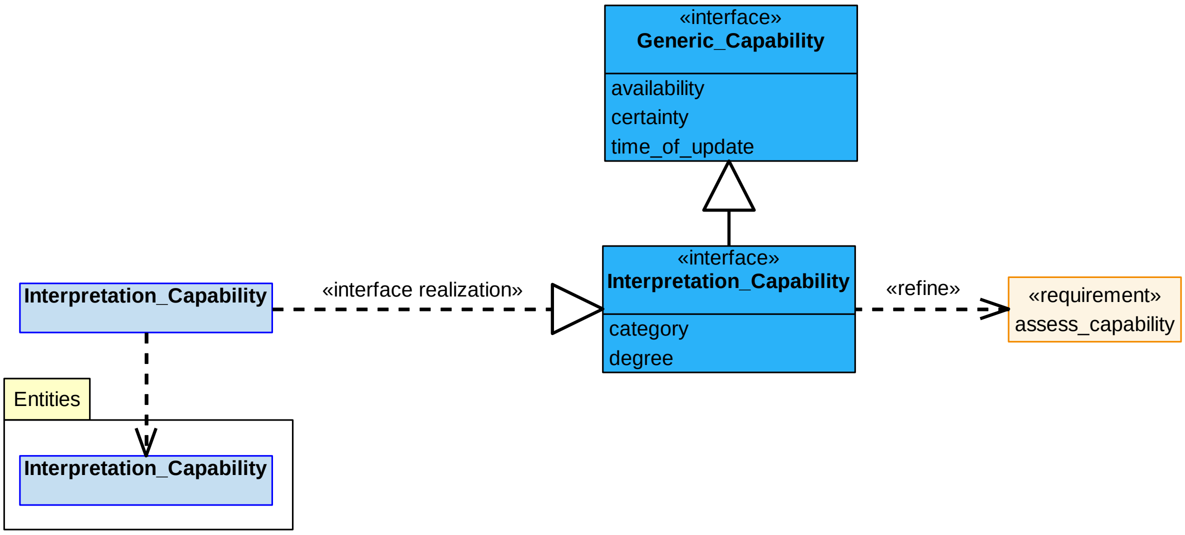


Figure 896: Interpretation_Capability Service Definition

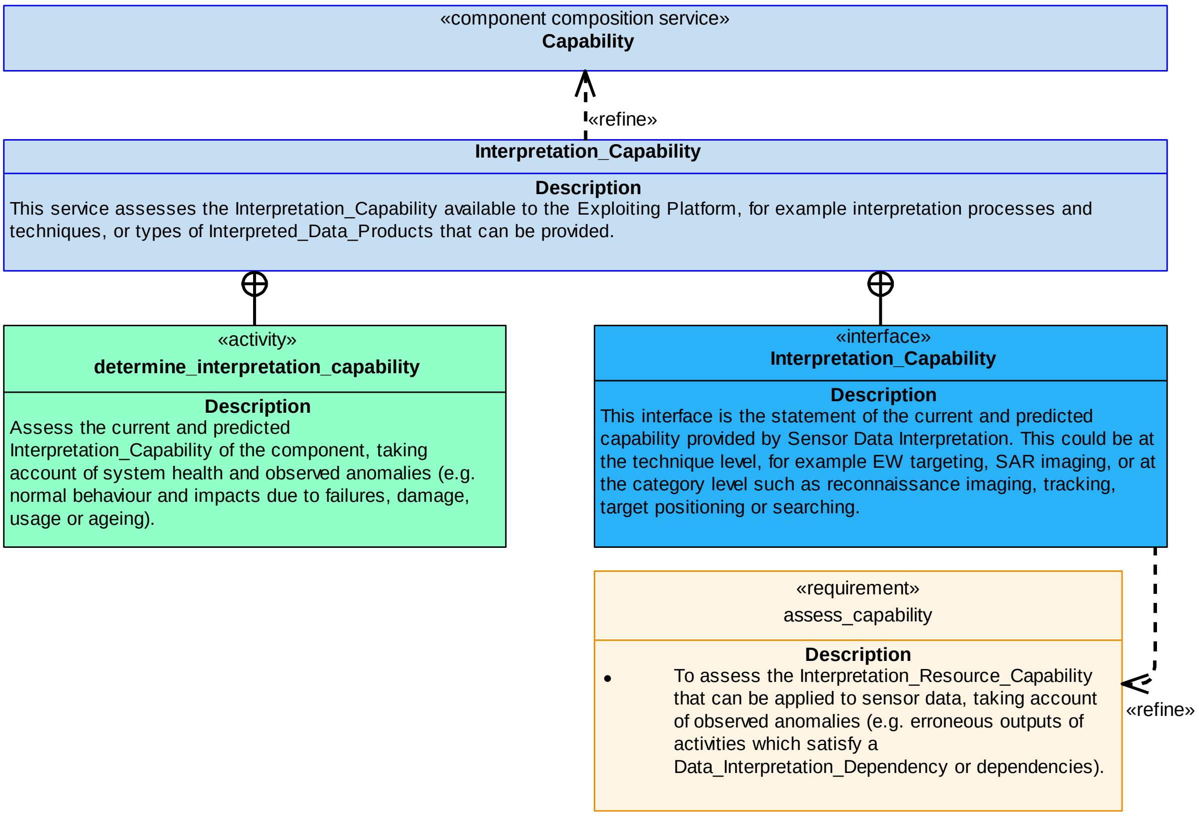

«component composition service»
Capability
«refine»
Interpretation_Capability
Description
This service assesses the Interpretation_Capability available to the Exploiting Platform, for example interpretation processes and techniques, or types of Interpreted_Data_Products that can be provided.
«activity»
determine_interpretation_capability
Description
Assess the current and predicted Interpretation_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Interpretation_Capability
Description
This interface is the statement of the current and predicted capability provided by Sensor Data Interpretation. This could be at the technique level, for example EW targeting, SAR imaging, or at the category level such as reconnaissance imaging, tracking, target positioning or searching.
«requirement»
assess_capability
Description
To assess the Interpretation_Resource_Capability that can be applied to sensor data, taking account of observed anomalies (e.g. erroneous outputs of activities which satisfy a Data_Interpretation_Dependency or dependencies).
«refine»

Figure 897: Interpretation_Capability Service Policy

Interpretation_Capability

This service assesses the Interpretation_Capability available to the Exploiting Platform, for example interpretation processes and techniques, or types of Interpreted_Data_Products that can be provided.

Interface

Interpretation_Capability

This interface is the statement of the current and predicted capability provided by Sensor Data Interpretation. This could be at the technique level, for example EW targeting, SAR imaging, or at the category level such as reconnaissance imaging, tracking, target positioning or searching.

Attributes

category	A type of Interpretation_Capability that can be provided.
degree	The level of performance or effectiveness that can be achieved for an available category of Interpretation_Capability.

Activity

determine_interpretation_capability

Assess the current and predicted Interpretation_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.51.7.1.7 Capability_Evidence

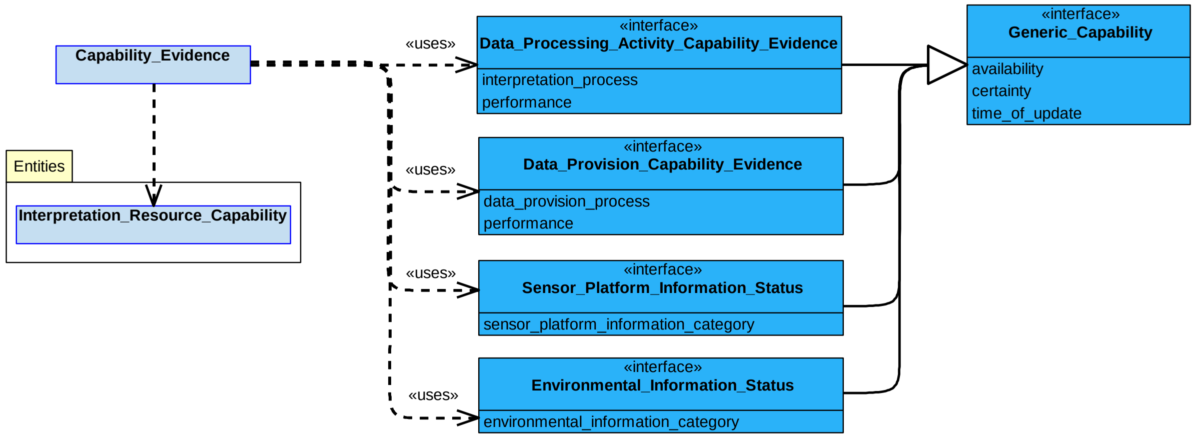
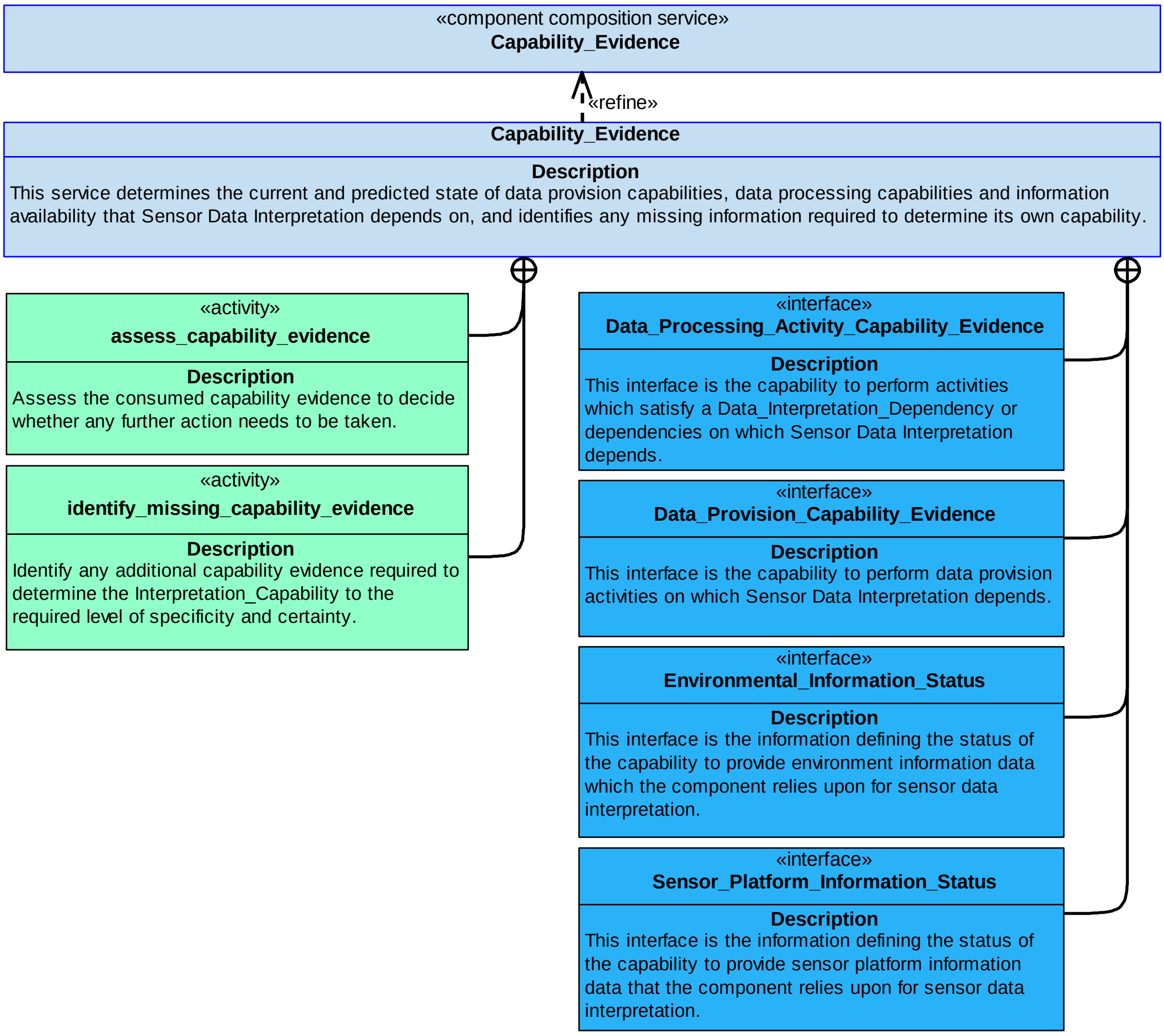

Capability_Evidence
Entities
Interpretation_Resource_Capability
«uses»
«uses»
«uses»
«uses»
«interface»
Data_Processing_Activity_Capability_Evidence
interpretation_process
performance
«interface»
Data_Provision_Capability_Evidence
data_provision_process
performance
«interface»
Sensor_Platform_Information_Status
sensor_platform_information_category
«interface»
Environmental_Information_Status
environmental_information_category
«interface»
Generic_Capability
availability
certainty
time_of_update

Figure 898: Capability_Evidence Service Definition



Description

This service determines the current and predicted state of data provision capabilities, data processing capabilities and information availability that Sensor Data Interpretation depends on, and identifies any missing information required to determine its own capability.

«activity»
assess_capability_evidence

Description

Assess the consumed capability evidence to decide whether any further action needs to be taken.

«activity»
identify_missing_capability_evidence

Description

Identify any additional capability evidence required to determine the Interpretation_Capability to the required level of specificity and certainty.

«interface»
Data_Processing_Activity_Capability_Evidence

Description

This interface is the capability to perform activities which satisfy a Data_Interpretation_Dependency or dependencies on which Sensor Data Interpretation depends.

«interface»
Data_Provision_Capability_Evidence

Description

This interface is the capability to perform data provision activities on which Sensor Data Interpretation depends.

«interface»
Environmental_Information_Status

Description

This interface is the information defining the status of the capability to provide environment information data which the component relies upon for sensor data interpretation.

«interface»
Sensor_Platform_Information_Status

Description

This interface is the information defining the status of the capability to provide sensor platform information data that the component relies upon for sensor data interpretation.

Figure 899: Capability_Evidence Service Policy

Capability_Evidence

This service determines the current and predicted state of data provision capabilities, data processing capabilities and information availability that Sensor Data Interpretation depends on, and identifies any missing information required to determine its own capability.

Interfaces

Data_Processing_Activity_Capability_Evidence

This interface is the capability to perform activities which satisfy a Data_Interpretation_Dependency or dependencies on which Sensor Data Interpretation depends.

Attributes

interpretation_process	The specific processing algorithm or technique about which a capability statement is being made. For example, track association algorithms, signal processing functions and the required parameters for successful operation.
performance	The level or degree of capability available for the associated Interpretation_Resource_Capability, taking into account the algorithms and processing power installed.

Data_Provision_Capability_Evidence

This interface is the capability to perform data provision activities on which Sensor Data Interpretation depends.

Attributes

data_provision_process	The specific data type or data source about which capability statements are being made. For example, capability for information gathering (e.g. image gathering or electronic surveillance gathering).
performance	The level or degree of capability available for this particular Sensor_Data_Provision_Dependency, taking into account the data provision resources installed.

Sensor_Platform_Information_Status

This interface is the information defining the status of the capability to provide sensor platform information data that the component relies upon for sensor data interpretation.

Attribute

sensor_platform_information_category	The category of sensor platform information used in the interpretation of sensor data.

Environmental_Information_Status

This interface is the information defining the status of the capability to provide environment information data which the component relies upon for sensor data interpretation.

Attribute

environmental_information_category	The category of environmental information used in the interpretation of sensor data.

Activities

assess_capability_evidence

Assess the consumed capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any additional capability evidence required to determine the Interpretation_Capability to the required level of specificity and certainty.

5.4.2.51.7.2 Service Dependencies

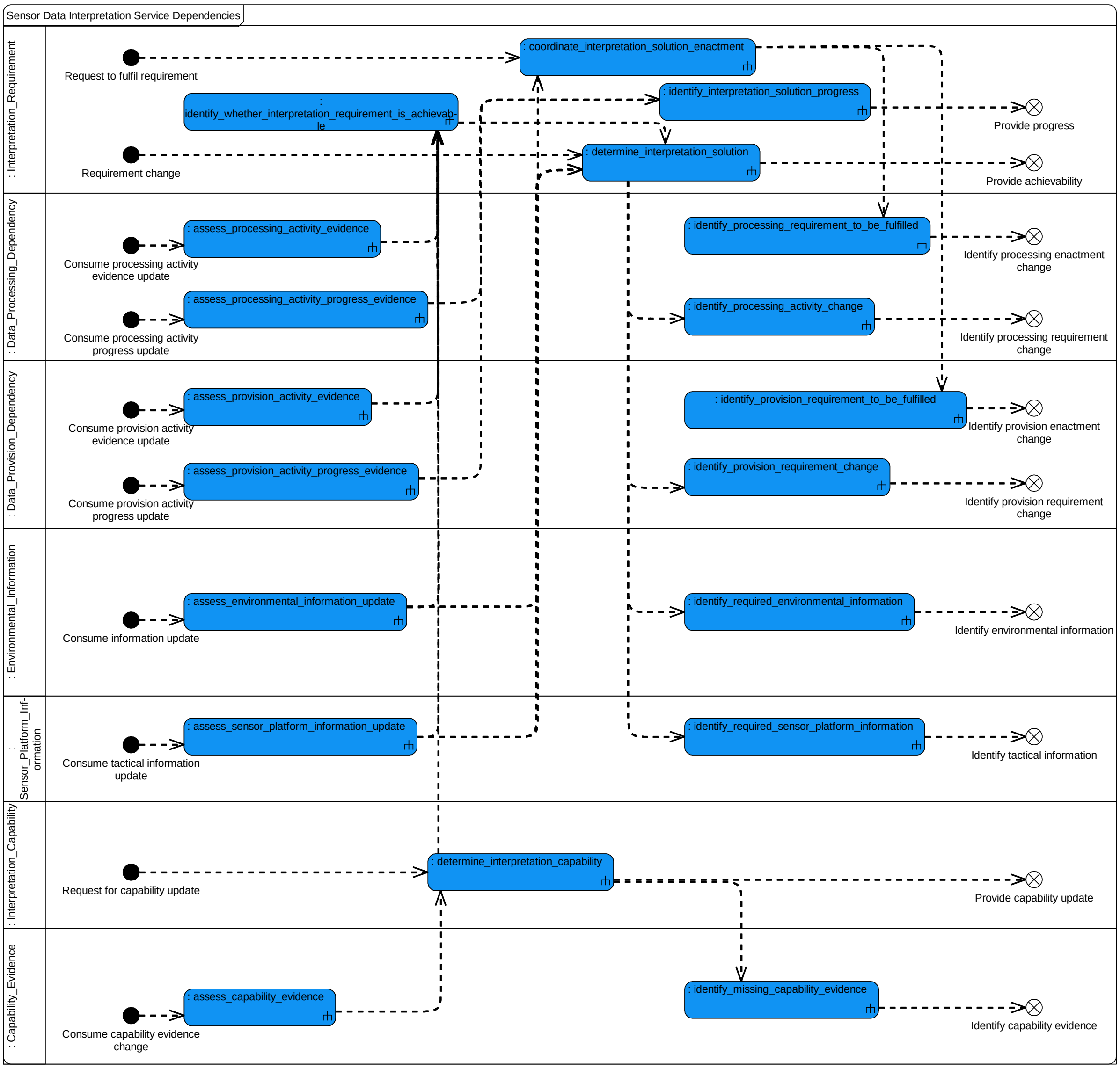

Sensor Data Interpretation Service Dependencies
: Interpretation_Requirement
: Data_Processing_Dependency
: Data_Provision_Dependency
: Environmental_Information
: Sensor_Platform_Inf-ormation
: Interpretation_Capability
: Capability_Evidence
Request to fulfil requirement
: coordinate_interpretation_solution_enactment
: identify_whether_interpretation_requirement_is_achievab-le
: identify_interpretation_solution_progress
Provide progress
Requirement change
: determine_interpretation_solution
Provide achievability
: assess_processing_activity_evidence
Consume processing activity evidence update
: identify_processing_requirement_to_be_fulfilled
Identify processing enactment change
: assess_processing_activity_progress_evidence
Consume processing activity progress update
: identify_processing_activity_change
Identify processing requirement change
: assess_provision_activity_evidence
Consume provision activity evidence update
: identify_provision_requirement_to_be_fulfilled
Identify provision enactment change
: assess_provision_activity_progress_evidence
Consume provision activity progress update
: identify_provision_requirement_change
Identify provision requirement change
: assess_environmental_information_update
Consume information update
: identify_required_environmental_information
Identify environmental information
: assess_sensor_platform_information_update
Consume tactical information update
: identify_required_sensor_platform_information
Identify tactical information
Request for capability update
: determine_interpretation_capability
Provide capability update
: assess_capability_evidence
Consume capability evidence change
: identify_missing_capability_evidence
Identify capability evidence

Figure 900: Sensor Data Interpretation Service Dependencies

5.4.2.52 Sensor Products

5.4.2.52.1 Role

The role of Sensor Products is to provide, manipulate and analyse sensor products, including the identification and feature characterisation of evidence of possible objects, and to maintain the traceability between any such generated artefacts and their source.

5.4.2.52.2 Overview

Control Architecture

Sensor Products is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Sensor Products receives a Requirement to provide, manipulate, or analyse a Sensor_Product, and in response provides any or all of the following:

- The raw or derived Sensor_Product.
- A characterisation of the Sensor_Product.
- The characterisation of one or more Features identified within the Sensor_Product.

Examples of Use

Sensor Products can be used to process various Sensor_Product_Types at different levels of abstraction to:

- Provide Sensor_Products directly with minimal processing, e.g. a video feed to an operator.
- Improve Sensor_Product Quality (e.g. noise reduction, or image enhancement).
- Identify Features within Sensor_Product images and characterise these to provide evidence of possible objects.
- Identify Features within Sensor_Product waveforms and characterise these to provide evidence of the bearing or location of emitters and their type.
- Combine imagery from multiple compatible sensors (e.g. to generate a multispectral image prior to further processing to identify Features, or to provide a panoramic composite image).

5.4.2.52.3 Service Summary

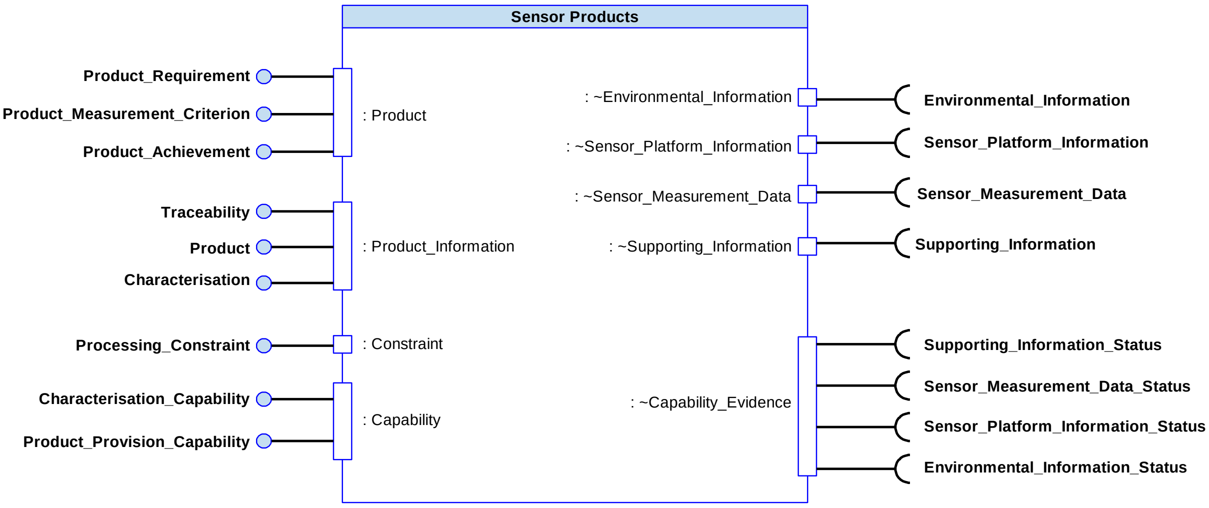

Sensor Products
Product_Requirement
Product_Measurement_Criterion
Product_Achievement
: Product
Traceability
Product
Characterisation
: Product_Information
Processing_Constraint
: Constraint
Characterisation_Capability
Product_Provision_Capability
: Capability
: ~Environmental_Information
: ~Sensor_Platform_Information
: ~Sensor_Measurement_Data
: ~Supporting_Information
: ~Capability_Evidence
Environmental_Information
Sensor_Platform_Information
Sensor_Measurement_Data
Supporting_Information
Supporting_Information_Status
Sensor_Measurement_Data_Status
Sensor_Platform_Information_Status
Environmental_Information_Status

Figure 901: Sensor Products Service Summary

5.4.2.52.4 Responsibilities

capture_requirements

- To capture Requirements to provide, manipulate or analyse Sensor_Products or to identify and characterise Features within Sensor_Products (e.g. identify and characterise shapes matching a specified pattern from imagery captured in a particular region, or identify and characterise a specific waveform pattern from an RF source).

capture_measurement_criteria

- To capture Measurement_Criterion for Sensor_Product provision, manipulation, or analysis (e.g. Quality).

capture_constraints

- To capture Constraints on Sensor_Product provision, manipulation, or analysis (e.g. restriction on measurement source).

determine_if_requirement_is_achievable

- To determine if a Sensor_Product requirement is achievable given current Capability and Constraints.

determine_solution

- To determine a solution which meets the Requirements and satisfies the Constraints for Sensor_Product provision, manipulation, or analysis.

capture_acquisition_characteristics

- To capture the acquisition characteristics of Sensor_Products (e.g. time of capture, spatial region captured, or spectral frequency captured).

execute_solution

- To provide, manipulate, or analyse Sensor_Products based on determined solutions.

maintain_traceability

- To maintain the lineage of Sensor_Products, Sensor_Product_Characterisations of Sensor_Products and the Feature_Characterisation of Features identified within Sensor_Products, including Traceability to measurement sources (e.g. an external system or local sensor) and precursor Sensor_Products.

enhance_sensor_products

- To enhance Sensor_Products in order to improve Quality (e.g. noise reduction or contrast enhancement).

combine_sensor_products

- To combine Sensor_Products from different sources or with different acquisition characteristics, e.g. pixel level combining of imagery.

characterise_sensor_products

- To characterise the nature of a Sensor_Product or Feature identified within a Sensor_Product (e.g. the Sensor_Product metadata, identified pattern, or confidence level).

determine_solution_quality

- To determine the Quality of a solution against the Measurement_Criterion.

determine_quality_of_outputs

- To determine the Quality of Sensor_Products, the Sensor_Product_Characterisation of Sensor_Products and the Feature_Characterisation of identified Features within Sensor_Products (e.g. the level of confidence associated with a specified pattern match).

capture_sensor_products

- To capture Sensor_Products.

assess_capability

- To assess the Capability to provide, manipulate, or analyse Sensor_Products, taking account of observed anomalies (e.g. measurement source availability).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of Sensor Products Capability over time and with use.

5.4.2.52.5 Subject Matter Semantics

The subject matter of Sensor Products is the data measured by a sensor, data derived or extracted from such a measurement, and the characterisation of such data.

Exclusions

The subject matter of Sensor Products does not include:

- The determination of sensing requirements, including the coordination of vehicle or sensor positioning, e.g. for the identification of Features within images, where a number of different viewing aspects of the same objects may be needed for higher confidence Feature identification and characterisation.
- The control of the quality of Sensor_Product, Sensor_Product_Characterisation, or Feature_Characterisation, where this depends on the quality of source data, which may in turn depend on Capture_Data such as vehicle position and orientation, data resolution and signal to noise ratio.
- The interpretation of identified Features to determine the identity, location and characteristics of tactical objects, or the behaviours and relationships between such objects, i.e. this component is limited to Feature_Characterisation of the identified Feature (e.g. pattern matched to a stated degree of confidence in an IR image at this time and location) without attempting to place tactical significance on any objects that might be identified from one or more sources of such evidence.

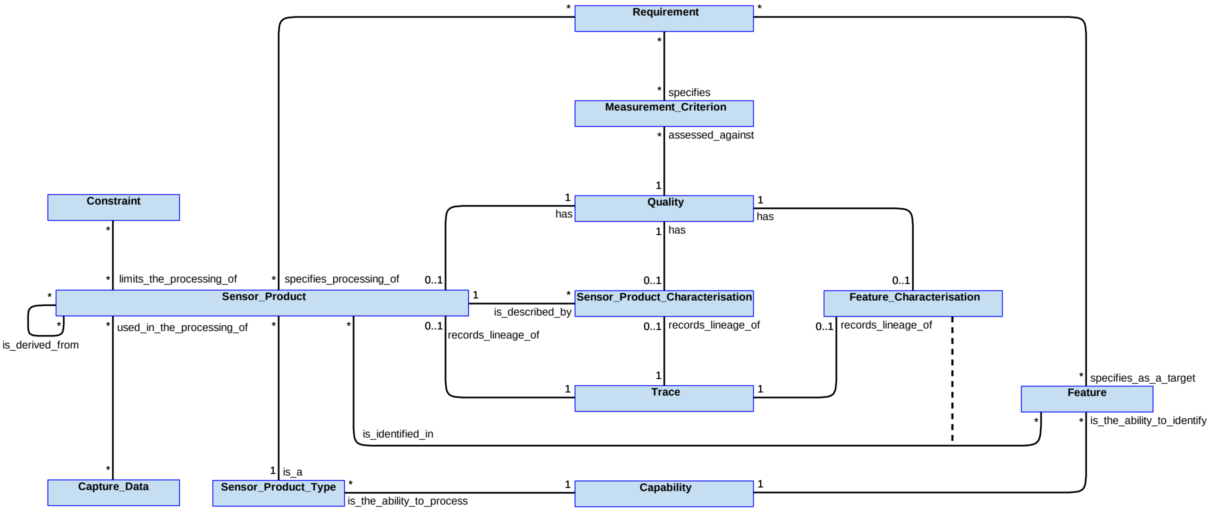


Figure 902: Sensor Products Semantics

5.4.2.52.5.1 Entities

Feature

A pattern that may be the target for identification within a Sensor_Product (e.g. a pattern of pixels or a particular pulse pattern).

Measurement_Criterion

A measure against which achievement of the Requirement can be assessed.

Requirement

A specification for the provision, manipulation, or analysis of a Sensor_Product, e.g. identify and characterise all instances of a specified pattern from images taken of a geographical area.

Sensor_Product

A measurement of the physical environment captured by a sensor, or data derived or extracted from such a measurement. This includes raw or unprocessed sensor data (e.g. a direct video stream), refined sensor data (e.g. noise reduced or contrast enhanced), extracted sensor data (e.g. an image clip or signal clip), or combined sensor data (e.g. a multispectral image combination, or composite panoramic image).

Sensor_Product_Type

A type of measurement of the physical environment captured by a sensor, or the type of data derived or extracted from such a measurement.

Constraint

A restriction on when or how a Sensor_Product is provided, manipulated or analysed (e.g. a restriction on which types/sources of sensor measurement should be used).

Capture_Data

Platform and environmental data, recorded at the time of measurement capture (e.g. light level, field of view, exposure, frequency range, time, azimuth, elevation, receive power or location).

Capability

The range of Sensor_Product types that the component is able to provide, manipulate, or analyse and the range of processing it is able to perform, including the range of Features that can be identified.

Quality

A measure of the effectiveness or adequacy of Sensor_Product provision, manipulation, or analysis; or of the artefacts resulting from such processing.

Trace

A record of the lineage of any artefacts provided, derived, or extracted from a Sensor_Product, or the lineage of descriptions of such artefacts.

Sensor_Product_Characterisation

A description of a Sensor_Product capturing the properties, attributes and associated meta data that provides information about the nature and content of the Sensor_Product and its acquisition.

Feature_Characterisation

A description of a Feature identified within a Sensor_Product, capturing the properties, attributes and associated meta data that may provide evidence of the identity and location/time of real world objects and be used in the determination of the quality of that evidence.

5.4.2.52.6 Design Rationale

5.4.2.52.6.1 Assumptions

None.

5.4.2.52.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Sensor Products:

- Data Driving - The range of Sensor_Product_Types and pre-defined Features is expected to vary between deployments; the use of data driving in this component should therefore be considered.
- Recording and Logging - logging operations and record retention will be carried out for audit purposes, for example recording the lineage of identified features.

Extensions

- The algorithms used in Sensor_Product manipulations and Feature identification and characterisation processing are likely to vary in terms of the data involved and the behaviour of the algorithm. As such it is suggested that the use of extension components for Sensor Products may be appropriate to cater for varying sensor types and data processing options associated with a particular Exploiting Programme**.**

Exploitation Considerations

- An exploitation may wish to include multiple instances of Sensor Products to cater for the varying sensor types associated with a particular Exploiting Programme**.**

5.4.2.52.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

- Failure of this component could result in incorrect geolocation of targets and so result in weapons impacting locations not intended by the crew and so result in unintended harm to third parties. This drives a DAL B indicative IDAL.

5.4.2.52.6.4 Security Considerations

The indicative security classification is SNEO.

This component is involved in the detection of Features within Sensor_Products, e.g. from pixel patterns or waveforms. Whilst some algorithms for this may be of lower classification, many algorithms and typical sensor data collected during a mission are considered more likely to be SNEO. The component is one of a group that will maintain the integrity and availability of the sensor data and its use. The integrity and availability of the output needs appropriate protection to prevent interference leading to incorrect detection of Features.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to the acquisition characteristics and of tracing products back to their source data.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness and to ensure the integrity of object identification and location that might be interpreted from Sensor Products outputs (e.g. targeting).

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.52.7 Services

5.4.2.52.7.1 Service Definitions

5.4.2.52.7.1.1 Product

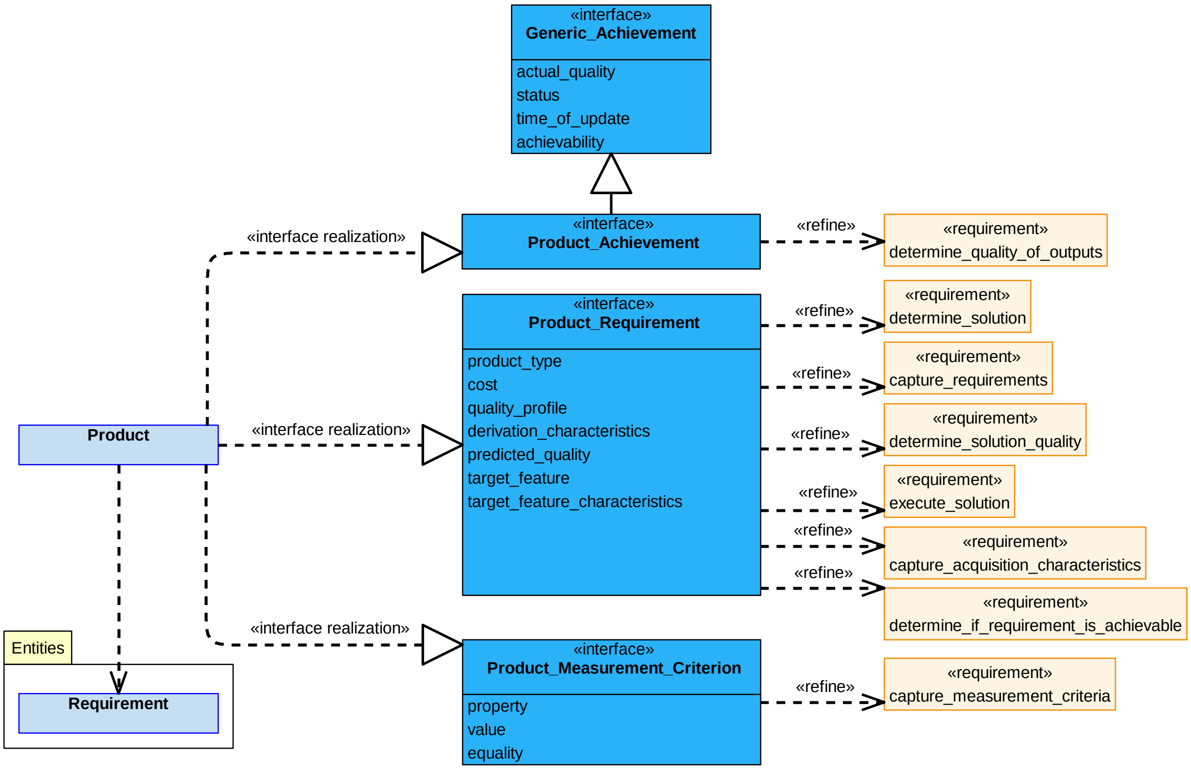


Figure 903: Product Service Definition

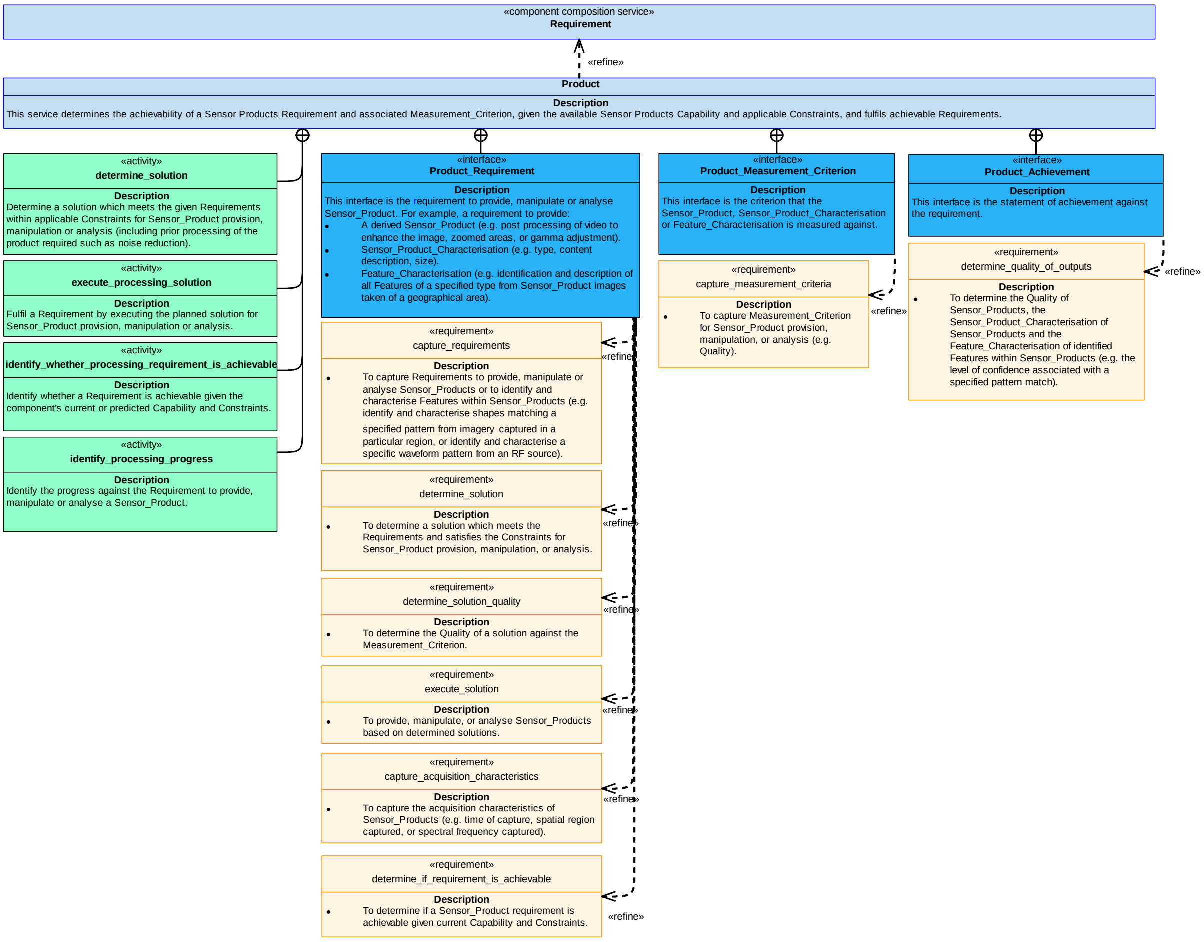

«component composition service»
Requirement
«refine»
Product
Description
This service determines the achievability of a Sensor Products Requirement and associated Measurement_Criterion, given the available Sensor Products Capability and applicable Constraints, and fulfils achievable Requirements.
«activity»
determine_solution
Description
Determine a solution which meets the given Requirements within applicable Constraints for Sensor_Product provision, manipulation or analysis (including prior processing of the product required such as noise reduction).
«activity»
execute_processing_solution
Description
Fulfil a Requirement by executing the planned solution for Sensor_Product provision, manipulation or analysis.
«activity»
identify_whether_processing_requirement_is_achievable
Description
Identify whether a Requirement is achievable given the component's current or predicted Capability and Constraints.
«activity»
identify_processing_progress
Description
Identify the progress against the Requirement to provide, manipulate or analyse a Sensor_Product.
«interface»
Product_Requirement
Description
This interface is the requirement to provide, manipulate or analyse Sensor_Product. For example, a requirement to provide:
• A derived Sensor_Product (e.g. post processing of video to enhance the image, zoomed areas, or gamma adjustment).
• Sensor_Product_Characterisation (e.g. type, content description, size).
• Feature_Characterisation (e.g. identification and description of all Features of a specified type from Sensor_Product images taken of a geographical area).
«requirement»
capture_requirements
Description
• To capture Requirements to provide, manipulate or analyse Sensor_Products or to identify and characterise Features within Sensor_Products (e.g. identify and characterise shapes matching a specified pattern from imagery captured in a particular region, or identify and characterise a specific waveform pattern from an RF source).
«refine»
«requirement»
determine_solution
Description
• To determine a solution which meets the Requirements and satisfies the Constraints for Sensor_Product provision, manipulation, or analysis.
«refine»
«requirement»
determine_solution_quality
Description
• To determine the Quality of a solution against the Measurement_Criterion.
«refine»
«requirement»
execute_solution
Description
• To provide, manipulate, or analyse Sensor_Products based on determined solutions.
«refine»
«requirement»
capture_acquisition_characteristics
Description
• To capture the acquisition characteristics of Sensor_Products (e.g. time of capture, spatial region captured, or spectral frequency captured).
«refine»
«requirement»
determine_if_requirement_is_achievable
Description
• To determine if a Sensor_Product requirement is achievable given current Capability and Constraints.
«refine»
«interface»
Product_Measurement_Criterion
Description
This interface is the criterion that the Sensor_Product, Sensor_Product_Characterisation or Feature_Characterisation is measured against.
«requirement»
capture_measurement_criteria
Description
• To capture Measurement_Criterion for Sensor_Product provision, manipulation, or analysis (e.g. Quality).
«refine»
«interface»
Product_Achievement
Description
This interface is the statement of achievement against the requirement.
«requirement»
determine_quality_of_outputs
Description
• To determine the Quality of Sensor_Products, the Sensor_Product_Characterisation of Sensor_Products and the Feature_Characterisation of identified Features within Sensor_Products (e.g. the level of confidence associated with a specified pattern match).
«refine»

Figure 904: Product Service Policy

Product

This service determines the achievability of a Sensor Products Requirement and associated Measurement_Criterion, given the available Sensor Products Capability and applicable Constraints, and fulfils achievable Requirements.

Interfaces

Product_Measurement_Criterion

This interface is the criterion that the Sensor_Product, Sensor_Product_Characterisation or Feature_Characterisation is measured against.

Attributes

property	The criterion property to be measured, e.g. a specific feature, a cost or quality factor.
value	The amount related to the property to be measured.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Product_Achievement

This interface is the statement of achievement against the requirement.

Product_Requirement

This interface is the requirement to provide, manipulate or analyse Sensor_Product. For example, a requirement to provide:

- A derived Sensor_Product (e.g. post processing of video to enhance the image, zoomed areas, or gamma adjustment).
- Sensor_Product_Characterisation (e.g. type, content description, size).
- Feature_Characterisation (e.g. identification and description of all Features of a specified type from Sensor_Product images taken of a geographical area).

Attributes

product_type	The Sensor_Product_Type to be provided, manipulated, or analysed.
cost	The cost of meeting the Sensor_Product Requirement, e.g. resources used or time taken.
quality_profile	Acceptable quality thresholds (i.e. minimum vs ideal level) of the Sensor_Product processing.
derivation_characteristics	The characteristics of the processing to be performed to produce a derived Sensor_Product, e.g. zoom an area, colour adjust or combine imagery from multiple sensors.
predicted_quality	How well the proposed Sensor_Product processing is predicted to satisfy the Requirement.
target_feature	The Feature(s) to be identified within the Sensor_Product.
target_feature_characteristics	The characteristics of the Feature(s) to be identified, e.g. size, shape, frequency, pulse width or polarisation.

Activities

determine_solution

Determine a solution which meets the given Requirements within applicable Constraints for Sensor_Product provision, manipulation or analysis (including prior processing of the product required such as noise reduction).

execute_processing_solution

Fulfil a Requirement by executing the planned solution for Sensor_Product provision, manipulation or analysis.

identify_whether_processing_requirement_is_achievable

Identify whether a Requirement is achievable given the component's current or predicted Capability and Constraints.

identify_processing_progress

Identify the progress against the Requirement to provide, manipulate or analyse a Sensor_Product.

5.4.2.52.7.1.2 Product_Information

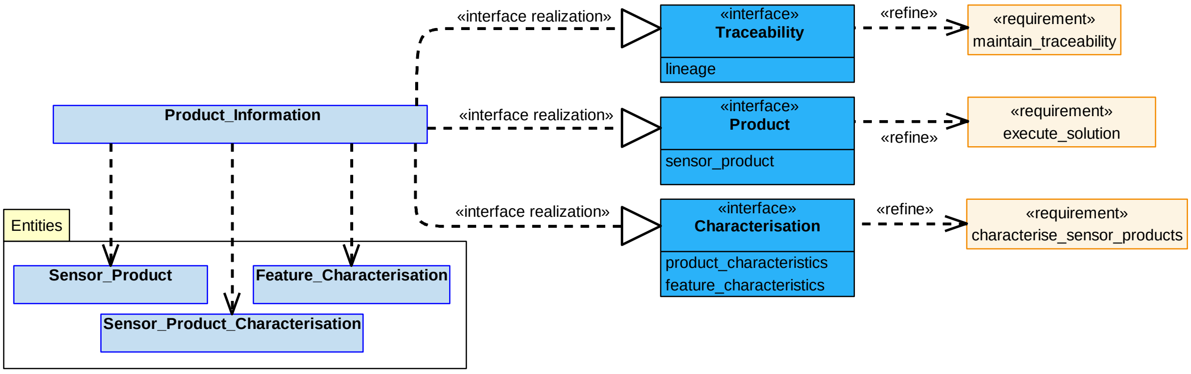
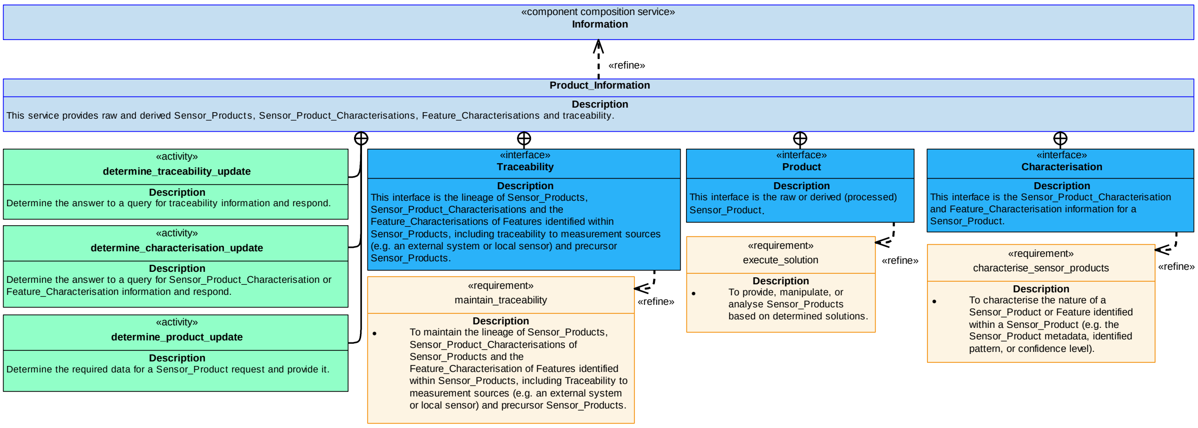


Figure 905: Product Information Service Definition


«component composition service»
Information
«refine»
Product_Information

Description

This service provides raw and derived Sensor_Products, Sensor_Product_Characterisations, Feature_Characterisations and traceability.

«activity»
determine_traceability_update

Description

Determine the answer to a query for traceability information and respond.

«activity»
determine_characterisation_update

Description

Determine the answer to a query for Sensor_Product_Characterisation or Feature_Characterisation information and respond.

Description

Determine the required data for a Sensor_Product request and provide it.

Description

This interface is the lineage of Sensor_Products, Sensor_Product_Characterisations and the Feature_Characterisations of Features identified within Sensor_Products, including traceability to measurement sources (e.g. an external system or local sensor) and precursor Sensor_Products.

Description

- To maintain the lineage of Sensor_Products, Sensor_Product_Characterisations of Sensor_Products and the Feature_Characterisation of Features identified within Sensor_Products, including Traceability to measurement sources (e.g. an external system or local sensor) and precursor Sensor_Products.

Description

This interface is the raw or derived (processed) Sensor_Product.

Description

- To provide, manipulate, or analyse Sensor_Products based on determined solutions.

Description

This interface is the Sensor_Product_Characterisation and Feature_Characterisation information for a Sensor_Product.

Description

- To characterise the nature of a Sensor_Product or Feature identified within a Sensor_Product (e.g. the Sensor_Product metadata, identified pattern, or confidence level).

Figure 906: Product Information Service Policy

Product_Information

This service provides raw and derived Sensor_Products, Sensor_Product_Characterisations, Feature_Characterisations and traceability.

Interfaces

Traceability

This interface is the lineage of Sensor_Products, Sensor_Product_Characterisations and the Feature_Characterisations of Features identified within Sensor_Products, including traceability to measurement sources (e.g. an external system or local sensor) and precursor Sensor_Products.

Attribute

lineage	The lineage of Sensor_Products, Sensor_Product_Characterisations and Feature_Characterisations, including traceability to measurement sources.

Product

This interface is the raw or derived (processed) Sensor_Product.

Attribute

sensor_product	The raw or derived Sensor_Product.

Characterisation

This interface is the Sensor_Product_Characterisation and Feature_Characterisation information for a Sensor_Product.

Attributes

product_characteristics	Information identified about a Sensor_Product, such as type, content description or size.
feature_characteristics	The characteristics of the identified Feature(s) (e.g. size, shape, frequency or pulse width or polarisation).

Activities

determine_traceability_update

Determine the answer to a query for traceability information and respond.

determine_characterisation_update

Determine the answer to a query for Sensor_Product_Characterisation or Feature_Characterisation information and respond.

determine_product_update

Determine the required data for a Sensor_Product request and provide it.

5.4.2.52.7.1.3 Environmental_Information

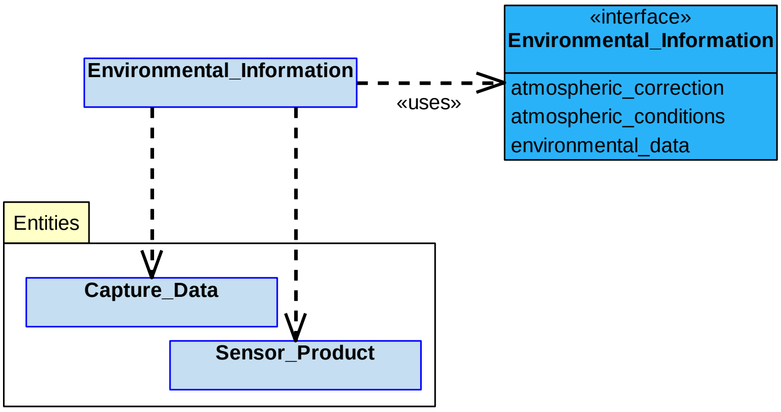


Figure 907: Environmental_Information Service Definition

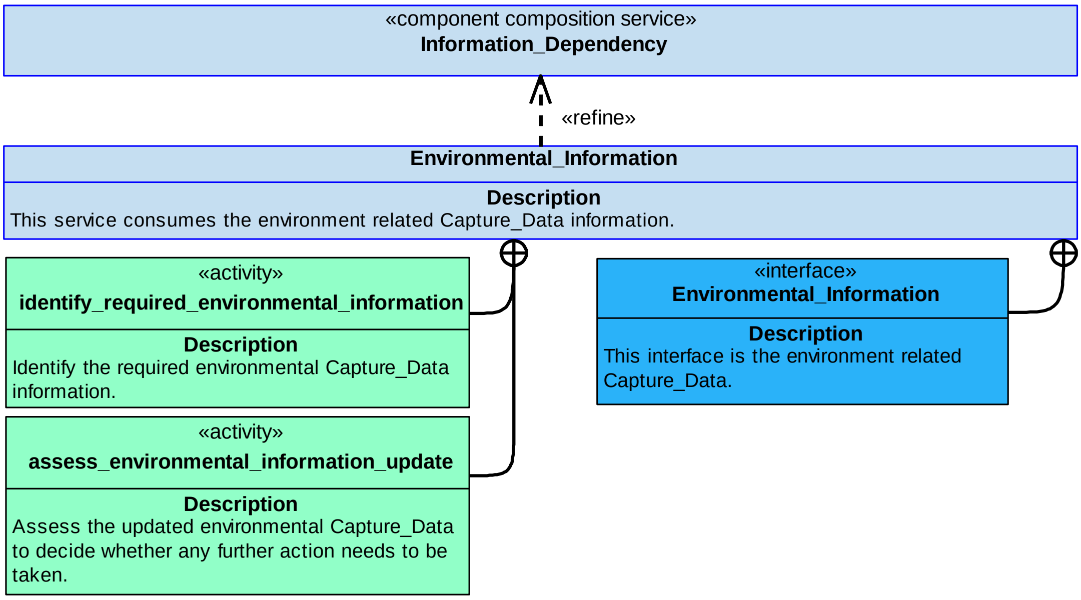

«component composition service»
Information_Dependency
«refine»
Environmental_Information
Description
This service consumes the environment related Capture_Data information.
«activity»
identify_required_environmental_information
Description
Identify the required environmental Capture_Data information.
«activity»
assess_environmental_information_update
Description
Assess the updated environmental Capture_Data to decide whether any further action needs to be taken.
«interface»
Environmental_Information
Description
This interface is the environment related Capture_Data.

Figure 908: Environmental_Information Service Policy

Environmental_Information

This service consumes the environment related Capture_Data information.

Interface

Environmental_Information

This interface is the environment related Capture_Data.

Attributes

atmospheric_correction	Spatial correction for sensors related to changes in atmospheric conditions that affect sensing processes and performance.
atmospheric_conditions	Weather conditions and features at the time of Sensor_Product capture.
environmental_data	Information describing surfaces and features in the environment within the location of Sensor_Product capture (e.g. land terrain, sea state or clutter).

Activities

assess_environmental_information_update

Assess the updated environmental Capture_Data to decide whether any further action needs to be taken.

identify_required_environmental_information

Identify the required environmental Capture_Data information.

5.4.2.52.7.1.4 Sensor_Platform_Information

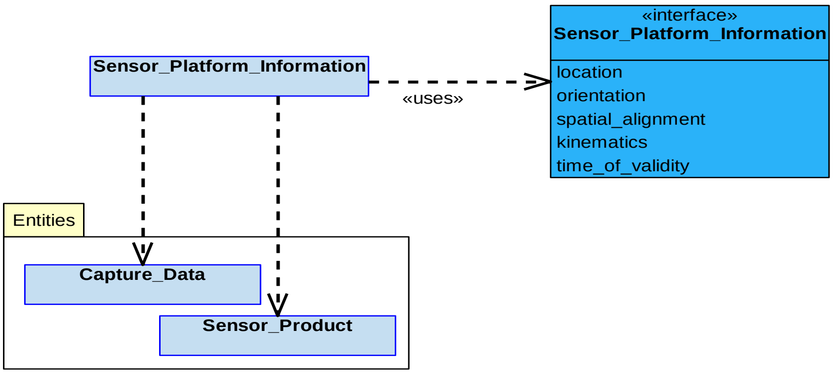


Figure 909: Sensor_Platform_Information Service Definition

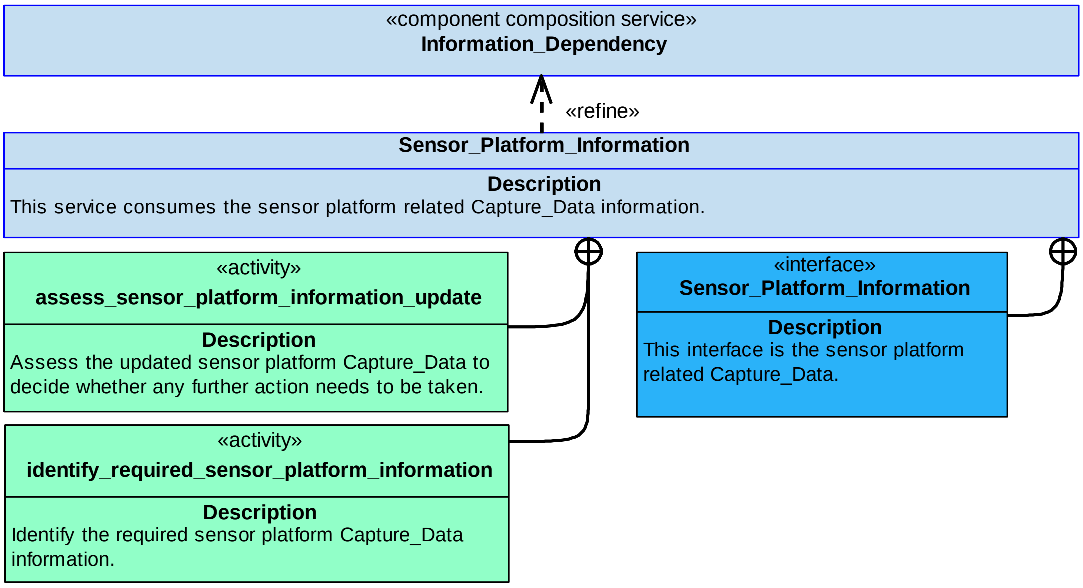

«component composition service»
Information_Dependency
«refine»
Sensor_Platform_Information
Description
This service consumes the sensor platform related Capture_Data information.
«activity»
assess_sensor_platform_information_update
Description
Assess the updated sensor platform Capture_Data to decide whether any further action needs to be taken.
«activity»
identify_required_sensor_platform_information
Description
Identify the required sensor platform Capture_Data information.
«interface»
Sensor_Platform_Information
Description
This interface is the sensor platform related Capture_Data.

Figure 910: Sensor_Platform_Information Service Policy

Sensor_Platform_Information

This service consumes the sensor platform related Capture_Data information.

Interface

Sensor_Platform_Information

This interface is the sensor platform related Capture_Data.

Attributes

location	The absolute location of the source at the time of Sensor_Product capture, which may be needed to determine the location of identified Features (e.g. latitude, longitude, altitude).
orientation	The orientation of the source at the time of Sensor_Product capture, which may be needed to determine the relative location or bearing and perspective of identified Features.

spatial_alignment	Spatial correction for sensor position relative to a reference point.
kinematics	Information relating to the motion of the source at the time of Sensor_Product capture (e.g. heading, speed or acceleration).
time_of_validity	The time when the measurement was taken.

Activities

identify_required_sensor_platform_information

Identify the required sensor platform Capture_Data information.

assess_sensor_platform_information_update

Assess the updated sensor platform Capture_Data to decide whether any further action needs to be taken.

5.4.2.52.7.1.5 Sensor_Measurement_Data

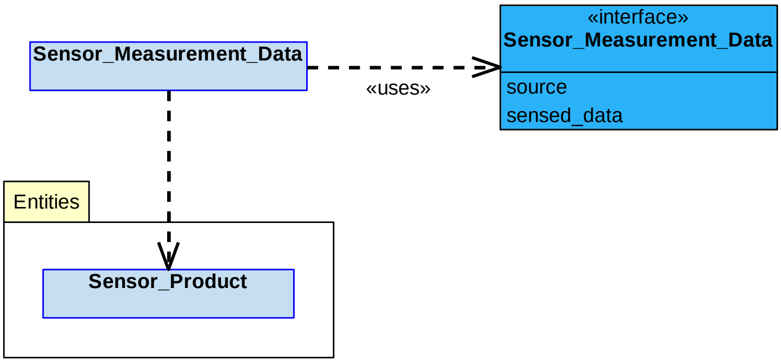


Figure 911: Sensor_Measurement_Data Service Definition

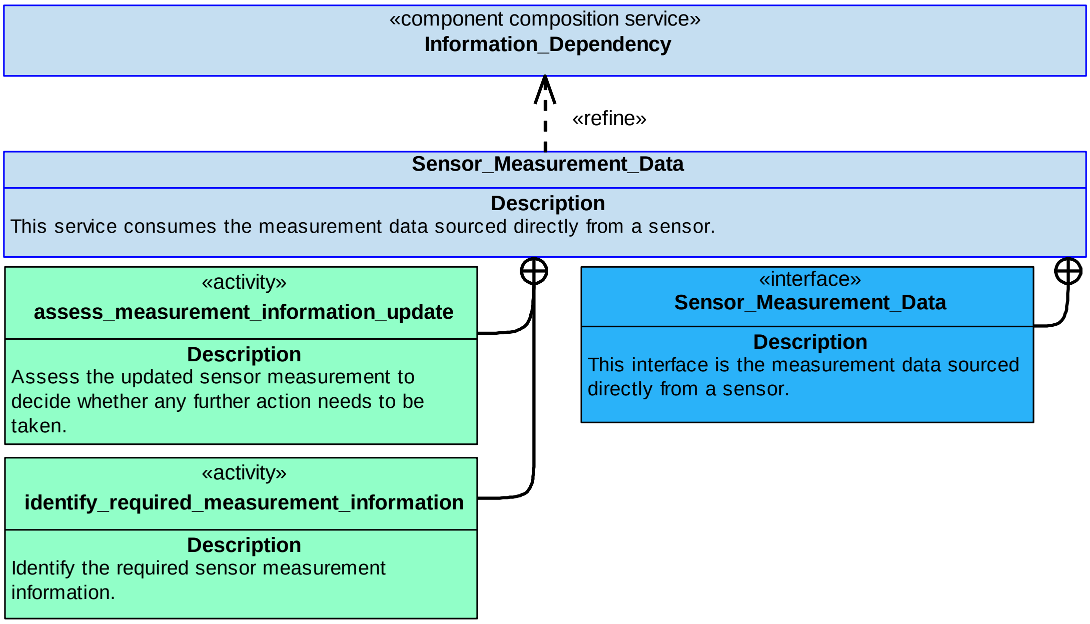

«component composition service»
Information_Dependency
«refine»
Sensor_Measurement_Data
Description
This service consumes the measurement data sourced directly from a sensor.
«activity»
assess_measurement_information_update
Description
Assess the updated sensor measurement to decide whether any further action needs to be taken.
«activity»
identify_required_measurement_information
Description
Identify the required sensor measurement information.
«interface»
Sensor_Measurement_Data
Description
This interface is the measurement data sourced directly from a sensor.

Figure 912: Sensor_Measurement_Data Service Policy

Sensor_Measurement_Data

This service consumes the measurement data sourced directly from a sensor.

Interface

Sensor_Measurement_Data

This interface is the measurement data sourced directly from a sensor.

Attributes

source	The source of the sensed data, e.g. a specific sensor and its location (which could be another sensor platform).
sensed_data	The data sourced from the sensor itself including associated metadata (e.g. radar record, image, video recording or EW record).

Activities

identify_required_measurement_information

Identify the required sensor measurement information.

assess_measurement_information_update

Assess the updated sensor measurement to decide whether any further action needs to be taken.

5.4.2.52.7.1.6 Supporting_Information

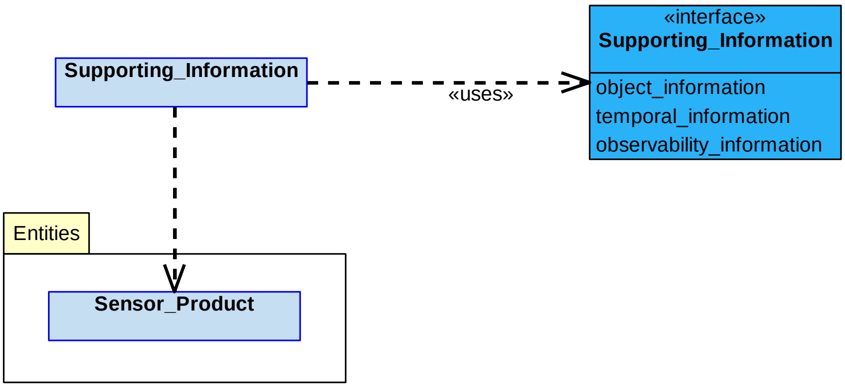


Figure 913: Supporting_Information Service Definition

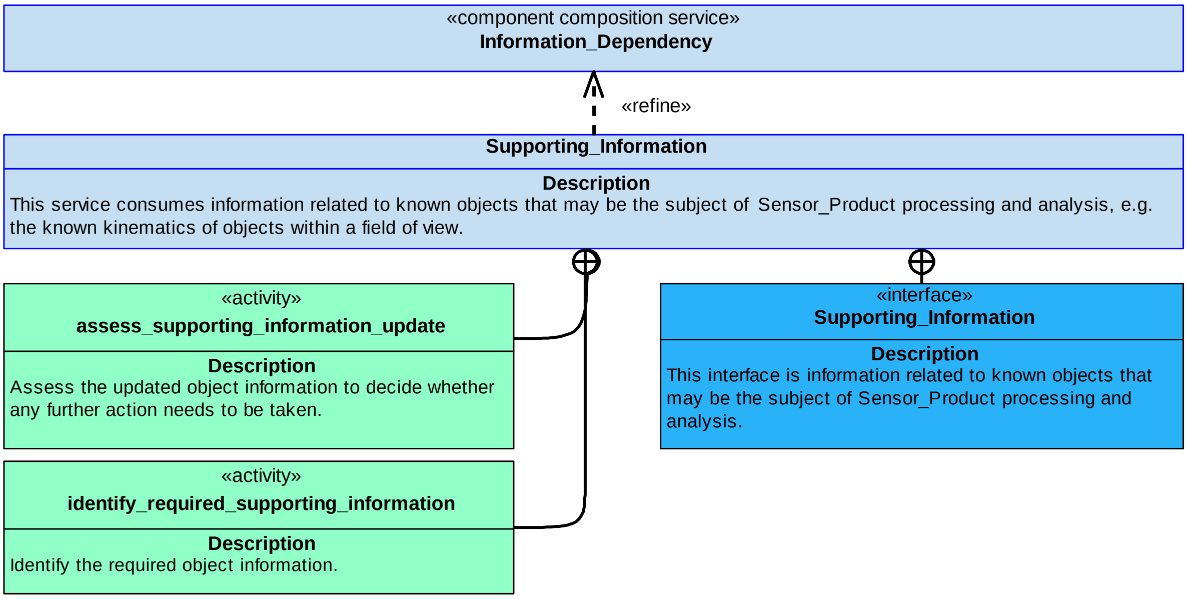

«component composition service»
Information_Dependency
«refine»
Supporting_Information
Description
This service consumes information related to known objects that may be the subject of Sensor_Product processing and analysis, e.g. the known kinematics of objects within a field of view.
«activity»
assess_supporting_information_update
Description
Assess the updated object information to decide whether any further action needs to be taken.
«activity»
identify_required_supporting_information
Description
Identify the required object information.
«interface»
Supporting_Information
Description
This interface is information related to known objects that may be the subject of Sensor_Product processing and analysis.

Figure 914: Supporting_Information Service Policy

Supporting_Information

This service consumes information related to known objects that may be the subject of Sensor_Product processing and analysis, e.g. the known kinematics of objects within a field of view.

Interface

Supporting_Information

This interface is information related to known objects that may be the subject of Sensor_Product processing and analysis.

Attributes

object_information	Kinetic information about sensing target criteria, location and movement, or that of surrounding objects. Knowledge of an object's motion at the time of Sensor_Product capture can influence subsequent processing activities.
temporal_information	Timing information required for low latency synchronisation of sensing techniques.
observability_information	Interactive low observability factors affecting the use of active sensing techniques.

Activities

assess_supporting_information_update

Assess the updated object information to decide whether any further action needs to be taken.

identify_required_supporting_information

Identify the required object information.

5.4.2.52.7.1.7 Constraint

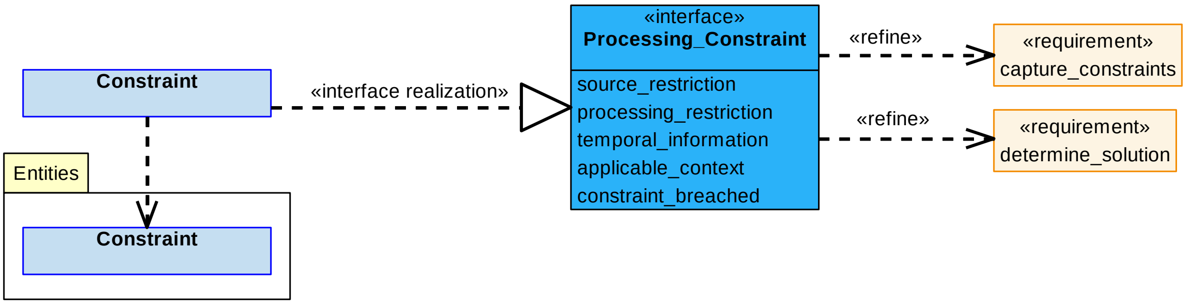


Figure 915: Constraint Service Definition

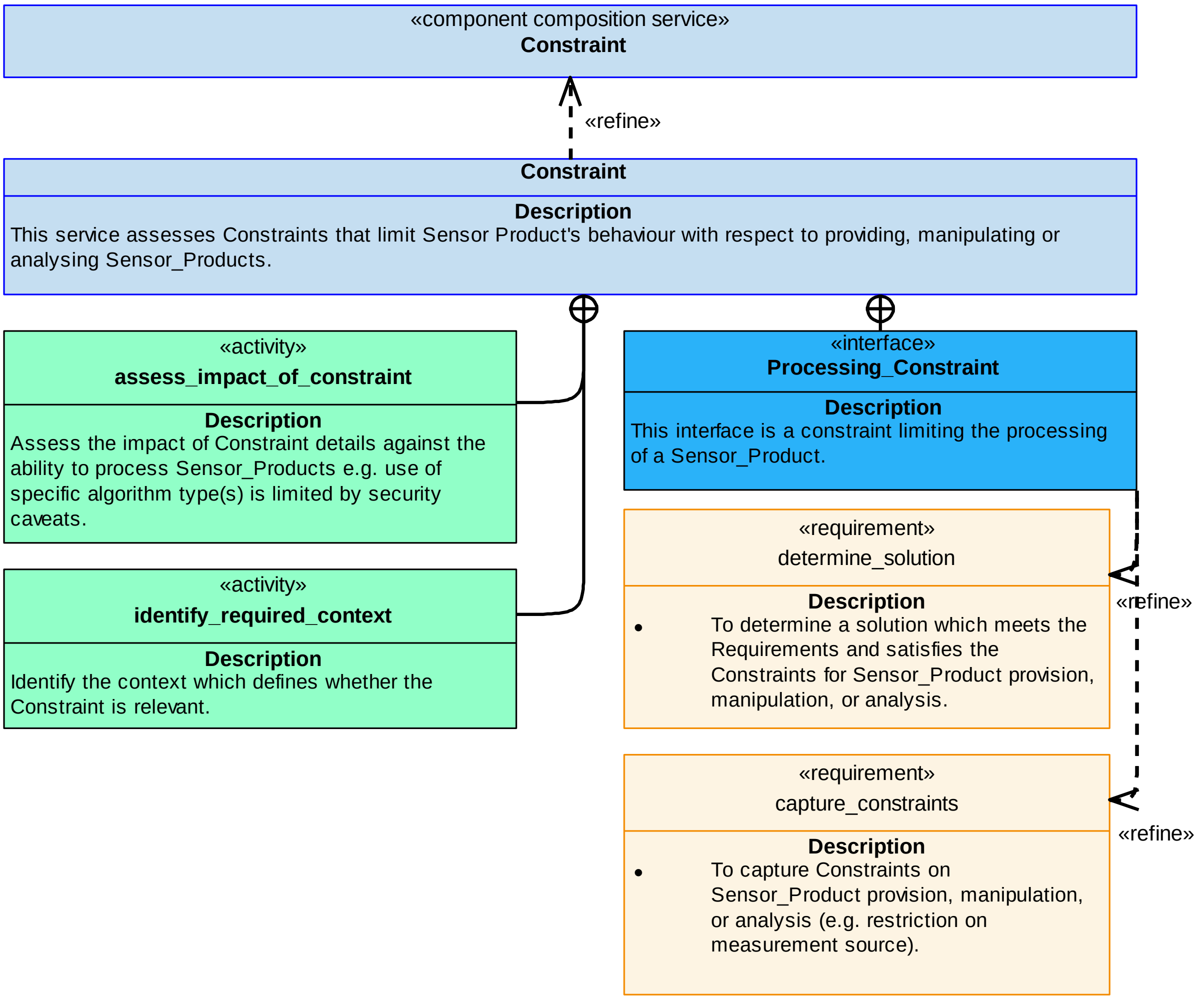

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints that limit Sensor Product's behaviour with respect to providing, manipulating or analysing Sensor_Products.
«activity»
assess_impact_of_constraint
Description
Assess the impact of Constraint details against the ability to process Sensor_Products e.g. use of specific algorithm type(s) is limited by security caveats.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraint is relevant.
«interface»
Processing_Constraint
Description
This interface is a constraint limiting the processing of a Sensor_Product.
«requirement»
determine_solution
Description
To determine a solution which meets the Requirements and satisfies the Constraints for Sensor_Product provision, manipulation, or analysis.
«refine»
«requirement»
capture_constraints
Description
To capture Constraints on Sensor_Product provision, manipulation, or analysis (e.g. restriction on measurement source).
«refine»

Figure 916: Constraint Service Policy

Constraint

This service assesses Constraints that limit Sensor Product's behaviour with respect to providing, manipulating or analysing Sensor_Products.

Interface

Processing_Constraint

This interface is a constraint limiting the processing of a Sensor_Product.

Attributes

source_restriction	A restriction on the usage of specific Sensor_Product_Types.
processing_restriction	A restriction on the application or use of an algorithm (e.g. noise reduction or frequency filtering) or data set (e.g. a specific Feature definition) in the processing of Sensor_Products.
temporal_information	Timing information pertaining to the periods of time when the constraint will be applicable, e.g. applicable for 30 minutes in an hour's time.
applicable_context	The context in which the constraint is applicable.

constraint_breached	This interface is the limitations imposed on the processing of Sensor Products and identification of whether these limitations have been breached.

Activities

assess_impact_of_constraint

Assess the impact of Constraint details against the ability to process Sensor_Products e.g. use of specific algorithm type(s) is limited by security caveats.

identify_required_context

Identify the context which defines whether the Constraint is relevant.

5.4.2.52.7.1.8 Capability

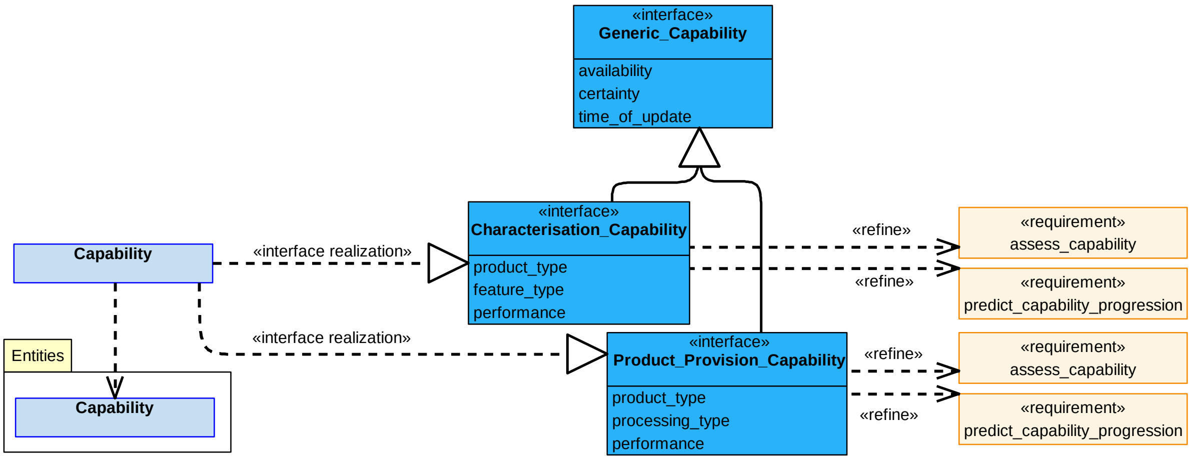


Figure 917: Capability Service Definition

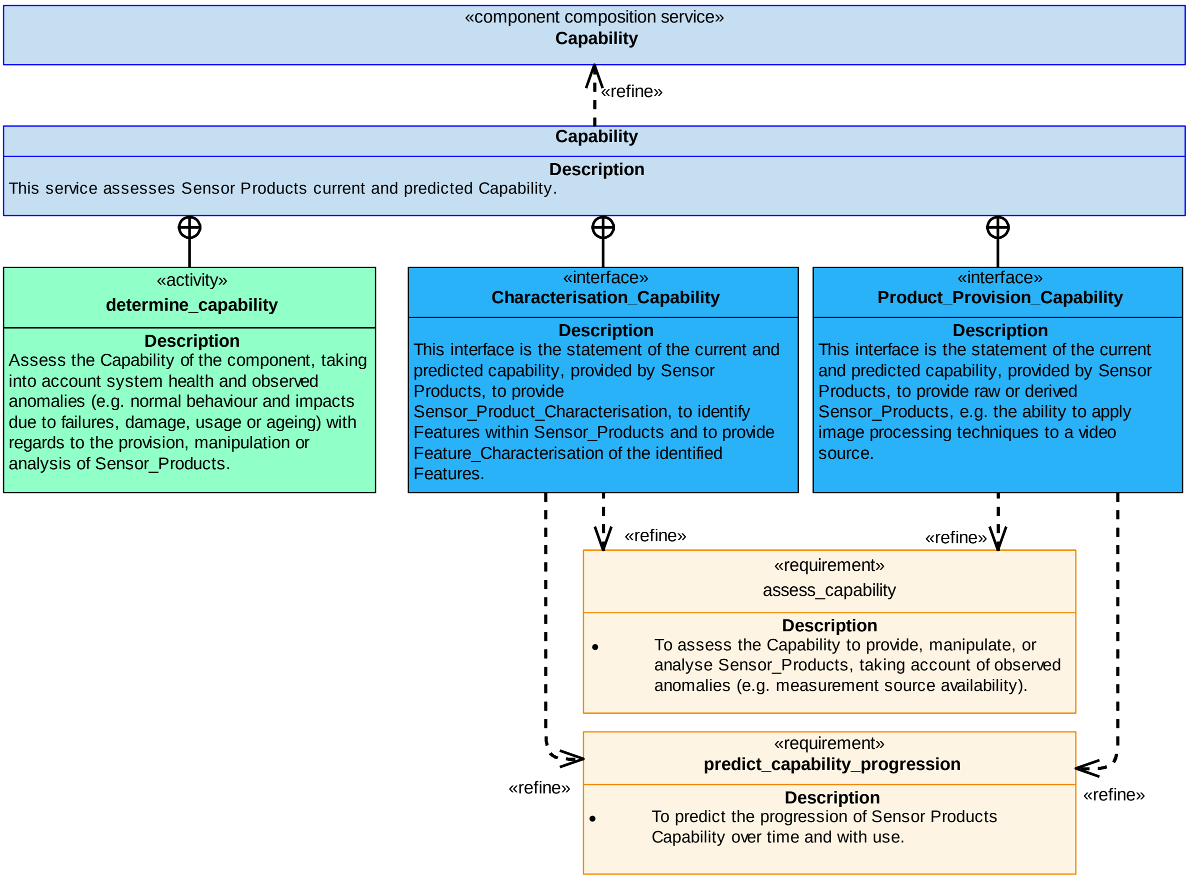

«component composition service»
Capability
«refine»
Capability
Description
This service assesses Sensor Products current and predicted Capability.
«activity»
determine_capability
Description
Assess the Capability of the component, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing) with regards to the provision, manipulation or analysis of Sensor_Products.
«interface»
Characterisation_Capability
Description
This interface is the statement of the current and predicted capability, provided by Sensor Products, to provide Sensor_Product_Characterisation, to identify Features within Sensor_Products and to provide Feature_Characterisation of the identified Features.
«interface»
Product_Provision_Capability
Description
This interface is the statement of the current and predicted capability, provided by Sensor Products, to provide raw or derived Sensor_Products, e.g. the ability to apply image processing techniques to a video source.
«refine»
«refine»
«requirement»
assess_capability
Description
To assess the Capability to provide, manipulate, or analyse Sensor_Products, taking account of observed anomalies (e.g. measurement source availability).
«requirement»
predict_capability_progression
Description
To predict the progression of Sensor Products Capability over time and with use.
«refine»
«refine»

Figure 918: Capability Service Policy

Capability

This service assesses Sensor Products current and predicted Capability.

Interfaces

Characterisation_Capability

This interface is the statement of the current and predicted capability, provided by Sensor Products, to provide Sensor_Product_Characterisation, to identify Features within Sensor_Products and to provide Feature_Characterisation of the identified Features.

Attributes

product_type	The Sensor_Product_Type to which the capability relates.
feature_type	The type or category of Feature that the component is capable of identifying, and characterising, within the Sensor_Product_Type, e.g. vehicles from an image or RF waveforms in a specific frequency range from an RF recording of a much wider frequency range.
performance	The level of performance or effectiveness that can be achieved when using an algorithm, e.g. the volume of data that can be processed in a particular time period.

Product_Provision_Capability

This interface is the statement of the current and predicted capability, provided by Sensor Products, to provide raw or derived Sensor_Products, e.g. the ability to apply image processing techniques to a video source.

Attributes

product_type	The Sensor_Product_Type to which the capability relates.
processing_type	The type or category of Sensor_Product processing that the component is capable of performing, e.g. gamma adjustment to a video stream.
performance	The level of performance or effectiveness that can be achieved, e.g. the frames per second or resolution achievable when processing a raw video stream.

Activity

determine_capability

Assess the Capability of the component, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing) with regards to the provision, manipulation or analysis of Sensor_Products.

5.4.2.52.7.1.9 Capability_Evidence

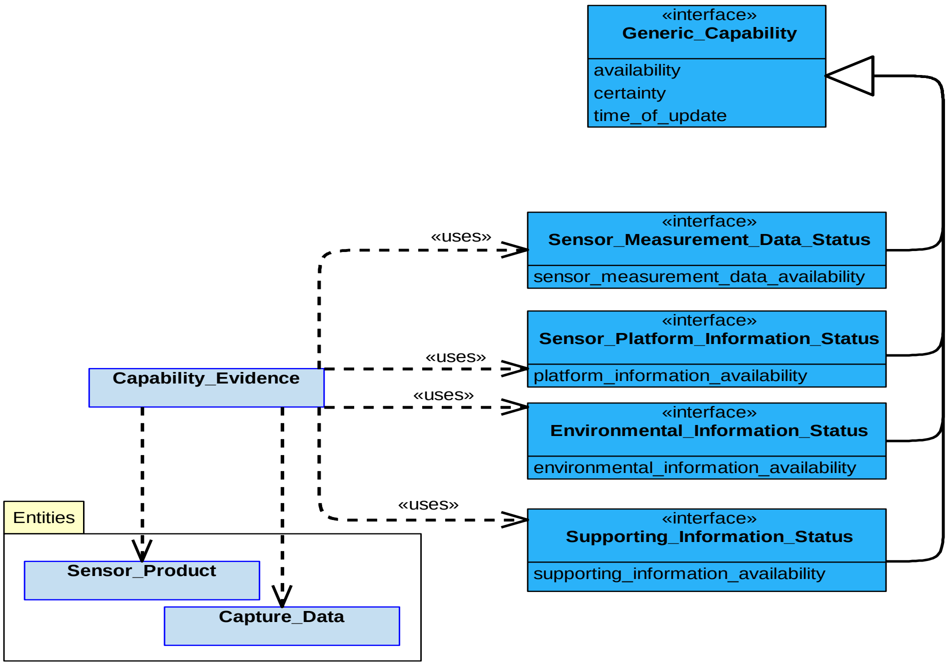

«interface»
Generic_Capability
availability
certainty
time_of_update
«uses»
«interface»
Sensor_Measurement_Data_Status
sensor_measurement_data_availability
«interface»
Sensor_Platform_Information_Status
platform_information_availability
Capability_Evidence
«uses»
«uses»
«interface»
Environmental_Information_Status
environmental_information_availability
Entities
«uses»
«interface»
Supporting_Information_Status
supporting_information_availability
Sensor_Product
Capture_Data

Figure 919: Capability_Evidence Service Definition

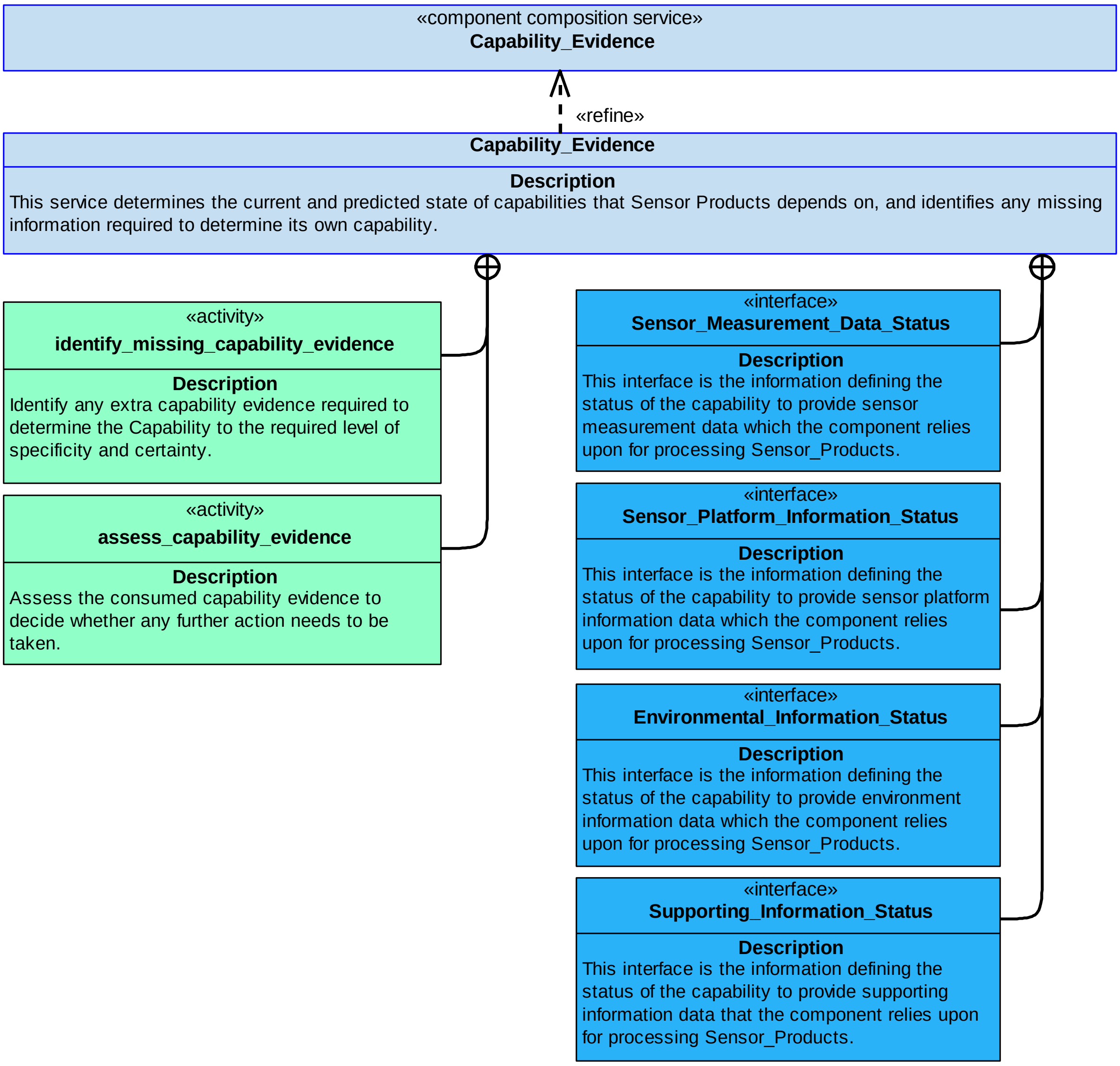

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service determines the current and predicted state of capabilities that Sensor Products depends on, and identifies any missing information required to determine its own capability.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«activity»
assess_capability_evidence
Description
Assess the consumed capability evidence to decide whether any further action needs to be taken.
«interface»
Sensor_Measurement_Data_Status
Description
This interface is the information defining the status of the capability to provide sensor measurement data which the component relies upon for processing Sensor_Products.
«interface»
Sensor_Platform_Information_Status
Description
This interface is the information defining the status of the capability to provide sensor platform information data which the component relies upon for processing Sensor_Products.
«interface»
Environmental_Information_Status
Description
This interface is the information defining the status of the capability to provide environment information data which the component relies upon for processing Sensor_Products.
«interface»
Supporting_Information_Status
Description
This interface is the information defining the status of the capability to provide supporting information data that the component relies upon for processing Sensor_Products.

Figure 920: Capability_Evidence Service Policy

Capability_Evidence

This service determines the current and predicted state of capabilities that Sensor Products depends on, and identifies any missing information required to determine its own capability.

Interfaces

Supporting_Information_Status

This interface is the information defining the status of the capability to provide supporting information data that the component relies upon for processing Sensor_Products.

Attribute

supporting_information_availability	The availability of supporting information used in the processing of Sensor_Products.

Sensor_Measurement_Data_Status

This interface is the information defining the status of the capability to provide sensor measurement data which the component relies upon for processing Sensor_Products.

Attribute

sensor_measurement_data_availability	The availability of sensor measurement data used in the production of Sensor_Products.

Sensor_Platform_Information_Status

This interface is the information defining the status of the capability to provide sensor platform information data which the component relies upon for processing Sensor_Products.

Attribute

platform_information_availability	The availability of sensor platform information used in the processing of Sensor_Products.

Environmental_Information_Status

This interface is the information defining the status of the capability to provide environment information data which the component relies upon for processing Sensor_Products.

Attribute

environmental_information_availability	The availability of environment information used in the processing of Sensor_Products.

Activities

assess_capability_evidence

Assess the consumed capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.52.7.2 Service Dependencies

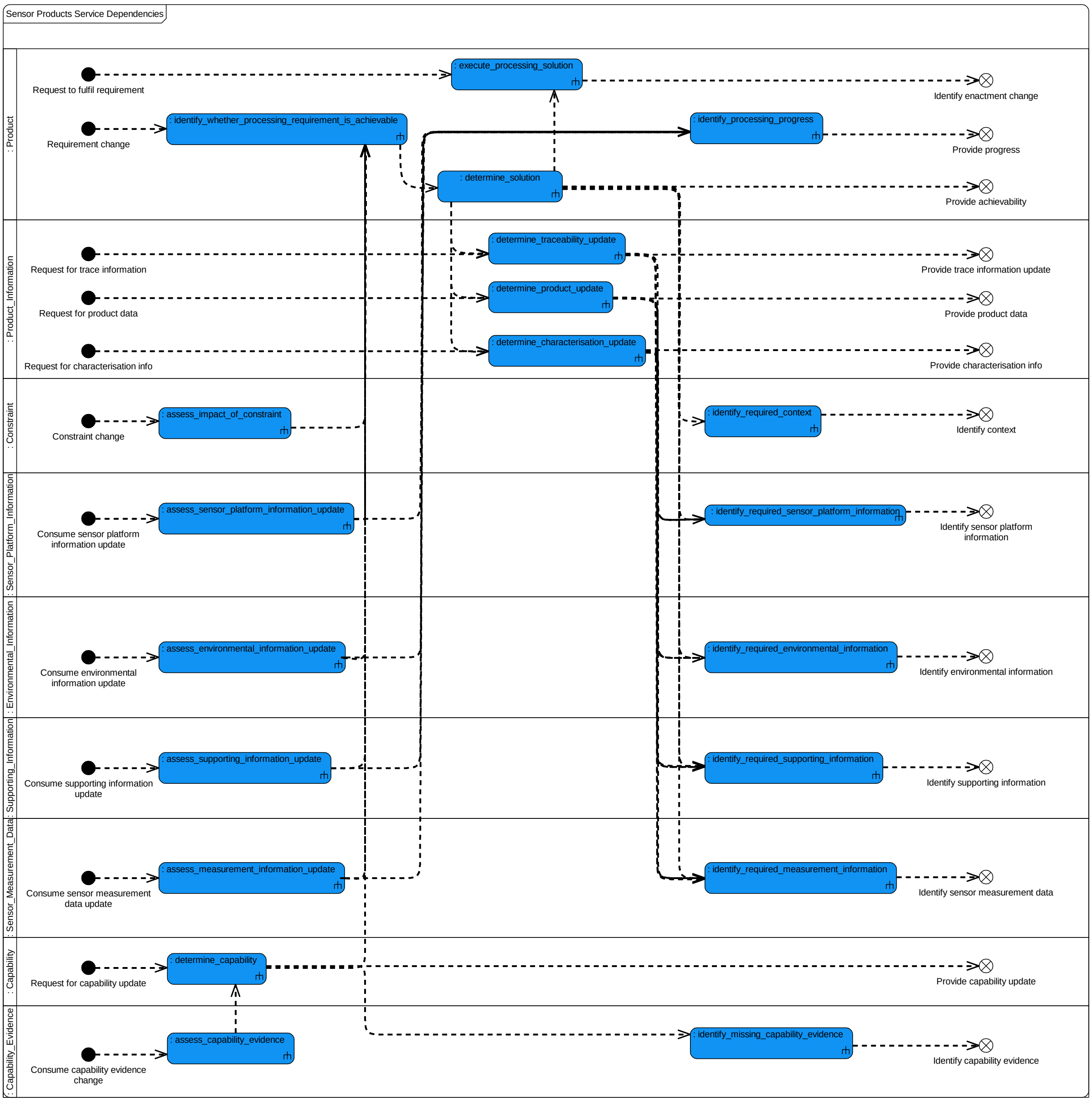

Sensor Products Service Dependencies
: Product
: Product_Information
: Constraint
: Sensor_Platform_Information
: Environmental_Information
: Supporting_Information
: Sensor_Measurement_Data
: Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
Request for trace information
Request for product data
Request for characterisation info
Constraint change
Consume sensor platform information update
Consume environmental information update
Consume supporting information update
Consume sensor measurement data update
Request for capability update
Consume capability evidence change
: execute_processing_solution
: identify_whether_processing_requirement_is_achievable
: identify_processing_progress
: determine_solution
: determine_traceability_update
: determine_product_update
: determine_characterisation_update
: assess_impact_of_constraint
: assess_sensor_platform_information_update
: assess_environmental_information_update
: assess_supporting_information_update
: assess_measurement_information_update
: determine_capability
: assess_capability_evidence
: identify_required_context
: identify_required_sensor_platform_information
: identify_required_environmental_information
: identify_required_supporting_information
: identify_required_measurement_information
: identify_missing_capability_evidence
Identify enactment change
Provide progress
Provide achievability
Provide trace information update
Provide product data
Provide characterisation info
Identify context
Identify sensor platform information
Identify environmental information
Identify supporting information
Identify sensor measurement data
Provide capability update
Identify capability evidence

Figure 921: Sensor Products Service Dependencies

5.4.2.53 Sensors

5.4.2.53.1 Role

The role of Sensors is to provide an interface to obtain measurements from sensors.

5.4.2.53.2 Overview

Sensors is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In response to a demand, a Sensor_Resource will capture a Sensor_Measurement.

Examples of Use

- This component will be required in a system which includes a Sensor_Resource to provide a measurement of an aspect of the physical environment, such as a fuel level sensor or a temperature sensor.
- It may be used to control complex equipment with a simple interface, such as a camera that accommodates itself to environmental conditions.

5.4.2.53.3 Service Summary

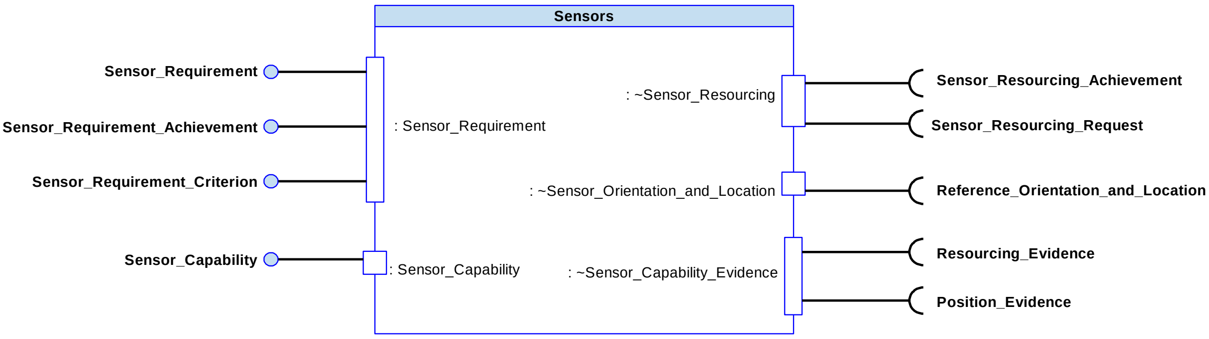

Sensors
Sensor_Requirement
Sensor_Requirement_Achievement
Sensor_Requirement_Criterion
Sensor_Capability
: Sensor_Requirement
: Sensor_Capability
: ~Sensor_Resourcing
: ~Sensor_Orientation_and_Location
: ~Sensor_Capability_Evidence
Sensor_Resourcing_Achievement
Sensor_Resourcing_Request
Reference_Orientation_and_Location
Resourcing_Evidence
Position_Evidence

Figure 922: Sensors Service Summary

5.4.2.53.4 Responsibilities

capture_requirements_for_sensor_resources

- To capture provided requirements (e.g. turn on/off or obtain a measurement) for use of Sensor_Resources.

capture_measurement_criteria

- To capture Measurement_Criterion for a Sensor_Measurement.

determine_sensor_solution

- To determine a solution for use of Sensor_Resources that will meet given Requirements.

determine_if_solution_remains_feasible

- To determine if a planned or ongoing Sensor_Solution remains feasible given current Capability.

control_use_of_sensor

- To control the use of Sensor_Resources to obtain a measurement.

capture_sensor_data

- To collect data from a Sensor_Resource (e.g. temperature from a thermometer or location of object in an area).

identify_progress

- To identify progress against a Requirement.

provide_sensor_data_feedback

- To provide the wider system with feedback data.

update_resource_usages

- To update the status of the sensors usage of the platform's resources.

assess_sensor_capability

- To assess the capability to perform sensing using Sensor_Resources, taking into account available resources and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of Sensor_Resource capability over time and with use.

5.4.2.53.5 Subject Matter Semantics

The subject matter of Sensors is measurements from Sensor_Resources.

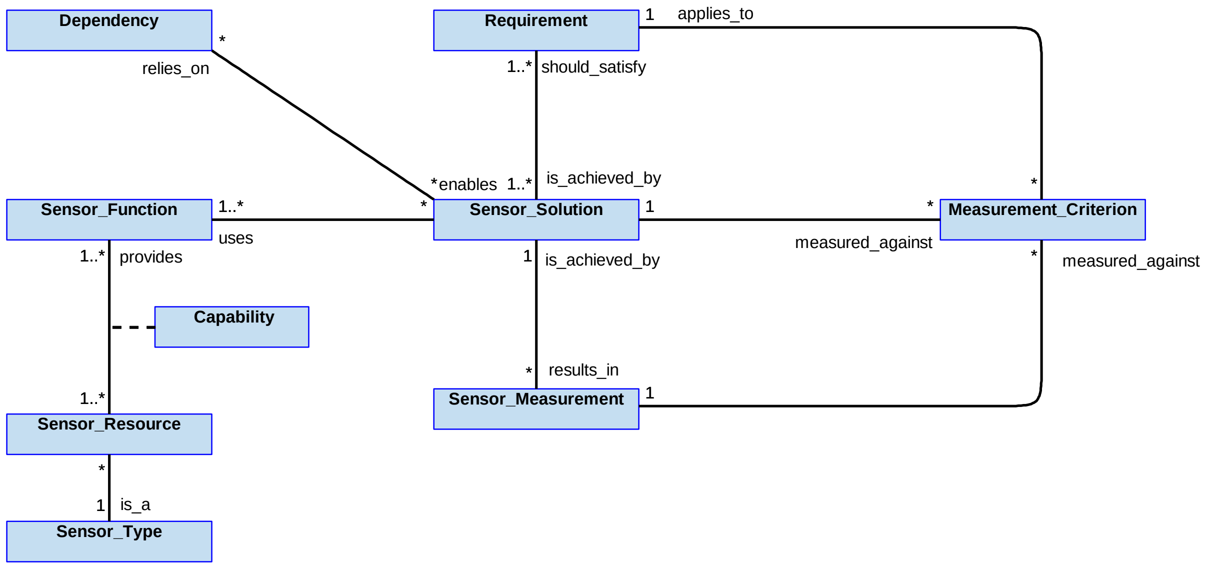


Figure 923: Sensors Semantics

5.4.2.53.5.1 Entities

Capability

The ability to determine a measurement of the physical environment in order to meet requirements. This takes into account the ability of Sensors to control the Sensor_Resource.

Measurement_Criterion

A criterion that a possible solution is measured against.

Requirement

A requirement to control a sensor or capture sensor data (e.g. turn on/off or obtain a measurement).

Sensor_Function

A sensor operation that can be performed by a sensor.

Sensor_Measurement

A measurement of the physical environment that can be obtained using a Sensor_Resource.

Sensor_Resource

Sensor equipment that is capable of obtaining a measurement of the physical environment.

Sensor_Type

A type of sensor that can be utilised (e.g. fuel flow, air pressure or temperature sensors).

Sensor_Solution

A solution to satisfy the Requirement for a Sensor_Measurement.

Dependency

Something that the component relies on for a successful Sensor_Solution outcome (e.g. for moveable sensors, information about their position relative to the platform datum's, or power and cooling needs to support a Sensor_Function).

5.4.2.53.6 Design Rationale

5.4.2.53.6.1 Assumptions

- The Sensor_Resources managed by the Sensors component may be simple pieces of sensing equipment such as temperature sensors.
- Sensor_Resources may represent a simple interface to sensing functions of a complex piece of equipment that may be highly sophisticated (such as a camera that accommodates to environmental conditions).
- Sensor_Resources do not represent the whole of complex pieces of equipment with multiple functions and a highly configurable interface. Such equipment is addressed in the Interaction with Equipment PYRAMID concept.
- A Sensor_Resource provides a measurement of a physical characteristic (e.g. EM spectrum, acoustics or optical).

5.4.2.53.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Sensors:

- Interaction with Equipment - This explains how the PRA accommodates complex sensors which provide a suite of functions with a high level of configurability.
- Data Driving - There are numerous types of Sensor_Resource, this component could be configured to support any of them using data driving.

Extensions

- It is possible that extension components will be developed to provide an interface to specific types of equipment.

Exploitation Considerations

- The types of Sensor_Measurement produced by Sensor_Resources of the same type are tied to the physical phenomenon being sensed and therefore are unlikely to change when one resource is replaced by another of the same type.
- The interface of the Sensors component with its Sensor_Resources is likely to be specific to the type of Sensor_Resource, and may change if one Sensor_Resource is replaced with another. This variation is expected to be handled by data driving or extensions. (Note that resources replaced on a like-for-like basis (with the same form, fit and function) may have

different failure modes. The PRA accommodates this using Anomaly Detection and Health Assessment as explained in the Health Management PYRAMID concept.)

- A new Sensor_Resource could be introduced during mission fit, e.g. different payload bay modules may be switched according to mission requirements, and these may incorporate different Sensor_Resources.

5.4.2.53.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component could fail to correctly measure a critical aspect of the physical environment. In the case of an air vehicle, failure of this component may cause uncontrolled flight of the air vehicle. For example, if the position of flaps, undercarriage or weapon bay doors was incorrectly reported then the air vehicle may be flown outside the safe flight envelope. This could lead to loss of aerodynamic control and / or loss of structural integrity of the air vehicle and result in an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

Where instances of this component are used to prevent hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.53.6.4 Security Considerations

The indicative security classification is O but will vary according to the sensor.

This component forms part of the control interface with sensors (turning it on or off, triggering it to perform a sensor measurement, etc.) and as such is dependent on the Sensor_Type being managed and their use; there are expected to be multiple instances of this component, for sensors ranging from simple flow or temperature sensors to complex tactical sensing equipment. The confidentiality, integrity and availability requirements will need to reflect this.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to sensor use during the mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** of the sensor state against the commanded operations. Unexpected activity may indicate that the Exploiting Platform has been compromised.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.53.7 Services

5.4.2.53.7.1 Service Definitions

5.4.2.53.7.1.1 Sensor_Requirement

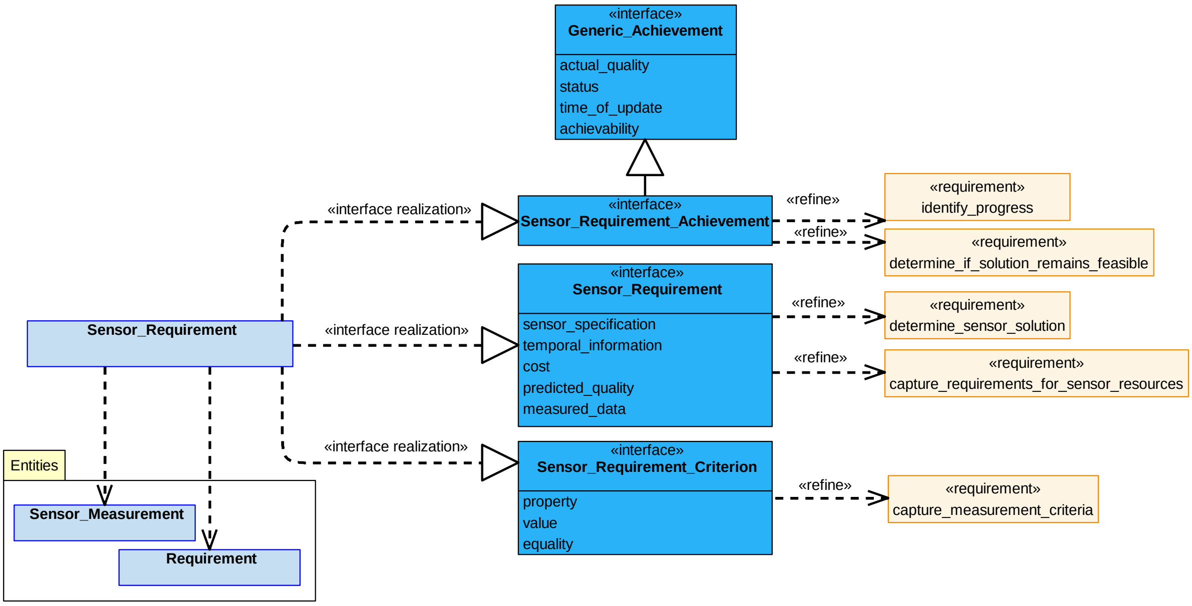


Figure 924: Sensor_Requirement Service Definition

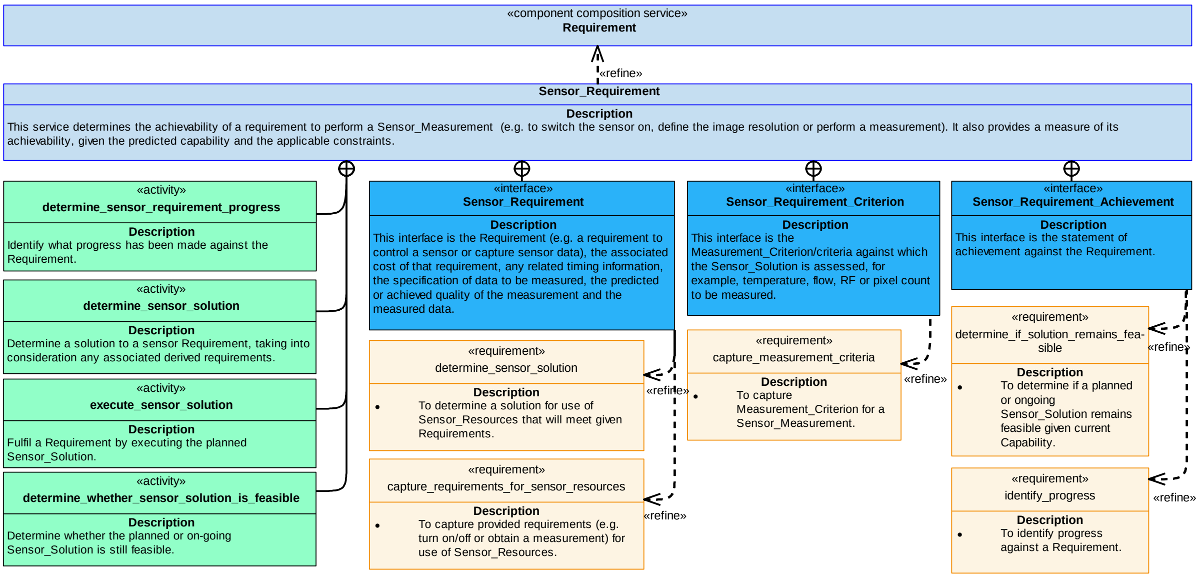

«component composition service»
Requirement
«refine»
Sensor_Requirement
Description
This service determines the achievability of a requirement to perform a Sensor_Measurement (e.g. to switch the sensor on, define the image resolution or perform a measurement). It also provides a measure of its achievability, given the predicted capability and the applicable constraints.
«activity»
determine_sensor_requirement_progress
Description
Identify what progress has been made against the Requirement.
«activity»
determine_sensor_solution
Description
Determine a solution to a sensor Requirement, taking into consideration any associated derived requirements.
«activity»
execute_sensor_solution
Description
Fulfil a Requirement by executing the planned Sensor_Solution.
«activity»
determine_whether_sensor_solution_is_feasible
Description
Determine whether the planned or on-going Sensor_Solution is still feasible.
«interface»
Sensor_Requirement
Description
This interface is the Requirement (e.g. a requirement to control a sensor or capture sensor data), the associated cost of that requirement, any related timing information, the specification of data to be measured, the predicted or achieved quality of the measurement and the measured data.
«requirement»
determine_sensor_solution
Description
To determine a solution for use of Sensor_Resources that will meet given Requirements.
«refine»
«requirement»
capture_requirements_for_sensor_resources
Description
To capture provided requirements (e.g. turn on/off or obtain a measurement) for use of Sensor_Resources.
«refine»
«interface»
Sensor_Requirement_Criterion
Description
This interface is the Measurement_Criterion/criteria against which the Sensor_Solution is assessed, for example, temperature, flow, RF or pixel count to be measured.
«requirement»
capture_measurement_criteria
Description
To capture Measurement_Criterion for a Sensor_Measurement.
«refine»
«interface»
Sensor_Requirement_Achievement
Description
This interface is the statement of achievement against the Requirement.
«requirement»
determine_if_solution_remains_feasible
Description
To determine if a planned or ongoing Sensor_Solution remains feasible given current Capability.
«refine»
«requirement»
identify_progress
Description
To identify progress against a Requirement.
«refine»

Figure 925: Sensor_Requirement Service Policy

Sensor_Requirement

This service determines the achievability of a requirement to perform a Sensor_Measurement (e.g. to switch the sensor on, define the image resolution or perform a measurement). It also provides a measure of its achievability, given the predicted capability and the applicable constraints.

Interfaces

Sensor_Requirement_Criterion

This interface is the Measurement_Criterion/criteria against which the Sensor_Solution is assessed, for example, temperature, flow, RF or pixel count to be measured.

Attributes

property	The property to be measured, e.g. a specific measurement such as temperature, flow, RF.
value	The amount related to the property to be measured, e.g. 50 degrees Celsius where a temperature is being measured.
equality	The relationship between the value and any limit on the measurement, e.g. less / more than, equal to or min / max limits.

Sensor_Requirement

This interface is the Requirement (e.g. a requirement to control a sensor or capture sensor data), the associated cost of that requirement, any related timing information, the specification of data to be measured, the predicted or achieved quality of the measurement and the measured data.

Attributes

sensor_specification	A specification of the control parameters of a Sensor_Resource and the data to be captured (e.g. field of regard or image resolution).
temporal_information	Information covering timing for the requested Sensor_Resource, e.g. start and end times.
cost	The cost of executing the Sensor_Solution, for example: sensor resources used or time taken.
predicted_quality	How well the proposed Sensor_Solution is predicted to satisfy the Requirement.
measured_data	The measured sensor data which may be a pointer to data measurement streams or locations where the data is stored.

Sensor_Requirement_Achievement

This interface is the statement of achievement against the Requirement.

Activities

determine_sensor_solution

Determine a solution to a sensor Requirement, taking into consideration any associated derived requirements.

execute_sensor_solution

Fulfil a Requirement by executing the planned Sensor_Solution.

determine_whether_sensor_solution_is_feasible

Determine whether the planned or on-going Sensor_Solution is still feasible.

determine_sensor_requirement_progress

Identify what progress has been made against the Requirement.

5.4.2.53.7.1.2 Sensor_Resourcing

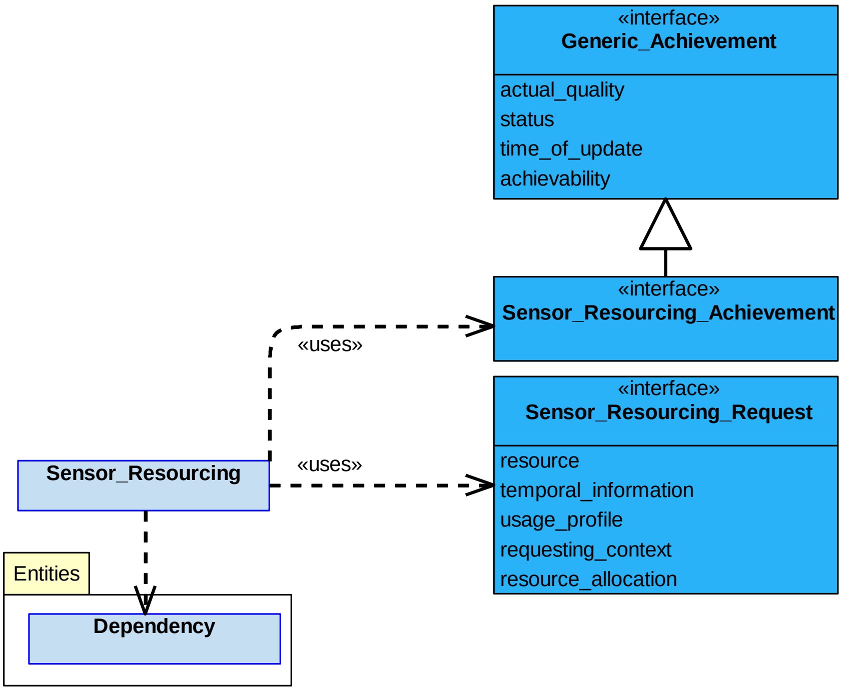


Figure 926: Sensor_Resourcing Service Definition

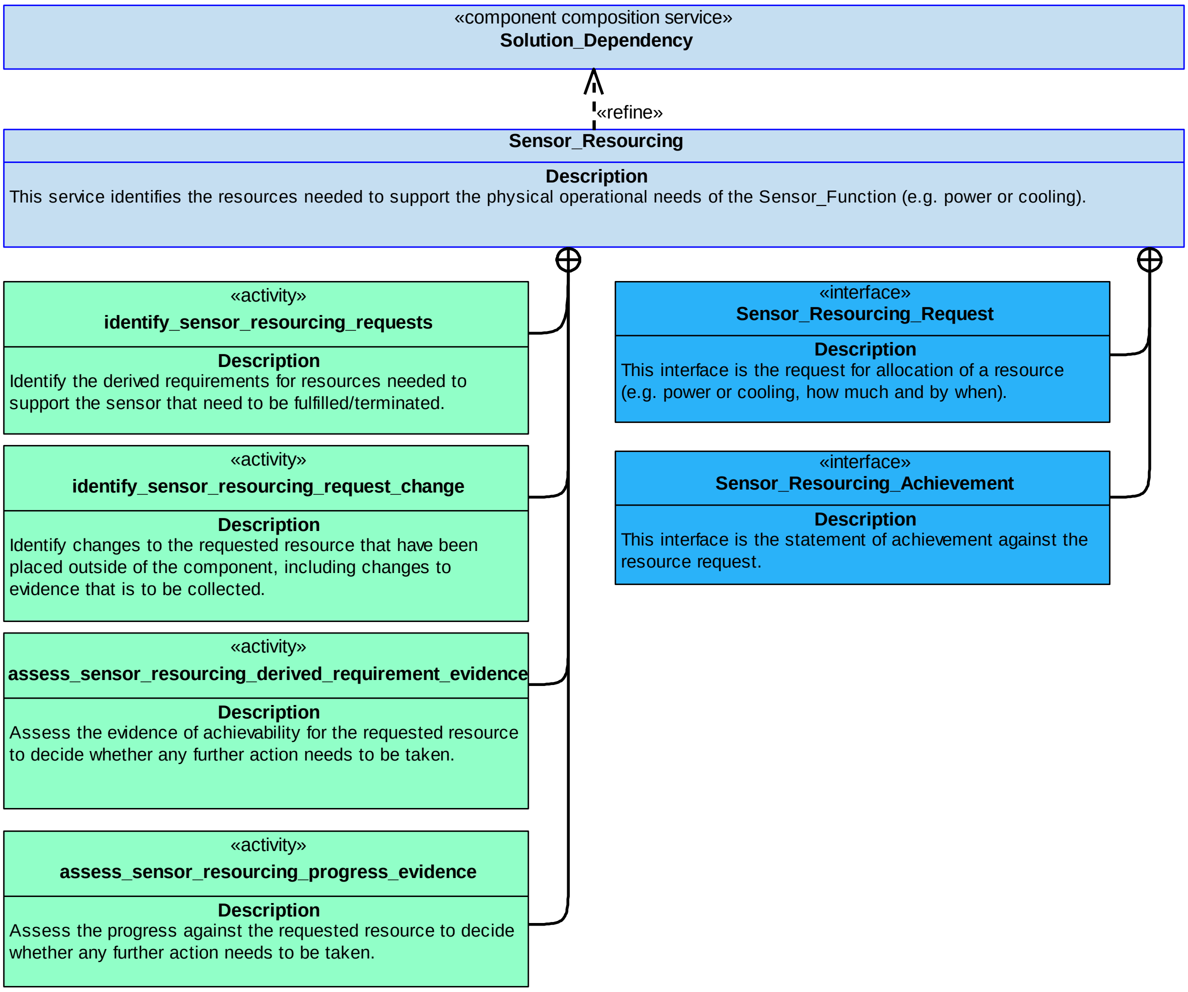

«component composition service»
Solution_Dependency
«refine»
Sensor_Resourcing
Description
This service identifies the resources needed to support the physical operational needs of the Sensor_Function (e.g. power or cooling).
«activity»
identify_sensor_resourcing_requests
Description
Identify the derived requirements for resources needed to support the sensor that need to be fulfilled/terminated.
«activity»
identify_sensor_resourcing_request_change
Description
Identify changes to the requested resource that have been placed outside of the component, including changes to evidence that is to be collected.
«activity»
assess_sensor_resourcing_derived_requirement_evidence
Description
Assess the evidence of achievability for the requested resource to decide whether any further action needs to be taken.
«activity»
assess_sensor_resourcing_progress_evidence
Description
Assess the progress against the requested resource to decide whether any further action needs to be taken.
«interface»
Sensor_Resourcing_Request
Description
This interface is the request for allocation of a resource (e.g. power or cooling, how much and by when).
«interface»
Sensor_Resourcing_Achievement
Description
This interface is the statement of achievement against the resource request.

Figure 927: Sensor_Resourcing Service Policy

Sensor_Resourcing

This service identifies the resources needed to support the physical operational needs of the Sensor_Function (e.g. power or cooling).

Interfaces

Sensor_Resourcing_Request

This interface is the request for allocation of a resource (e.g. power or cooling, how much and by when).

Attributes

resource	The resource being requested (e.g. power or cooling).
temporal_information	Information covering timing for the requested resource, such as start and end times. This might include segments of a requested time window that must not be interrupted, etc.
usage_profile	The quantity of resource requested for use, e.g. a one-off amount or a variable amount, an example being 10 kW for a specified period of time.
requesting_context	The information that identifies the source or reason for the request.

resource_allocation	The actual allocated resource quantity required to meet the usage_profile.

Sensor_Resourcing_Achievement

This interface is the statement of achievement against the resource request.

Activities

identify_sensor_resourcing_requests

Identify the derived requirements for resources needed to support the sensor that need to be fulfilled/terminated.

identify_sensor_resourcing_request_change

Identify changes to the requested resource that have been placed outside of the component, including changes to evidence that is to be collected.

assess_sensor_resourcing_derived_requirement_evidence

Assess the evidence of achievability for the requested resource to decide whether any further action needs to be taken.

assess_sensor_resourcing_progress_evidence

Assess the progress against the requested resource to decide whether any further action needs to be taken.

5.4.2.53.7.1.3 Sensor_Orientation_and_Location

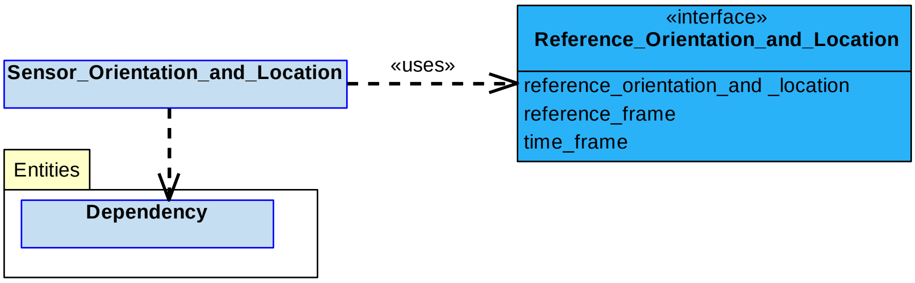

Sensor_Orientation_and_Location
«uses»
«interface»
Reference_Orientation_and_Location
reference_orientation_and _location
reference_frame
time_frame
Entities
Dependency

Figure 928: Sensor_Orientation_and_Location Service Definition

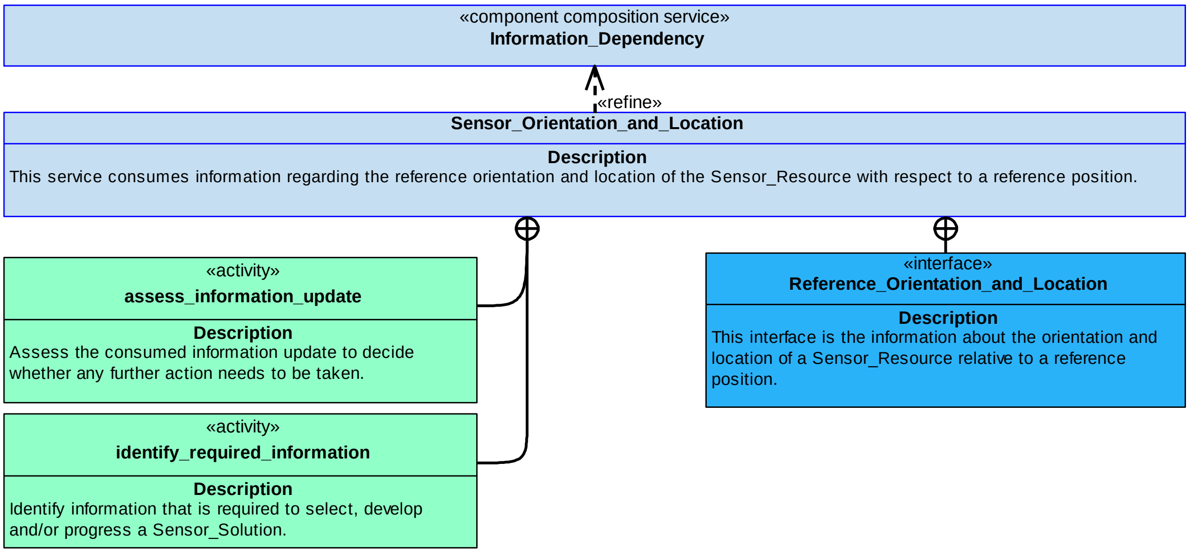

«component composition service»
Information_Dependency
«refine»
Sensor_Orientation_and_Location
Description
This service consumes information regarding the reference orientation and location of the Sensor_Resource with respect to a reference position.
«activity»
assess_information_update
Description
Assess the consumed information update to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify information that is required to select, develop and/or progress a Sensor_Solution.
«interface»
Reference_Orientation_and_Location
Description
This interface is the information about the orientation and location of a Sensor_Resource relative to a reference position.

Figure 929: Sensor_Orientation_and_Location Service Policy

Sensor_Orientation_and_Location

This service consumes information regarding the reference orientation and location of the Sensor_Resource with respect to a reference position.

Interface

Reference_Orientation_and_Location

This interface is the information about the orientation and location of a Sensor_Resource relative to a reference position.

Attributes

reference_orientation_and_location	This is the offset between the reference position and the Sensor_Resource position.
reference_frame	A physical reference point for the data consumed.
time_frame	A temporal reference point for the data consumed.

Activities

assess_information_update

Assess the consumed information update to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to select, develop and/or progress a Sensor_Solution.

5.4.2.53.7.1.4 Sensor_Capability

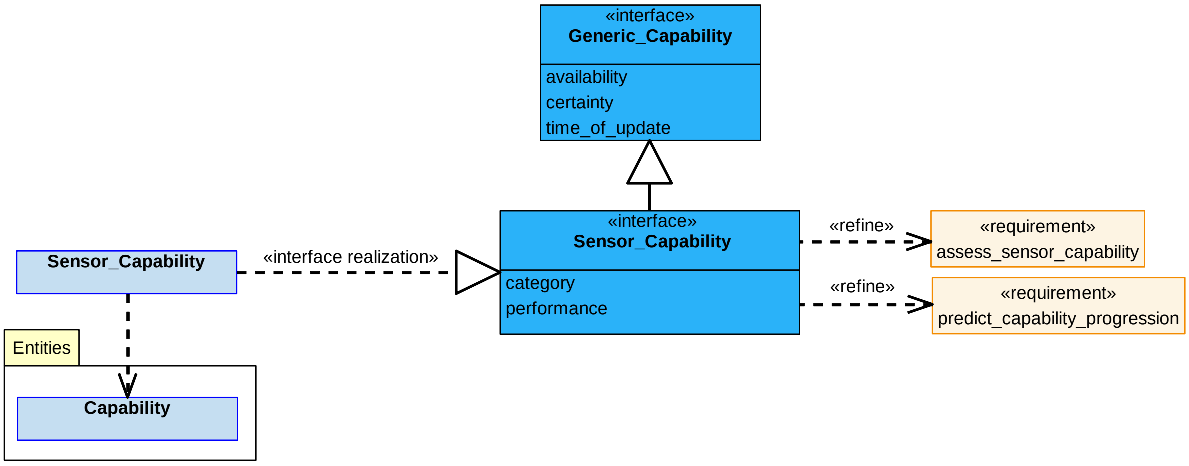


Figure 930: Sensor_Capability Service Definition

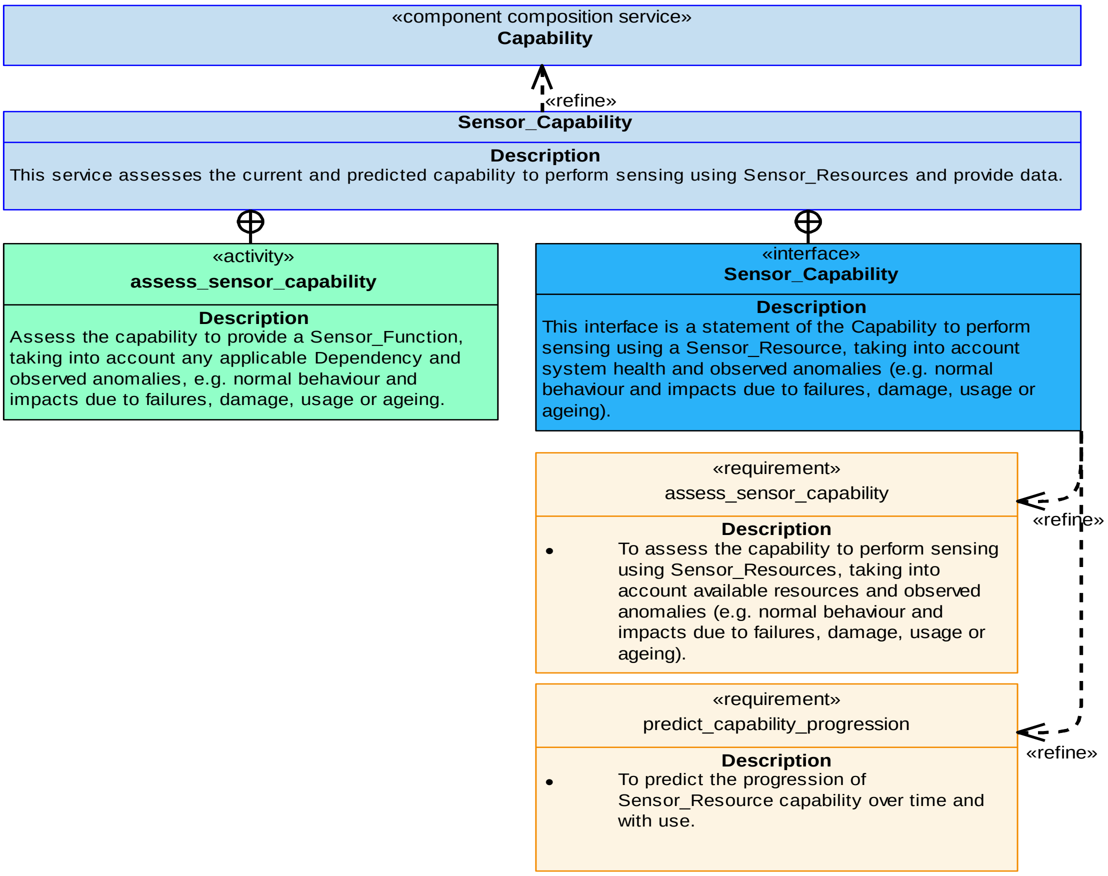

«component composition service»
Capability
«refine»
Sensor_Capability
Description
This service assesses the current and predicted capability to perform sensing using Sensor_Resources and provide data.
«activity»
assess_sensor_capability
Description
Assess the capability to provide a Sensor_Function, taking into account any applicable Dependency and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.
«interface»
Sensor_Capability
Description
This interface is a statement of the Capability to perform sensing using a Sensor_Resource, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«requirement»
assess_sensor_capability
Description
To assess the capability to perform sensing using Sensor_Resources, taking into account available resources and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of Sensor_Resource capability over time and with use.
«refine»

Figure 931: Sensor_Capability Service Policy

Sensor_Capability

This service assesses the current and predicted capability to perform sensing using Sensor_Resources and provide data.

Interface

Sensor_Capability

This interface is a statement of the Capability to perform sensing using a Sensor_Resource, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Attributes

category	The type of Sensor_Function (e.g. passive radar detection, electro-optics imaging or thermal Imaging).
performance	The level of performance or effectiveness that the Sensor_Function can provide (e.g. the spectrum available, latency or fidelity of surveillance).

Activity

assess_sensor_capability

Assess the capability to provide a Sensor_Function, taking into account any applicable Dependency and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.

5.4.2.53.7.1.5 Sensor_Capability_Evidence

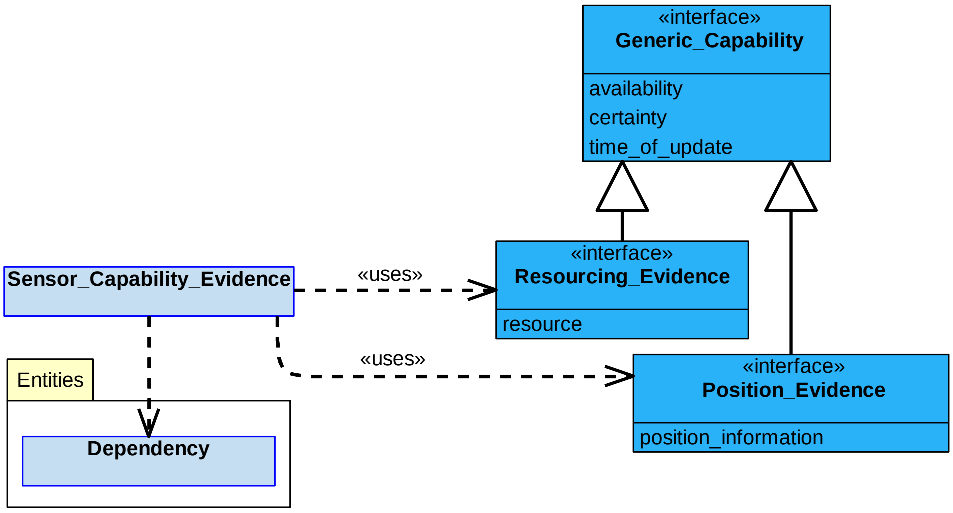


Figure 932: Sensor_Capability_Evidence Service Definition

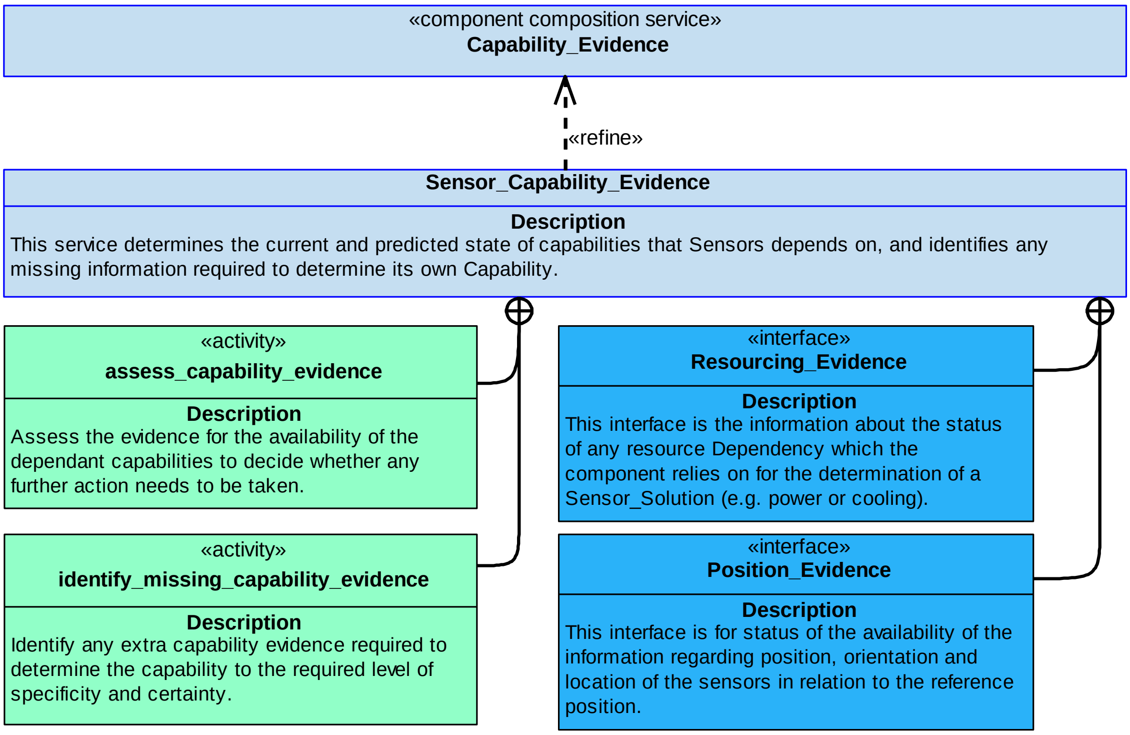

«component composition service»
Capability_Evidence
«refine»
Sensor_Capability_Evidence
Description
This service determines the current and predicted state of capabilities that Sensors depends on, and identifies any missing information required to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the evidence for the availability of the dependant capabilities to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.
«interface»
Resourcing_Evidence
Description
This interface is the information about the status of any resource Dependency which the component relies on for the determination of a Sensor_Solution (e.g. power or cooling).
«interface»
Position_Evidence
Description
This interface is for status of the availability of the information regarding position, orientation and location of the sensors in relation to the reference position.

Figure 933: Sensor_Capability_Evidence Service Policy

Sensor_Capability_Evidence

This service determines the current and predicted state of capabilities that Sensors depends on, and identifies any missing information required to determine its own Capability.

Interfaces

Resourcing_Evidence

This interface is the information about the status of any resource Dependency which the component relies on for the determination of a Sensor_Solution (e.g. power or cooling).

Attribute

resource	The resource used to support a Sensor_Resource, e.g. power or cooling.

Position_Evidence

This interface is for status of the availability of the information regarding position, orientation and location of the sensors in relation to the reference position.

Attribute

position_information	The type of information relating to position, orientation and location.

Activities

assess_capability_evidence

Assess the evidence for the availability of the dependant capabilities to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.53.7.2 Service Dependencies

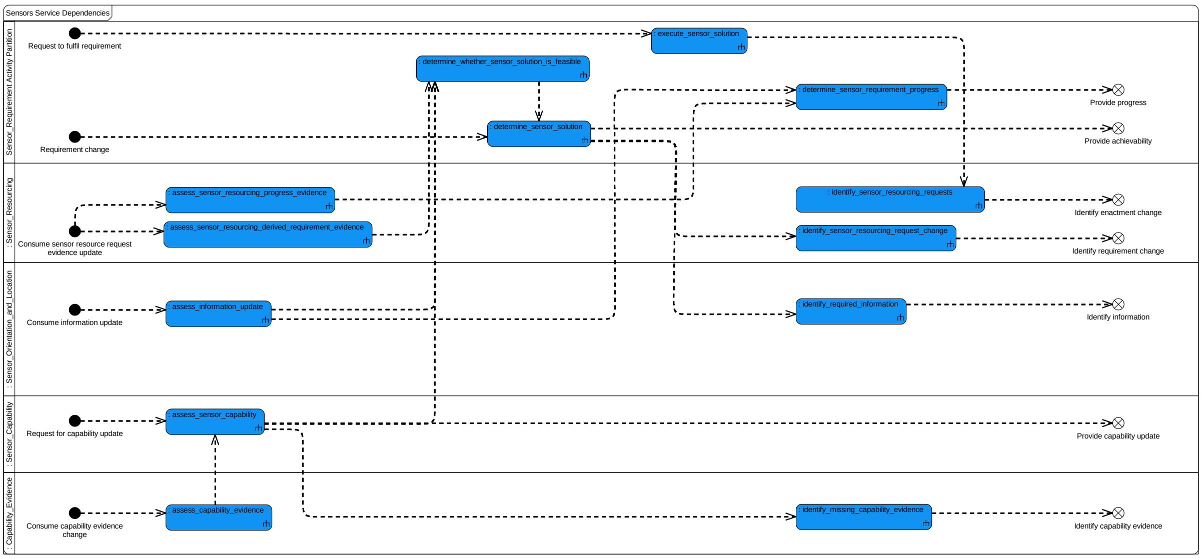

Sensors Service Dependencies
Sensor_Requirement Activity Partition
: Sensor_Resourcing
: Sensor_Orientation_and_Location
: Sensor_Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
Consume sensor resource request evidence update
Consume information update
Request for capability update
Consume capability evidence change
: execute_sensor_solution
: determine_whether_sensor_solution_is_feasible
: determine_sensor_solution
: determine_sensor_requirement_progress
: assess_sensor_resourcing_progress_evidence
: assess_sensor_resourcing_derived_requirement_evidence
: identify_sensor_resourcing_requests
: identify_sensor_resourcing_request_change
: assess_information_update
: identify_required_information
: assess_sensor_capability
: assess_capability_evidence
: identify_missing_capability_evidence
Provide progress
Provide achievability
Identify enactment change
Identify requirement change
Identify information
Provide capability update
Identify capability evidence

Figure 934: Sensors Service Dependencies

5.4.2.54 Signature

5.4.2.54.1 Role

The role of Signature is to calculate or estimate the signatures of objects, irrespective of whether an object is external to the host system.

5.4.2.54.2 Overview

Control Architecture

Signature is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Signature provides tactical information about an Object's Signature across a range of Phenomenon, and determines the Signature arising from changes to the Object's State. The Signature applies to a given Aspect of the Object.

Examples of Use

Signature will be used in order to:

- Identify own vehicle signatures, such as the RCS under a specific State (e.g. with apertures open or whilst emitting).
- Determine signatures of other objects, such as the heat signature of a flare package proposed to defeat an incoming missile.

5.4.2.54.3 Service Summary

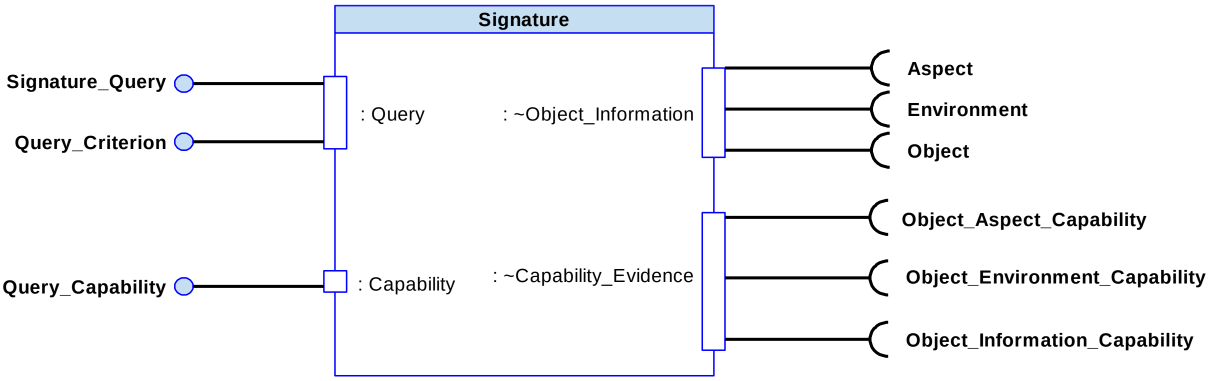


Figure 935: Signature Service Summary

5.4.2.54.4 Responsibilities

capture_requirements_for_signature_calculations

- To capture provided Signature requirements (e.g. determine own vehicle infrared emissions for a given Aspect) for the Signature calculations.

capture_measurement_criteria_for_signature_calculations

- To capture provided Measurement_Criterion/criteria for Signatures (i.e. that the signature determination is within a specified margin of error).

determine_object_emissions

- To determine the emissions of an Object in a given State.

determine_object_reflections

- To determine the reflections from an Object in a given State from a given Aspect.

determine_signature_for_provided_configuration

- To determine the Signature of an Object that would result from the Object implementing or enacting a proposed State.

determine_configuration

- To determine a State that satisfies a required Signature level.

determine_quality_of_deliverables

- To determine the quality of the deliverables provided by Signature during execution, measured against given requirements and the Measurement_Criterion/criteria.

assess_signature_capability

- To assess the Capability to provide the Signature calculations taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Signature assessment Capability.

predict_capability_progression

- To predict the progression of Signature's Capability over time and with use.

5.4.2.54.5 Subject Matter Semantics

The subject matter of Signature is the spectral signature of objects.

Exclusions

The subject matter of Signature does not include:

- Whether the Object can be detected, only the determination of its Signature is included.
- The commandment of vehicle State changes in order to alter the signature of an Object.

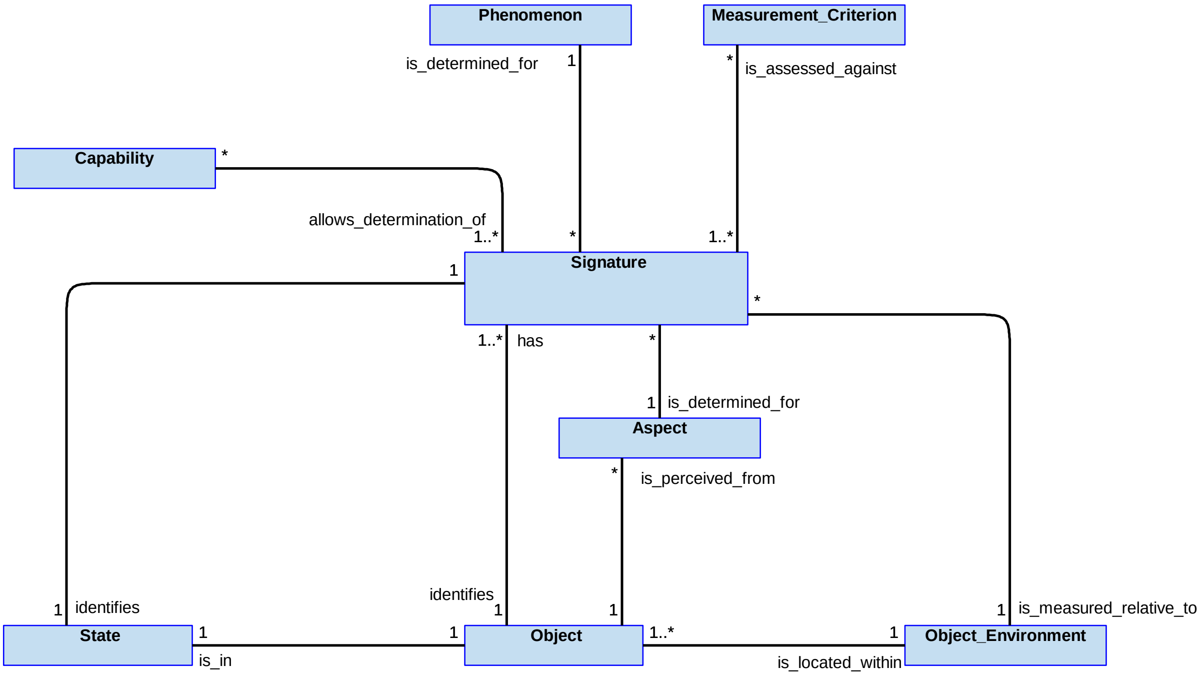


Figure 936: Signature Semantics

5.4.2.54.5.1 Entities

Aspect

The direction (with respect to some object axis) which an Object can be perceived from.

Capability

The capability of the Signature component based upon the availability of the information required to respond to queries.

Measurement_Criterion

Something by which the quality of the determined Signature will be measured.

Object

An item that can emit or reflect energy (e.g. a vehicle or an on-board emitter).

Phenomenon

An energy transmission mechanism (e.g. electromagnetic, acoustic, magnetism or radioactivity).

Signature

A unique identification of an Object, element of an Object or the State of an Object based on the Object's energy emission or reflection characteristics. The energy may be measured relative to the background environment.

Object_Environment

The environment local to an Object, e.g. the air density due to the altitude.

State

An arrangement, mode or configuration (e.g. an aperture is open or an engine is in reheat).

5.4.2.54.6 Design Rationale

5.4.2.54.6.1 Assumptions

- The observability of objects will be determined by another component.

5.4.2.54.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Signature:

- Data Driving - To mitigate the significant variances in the Signature Phenomenon types and attributes (e.g. the radar cross section or IR signature at different frequencies of emission or reflection).

Extensions

- The Signature component primarily calculates the signature for a wide variety of signature types (e.g. electromagnetic, acoustic, thermal or magnetic). The component could use multiple extension components to handle different features and any associated algorithms.

5.4.2.54.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component could result in the air vehicle being observed by enemy forces when not intended. Therefore, the air vehicle may be subjected to physical attack from enemy forces (e.g. missile attack). However, this is normally excluded from safety analysis. Therefore, an IDAL no more onerous than DAL C, is considered appropriate for this component.

5.4.2.54.6.4 Security Considerations

The indicative security classification is SNEO.

The component deals with Signature information for different Objects (including the Exploiting Platform). This can be to optimise own stealth characteristics or to aid identification of others based on their signature. The algorithms involved are likely to be SNEO. Where necessary, there may be instances in different security domains, e.g. to cater for different sensors or intelligence data. These instances may need to communicate with each other to provide a full signature assessment. If so, separation will be handled externally to the component. Any loss of integrity or availability in the output of this component may lead to the Exploiting Platform placing itself in a situation where its signature may be observable by hostile forces, or to the misidentification of others. The confidentiality, integrity and availability requirements will need to reflect this. Where algorithms are data-driven, the associated configuration data will also carry appropriate confidentiality requirements.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of signature assessments made during the course of a mission.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.54.7 Services

5.4.2.54.7.1 Service Definitions

5.4.2.54.7.1.1 Query

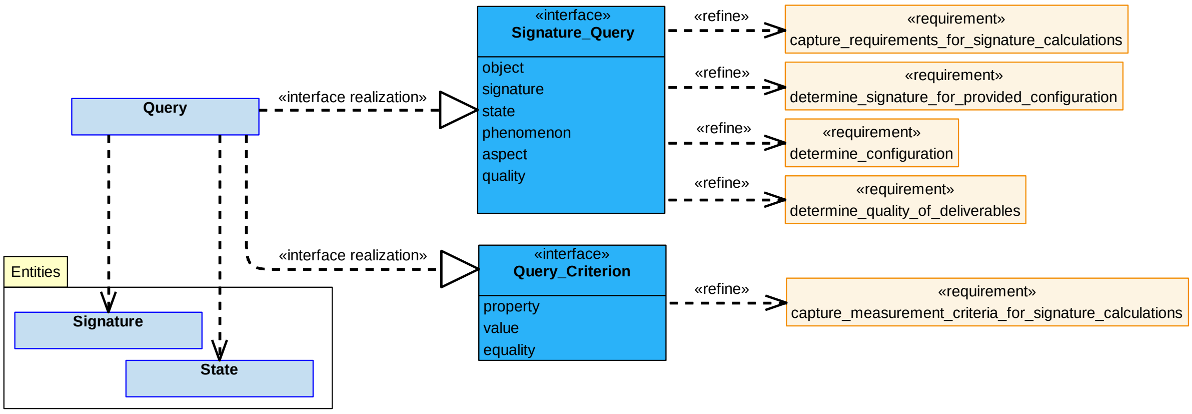


Figure 937: Query Service Definition

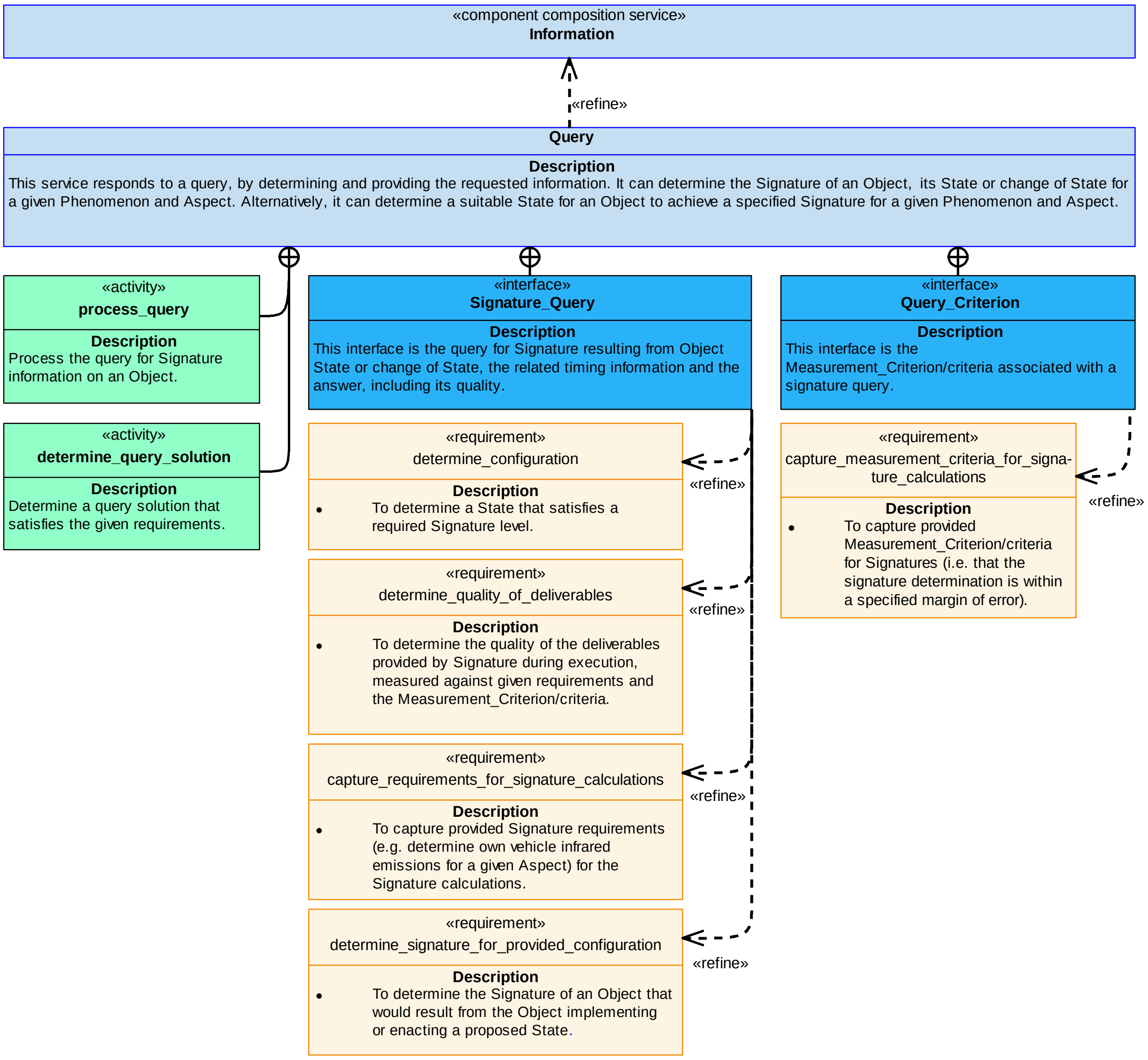

«component composition service»
Information
«refine»
Query
Description
This service responds to a query, by determining and providing the requested information. It can determine the Signature of an Object, its State or change of State for a given Phenomenon and Aspect. Alternatively, it can determine a suitable State for an Object to achieve a specified Signature for a given Phenomenon and Aspect.
«activity»
process_query
Description
Process the query for Signature information on an Object.
«activity»
determine_query_solution
Description
Determine a query solution that satisfies the given requirements.
«interface»
Signature_Query
Description
This interface is the query for Signature resulting from Object State or change of State, the related timing information and the answer, including its quality.
«interface»
Query_Criterion
Description
This interface is the Measurement_Criterion/criteria associated with a signature query.
«requirement»
determine_configuration
Description
To determine a State that satisfies a required Signature level.
«refine»
«requirement»
capture_measurement_criteria_for_signature_calculations
Description
To capture provided Measurement_Criterion/criteria for Signatures (i.e. that the signature determination is within a specified margin of error).
«refine»
«requirement»
determine_quality_of_deliverables
Description
To determine the quality of the deliverables provided by Signature during execution, measured against given requirements and the Measurement_Criterion/criteria.
«refine»
«requirement»
capture_requirements_for_signature_calculations
Description
To capture provided Signature requirements (e.g. determine own vehicle infrared emissions for a given Aspect) for the Signature calculations.
«refine»
«requirement»
determine_signature_for_provided_configuration
Description
To determine the Signature of an Object that would result from the Object implementing or enacting a proposed State.
«refine»

Figure 938: Query Service Policy

Query

This service responds to a query, by determining and providing the requested information. It can determine the Signature of an Object, its State or change of State for a given Phenomenon and Aspect. Alternatively, it can determine a suitable State for an Object to achieve a specified Signature for a given Phenomenon and Aspect.

Interfaces

Signature_Query

This interface is the query for Signature resulting from Object State or change of State, the related timing information and the answer, including its quality.

Attributes

object	The Object that the query is about.
signature	Information relating to the Objects Signature for the query.

state	The current or proposed State of the Object.
phenomenon	The Phenomenon being considered for the query.
aspect	Information relating to the Object Aspect for the query.
quality	The quality of the determined query response.

Query_Criterion

This interface is the Measurement_Criterion/criteria associated with a signature query.

Attributes

property	The criterion property to be determined.
value	The amount related to the property to be measured.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

process_query

Process the query for Signature information on an Object.

determine_query_solution

Determine a query solution that satisfies the given requirements.

5.4.2.54.7.1.2 Object_Information

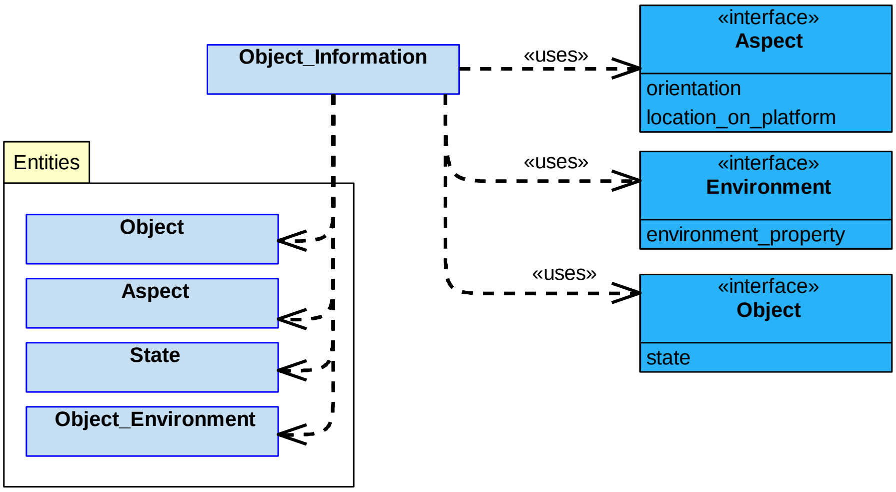


Figure 939: Object_Information Service Definition

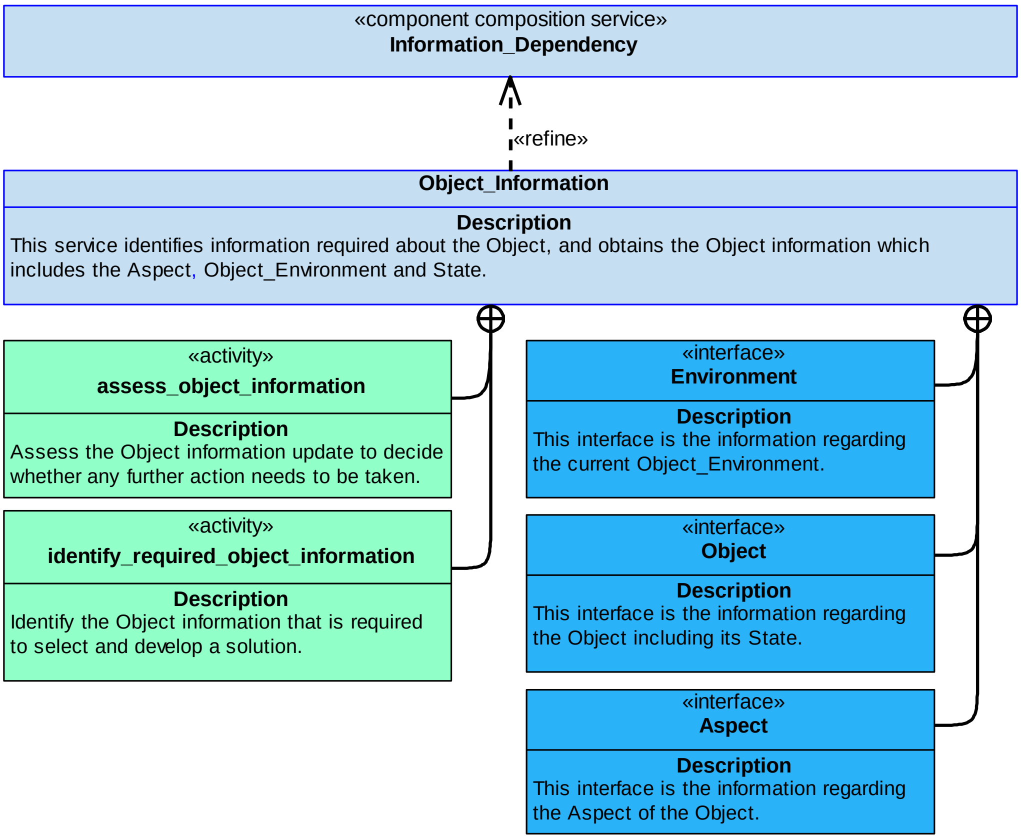

«component composition service»
Information_Dependency
«refine»
Object_Information
Description
This service identifies information required about the Object, and obtains the Object information which includes the Aspect, Object_Environment and State.
«activity»
assess_object_information
Description
Assess the Object information update to decide whether any further action needs to be taken.
«activity»
identify_required_object_information
Description
Identify the Object information that is required to select and develop a solution.
«interface»
Environment
Description
This interface is the information regarding the current Object_Environment.
«interface»
Object
Description
This interface is the information regarding the Object including its State.
«interface»
Aspect
Description
This interface is the information regarding the Aspect of the Object.

Figure 940: Object Information Service Policy

Object_Information

This service identifies information required about the Object, and obtains the Object information which includes the Aspect, Object_Environment and State.

Interfaces

Object

This interface is the information regarding the Object including its State.

Attribute

state	A parameter relating to the State of the Object.

Environment

This interface is the information regarding the current Object_Environment.

Attribute

environment_property	A parameter relating to the Object_Environment.

Aspect

This interface is the information regarding the Aspect of the Object.

Attributes

orientation	A parameter relating to the orientation of the Object, with respect to some object axis.
location_on_platform	For Objects that are located on a larger platform (e.g. a transmitter located on an air vehicle), this parameter relates to the location of the Object relative to the platform.

Activities

identify_required_object_information

Identify the Object information that is required to select and develop a solution.

assess_object_information

Assess the Object information update to decide whether any further action needs to be taken.

5.4.2.54.7.1.3 Capability

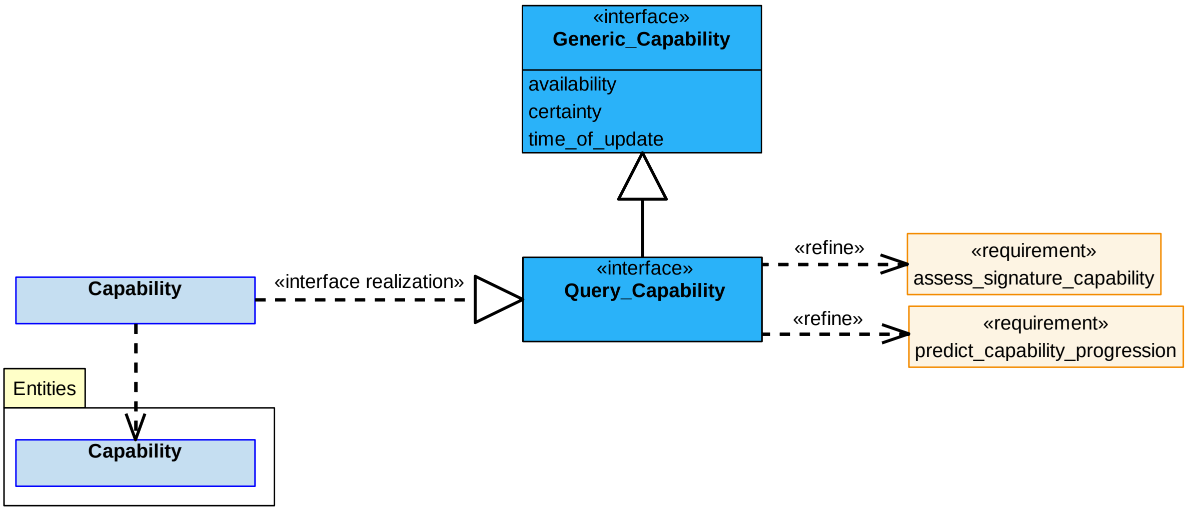


Figure 941: Capability Service Definition

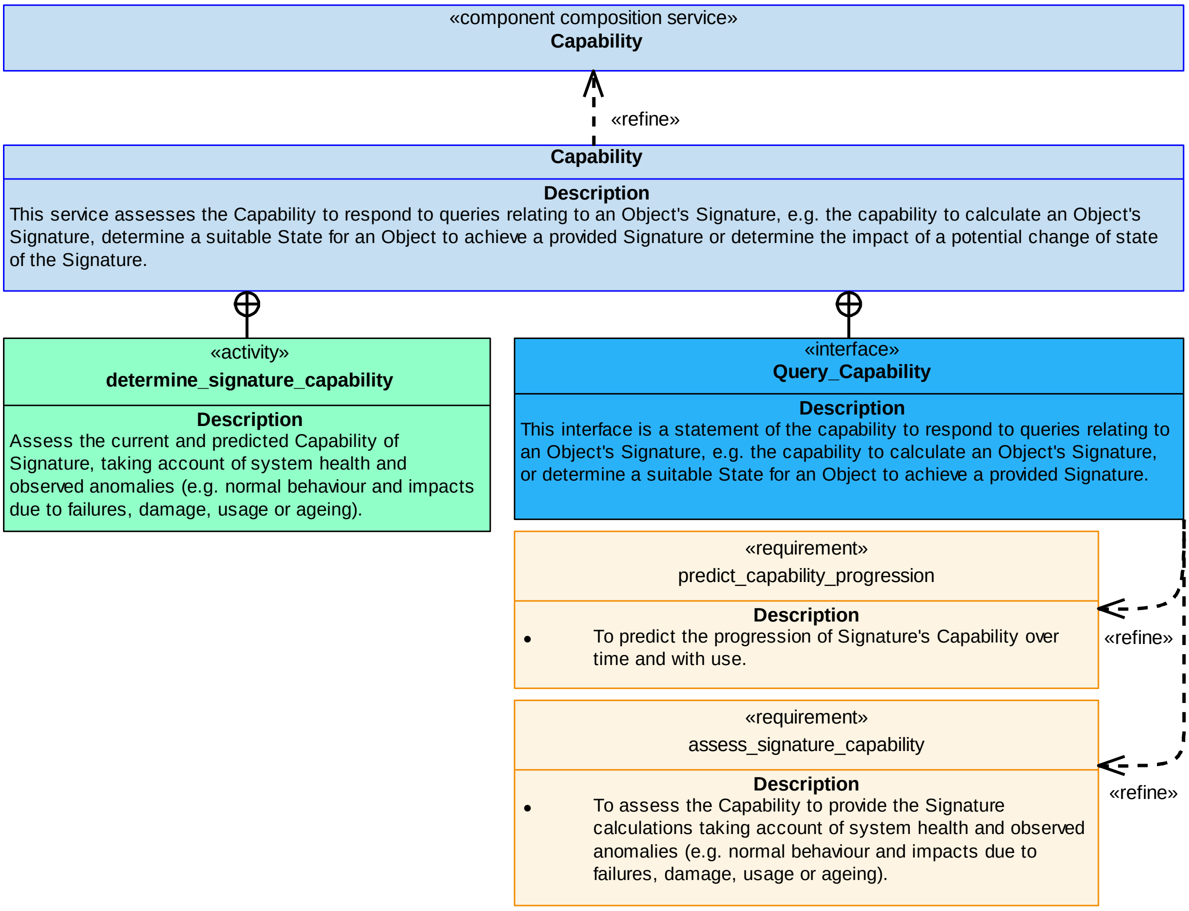

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the Capability to respond to queries relating to an Object's Signature, e.g. the capability to calculate an Object's Signature, determine a suitable State for an Object to achieve a provided Signature or determine the impact of a potential change of state of the Signature.
«activity»
determine_signature_capability
Description
Assess the current and predicted Capability of Signature, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Query_Capability
Description
This interface is a statement of the capability to respond to queries relating to an Object's Signature, e.g. the capability to calculate an Object's Signature, or determine a suitable State for an Object to achieve a provided Signature.
«requirement»
predict_capability_progression
Description
To predict the progression of Signature's Capability over time and with use.
«refine»
«requirement»
assess_signature_capability
Description
To assess the Capability to provide the Signature calculations taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 942: Capability Service Policy

Capability

This service assesses the Capability to respond to queries relating to an Object's Signature, e.g. the capability to calculate an Object's Signature, determine a suitable State for an Object to achieve a provided Signature or determine the impact of a potential change of state of the Signature.

Interface

Query_Capability

This interface is a statement of the capability to respond to queries relating to an Object's Signature, e.g. the capability to calculate an Object's Signature, or determine a suitable State for an Object to achieve a provided Signature.

Activity

determine_signature_capability

Assess the current and predicted Capability of Signature, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.54.7.1.4 Capability_Evidence

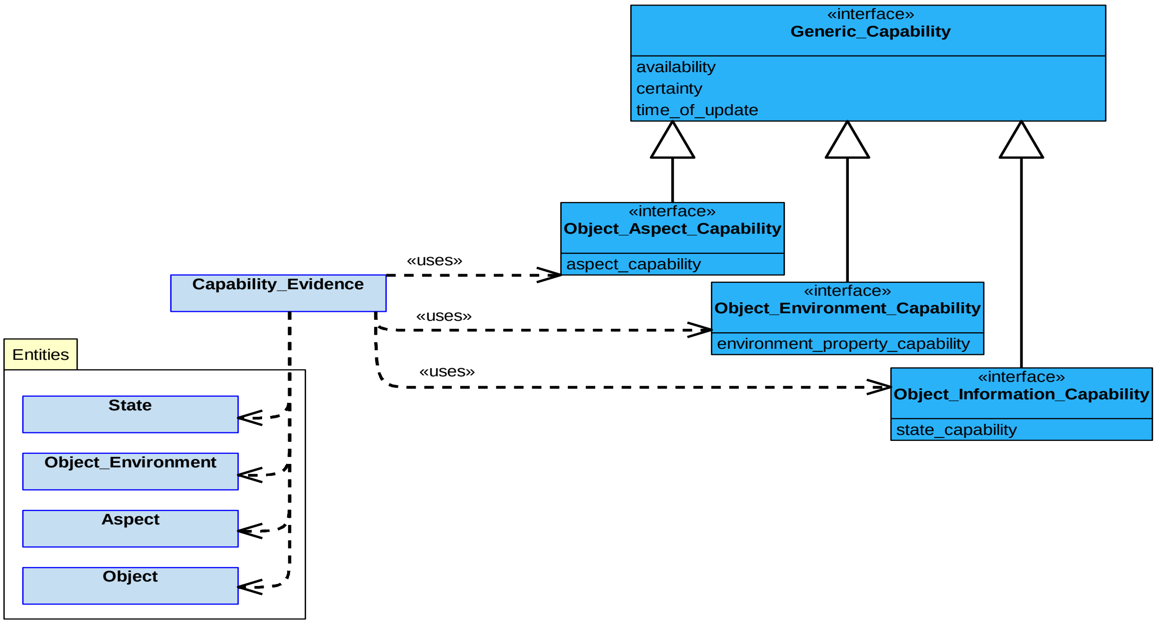


Figure 943: Capability_Evidence Service Definition

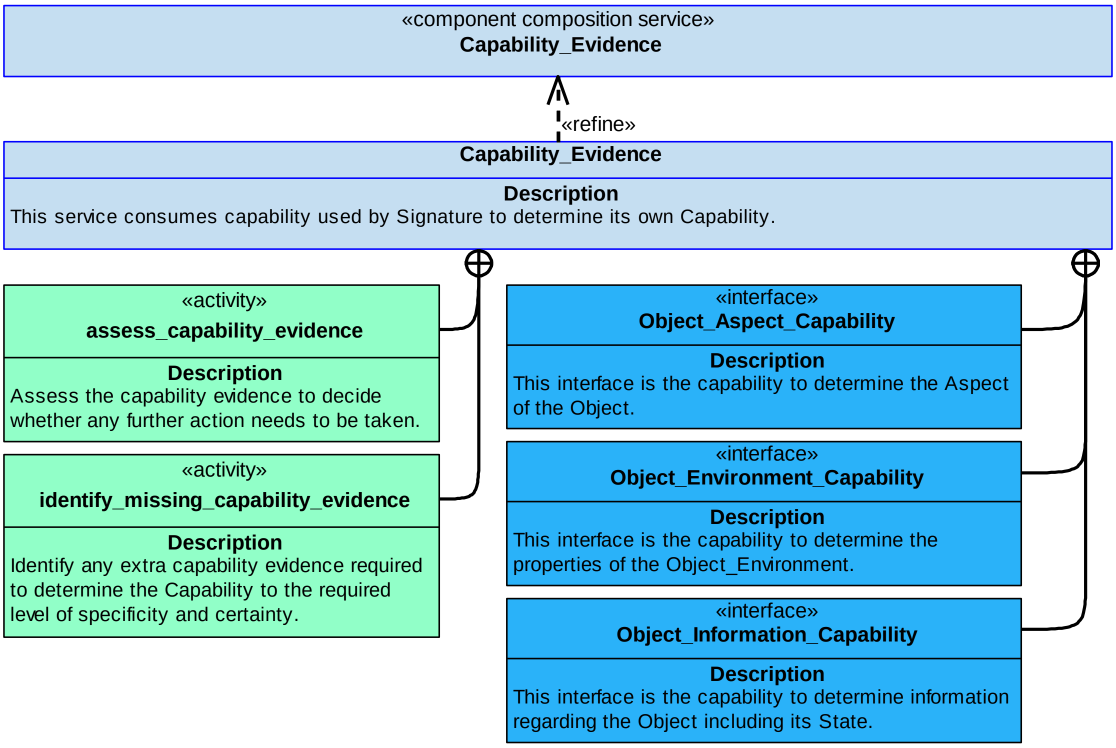

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capability used by Signature to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Object_Aspect_Capability
Description
This interface is the capability to determine the Aspect of the Object.
«interface»
Object_Environment_Capability
Description
This interface is the capability to determine the properties of the Object_Environment.
«interface»
Object_Information_Capability
Description
This interface is the capability to determine information regarding the Object including its State.

Figure 944: Capability Evidence Service Policy

Capability_Evidence

This service consumes capability used by Signature to determine its own Capability.

Interfaces

Object_Environment_Capability

This interface is the capability to determine the properties of the Object_Environment.

Attribute

environment_property_capability	A parameter relating to the capability to determine the properties of the Object_Environment.

Object_Information_Capability

This interface is the capability to determine information regarding the Object including its State.

Attribute

state_capability	A parameter relating to the capability to determine the State of the Object.

Object_Aspect_Capability

This interface is the capability to determine the Aspect of the Object.

Attribute

aspect_capability	A parameter relating to the capability to determine the Aspect of the Object.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.54.7.2 Service Dependencies

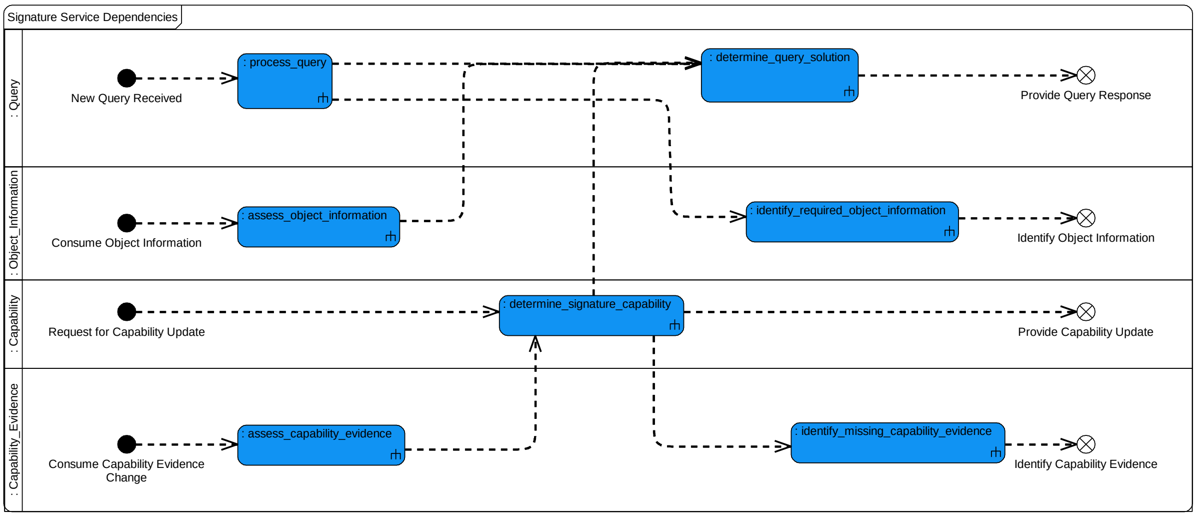

Signature Service Dependencies
: Query
: Object_Information
: Capability
: Capability_Evidence
New Query Received
: process_query
: determine_query_solution
Provide Query Response
Consume Object Information
: assess_object_information
: identify_required_object_information
Identify Object Information
Request for Capability Update
: determine_signature_capability
Provide Capability Update
Consume Capability Evidence Change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify Capability Evidence

Figure 945: Signature Service Dependencies

5.4.2.55 Spatial Correction

5.4.2.55.1 Role

The role of Spatial Correction is to determine the correction to the spatial relationship between different positions to account for physical effects.

5.4.2.55.2 Overview

Control Architecture

Spatial Correction is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Following a Requirement for a correction in response to a specific type of Physical_Effect, Spatial Correction uses Condition information to determine the Spatial_Correction needed to accommodate its impact.

Examples of Use

This component can be used where there is a need to:

- Align a sensor or weapon with a reference point in a Coordinate_Frame or to align multiple sensors or weapons, to allow for aero-elastic deformation of the airframe.
- Compensate for Physical_Effects on emissions to or from the Exploiting Platform, such as to calculate radio wave propagation between two antennas.

5.4.2.55.3 Service Summary

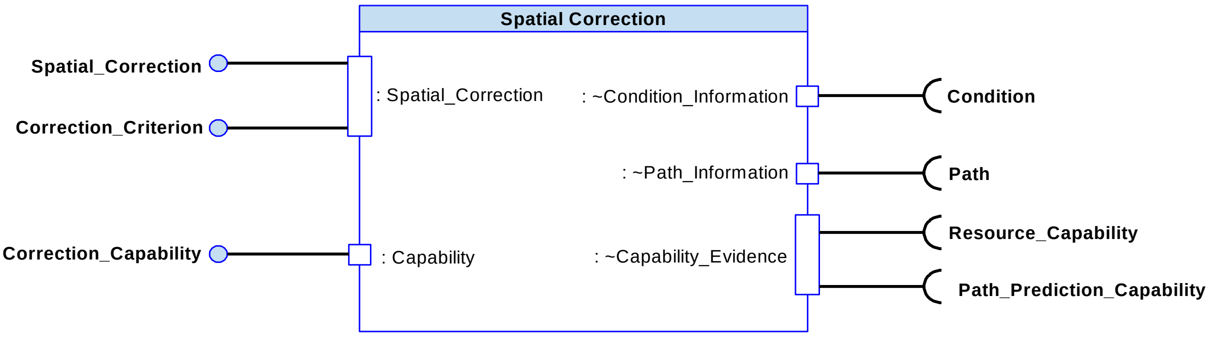


Figure 946: Spatial Correction Service Summary

5.4.2.55.4 Responsibilities

capture_requirements_for_spatial_correction

- To capture provided Requirements for calculating Spatial_Corrections.

capture_measurement_criteria

- To capture given Measurement_Criterion.

determine_spatial_correction

- To determine the Spatial_Corrections resulting from a Physical_Effect.

determine_quality_of_spatial_correction

- To determine the quality of the Spatial_Correction against the Measurement_Criterion.

assess_capability_to_provide_spatial_correction

- To assess the Capability to determine Spatial_Corrections, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the Capability over time and with use.

5.4.2.55.5 Subject Matter Semantics

The subject matter of Spatial Correction is Physical_Effects that are compensated for by Spatial_Corrections.

Exclusions

The subject matter of Spatial Correction does not include:

- The calculations/methods used by resources to provide any Conditions (e.g. measurements).
- Implementation of the Spatial_Corrections (e.g. a Spatial_Correction may be determined for the position of a sensor, however, the component is not concerned with any changes to the sensor outputs as a result of the Spatial_Correction).

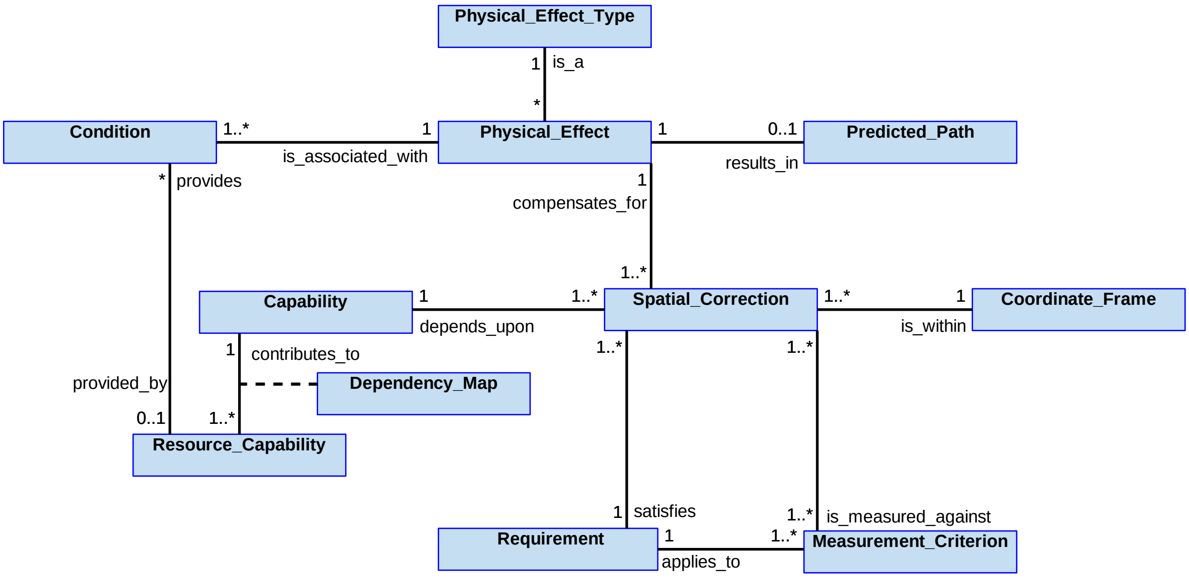


Figure 947: Spatial Correction Semantics

5.4.2.55.5.1 Entities

Capability

The ability of the component to calculate a Spatial_Correction to account for a Physical_Effect.

Condition

Something used to determine the Physical_Effect's magnitude, i.e. measurements (such as weight of a store on a wing pylon) and information derived from measurements (e.g. airspeed, environmental conditions or operating within a given range band).

Coordinate_Frame

The frame of reference used (e.g. aircraft body frame, sensor frame or local geodetic frame).

Dependency_Map

Mapping of how the Capability is dependent on the Resource_Capability.

Physical_Effect

A phenomenon or physical process that causes an observable effect that needs to be compensated for, such as the aeroelastic deformation of an Exploiting Platform and radio wave or acoustic propagation.

Physical_Effect_Type

The kinds of phenomenon or physical process that Spatial Correction knows how to compensate for.

Requirement

A requirement to calculate a correction to a position or spatial relationship caused by a known phenomenon, e.g. of a point on the Exploiting Platform's physical structure affected by aeroelastic deformation.

Resource_Capability

The resource capability (e.g. from sensors or other data sources) Spatial Correction depends upon to provide the input information necessary to calculate the Spatial_Corrections.

Spatial_Correction

The correction required to address a change in a position or spatial relationship. Both angular and linear elements may be accounted for, and may be provided as a corrected value or the delta between uncorrected and corrected values.

Measurement_Criterion

A criterion by which the quality of a Spatial_Correction will be measured against (e.g. any thresholds or update rates that apply to a Spatial_Correction, or the confidence in a determined Spatial_Correction).

Predicted_Path

The path that an object is predicted to follow given Physical_Effects, e.g. the predicted path of the highest power part of an RF signal over a specified distance.

5.4.2.55.6 Design Rationale

5.4.2.55.6.1 Assumptions

- The Exploiting Platform operates within a performance envelope where a Physical_Effect, such as aero-elastic deformation, remains predictable. Where limits of predictable behaviour are breached, this might result in limited or no Capability to determine a Spatial_Correction.

5.4.2.55.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Spatial Correction:

- Data Driving - It may be appropriate to update the information relating to positions of interest within the Coordinate_Frame and how they may be affected by various phenomena through data driving. For example, this may include data relating to a new type of role-fit sensor, including its datum points, weight and aerodynamic properties, and the way wing bending or flutter affects its position. Similarly, any updates to a model used to predict the effects may be suitable candidates for data driving, for example, any variation to assumed conditions, mathematical coefficients or constants.

Extensions

- Extensions may be used for the different types of Physical_Effect that may be addressed.

Exploitation Considerations

- The corrections may be derived by different means, for example, this could mean wing deflection is derived from a model using airspeed and altitude, or though the extrapolation of measurements of wing angle at known points.

- Where Physical_Effects apply differently in different parts of the envelope or range bands, it may be necessary to have multiple models to calculate the Spatial_Correction.

5.4.2.55.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Failure of this component could cause uncontrolled flight of the Exploiting Platform if it leads to incorrect demands being placed on Mechanical Positioning relating to control surfaces. Whilst the Exploiting Platform would be within its aerodynamic limits, the control surfaces would not be positioned as intended and hence the path of the Exploiting Platform would not be controlled. The result is likely to be loss of the Exploiting Platform and fatalities. Therefore, the indicative DAL is A.
- Additionally, failure of this component could cause the incorrect geolocation of an object that is subsequently targeted by weapons or misdirection of support to a weapon post release from the host Exploiting Platform (e.g. laser designation). This could cause the weapon to strike a location not intended by the crew, resulting in unintended harm to third parties. DAL B is appropriate for this failure condition, but it is not the driving case.

Where instances of this component contribute to hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

5.4.2.55.6.4 Security Considerations

The indicative security classification is O.

This component determines the Spatial_Corrections required in order to compensate for Physical_Effects, such as airframe deformation or acoustic propagation. It requires no knowledge of the Exploiting Platform’s functional capabilities, only certain physical characteristics (e.g. the weight of a store attached to the wing and the wings stiffness) and as such this component is considered likely to be O. Where those characteristics do provide additional insight into the overall Exploiting Platform's capability (e.g. for RF propagation), this will drive a higher requirement for confidentiality. Loss of integrity or availability may lead to incorrect or no Spatial_Correction being performed, with possible consequences for safety, performance and operational capability.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to changes in conditions and data, used in determining the correction.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.55.7 Services

5.4.2.55.7.1 Service Definitions

5.4.2.55.7.1.1 Spatial_Correction

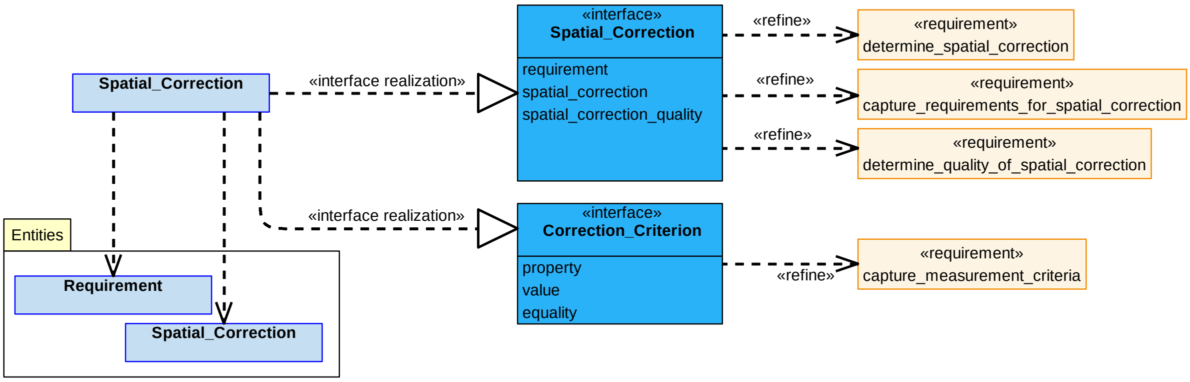


Figure 948: Spatial_Correction Service Definition

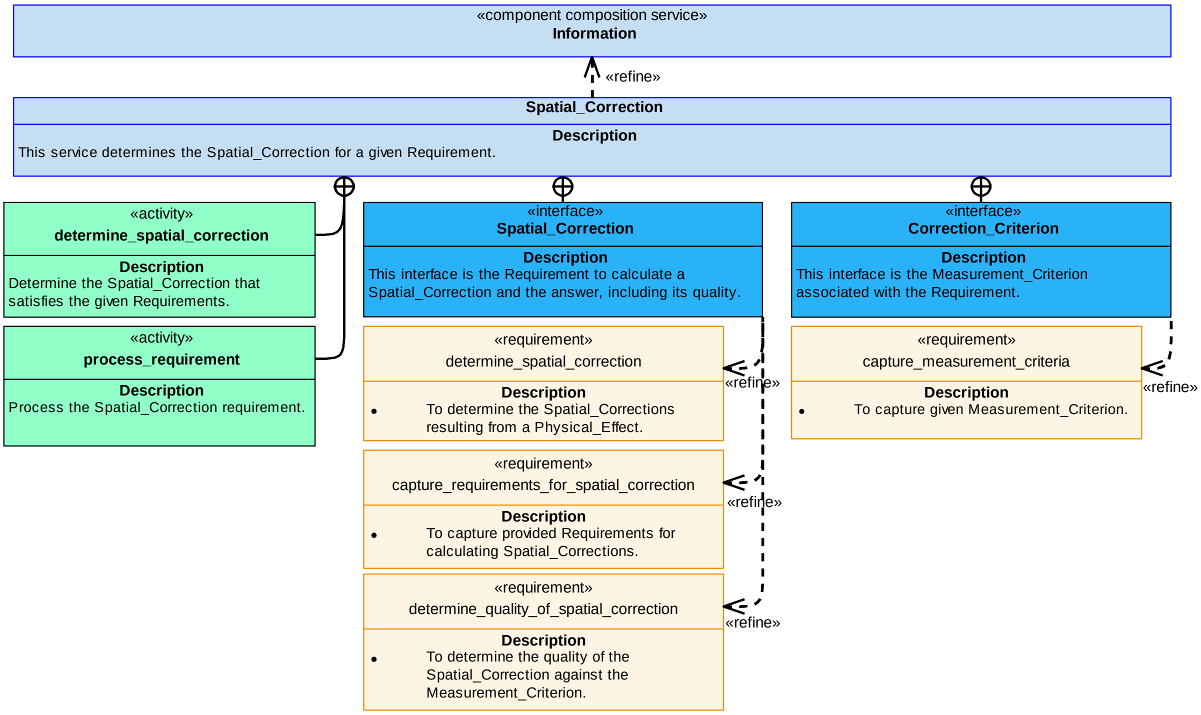

«component composition service»
Information
«refine»
Spatial_Correction
Description
This service determines the Spatial_Correction for a given Requirement.
«activity»
determine_spatial_correction
Description
Determine the Spatial_Correction that satisfies the given Requirements.
«activity»
process_requirement
Description
Process the Spatial_Correction requirement.
«interface»
Spatial_Correction
Description
This interface is the Requirement to calculate a Spatial_Correction and the answer, including its quality.
«interface»
Correction_Criterion
Description
This interface is the Measurement_Criterion associated with the Requirement.
«requirement»
determine_spatial_correction
Description
To determine the Spatial_Corrections resulting from a Physical_Effect.
«refine»
«requirement»
capture_measurement_criteria
Description
To capture given Measurement_Criterion.
«refine»
«requirement»
capture_requirements_for_spatial_correction
Description
To capture provided Requirements for calculating Spatial_Corrections.
«refine»
«requirement»
determine_quality_of_spatial_correction
Description
To determine the quality of the Spatial_Correction against the Measurement_Criterion.
«refine»

Figure 949: Spatial_Correction Service Policy

Spatial_Correction

This service determines the Spatial_Correction for a given Requirement.

Interfaces

Spatial_Correction

This interface is the Requirement to calculate a Spatial_Correction and the answer, including its quality.

Attributes

requirement	The Requirement under consideration.
spatial_correction	The Spatial_Correction.
spatial_correction_quality	The quality of the Spatial_Correction (e.g. 90% confident).

Correction_Criterion

This interface is the Measurement_Criterion associated with the Requirement.

Attributes

property	The criterion property to be determined, e.g. confidence.
value	The amount related to the property to be measured, e.g. a percentage.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Activities

process_requirement

Process the Spatial_Correction requirement.

determine_spatial_correction

Determine the Spatial_Correction that satisfies the given Requirements.

5.4.2.55.7.1.2 Condition_Information

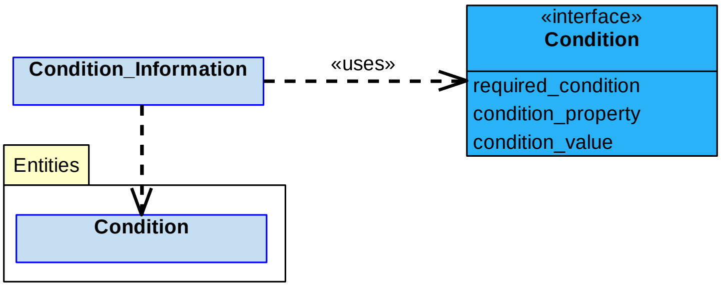


Figure 950: Condition_Information Service Definition

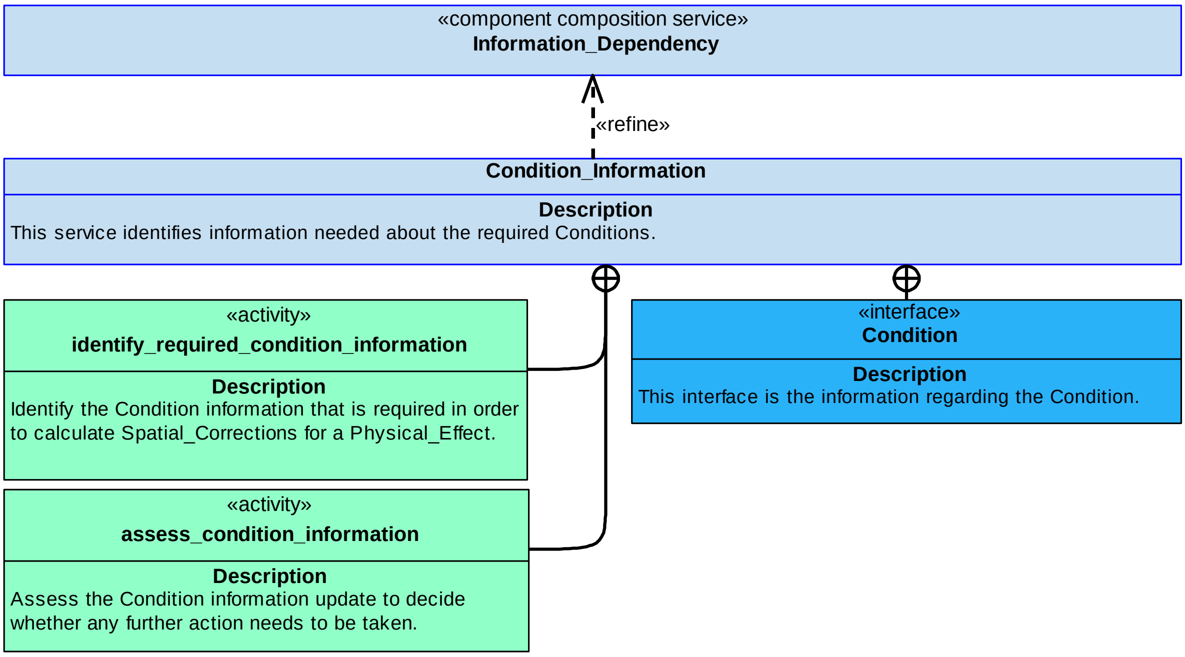

«component composition service»
Information_Dependency
«refine»
Condition_Information
Description
This service identifies information needed about the required Conditions.
«activity»
identify_required_condition_information
Description
Identify the Condition information that is required in order to calculate Spatial_Corrections for a Physical_Effect.
«activity»
assess_condition_information
Description
Assess the Condition information update to decide whether any further action needs to be taken.
«interface»
Condition
Description
This interface is the information regarding the Condition.

Figure 951: Condition_Information Service Policy

Condition_Information

This service identifies information needed about the required Conditions.

Interface

Condition

This interface is the information regarding the Condition.

Attributes

required_condition	The Condition information that is required in order to calculate a Spatial_Correction.
condition_property	A property relating to the Condition (e.g. mass, weight or frequency).
condition_value	The value of the condition_property.

Activities

identify_required_condition_information

Identify the Condition information that is required in order to calculate Spatial_Corrections for a Physical_Effect.

assess_condition_information

Assess the Condition information update to decide whether any further action needs to be taken.

5.4.2.55.7.1.3 Path_Information

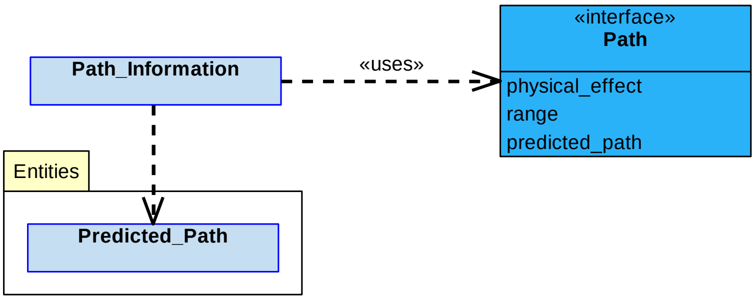


Figure 952: Path_Information Service Definition

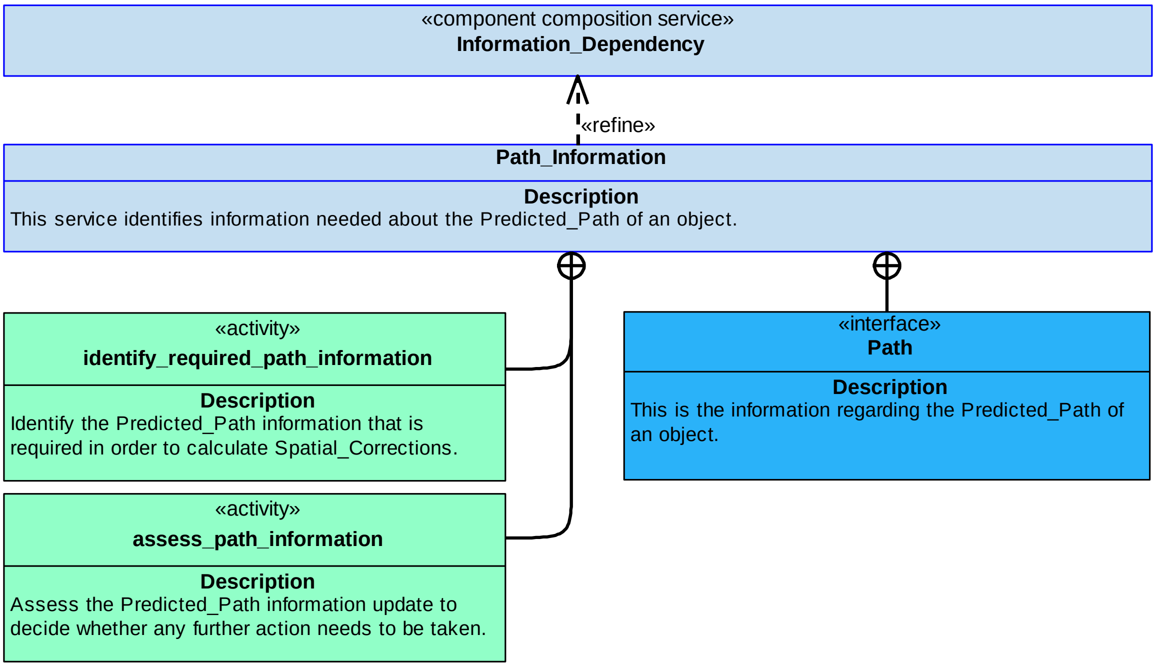

«component composition service»
Information_Dependency
«refine»
Path_Information
Description
This service identifies information needed about the Predicted_Path of an object.
«activity»
identify_required_path_information
Description
Identify the Predicted_Path information that is required in order to calculate Spatial_Corrections.
«interface»
Path
Description
This is the information regarding the Predicted_Path of an object.
«activity»
assess_path_information
Description
Assess the Predicted_Path information update to decide whether any further action needs to be taken.

Figure 953: Path_Information Service Policy

Path_Information

This service identifies information needed about the Predicted_Path of an object.

Interface

Path

This is the information regarding the Predicted_Path of an object.

Attributes

physical_effect	The Physical_Effect.
range	The range, over which the predicted_path is required.
predicted_path	The Predicted_Path.

Activities

identify_required_path_information

Identify the Predicted_Path information that is required in order to calculate Spatial_Corrections.

assess_path_information

Assess the Predicted_Path information update to decide whether any further action needs to be taken.

5.4.2.55.7.1.4 Capability

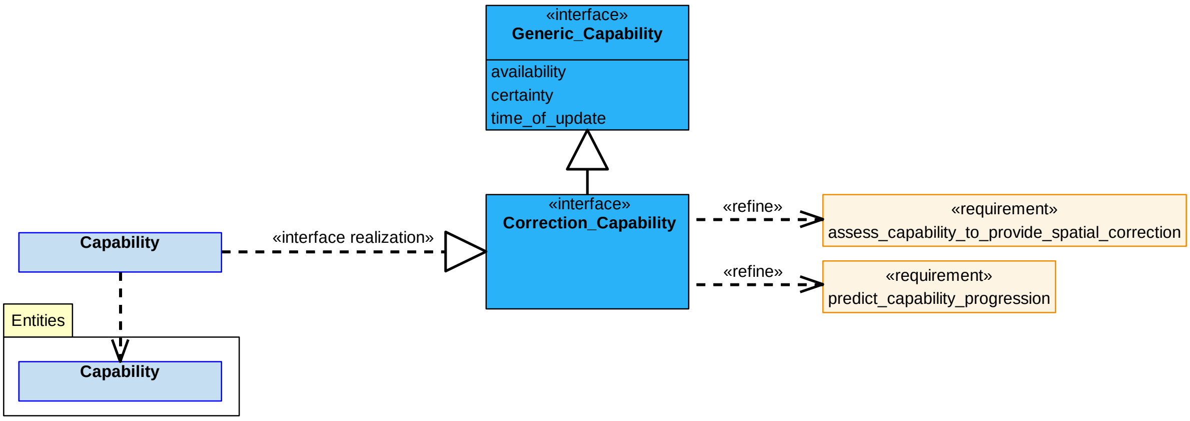


Figure 954: Capability Service Definition

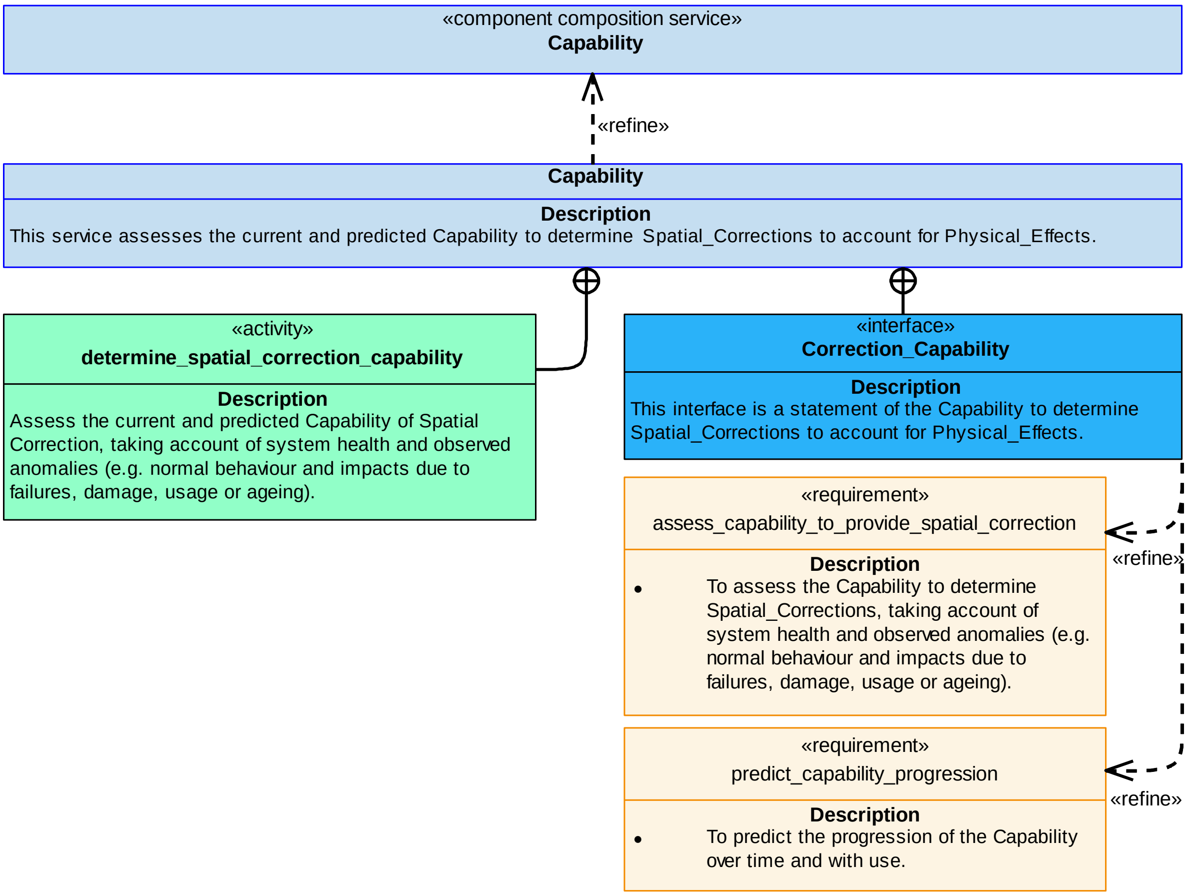

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to determine Spatial_Corrections to account for Physical_Effects.
«activity»
determine_spatial_correction_capability
Description
Assess the current and predicted Capability of Spatial Correction, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Correction_Capability
Description
This interface is a statement of the Capability to determine Spatial_Corrections to account for Physical_Effects.
«requirement»
assess_capability_to_provide_spatial_correction
Description
To assess the Capability to determine Spatial_Corrections, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the Capability over time and with use.
«refine»

Figure 955: Capability Service Policy

Capability

This service assesses the current and predicted Capability to determine Spatial_Corrections to account for Physical_Effects.

Interface

Correction_Capability

This interface is a statement of the Capability to determine Spatial_Corrections to account for Physical_Effects.

Activity

determine_spatial_correction_capability

Assess the current and predicted Capability of Spatial Correction, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.55.7.1.5 Capability_Evidence

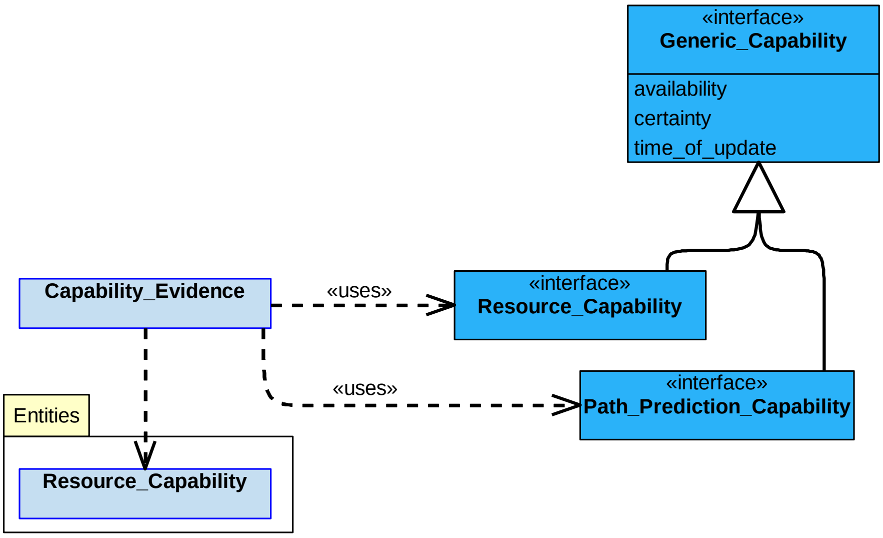


Figure 956: Capability_Evidence Service Definition

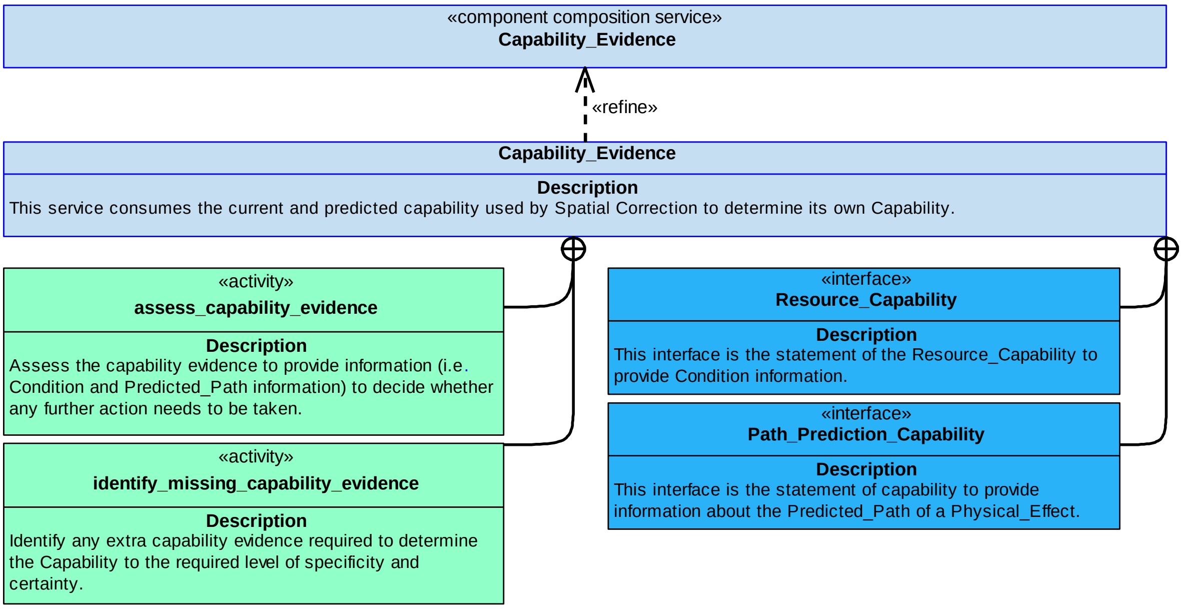

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted capability used by Spatial Correction to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to provide information (i.e. Condition and Predicted_Path information) to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Resource_Capability
Description
This interface is the statement of the Resource_Capability to provide Condition information.
«interface»
Path_Prediction_Capability
Description
This interface is the statement of capability to provide information about the Predicted_Path of a Physical_Effect.

Figure 957: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capability used by Spatial Correction to determine its own Capability.

Interfaces

Resource_Capability

This interface is the statement of the Resource_Capability to provide Condition information.

Path_Prediction_Capability

This interface is the statement of capability to provide information about the Predicted_Path of a Physical_Effect.

<u>Activities</u>

assess_capability_evidence

Assess the capability evidence to provide information (i.e. Condition and Predicted_Path information) to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.55.7.2 Service Dependencies

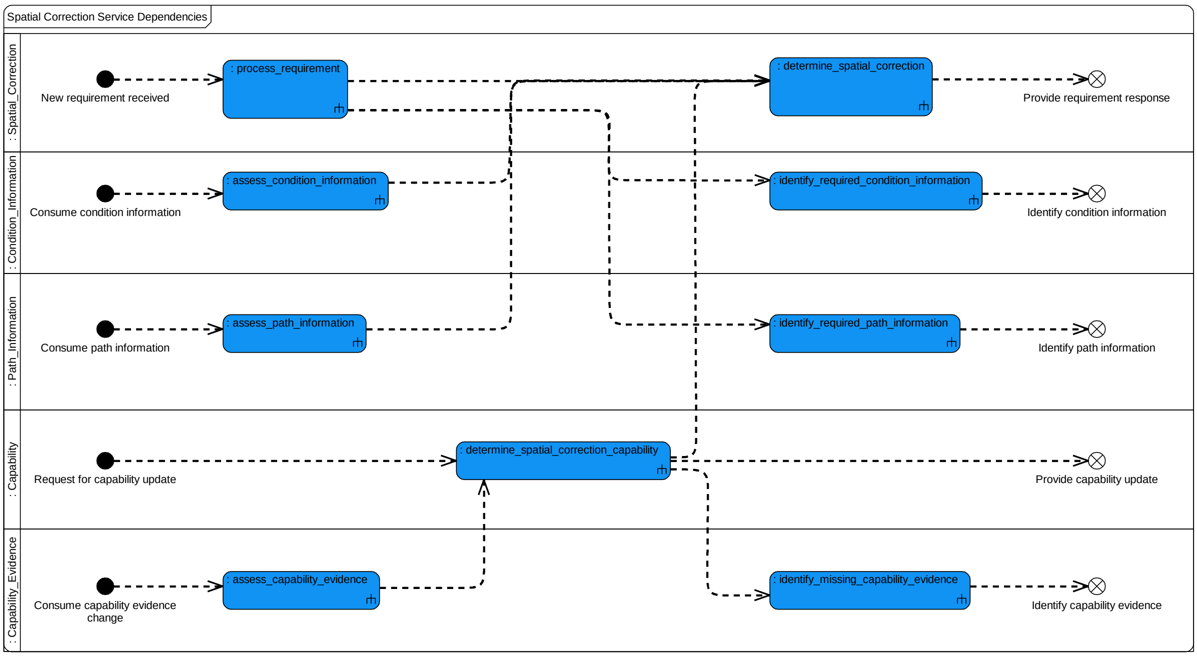

Spatial Correction Service Dependencies
: Spatial_Correction
: Condition_Information
: Path_Information
: Capability
: Capability_Evidence
New requirement received
: process_requirement
: determine_spatial_correction
Provide requirement response
Consume condition information
: assess_condition_information
: identify_required_condition_information
Identify condition information
Consume path information
: assess_path_information
: identify_required_path_information
Identify path information
Request for capability update
determine_spatial_correction_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 958: Spatial Correction Service Dependencies

5.4.2.56 Spectrum

5.4.2.56.1 Role

The role of Spectrum is to allocate spectrum (i.e. electromagnetic and/or acoustic) to meet demands taking into account internal and external sources of interference.

5.4.2.56.2 Overview

Control Architecture

Spectrum is a resource component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Spectrum is used to coordinate the use of spectrum resources (i.e. acoustic and/or electromagnetic spectrum). This may take the form of an allocation request (e.g. a frequency range, at a given time, in a particular direction, at a given power level). Spectrum will determine allocation solutions to satisfy spectrum demands taking into account existing Spectrum_Constraints and will identify conflicts where a solution cannot be found to satisfy all demands.

Examples of Use

Spectrum will be required to:

- Prevent multiple devices from using frequencies that will cause interference above acceptable levels.
- Prevent or limit the use of a portion of spectrum in accordance with operational conditions (e.g. the presence of a threat or regulatory permissions).

5.4.2.56.3 Service Summary

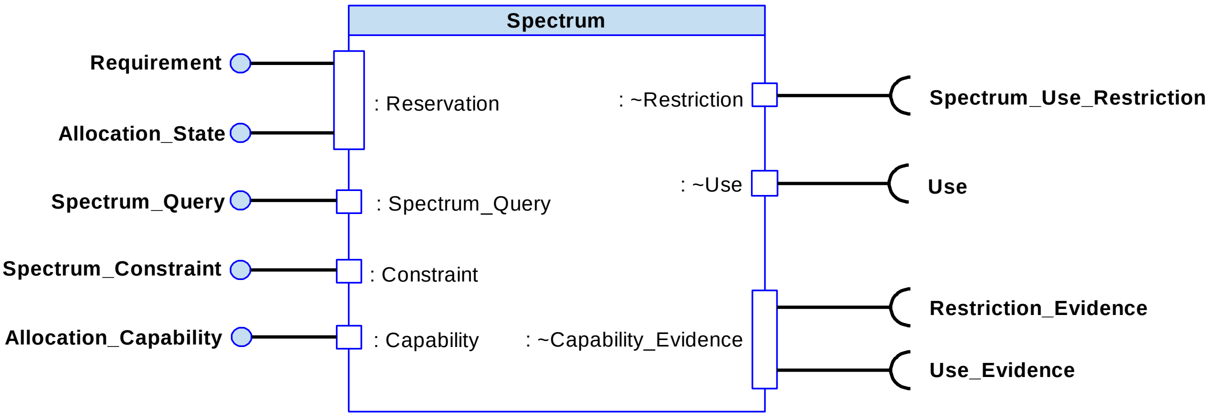


Figure 959: Spectrum Service Summary

5.4.2.56.4 Responsibilities

capture_requirements

- To capture provided Requirements for Spectrum use.

capture_constraints

- To capture provided constraints (e.g. EMCON).

determine_allocation_solution

- To determine a Spectrum_Element_Allocation to satisfy Requirements and Interoperability_Criteria within the Spectrum_Constraints.

determine_available_spectrum

- To determine parts of a Spectrum available for use.

assess_spectrum_capability

- To assess the Capability to allocate Spectrum for use.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

identify_whether_solution_remains_feasible

- To identify whether a planned or ongoing Spectrum_Element_Allocation remains feasible given currently available Spectrum and Spectrum_Constraints.

5.4.2.56.5 Subject Matter Semantics

The subject matter of Spectrum is the interference between uses of electromagnetic and acoustic spectrum and the allocation of elements of the electromagnetic and acoustic spectrum.

Exclusions

The subject matter of Spectrum does not include considerations beyond signal interference and allocation, including:

- Ensuring that spectrum users adhere to their allotted Spectrum_Element_Allocation.
- Ensuring that the use of spectrum will achieve a required function or performance, including selection of appropriate signal characteristics and ensuring observability of signals.
- Preventing transmission that could result in harm (e.g. operating a high power transmitter, on the ground, in the vicinity of the ground crew).

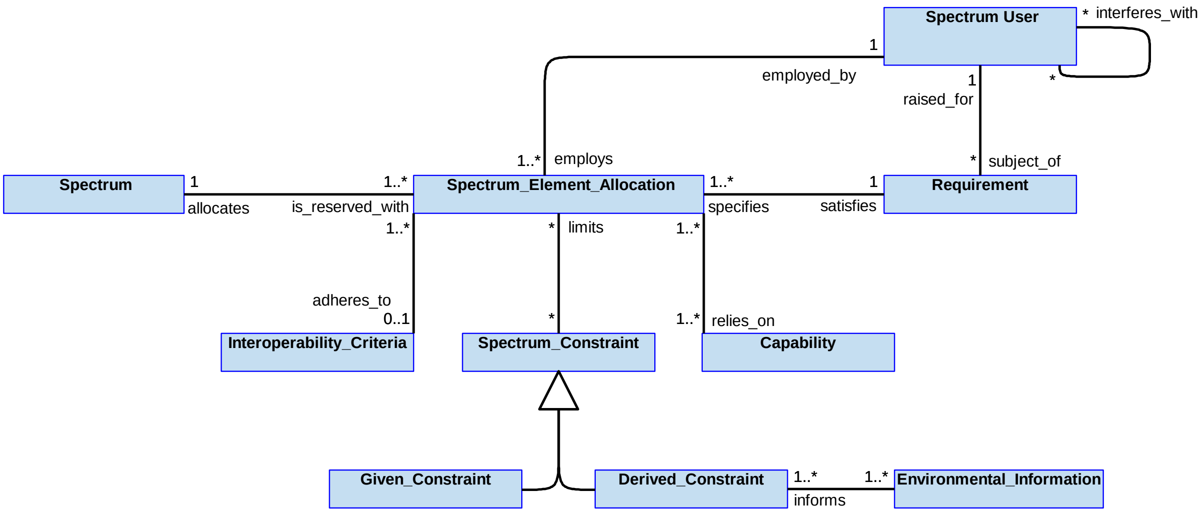


Figure 960: Spectrum Semantics

5.4.2.56.5.1 Entities

Spectrum User

An entity that utilises acoustic or electromagnetic energy in a frequency managed by this component. This includes users of Spectrum that are outside the control of the system.

Requirement

A need for the use of spectrum; this may be a specific frequency range, power level, directionality, time of use and nature of use, e.g. transmission or reception.

Spectrum_Element_Allocation

An allocated portion of spectrum and its defined properties of use, e.g. power, direction, time period.

Spectrum

The spectrum that is being monitored and/or managed (i.e. electromagnetic or acoustic).

Spectrum_Constraint

A limitation on the Spectrum_Element_Allocation and its properties of use.

Capability

The ability to allocate, restrict and confirm availability of Spectrum.

Interoperability_Criteria

The defined criteria in which spectrum users can operate together. This can be in the form of thresholds for acceptable levels of interference, e.g. maximum signal to noise ratio.

Given_Constraint

An imposed limitation by the system on the Spectrum_Element_Allocation, e.g. transmission not being allowed in a particular direction, due to the presence of a threat.

Derived_Constraint

A Spectrum_Constraint determined and defined by this component, taking into account appropriate factors.

Environmental_Information

Information about the local external spectrum conditions and situations to be considered that will influence the derivation of constraints.

5.4.2.56.6 Design Rationale

5.4.2.56.6.1 Assumptions

- Spectrum will cover constructive and destructive interference effects.
- Spectrum will be part of the solution to EM spectrum interoperability, where it can be used in frequency allocation planning and to triage the real time effects of a platform's usage.
- Frequency planning is likely to identify electromagnetic and acoustic reservations for use by equipment across multiple platforms, these can then be claimed at the time that they are needed.
- Not all uses of spectrum will be under the control of this component but the component may still have knowledge about external sources, e.g. through a Given_Constraint.

5.4.2.56.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Spectrum:

- Data Driving - Spectrum allocation interoperability can vary based on the attributes of the fitted equipment, therefore these attributes may be data-driven.
- Multi-Vehicle Coordination - Multiple instances of this component could be used to manage the interactions across multiple vehicles.
- Resource Management - This component manages Spectrum resources in alignment with this PYRAMID concept.

Extensions

- Extension components could be developed to accommodate a variety of emitters, sensor profiles and allocation algorithms.

Exploitation Considerations

- The approach to providing reservations needs to be considered. For example, a notification could be provided to alert the user when a reservation window opens/closes.
- The Spectrum_Query service allows for a potential user of a spectrum to ask about a frequency range availability, to tailor its expectations before making an electromagnetic or acoustic allocation request.

- Directionality is relative to the subject of analysis (e.g. node, antenna, or set of antenna).
- The level of optimisation verses design simplicity suitable for a deployment will be an important consideration, and may need to take account of factors such as the following to an appropriate degree:
    - The dynamically variability of interference between cooperating platforms, such as due to changes in their relative ranges and their relative transmission and reception angles.
    - The presence of, potentially significant, side band and/or out-of-band spectral energy on some transmissions.
    - The potential variability of receiver sensitivity to in-band and out-of-band interference, which may be based on operating modes and settings.
    - The fact that the effects of some types and levels of interference can be more effectively mitigated than other types and levels of interference or by different equipment.

5.4.2.56.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component could cause interference between transmitters and receivers. This could cause the loss of the functions dependant on the transmitters / receivers. This is considered a "large reduction in safety margins" (critical severity) and so the indicative IDAL is DAL B.

Note: It is not a responsibility of the Spectrum component to prevent high power transmitters causing catastrophic accidents if operated in the wrong circumstances (e.g. near ground crew or another air vehicle). In these cases, it is expected that the Interlocks component would inhibit any high power transmission.

5.4.2.56.6.4 Security Considerations

The indicative security classification is SNEO.

This component may have access to highly classified spectrum data and algorithms that, in addition to resolving interoperability issues, may be used to counter threats. Awareness of such data may also reveal the capabilities of the Exploiting Platform should its confidentiality be compromised. The indicative component security classification is therefore thought to be SNEO. Appropriate protective measures will be required.

The component is expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** for later forensic examination.
- **Maintaining Audit Records** to support accountability of spectrum related allocation decisions during the course of a mission.

- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

Supports security enforcing functions by:

- **Applying Emission Control Rules** applicable to the electromagnetic and acoustic spectrum, for example from EMCON.

5.4.2.56.7 Services

5.4.2.56.7.1 Service Definitions

5.4.2.56.7.1.1 Reservation

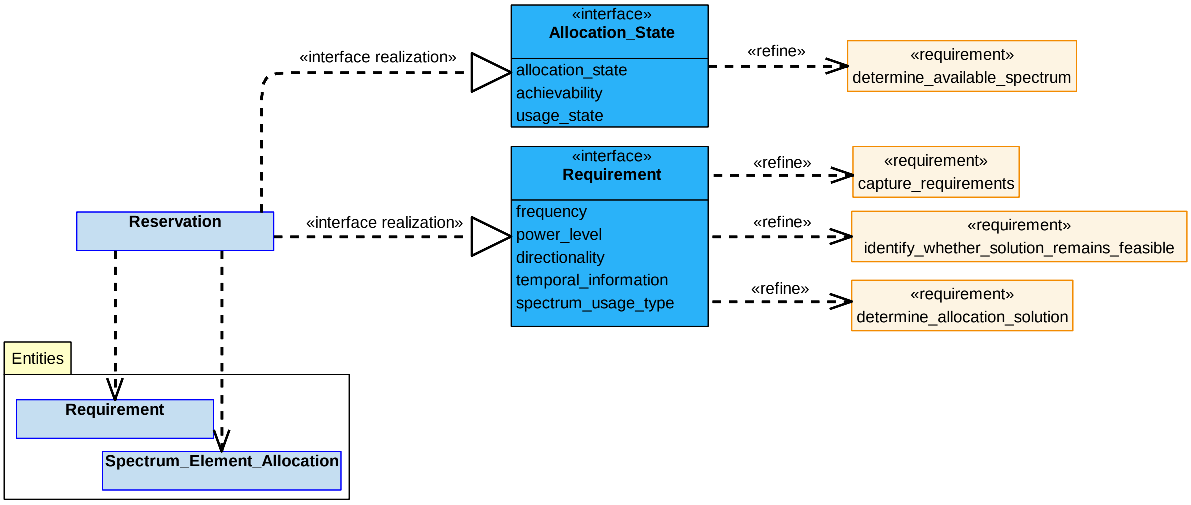


Figure 961: Reservation Service Definition

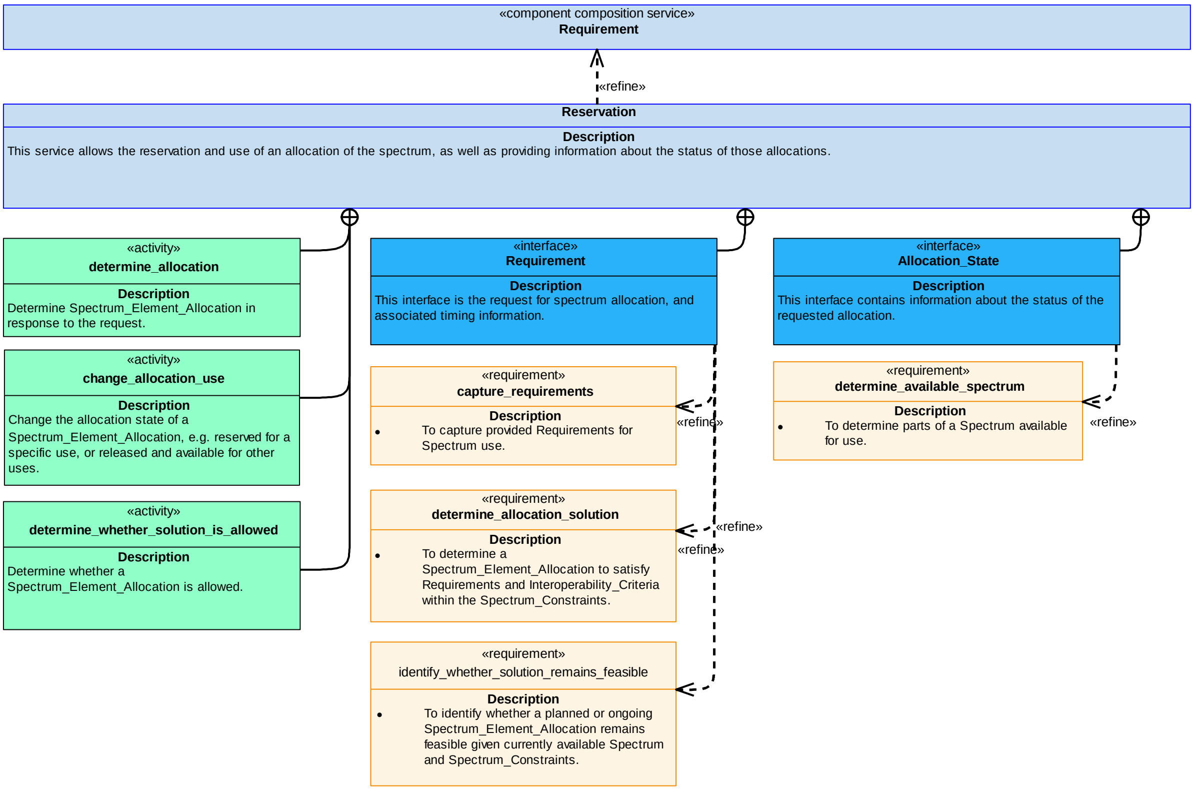

«component composition service»
Requirement
«refine»
Reservation
Description
This service allows the reservation and use of an allocation of the spectrum, as well as providing information about the status of those allocations.
«activity»
determine_allocation
Description
Determine Spectrum_Element_Allocation in response to the request.
«activity»
change_allocation_use
Description
Change the allocation state of a Spectrum_Element_Allocation, e.g. reserved for a specific use, or released and available for other uses.
«activity»
determine_whether_solution_is_allowed
Description
Determine whether a Spectrum_Element_Allocation is allowed.
«interface»
Requirement
Description
This interface is the request for spectrum allocation, and associated timing information.
«interface»
Allocation_State
Description
This interface contains information about the status of the requested allocation.
«requirement»
capture_requirements
Description
To capture provided Requirements for Spectrum use.
«refine»
«requirement»
determine_allocation_solution
Description
To determine a Spectrum_Element_Allocation to satisfy Requirements and Interoperability_Criteria within the Spectrum_Constraints.
«refine»
«refine»
«requirement»
identify_whether_solution_remains_feasible
Description
To identify whether a planned or ongoing Spectrum_Element_Allocation remains feasible given currently available Spectrum and Spectrum_Constraints.
«requirement»
determine_available_spectrum
Description
To determine parts of a Spectrum available for use.
«refine»

Figure 962: Reservation Service Policy

Reservation

This service allows the reservation and use of an allocation of the spectrum, as well as providing information about the status of those allocations.

Interfaces

Requirement

This interface is the request for spectrum allocation, and associated timing information.

Attributes

frequency	The Spectrum of interest (i.e. frequency, frequency range, and tolerance).
power_level	The required power level.
directionality	The direction and spread. For example, directional satellite comms may only be radiating upwards in a tight beam, allowing sensors to look down and forward in the same range of spectrum.
temporal_information	Information covering timing for the requested Spectrum, such as start and end times. This may include segments of a requested time window that must not be interrupted.
spectrum_usage_type	If the usage is for transmitting, receiving or both.

Allocation_State

This interface contains information about the status of the requested allocation.

Attributes

allocation_state	The current state of a Spectrum_Element_Allocation (e.g. requested, allocated, rejected, claimed, or released).
achievability	The achievability of a particular requirement (e.g. cannot find a solution). The response can include conditions/constraints that are non-binary allowing for operationally acceptable consequences for the requester.
usage_state	The state of a portion of the spectrum from an availability point of view. For example, it is assigned to a requirement and the window for use is open.

Activities

determine_allocation

Determine Spectrum_Element_Allocation in response to the request.

change_allocation_use

Change the allocation state of a Spectrum_Element_Allocation, e.g. reserved for a specific use, or released and available for other uses.

determine_whether_solution_is_allowed

Determine whether a Spectrum_Element_Allocation is allowed.

5.4.2.56.7.1.2 Restriction

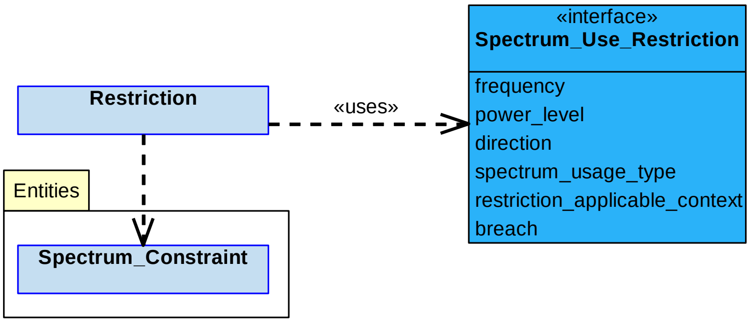


Figure 963: Restriction Service Definition

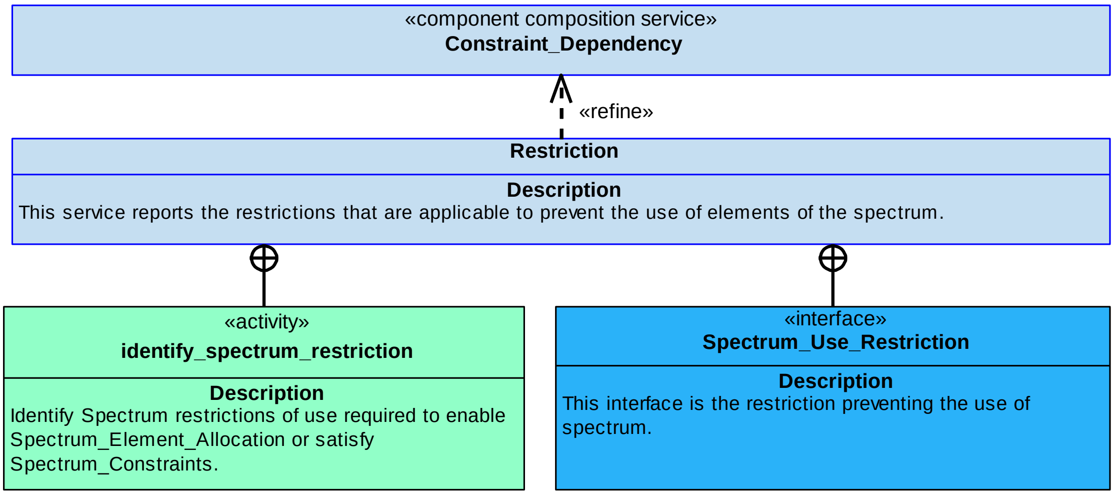

«component composition service»
Constraint_Dependency
«refine»
Restriction
Description
This service reports the restrictions that are applicable to prevent the use of elements of the spectrum.
«activity»
identify_spectrum_restriction
Description
Identify Spectrum restrictions of use required to enable Spectrum_Element_Allocation or satisfy Spectrum_Constraints.
«interface»
Spectrum_Use_Restriction
Description
This interface is the restriction preventing the use of spectrum.

Figure 964: Restriction Service Policy

Restriction

This service reports the restrictions that are applicable to prevent the use of elements of the spectrum.

Interface

Spectrum_Use_Restriction

This interface is the restriction preventing the use of spectrum.

Attributes

frequency	The spectrum restriction of interest (i.e. frequency, frequency range, and tolerance).
power_level	The restriction on the usable power level or levels.
direction	The restriction on directionality relative to the platform or other frame of reference. For example, any comms directed straight up to satellites is restricted.
spectrum_usage_type	A statement of whether the usage restriction is for transmitting, receiving or both.
restriction_applicable_context	The applicable context of a restriction, such as whether it is currently applicable and when the restriction may be lifted.
breach	A statement on whether a restriction has been breached, or is likely to breach a restriction if enforced.

Activity

identify_spectrum_restriction

Identify Spectrum restrictions of use required to enable Spectrum_Element_Allocation or satisfy Spectrum_Constraints.

5.4.2.56.7.1.3 Spectrum_Query

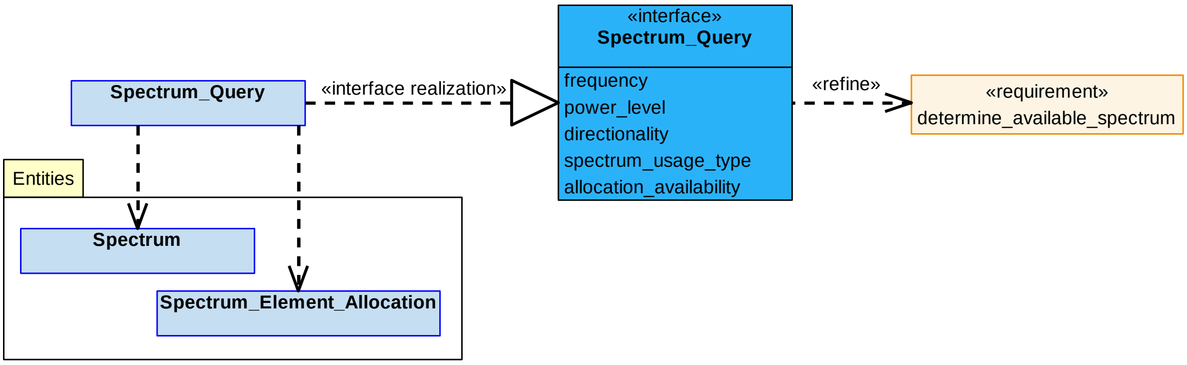


Figure 965: Spectrum_Query Service Definition

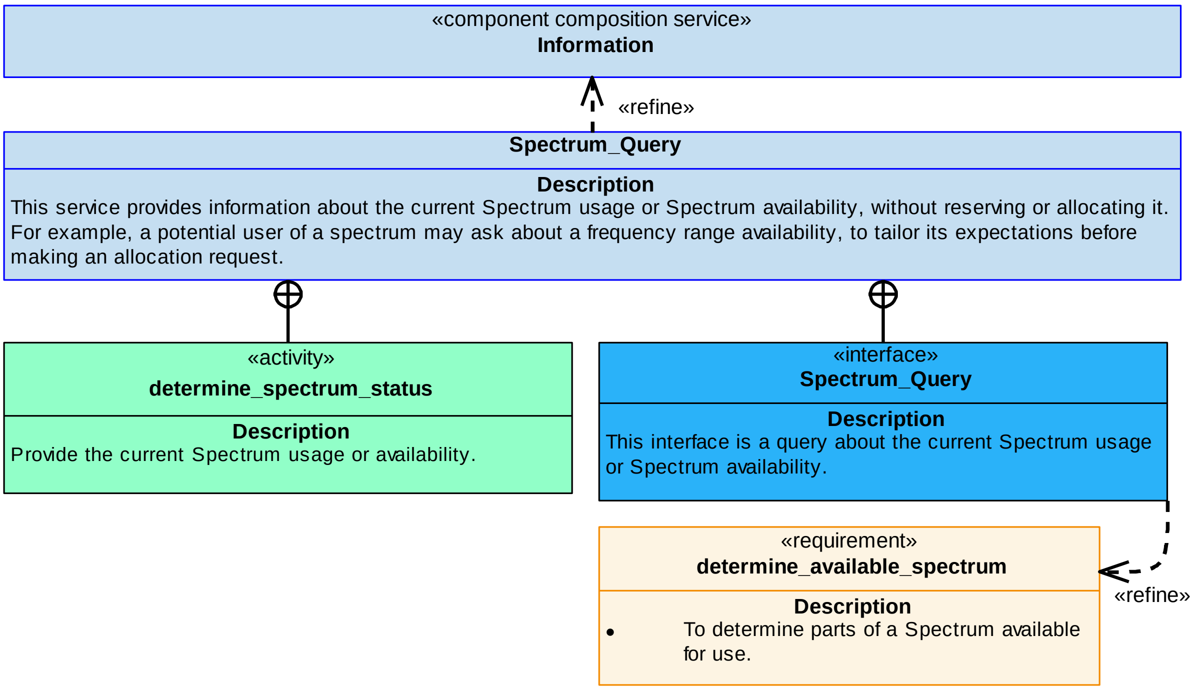

«component composition service»
Information
«refine»
Spectrum_Query
Description
This service provides information about the current Spectrum usage or Spectrum availability, without reserving or allocating it. For example, a potential user of a spectrum may ask about a frequency range availability, to tailor its expectations before making an allocation request.
«activity»
determine_spectrum_status
Description
Provide the current Spectrum usage or availability.
«interface»
Spectrum_Query
Description
This interface is a query about the current Spectrum usage or Spectrum availability.
«requirement»
determine_available_spectrum
Description
To determine parts of a Spectrum available for use.
«refine»

Figure 966: Spectrum_Query Service Policy

Spectrum_Query

This service provides information about the current Spectrum usage or Spectrum availability, without reserving or allocating it. For example, a potential user of a spectrum may ask about a frequency range availability, to tailor its expectations before making an allocation request.

<u>Interface</u>

Spectrum_Query

This interface is a query about the current Spectrum usage or Spectrum availability.

<u>Attributes</u>

frequency	The spectrum of interest (i.e. frequency, frequency range, and tolerance).

power_level	The power level, i.e. in use or available for use.
directionality	The direction and spread. For example, a tight beam pattern in a specified direction from the source.
spectrum_usage_type	If the usage is for transmitting, receiving or both.
allocation_availability	The state of the allocation, from an availability point of view. For example, it is assigned to a requirement and the window for use is open.

Activity

determine_spectrum_status

Provide the current Spectrum usage or availability.

5.4.2.56.7.1.4 Use

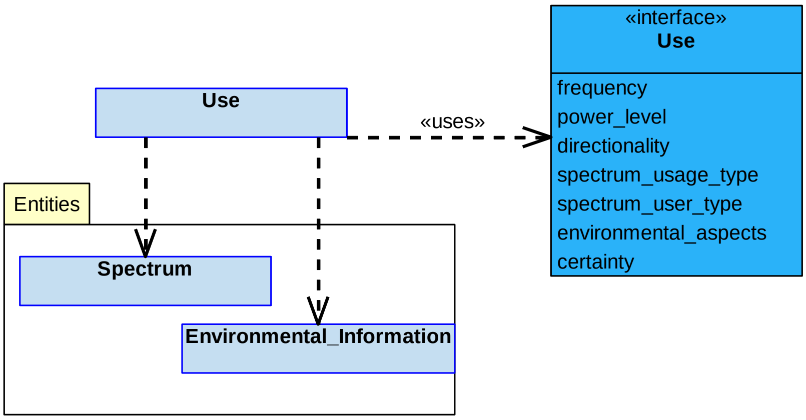


Figure 967: Use Service Definition

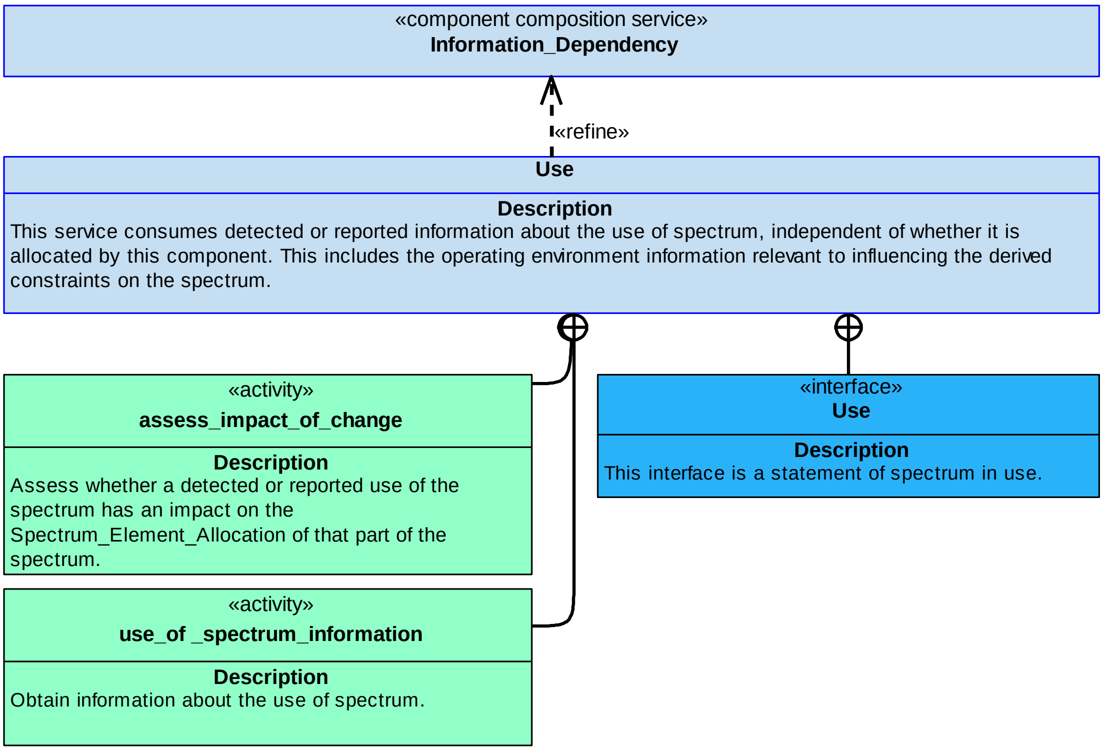

«component composition service»
Information_Dependency
«refine»
Use
Description
This service consumes detected or reported information about the use of spectrum, independent of whether it is allocated by this component. This includes the operating environment information relevant to influencing the derived constraints on the spectrum.
«activity»
assess_impact_of_change
Description
Assess whether a detected or reported use of the spectrum has an impact on the Spectrum_Element_Allocation of that part of the spectrum.
«activity»
use_of _spectrum_information
Description
Obtain information about the use of spectrum.
«interface»
Use
Description
This interface is a statement of spectrum in use.

Figure 968: Use Service Policy

Use

This service consumes detected or reported information about the use of spectrum, independent of whether it is allocated by this component. This includes the operating environment information relevant to influencing the derived constraints on the spectrum.

<u>Interface</u>

Use

This interface is a statement of spectrum in use.

<u>Attributes</u>

frequency	The spectrum of interest (i.e. frequency, frequency range, and tolerance).
power_level	The power level.
directionality	The direction and spread. For example a tight beam pattern, in a specified direction from the source.
spectrum_usage_type	If the usage is for transmitting, receiving or both.
spectrum_user_type	This attribute describes what is known about the user of the spectrum. e.g. not a known user (because it was detected by a sensor) or it is a known user (as it was detected through another instance of this component).
environmental_aspects	The impact of influencing external environment conditions on the spectrum at a given time and place e.g. the impact of interference due to a lighting storm or a building.
certainty	The level of certainty of the environmental information.

Activities

assess_impact_of_change

Assess whether a detected or reported use of the spectrum has an impact on the Spectrum_Element_Allocation of that part of the spectrum.

use_of _spectrum_information

Obtain information about the use of spectrum.

5.4.2.56.7.1.5 Constraint

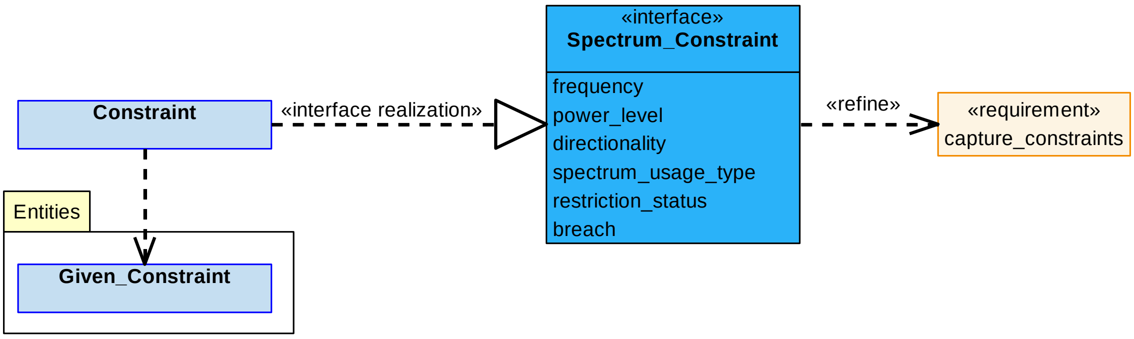


Figure 969: Constraint Service Definition

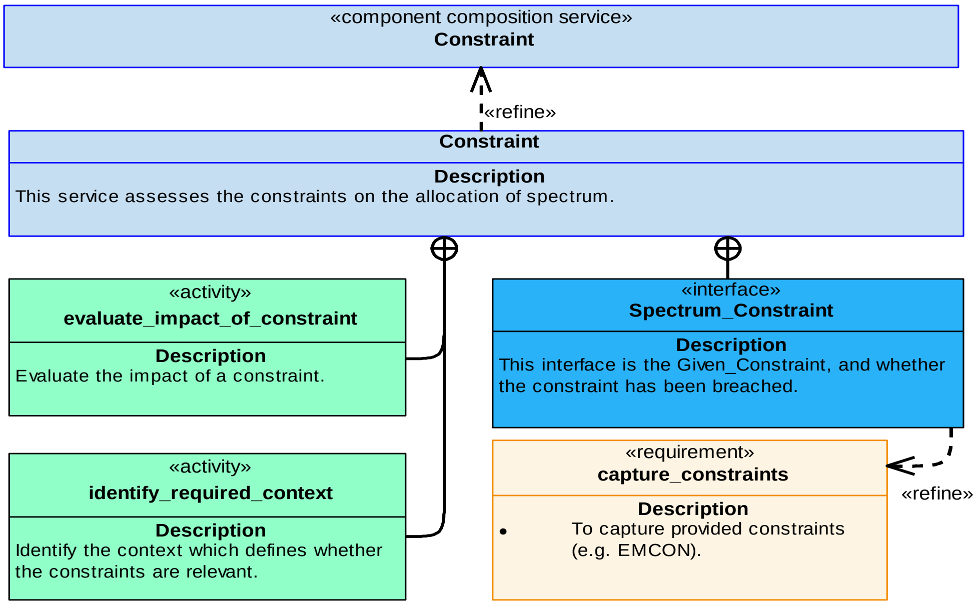

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the constraints on the allocation of spectrum.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of a constraint.
«activity»
identify_required_context
Description
Identify the context which defines whether the constraints are relevant.
«interface»
Spectrum_Constraint
Description
This interface is the Given_Constraint, and whether the constraint has been breached.
«requirement»
capture_constraints
Description
To capture provided constraints (e.g. EMCON).
«refine»

Figure 970: Constraint Service Policy

Constraint

This service assesses the constraints on the allocation of spectrum.

Interface

Spectrum_Constraint

This interface is the Given_Constraint, and whether the constraint has been breached.

Attributes

frequency	The spectrum of interest (i.e. frequency, frequency range, and tolerance).
power_level	The power level limitation.
directionality	The constrained direction and spread.
spectrum_usage_type	If the usage constraint is for transmitting, receiving or both.
restriction_status	The status of a Given_Constraint. For example, whether it is currently applicable and how long it will be applicable for.
breach	A statement that the constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of a constraint.

identify_required_context

Identify the context which defines whether the constraints are relevant.

5.4.2.56.7.1.6 Capability

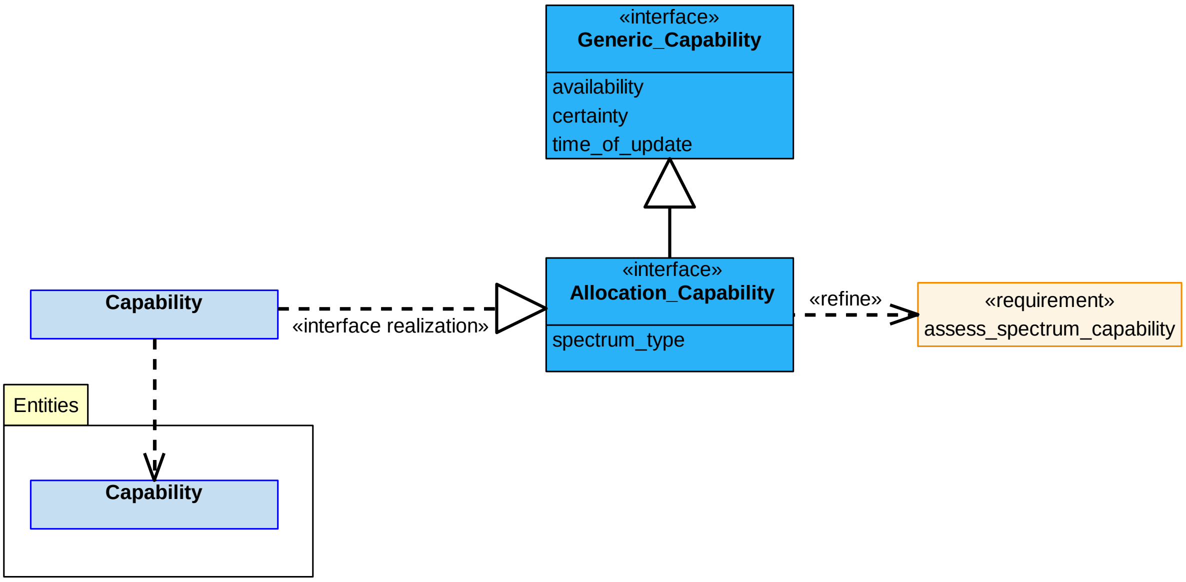


Figure 971: Capability Service Definition

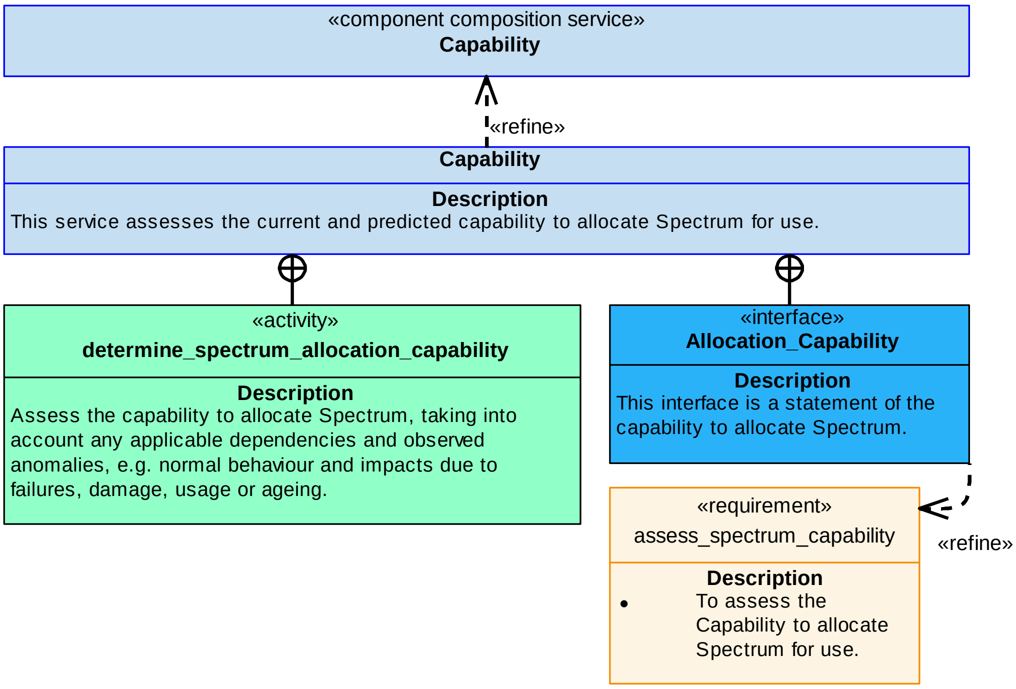

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability to allocate Spectrum for use.
«activity»
determine_spectrum_allocation_capability
Description
Assess the capability to allocate Spectrum, taking into account any applicable dependencies and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.
«interface»
Allocation_Capability
Description
This interface is a statement of the capability to allocate Spectrum.
«requirement»
assess_spectrum_capability
Description
To assess the Capability to allocate Spectrum for use.
«refine»

Figure 972: Capability Service Policy

Capability

This service assesses the current and predicted capability to allocate Spectrum for use.

Interface

Allocation_Capability

This interface is a statement of the capability to allocate Spectrum.

Attribute

spectrum_type	The type of spectrum, e.g. electromagnetic or acoustic spectrum.

Activity

determine_spectrum_allocation_capability

Assess the capability to allocate Spectrum, taking into account any applicable dependencies and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.

5.4.2.56.7.1.7 Capability_Evidence

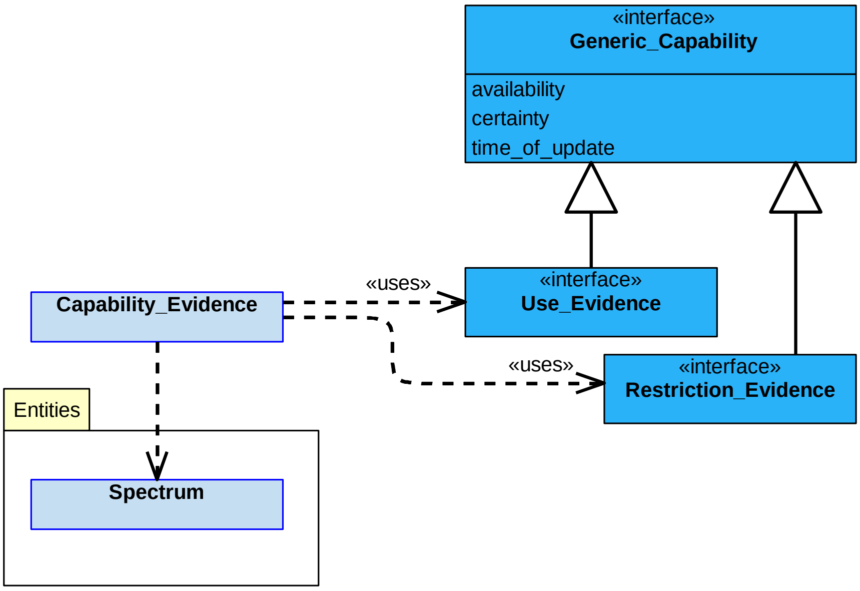


Figure 973: Capability_Evidence Service Definition

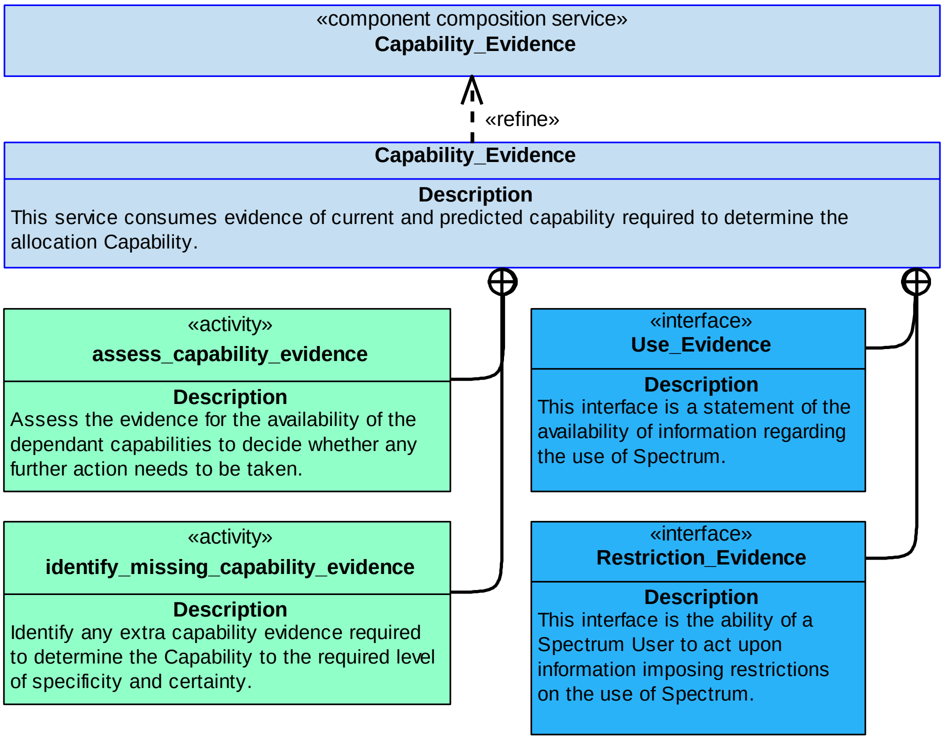

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes evidence of current and predicted capability required to determine the allocation Capability.
«activity»
assess_capability_evidence
Description
Assess the evidence for the availability of the dependant capabilities to decide whether any further action needs to be taken.
«interface»
Use_Evidence
Description
This interface is a statement of the availability of information regarding the use of Spectrum.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Restriction_Evidence
Description
This interface is the ability of a Spectrum User to act upon information imposing restrictions on the use of Spectrum.

Figure 974: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence of current and predicted capability required to determine the allocation Capability.

Interfaces

Use_Evidence

This interface is a statement of the availability of information regarding the use of Spectrum.

Restriction_Evidence

This interface is the ability of a Spectrum User to act upon information imposing restrictions on the use of Spectrum.

Activities

assess_capability_evidence

Assess the evidence for the availability of the dependant capabilities to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.56.7.2 Service Dependencies

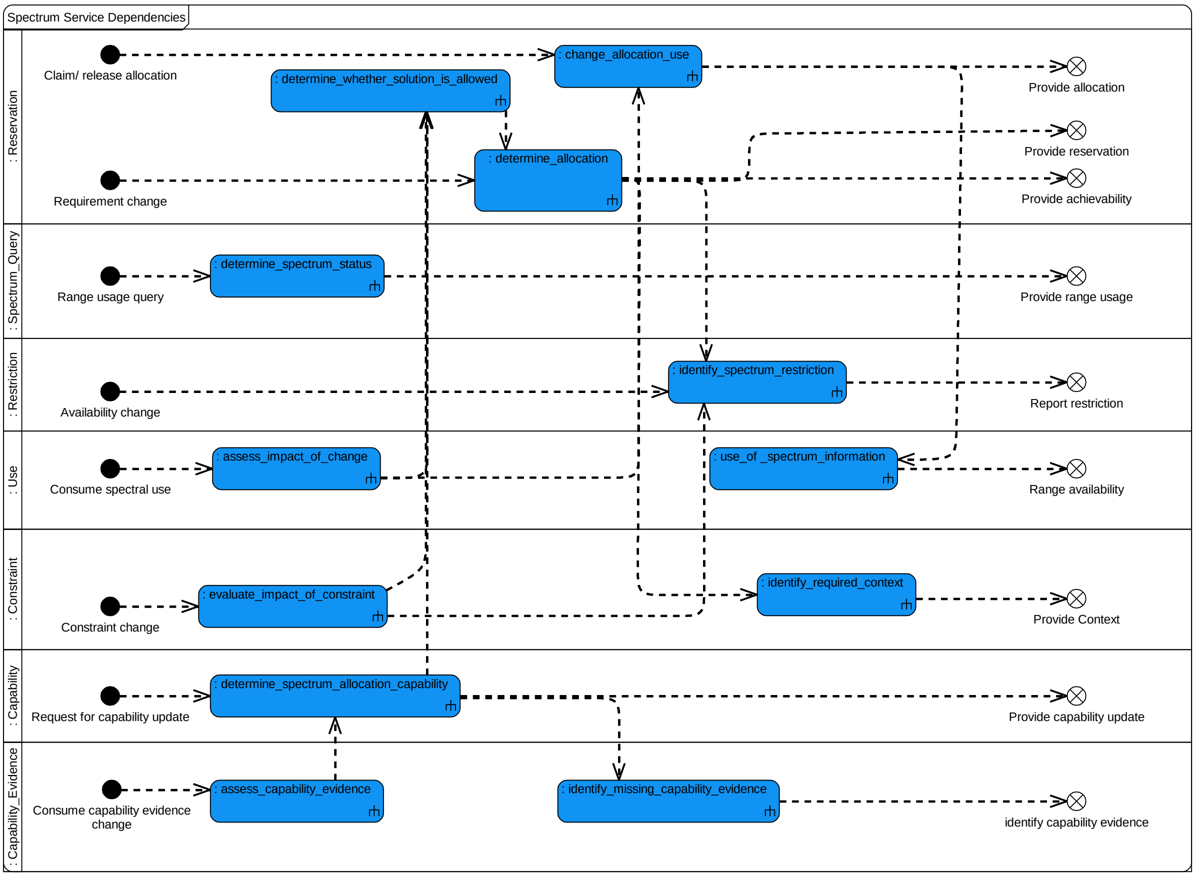

Spectrum Service Dependencies
: Reservation
: Spectrum_Query
: Restriction
: Use
: Constraint
: Capability
: Capability_Evidence
Claim/ release allocation
Requirement change
Range usage query
Availability change
Consume spectral use
Constraint change
Request for capability update
Consume capability evidence change
: change_allocation_use
: determine_whether_solution_is_allowed
: determine_allocation
: determine_spectrum_status
: identify_spectrum_restriction
: assess_impact_of_change
: use_of _spectrum_information
: evaluate_impact_of_constraint
: identify_required_context
: determine_spectrum_allocation_capability
: assess_capability_evidence
: identify_missing_capability_evidence
Provide allocation
Provide reservation
Provide achievability
Provide range usage
Report restriction
Range availability
Provide Context
Provide capability update
identify capability evidence

Figure 975: Spectrum Service Dependencies

5.4.2.57 Storage

5.4.2.57.1 Role

The role of Storage is to manage the storage infrastructure.

5.4.2.57.2 Overview

Control Architecture

Storage is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

The Storage component, with knowledge of the Storage_Users, their associated Requirements and the state of the Storage_Usage, will implement a Storage_Solution that will be applied to the Storage_Space. This can result in or be caused by a change to the system's current Capability and Constraint(s).

Examples of Use

Storage will be used to:

- Provide components with transparent access to Storage_Space.
- Manage the storage infrastructure and available Storage_Space capability.
- Maintain a view of the usage of the Storage_Space.
- Initiate disposal of data items.
- Trigger the sanitisation of Storage_Space.

5.4.2.57.3 Service Summary

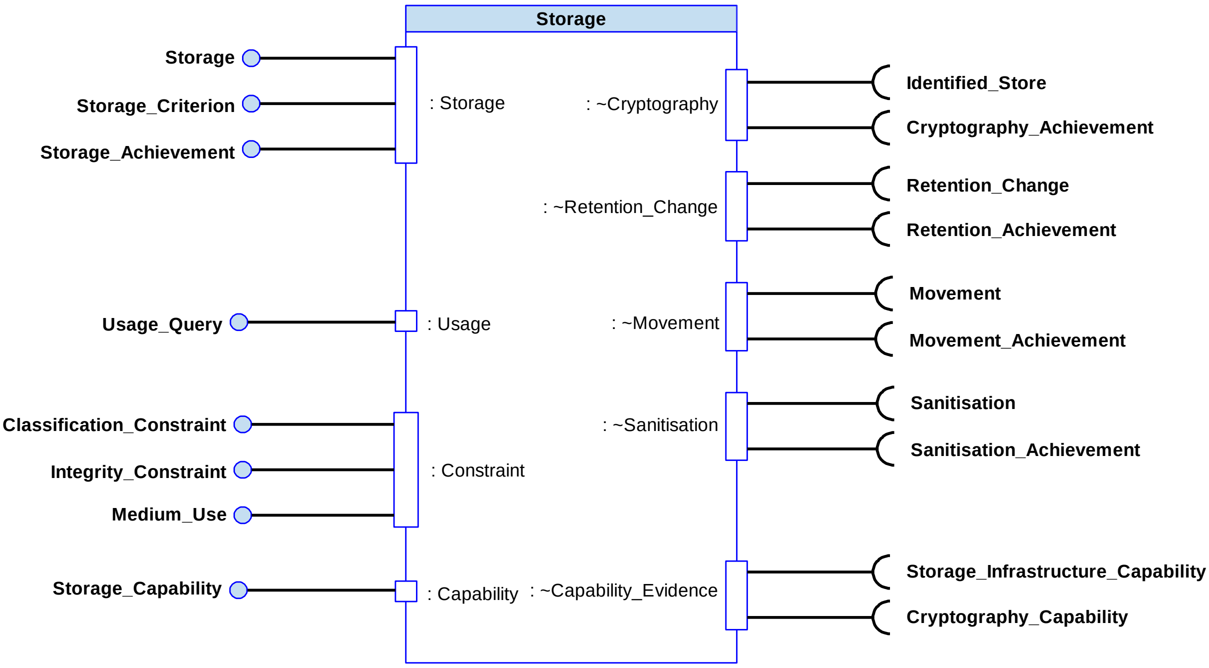


Figure 976: Storage Service Summary

5.4.2.57.4 Responsibilities

capture_requirements_for_sanitisation

- To capture provided Requirement(s) for sanitisation.

capture_requirements_for_storage

- To capture provided Requirement(s) (e.g. desired effect or output, timing or balancing) for storage.

capture_measurement_criteria_for_storage

- To capture given Measure_of_Achievement.

capture_storage_constraints

- To capture Constraint(s) on the manner in which data is to be stored (e.g. security, accessibility or redundancy).

determine_storage_solution

- To determine how to meet the given Requirement(s) and Constraint(s) for storage, e.g. moving data to balance the Storage_Space whilst observing security constraints.

determine_if_storage_solution_remains_feasible

- To identify whether the planned or in progress Storage_Solution against a Requirement is still feasible given current or predicted Capability and conditions.

deliver_storage_solution

- To deliver a Storage_Solution to meet the Requirement(s) within the Constraint(s).

initiate_sanitisation_of_storage_space

- To initiate the sanitisation of a Storage_Space.

identify_progress

- To identify the progress against the Requirement.

assess_storage_capability

- To assess the Capability to provide storage taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the Capability assessment.

predict_storage_capability_progression

- To predict the progression of the storage Capability over time and with use.

5.4.2.57.5 Subject Matter Semantics

The subject matter of Storage is the data Storage_Space available to the system.

Exclusions

The subject matter of Storage does not include:

- What data is held on Storage_Space and which components can access it, only how much there is and where it is held.
- How encryption of a Storage_Space is achieved, only that it may be required.
- The retrieval or storage of particular data held on the Storage_Space.

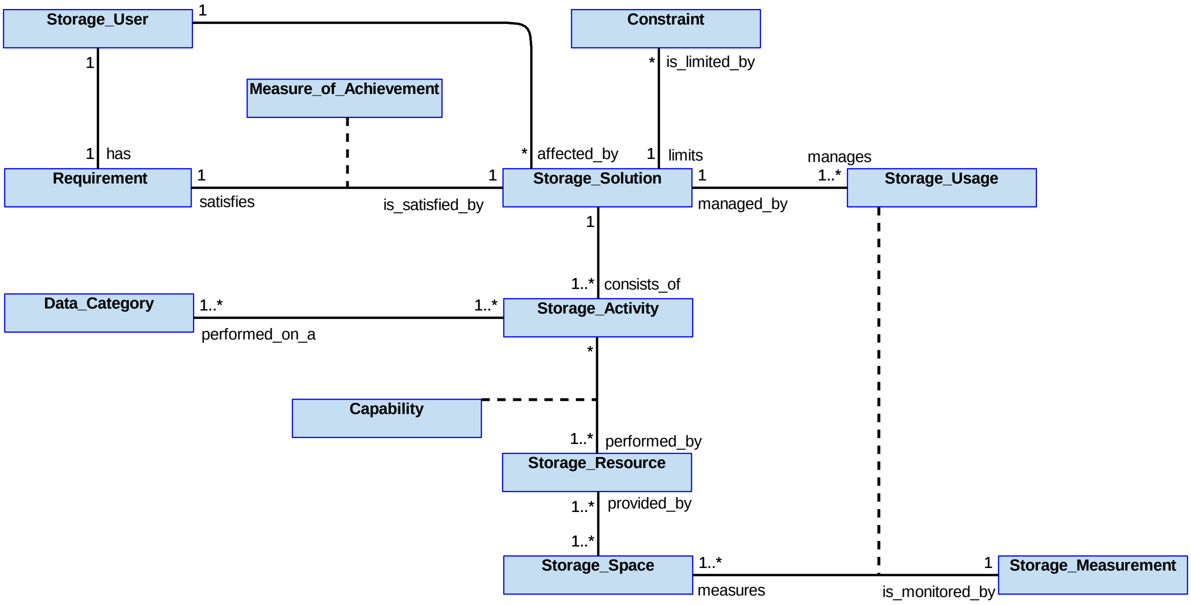


Figure 977: Storage Semantics

5.4.2.57.5.1 Entities

Capability

The range of storage activities that can be performed with the available Storage_Resources. This excludes the available storage capacity.

Constraint

An externally imposed restriction (e.g. rules of security classification or use of medium).

Data_Category

A categorisation of data (e.g. size, classification, type, date created or ID) that can be managed to perform a Storage_Activity.

Measure_of_Achievement

An evaluation of how well the Storage_Solution satisfies a Requirement.

Requirement

A need to manage Storage_Resources, e.g. the need to store 100TB at a write rate of 100MB/s or to initiate sanitisation of a Storage_Space.

Storage_Activity

An activity performed on stored data (e.g. transfer, duplication or deletion).

Storage_Resource

Something that can be instructed to act on data or retain storage capacity for a specific purpose, i.e. a part of the storage infrastructure.

Storage_Measurement

A measure of the storage capabilities (e.g. storage volume, data read/write rates, fragmentation or quality of service).

Storage_Solution

The resultant actions taken by the Storage component to manage the storage infrastructure, e.g. triggering a Storage_User to delete data.

Storage_Space

A medium data is read from or written to (e.g. physical drive) including the total available capacity contained therein.

Storage_Usage

The relationship between measured and available storage capacity (e.g. storage volume or data read/write rates), such as the level of capacity remaining in proportion to the total capacity (free space).

Storage_User

An entity that requires storage capability.

5.4.2.57.6 Design Rationale

5.4.2.57.6.1 Assumptions

- Data stored can be of any classification, provided the Exploiting Programme has ensured the correct security provisions have been made.
- There will be a Storage component in each node requiring the management of storage.
- Stored data will include classification, time stamps, and other pertinent metadata as required.

5.4.2.57.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Storage:

- Storage - As it describes the management of the storage media infrastructure.
- Data Driving - As data storage constraints that govern Storage can vary depending on mission, platform, etc.

Extensions

- The use of extension components for Storage components may be appropriate if different forms of Storage_Space are used.

Exploitation Considerations

- This Storage component is responsible for instigating the transfer of data from one Storage_Space to another.

- It is the responsibility of the Exploiting Programme to ensure that correct security provisions have been made and that data is written to the correct, appropriately security accredited, Storage_Space and to manage gateways, etc.

5.4.2.57.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

Failure of this component may result in a failure to provide data required for external use. This would include:

- Crash recording data.
- Life and usage data.

Life and usage data is considered the most significant. It is assumed that any data recorded would have been protected from corruption prior to processing by this component (using the hashing function of the Cryptographic Methods component). Therefore, the worst case consequence would be loss of the recorded data. Where the life and usage data relates to safety critical systems it is reasonable to expect that even when no critical events have occurred that a data file is stored for every flight. Therefore, if the data file is not found the loss of data can be detected and either worst case usage assumed or inspections of the Exploiting Platform performed. Therefore it is concluded that failure of this component may result a no worse than a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.

If an Exploiting Programme places data considered flight safety critical within storage devices that are able to be sanitised, the DAL will need to be increased to reflect the criticality of the data.

5.4.2.57.6.4 Security Considerations

The indicative security classification is O-S.

Whilst this component is unlikely to be highly classified itself, instances will be involved in each security domain that has access to a data store. Whilst this component does not handle data itself, it should be treated as per the confidentiality, integrity and availability needs of the data being stored during all mission phases. Any compromise of this component may mean that stored data, including security logs and audit data, cannot be stored or cannot be trusted.

The component implements security related functions relating to:

- **Back-up and Recovery** of data stored on the storage devices the component interacts with.
- Confidentiality and separation based on the **Classification of Data**; this component will not itself classify the data but will be cognisant of the security domain from which the data originates and allocate an appropriate Storage_Space.
- **System Status and Monitoring** through the monitoring of access and usage of the store, with any unexpected levels being indicative of a possible cyber attack.

The **Logging of Data** and **Maintaining Audit Records** functions will primarily be handled by the components that own the data through their retention policies, although Storage may affect the individual retention policies, e.g. should the available Storage_Space become limited. See the Recording and Logging PYRAMID concept for further details.

Implements security enforcing functions by:

- Implementing the Storage_Solution, including identifying where encrypting of storage spaces is required; actual encryption is handled by cryptographic components.
- **Rendering Sensitive Data Inaccessible** through requesting sanitisation of an entire Storage_Space, when required. This might involve the cryptographic components or be directly through the infrastructure to trigger an emergency purge (e.g. by destroying the hardware). See the Storage PYRAMID concept for further details.

5.4.2.57.7 Services

5.4.2.57.7.1 Service Definitions

5.4.2.57.7.1.1 Storage

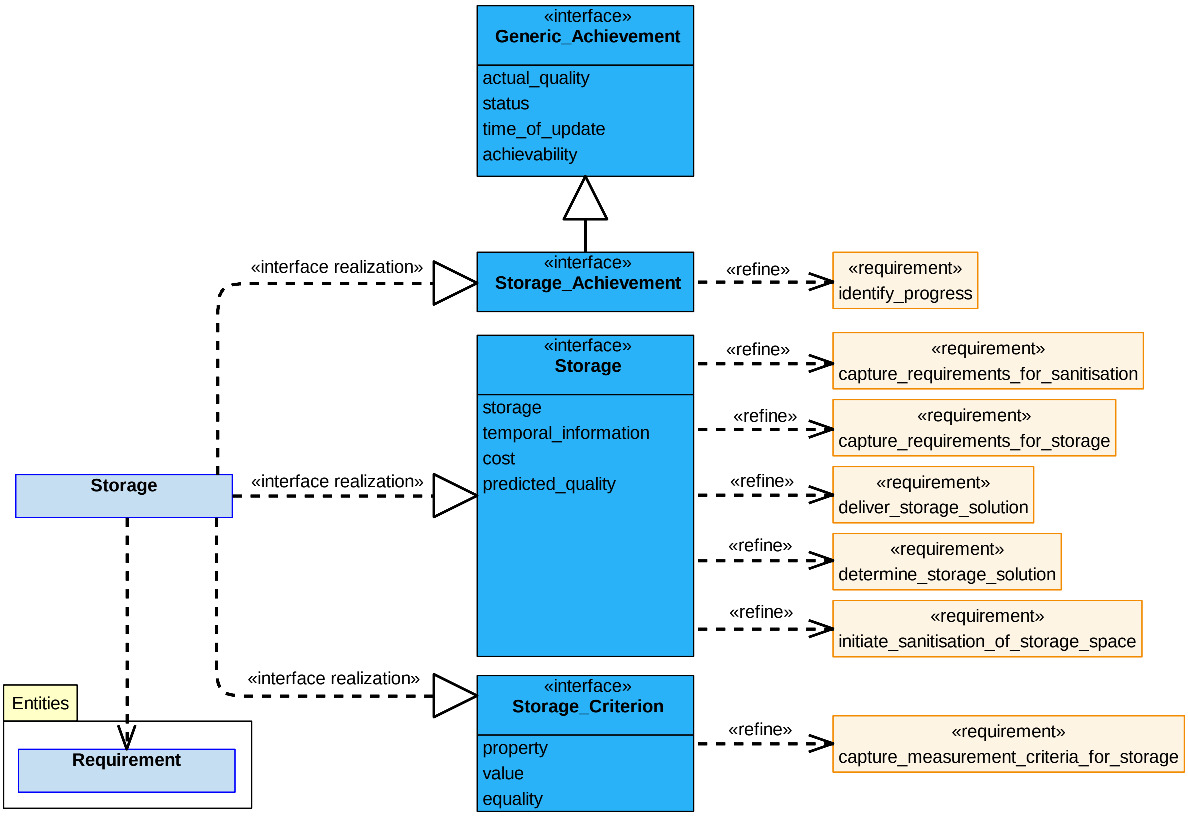


Figure 978: Storage Service Definition

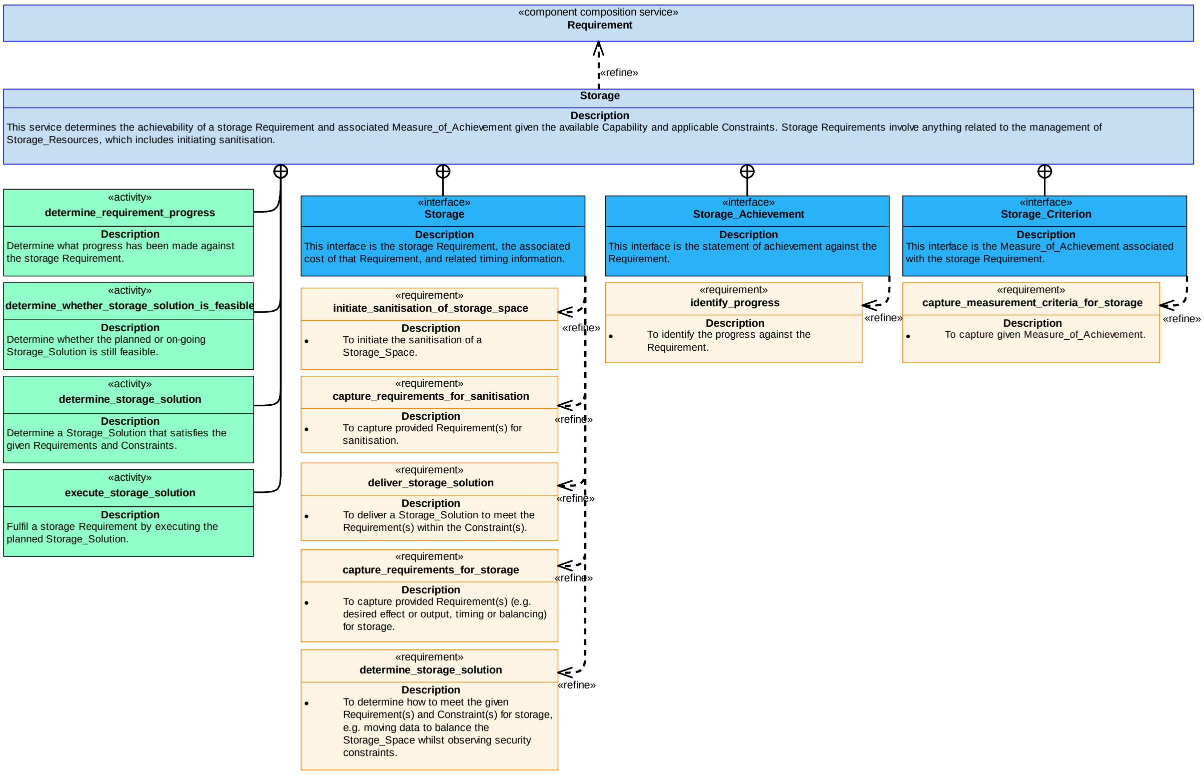

«component composition service»
Requirement
«refine»
Storage
Description
This service determines the achievability of a storage Requirement and associated Measure_of_Achievement given the available Capability and applicable Constraints. Storage Requirements involve anything related to the management of Storage_Resources, which includes initiating sanitisation.
«activity»
determine_requirement_progress
Description
Determine what progress has been made against the storage Requirement.
«activity»
determine_whether_storage_solution_is_feasible
Description
Determine whether the planned or on-going Storage_Solution is still feasible.
«activity»
determine_storage_solution
Description
Determine a Storage_Solution that satisfies the given Requirements and Constraints.
«activity»
execute_storage_solution
Description
Fulfil a storage Requirement by executing the planned Storage_Solution.
«interface»
Storage
Description
This interface is the storage Requirement, the associated cost of that Requirement, and related timing information.
«interface»
Storage_Achievement
Description
This interface is the statement of achievement against the Requirement.
«interface»
Storage_Criterion
Description
This interface is the Measure_of_Achievement associated with the storage Requirement.
«requirement»
initiate_sanitisation_of_storage_space
Description
To initiate the sanitisation of a Storage_Space.
«refine»
«requirement»
capture_requirements_for_sanitisation
Description
To capture provided Requirement(s) for sanitisation.
«refine»
«requirement»
deliver_storage_solution
Description
To deliver a Storage_Solution to meet the Requirement(s) within the Constraint(s).
«refine»
«requirement»
capture_requirements_for_storage
Description
To capture provided Requirement(s) (e.g. desired effect or output, timing or balancing) for storage.
«refine»
«requirement»
determine_storage_solution
Description
To determine how to meet the given Requirement(s) and Constraint(s) for storage, e.g. moving data to balance the Storage_Space whilst observing security constraints.
«refine»
«requirement»
identify_progress
Description
To identify the progress against the Requirement.
«refine»
«requirement»
capture_measurement_criteria_for_storage
Description
To capture given Measure_of_Achievement.
«refine»

Figure 979: Storage Service Policy

Storage

This service determines the achievability of a storage Requirement and associated Measure_of_Achievement given the available Capability and applicable Constraints. Storage Requirements involve anything related to the management of Storage_Resources, which includes initiating sanitisation.

Interfaces

Storage

This interface is the storage Requirement, the associated cost of that Requirement, and related timing information.

Attributes

storage	The Requirement to provide a Storage_Solution (e.g. storage of data, or moving data between storage devices/media).
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used or time taken.
predicted_quality	How well the planned Storage_Solution is predicted to satisfy the storage Requirement.

Storage_Criterion

This interface is the Measure_of_Achievement associated with the storage Requirement.

Attributes

property	The property to be measured, e.g. the data or data partition/store to be sanitised.
value	The measured value of the property, e.g. 100MB.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Storage_Achievement

This interface is the statement of achievement against the Requirement.

Activities

determine_requirement_progress

Determine what progress has been made against the storage Requirement.

determine_storage_solution

Determine a Storage_Solution that satisfies the given Requirements and Constraints.

execute_storage_solution

Fulfil a storage Requirement by executing the planned Storage_Solution.

determine_whether_storage_solution_is_feasible

Determine whether the planned or on-going Storage_Solution is still feasible.

5.4.2.57.7.1.2 Cryptography

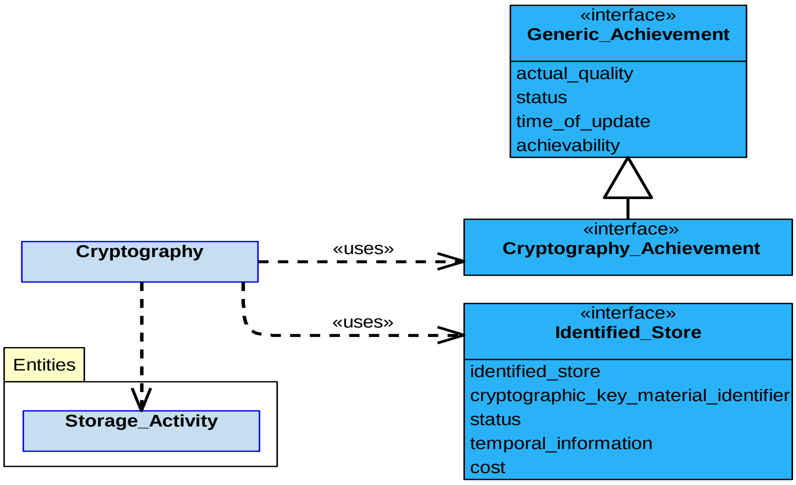


Figure 980: Cryptography Service Definition

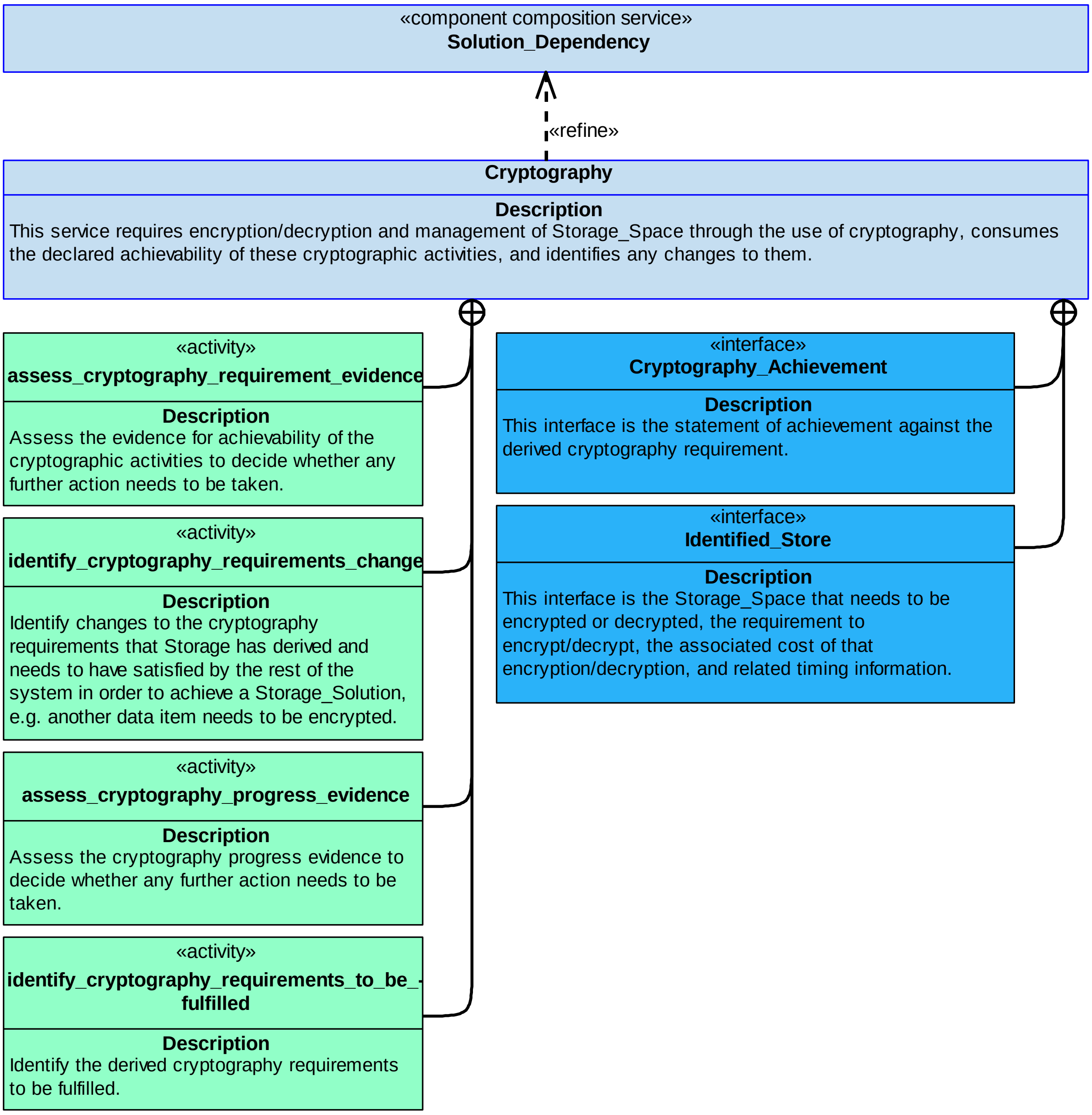

«component composition service»
Solution_Dependency
«refine»
Cryptography
Description
This service requires encryption/decryption and management of Storage_Space through the use of cryptography, consumes the declared achievability of these cryptographic activities, and identifies any changes to them.
«activity»
assess_cryptography_requirement_evidence
Description
Assess the evidence for achievability of the cryptographic activities to decide whether any further action needs to be taken.
«activity»
identify_cryptography_requirements_change
Description
Identify changes to the cryptography requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be encrypted.
«activity»
assess_cryptography_progress_evidence
Description
Assess the cryptography progress evidence to decide whether any further action needs to be taken.
«activity»
identify_cryptography_requirements_to_be_fulfilled
Description
Identify the derived cryptography requirements to be fulfilled.
«interface»
Cryptography_Achievement
Description
This interface is the statement of achievement against the derived cryptography requirement.
«interface»
Identified_Store
Description
This interface is the Storage_Space that needs to be encrypted or decrypted, the requirement to encrypt/decrypt, the associated cost of that encryption/decryption, and related timing information.

Figure 981: Cryptography Service Policy

Cryptography

This service requires encryption/decryption and management of Storage_Space through the use of cryptography, consumes the declared achievability of these cryptographic activities, and identifies any changes to them.

Interfaces

Identified_Store

This interface is the Storage_Space that needs to be encrypted or decrypted, the requirement to encrypt/decrypt, the associated cost of that encryption/decryption, and related timing information.

Attributes

identified_store	The identified Storage_Space to be encrypted or decrypted, e.g. a particular store.
cryptographic_key_material_identifier	An identifier for the cryptographic material (e.g. certificate) to be used.
status	The status of the identified Storage_Space to be encrypted/decrypted, e.g. compromised.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the cryptographic solution, e.g. resources used or time taken.

Cryptography_Achievement

This interface is the statement of achievement against the derived cryptography requirement.

Activities

assess_cryptography_requirement_evidence

Assess the evidence for achievability of the cryptographic activities to decide whether any further action needs to be taken.

assess_cryptography_progress_evidence

Assess the cryptography progress evidence to decide whether any further action needs to be taken.

identify_cryptography_requirements_change

Identify changes to the cryptography requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be encrypted.

identify_cryptography_requirements_to_be_fulfilled

Identify the derived cryptography requirements to be fulfilled.

5.4.2.57.7.1.3 Retention_Change

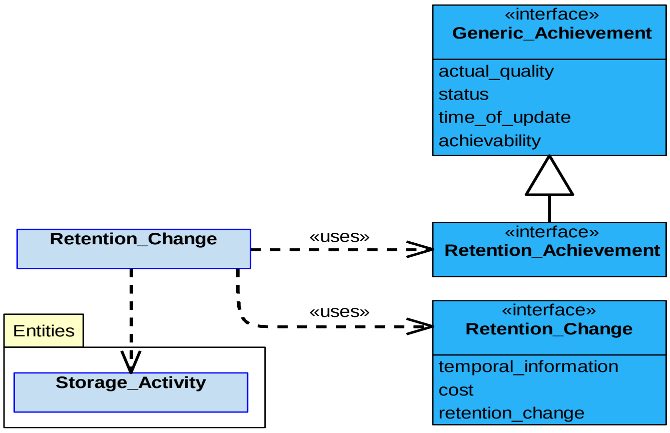


Figure 982: Retention_Change Service Definition

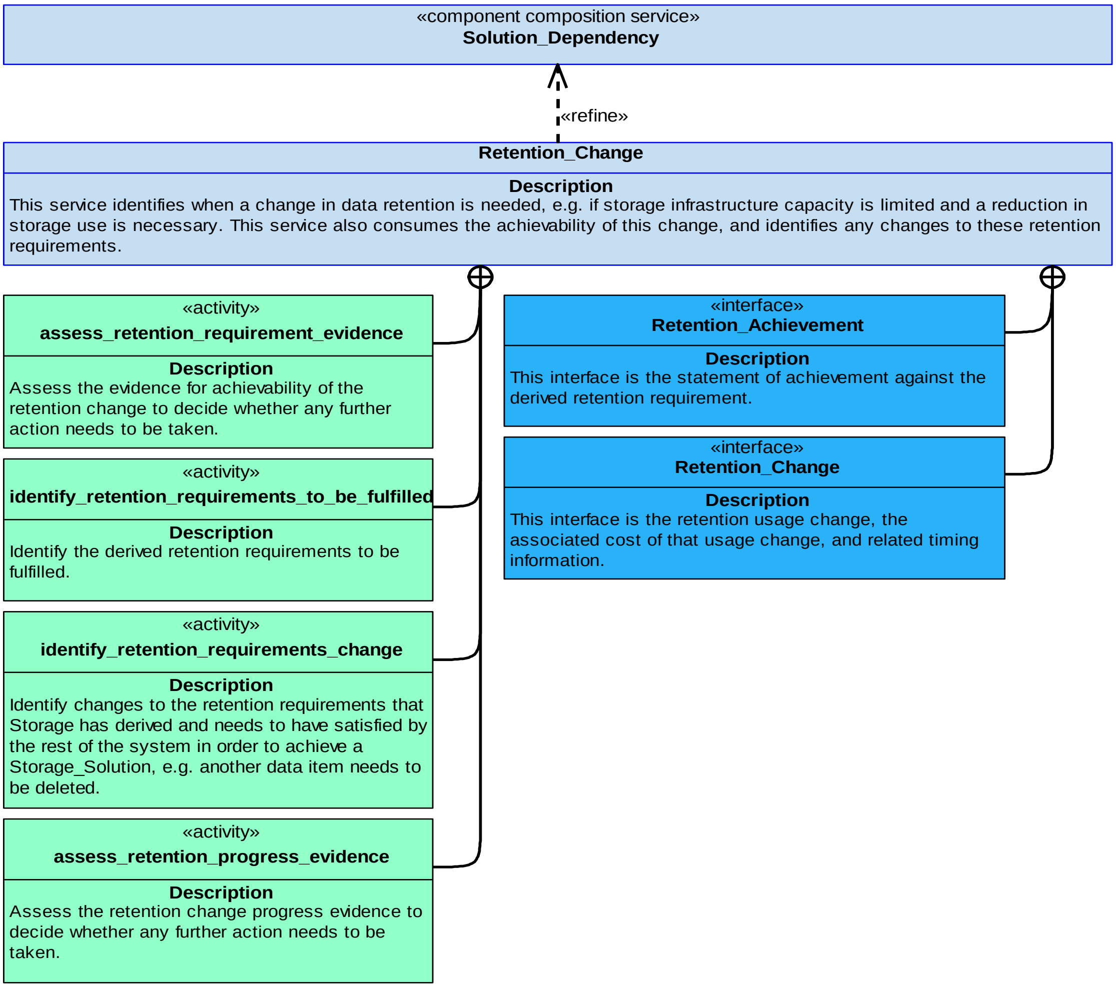

«component composition service»
Solution_Dependency
«refine»
Retention_Change
Description
This service identifies when a change in data retention is needed, e.g. if storage infrastructure capacity is limited and a reduction in storage use is necessary. This service also consumes the achievability of this change, and identifies any changes to these retention requirements.
«activity»
assess_retention_requirement_evidence
Description
Assess the evidence for achievability of the retention change to decide whether any further action needs to be taken.
«activity»
identify_retention_requirements_to_be_fulfilled
Description
Identify the derived retention requirements to be fulfilled.
«activity»
identify_retention_requirements_change
Description
Identify changes to the retention requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be deleted.
«activity»
assess_retention_progress_evidence
Description
Assess the retention change progress evidence to decide whether any further action needs to be taken.
«interface»
Retention_Achievement
Description
This interface is the statement of achievement against the derived retention requirement.
«interface»
Retention_Change
Description
This interface is the retention usage change, the associated cost of that usage change, and related timing information.

Figure 983: Retention_Change Service Policy

Retention_Change

This service identifies when a change in data retention is needed, e.g. if storage infrastructure capacity is limited and a reduction in storage use is necessary. This service also consumes the achievability of this change, and identifies any changes to these retention requirements.

Interfaces

Retention_Change

This interface is the retention usage change, the associated cost of that usage change, and related timing information.

Attributes

temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, e.g. resources used or time taken.
retention_change	The retention change with respect to storage usage.

Retention_Achievement

This interface is the statement of achievement against the derived retention requirement.

Activities

assess_retention_requirement_evidence

Assess the evidence for achievability of the retention change to decide whether any further action needs to be taken.

assess_retention_progress_evidence

Assess the retention change progress evidence to decide whether any further action needs to be taken.

identify_retention_requirements_change

Identify changes to the retention requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be deleted.

identify_retention_requirements_to_be_fulfilled

Identify the derived retention requirements to be fulfilled.

5.4.2.57.7.1.4 Movement

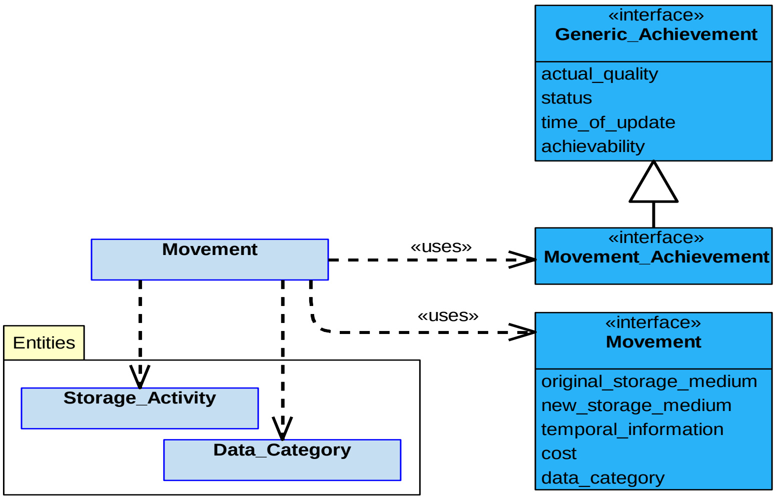


Figure 984: Movement Service Definition

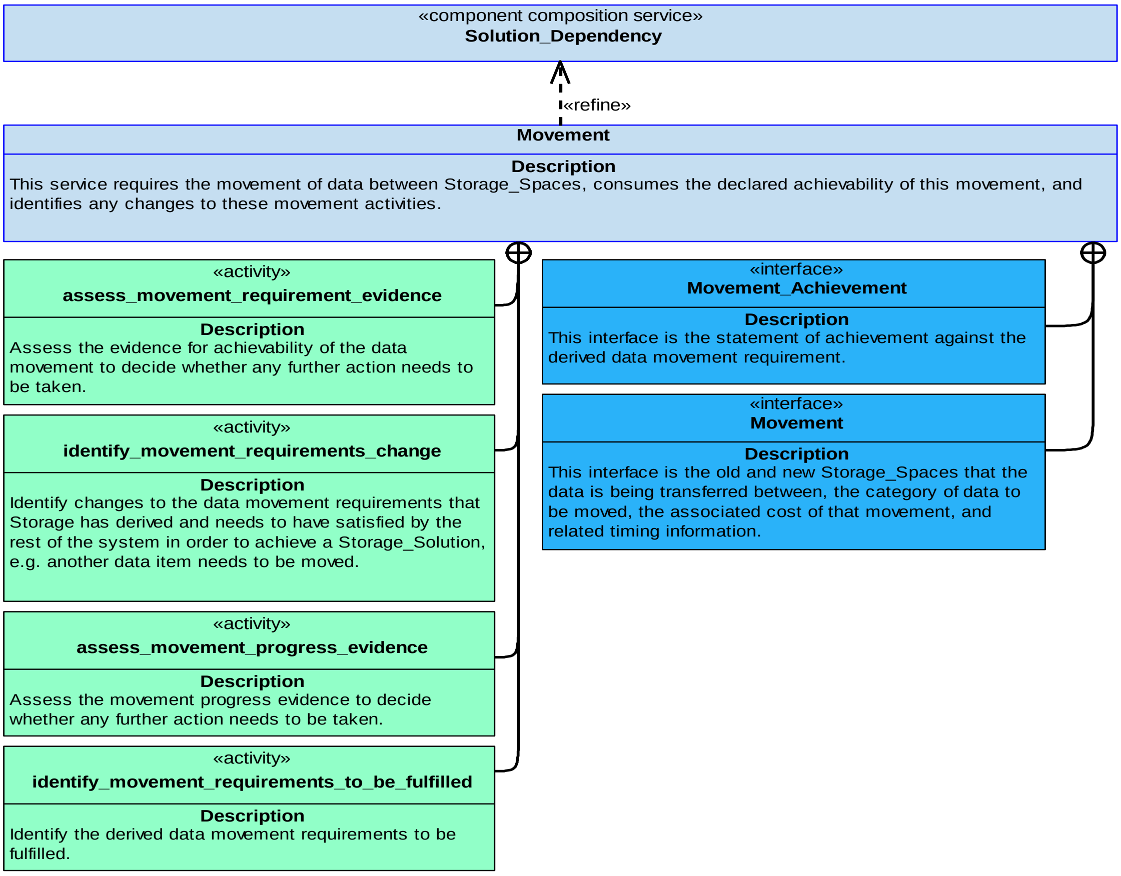

«component composition service»
Solution_Dependency
«refine»
Movement
Description
This service requires the movement of data between Storage_Spaces, consumes the declared achievability of this movement, and identifies any changes to these movement activities.
«activity»
assess_movement_requirement_evidence
Description
Assess the evidence for achievability of the data movement to decide whether any further action needs to be taken.
«activity»
identify_movement_requirements_change
Description
Identify changes to the data movement requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be moved.
«activity»
assess_movement_progress_evidence
Description
Assess the movement progress evidence to decide whether any further action needs to be taken.
«activity»
identify_movement_requirements_to_be_fulfilled
Description
Identify the derived data movement requirements to be fulfilled.
«interface»
Movement_Achievement
Description
This interface is the statement of achievement against the derived data movement requirement.
«interface»
Movement
Description
This interface is the old and new Storage_Spaces that the data is being transferred between, the category of data to be moved, the associated cost of that movement, and related timing information.

Figure 985: Movement Service Policy

Movement

This service requires the movement of data between Storage_Spaces, consumes the declared achievability of this movement, and identifies any changes to these movement activities.

Interfaces

Movement

This interface is the old and new Storage_Spaces that the data is being transferred between, the category of data to be moved, the associated cost of that movement, and related timing information.

Attributes

original_storage_medium	The Storage_Space to move data from.
new_storage_medium	The Storage_Space to move data to.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the movement solution, e.g. resources used or time taken.
data_category	The category of data that is to be moved (e.g. all of the data above a classification of Official-Sensitive, date_created, ID, size or type).

Movement_Achievement

This interface is the statement of achievement against the derived data movement requirement.

Activities

assess_movement_requirement_evidence

Assess the evidence for achievability of the data movement to decide whether any further action needs to be taken.

assess_movement_progress_evidence

Assess the movement progress evidence to decide whether any further action needs to be taken.

identify_movement_requirements_change

Identify changes to the data movement requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be moved.

identify_movement_requirements_to_be_fulfilled

Identify the derived data movement requirements to be fulfilled.

5.4.2.57.7.1.5 Sanitisation

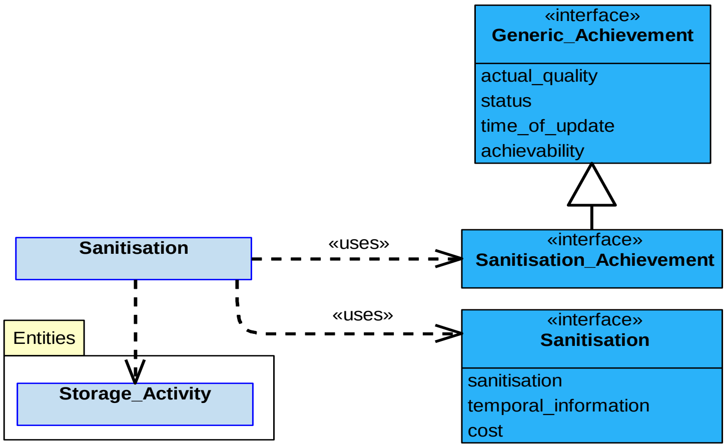


Figure 986: Sanitisation Service Definition

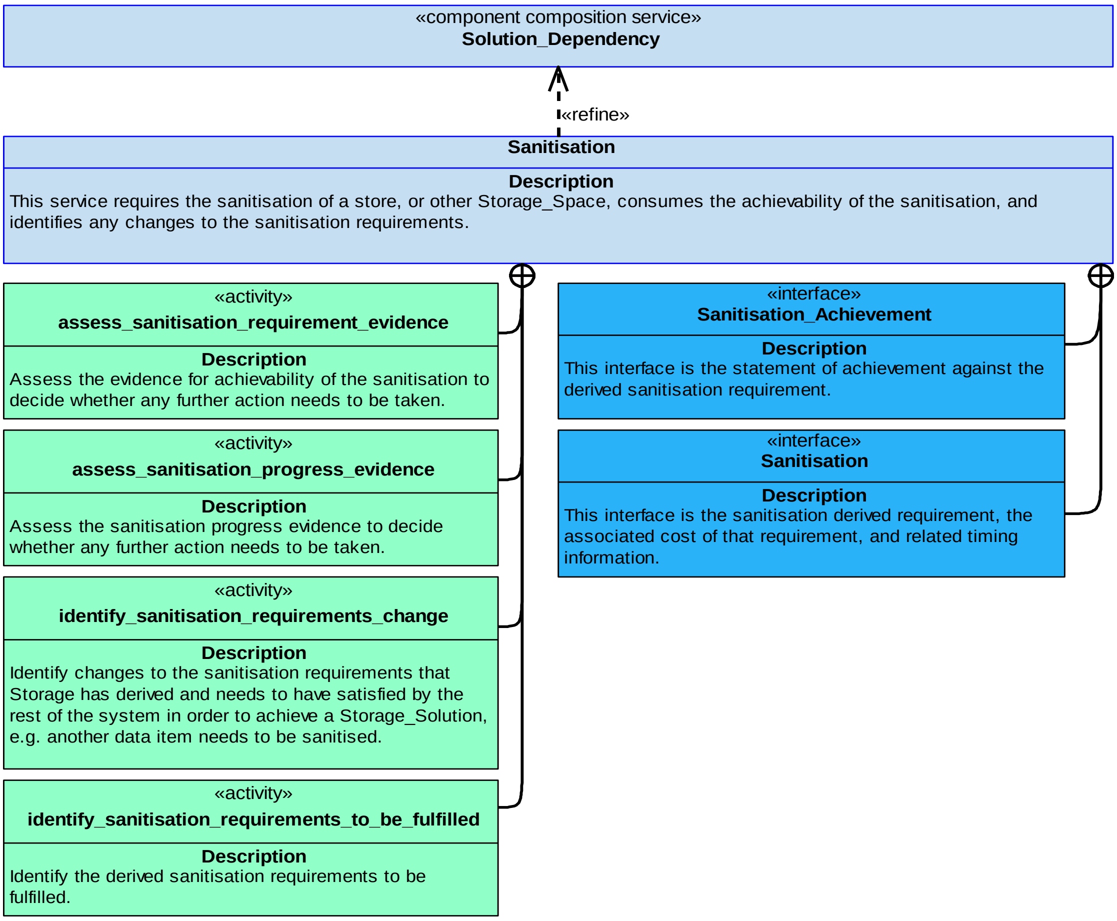

«component composition service»
Solution_Dependency
«refine»
Sanitisation
Description
This service requires the sanitisation of a store, or other Storage_Space, consumes the achievability of the sanitisation, and identifies any changes to the sanitisation requirements.
«activity»
assess_sanitisation_requirement_evidence
Description
Assess the evidence for achievability of the sanitisation to decide whether any further action needs to be taken.
«activity»
assess_sanitisation_progress_evidence
Description
Assess the sanitisation progress evidence to decide whether any further action needs to be taken.
«activity»
identify_sanitisation_requirements_change
Description
Identify changes to the sanitisation requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be sanitised.
«activity»
identify_sanitisation_requirements_to_be_fulfilled
Description
Identify the derived sanitisation requirements to be fulfilled.
«interface»
Sanitisation_Achievement
Description
This interface is the statement of achievement against the derived sanitisation requirement.
«interface»
Sanitisation
Description
This interface is the sanitisation derived requirement, the associated cost of that requirement, and related timing information.

Figure 987: Sanitisation Service Policy

Sanitisation

This service requires the sanitisation of a store, or other Storage_Space, consumes the achievability of the sanitisation, and identifies any changes to the sanitisation requirements.

Interfaces

Sanitisation

This interface is the sanitisation derived requirement, the associated cost of that requirement, and related timing information.

Attributes

sanitisation	The derived requirement to sanitise a store, drive, or part of a Storage_Space.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the sanitisation solution, e.g. resources used, time taken.

Sanitisation_Achievement

This interface is the statement of achievement against the derived sanitisation requirement.

Activities

assess_sanitisation_requirement_evidence

Assess the evidence for achievability of the sanitisation to decide whether any further action needs to be taken.

assess_sanitisation_progress_evidence

Assess the sanitisation progress evidence to decide whether any further action needs to be taken.

identify_sanitisation_requirements_change

Identify changes to the sanitisation requirements that Storage has derived and needs to have satisfied by the rest of the system in order to achieve a Storage_Solution, e.g. another data item needs to be sanitised.

identify_sanitisation_requirements_to_be_fulfilled

Identify the derived sanitisation requirements to be fulfilled.

5.4.2.57.7.1.6 Usage

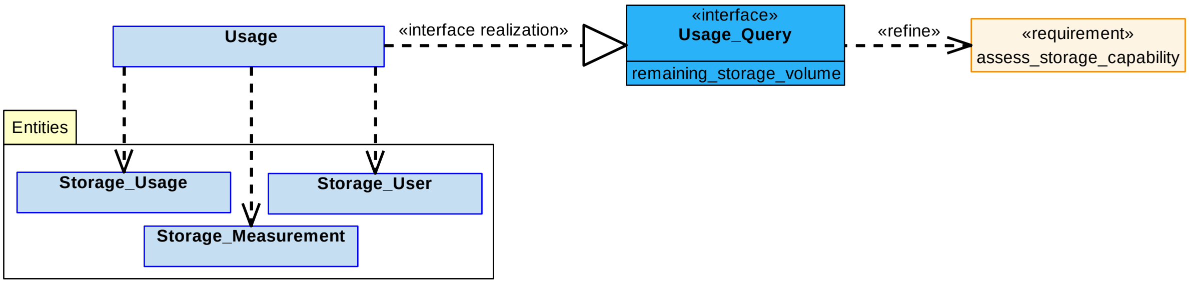


Figure 988: Usage Service Definition

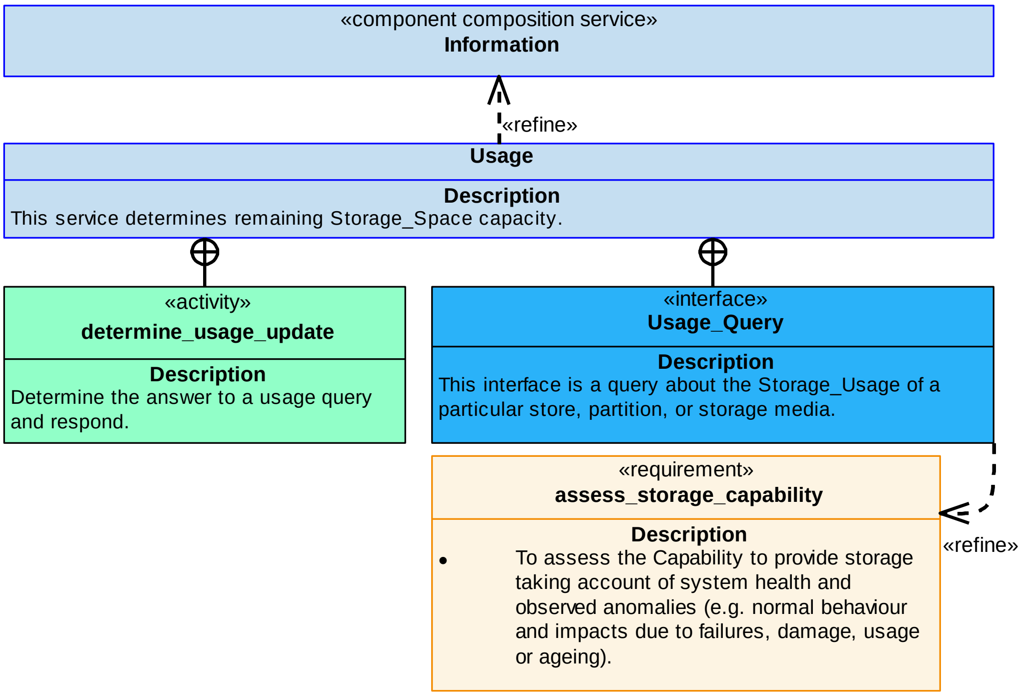

«component composition service»
Information
«refine»
Usage
Description
This service determines remaining Storage_Space capacity.
«activity»
determine_usage_update
Description
Determine the answer to a usage query and respond.
«interface»
Usage_Query
Description
This interface is a query about the Storage_Usage of a particular store, partition, or storage media.
«requirement»
assess_storage_capability
Description
• To assess the Capability to provide storage taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 989: Usage Service Policy

Usage

This service determines remaining Storage_Space capacity.

Interface

Usage_Query

This interface is a query about the Storage_Usage of a particular store, partition, or storage media.

Attribute

remaining_storage_volume	The remaining capacity for storage of a particular store, partition, or storage media.

Activity

determine_usage_update

Determine the answer to a usage query and respond.

5.4.2.57.7.1.7 Constraint

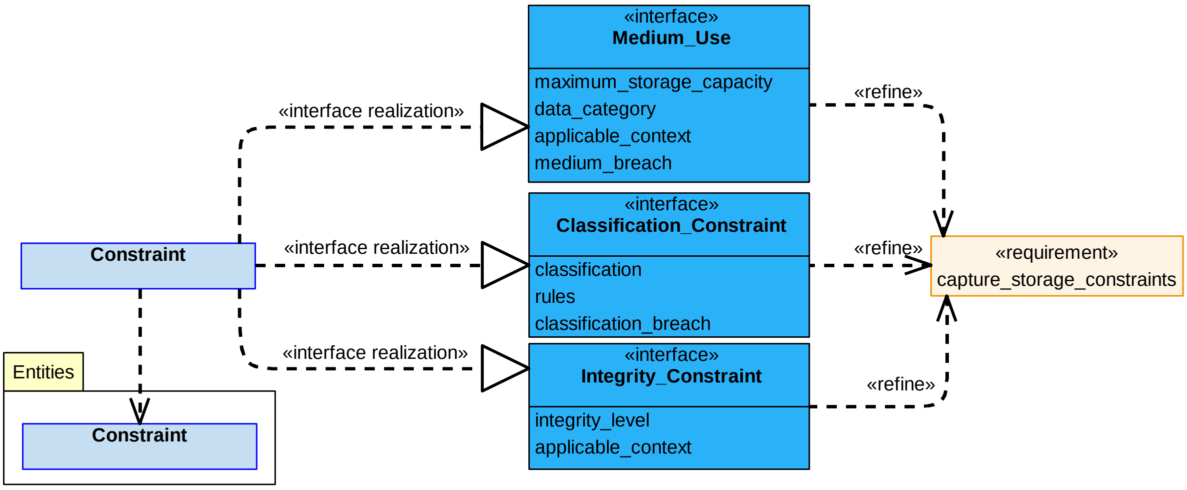


Figure 990: Constraint Service Definition

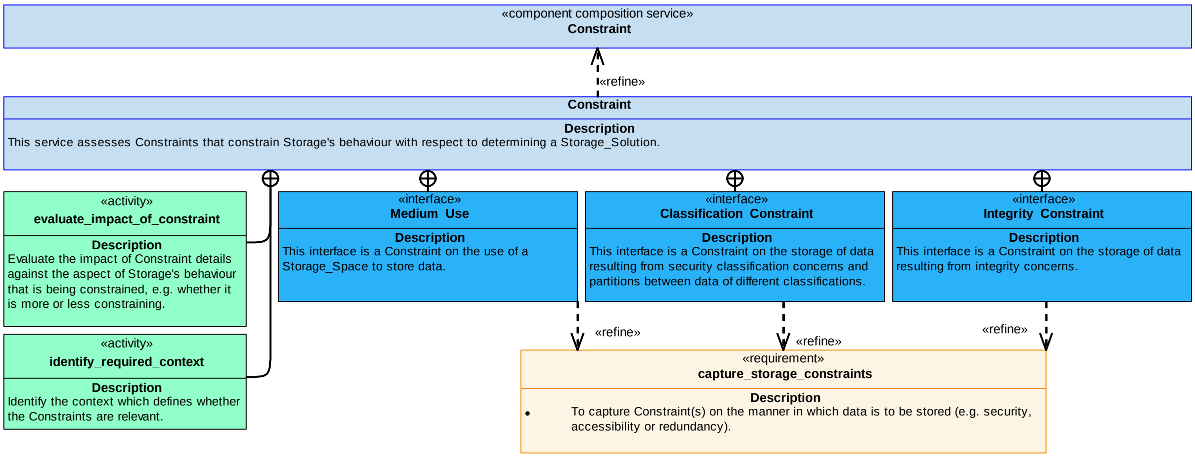

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses Constraints that constrain Storage's behaviour with respect to determining a Storage_Solution.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of Storage's behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Medium_Use
Description
This interface is a Constraint on the use of a Storage_Space to store data.
«interface»
Classification_Constraint
Description
This interface is a Constraint on the storage of data resulting from security classification concerns and partitions between data of different classifications.
«interface»
Integrity_Constraint
Description
This interface is a Constraint on the storage of data resulting from integrity concerns.
«refine»
«refine»
«refine»
«requirement»
capture_storage_constraints
Description
To capture Constraint(s) on the manner in which data is to be stored (e.g. security, accessibility or redundancy).

Figure 991: Constraint Service Policy

Constraint

This service assesses Constraints that constrain Storage's behaviour with respect to determining a Storage_Solution.

Interfaces

Medium_Use

This interface is a Constraint on the use of a Storage_Space to store data.

Attributes

maximum_storage_capacity	The maximum amount of a Storage_Space that is allowed to be used.
data_category	Which Data_Category(s) can be stored in a particular Storage_Space.
applicable_context	The context in which the Constraint is applicable.

medium_breach	A statement that the Constraint on the use of a Storage_Space has been breached.

Classification_Constraint

This interface is a Constraint on the storage of data resulting from security classification concerns and partitions between data of different classifications.

Attributes

classification	The security classification threshold for the storage of data, e.g. data must not be stored below Official-Sensitive.
rules	The rules by which data at different levels of security classification must be stored.
classification_breach	A statement that the classification Constraint has been breached.

Integrity_Constraint

This interface is a Constraint on the storage of data resulting from integrity concerns.

Attributes

integrity_level	The level of integrity that the data should be stored with, e.g. on a high integrity store because the data must be retained for a specific reason.
applicable_context	The context in which the Constraint is applicable.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of Storage's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.57.7.1.8 Capability

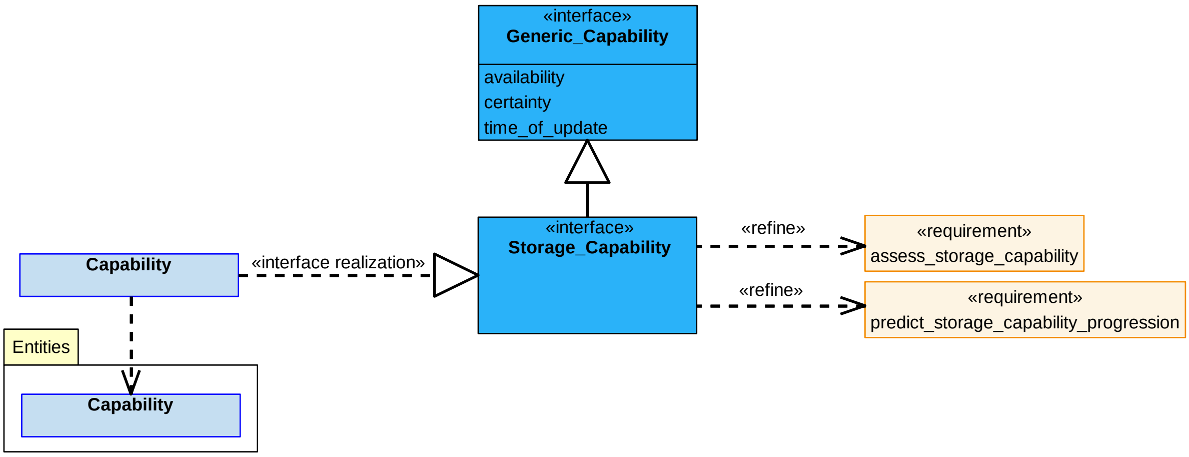


Figure 992: Capability Service Definition

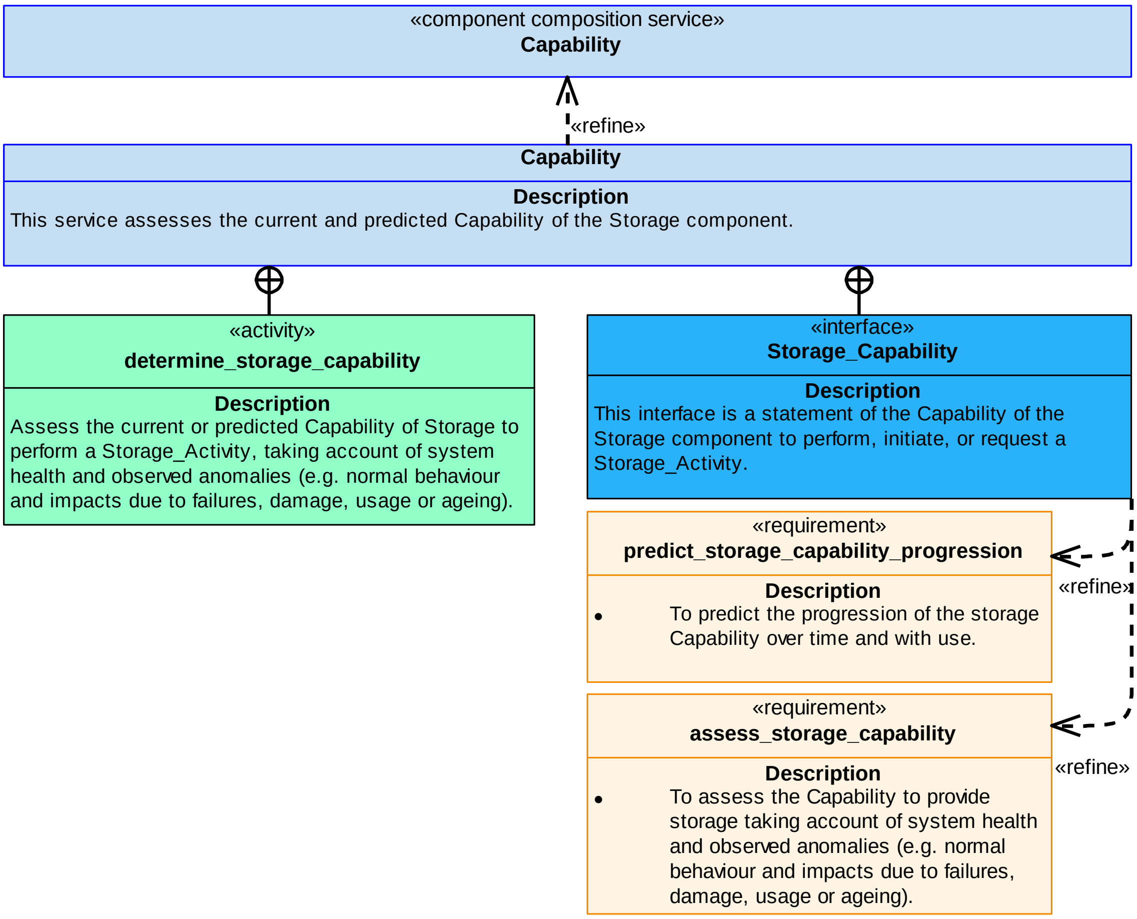

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability of the Storage component.
«activity»
determine_storage_capability
Description
Assess the current or predicted Capability of Storage to perform a Storage_Activity, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Storage_Capability
Description
This interface is a statement of the Capability of the Storage component to perform, initiate, or request a Storage_Activity.
«requirement»
predict_storage_capability_progression
Description
To predict the progression of the storage Capability over time and with use.
«refine»
«requirement»
assess_storage_capability
Description
To assess the Capability to provide storage taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 993: Capability Service Policy

Capability

This service assesses the current and predicted Capability of the Storage component.

Interface

Storage_Capability

This interface is a statement of the Capability of the Storage component to perform, initiate, or request a Storage_Activity.

Activity

determine_storage_capability

Assess the current or predicted Capability of Storage to perform a Storage_Activity, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.57.7.1.9 Capability_Evidence

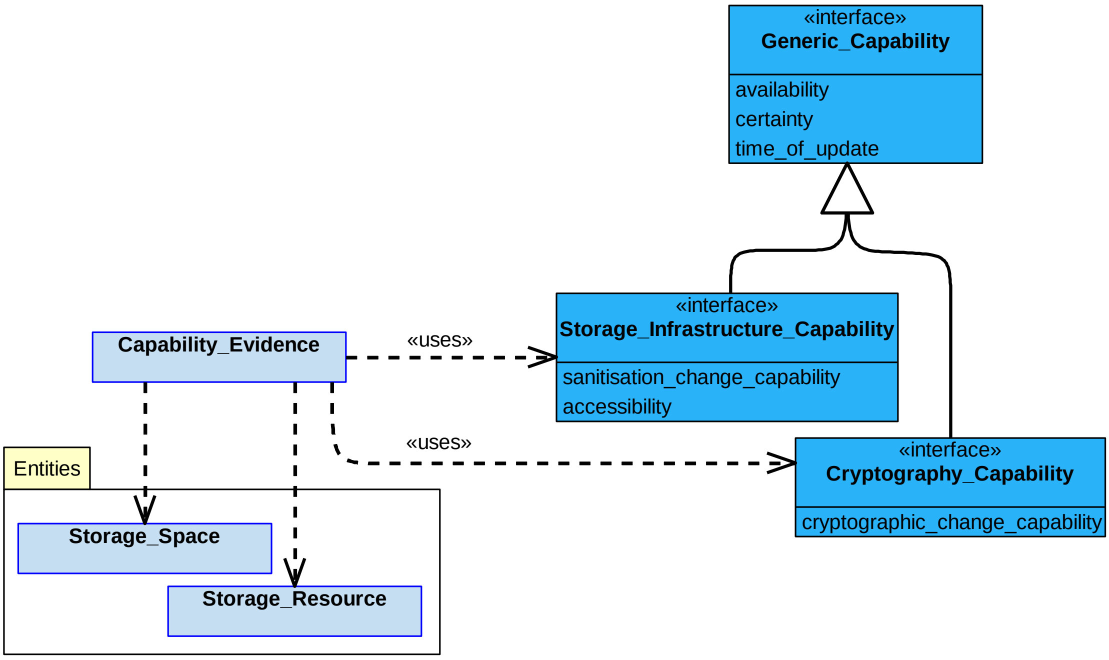


Figure 994: Capability_Evidence Service Definition

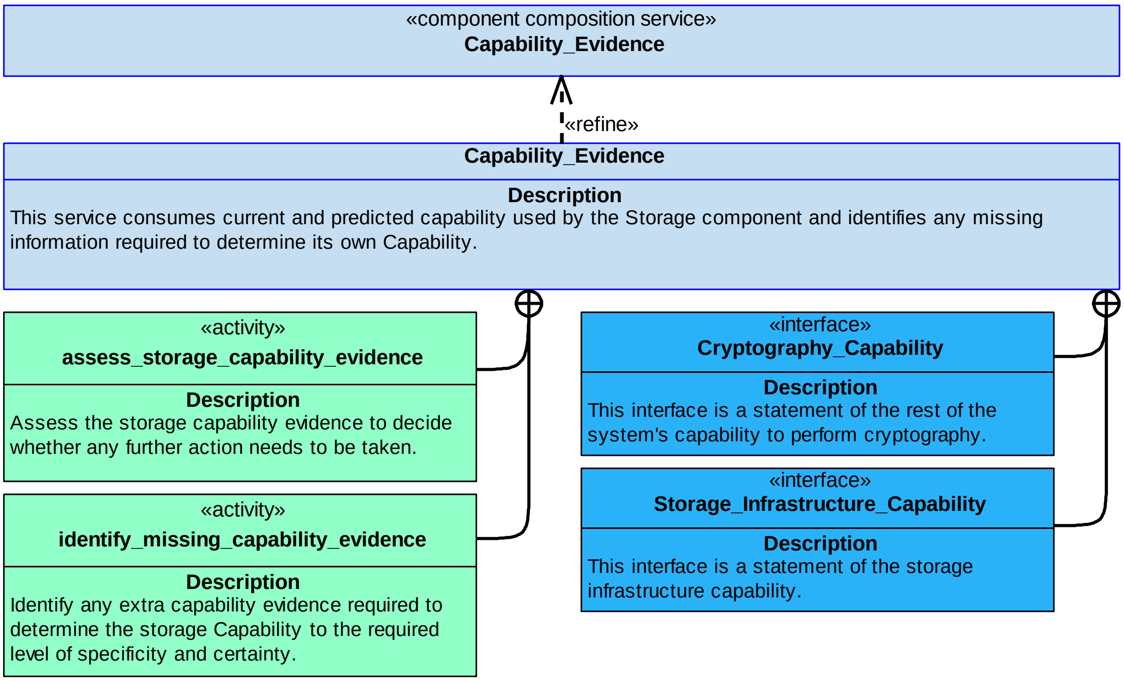

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability used by the Storage component and identifies any missing information required to determine its own Capability.
«activity»
assess_storage_capability_evidence
Description
Assess the storage capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the storage Capability to the required level of specificity and certainty.
«interface»
Cryptography_Capability
Description
This interface is a statement of the rest of the system's capability to perform cryptography.
«interface»
Storage_Infrastructure_Capability
Description
This interface is a statement of the storage infrastructure capability.

Figure 995: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability used by the Storage component and identifies any missing information required to determine its own Capability.

Interfaces

Storage_Infrastructure_Capability

This interface is a statement of the storage infrastructure capability.

Attributes

sanitisation_change_capability	An indication of whether a Storage_Space can be sanitised or not. This could be for a single data item, a store, partition or full storage media.
accessibility	An indication of the accessibility of the data, for example, inaccessibility resulting from the store data is located in being corrupt.

Cryptography_Capability

This interface is a statement of the rest of the system's capability to perform cryptography.

Attribute

cryptographic_change_capability	An indication of whether a Storage_Space can be encrypted or decrypted, or not. This could be for a single data item, a store, partition or full storage media.

Activities

assess_storage_capability_evidence

Assess the storage capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the storage Capability to the required level of specificity and certainty.

5.4.2.57.7.2 Service Dependencies

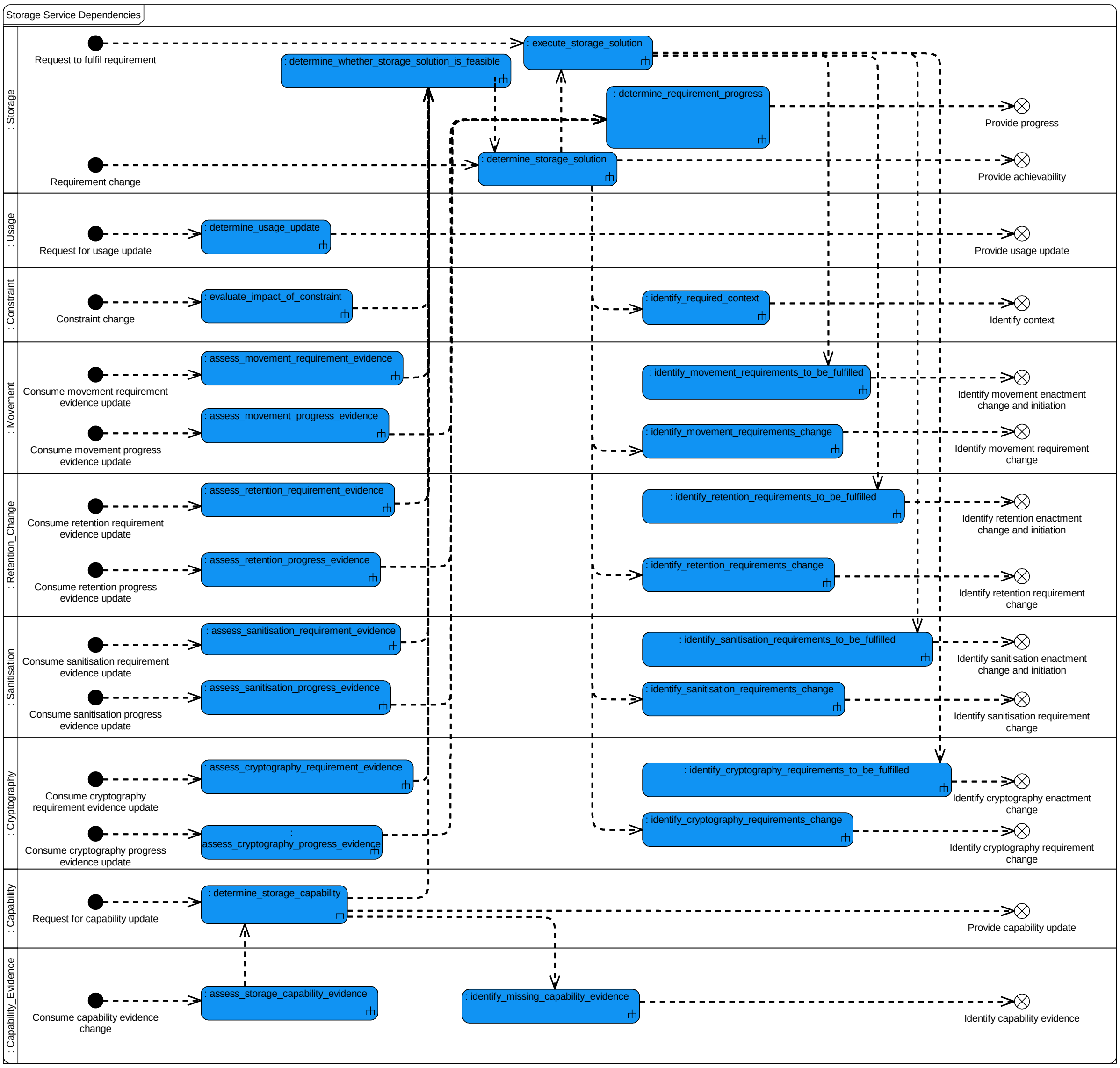

Storage Service Dependencies
: Storage
Request to fulfil requirement
: execute_storage_solution
: determine_whether_storage_solution_is_feasible
: determine_requirement_progress
Provide progress
Requirement change
: determine_storage_solution
Provide achievability
: Usage
Request for usage update
: determine_usage_update
Provide usage update
: Constraint
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
: Movement
Consume movement requirement evidence update
: assess_movement_requirement_evidence
: identify_movement_requirements_to_be_fulfilled
Identify movement enactment change and initiation
Consume movement progress evidence update
: assess_movement_progress_evidence
: identify_movement_requirements_change
Identify movement requirement change
: Retention_Change
Consume retention requirement evidence update
: assess_retention_requirement_evidence
: identify_retention_requirements_to_be_fulfilled
Identify retention enactment change and initiation
Consume retention progress evidence update
: assess_retention_progress_evidence
: identify_retention_requirements_change
Identify retention requirement change
: Sanitisation
Consume sanitisation requirement evidence update
: assess_sanitisation_requirement_evidence
: identify_sanitisation_requirements_to_be_fulfilled
Identify sanitisation enactment change and initiation
Consume sanitisation progress evidence update
: assess_sanitisation_progress_evidence
: identify_sanitisation_requirements_change
Identify sanitisation requirement change
: Cryptography
Consume cryptography requirement evidence update
: assess_cryptography_requirement_evidence
: identify_cryptography_requirements_to_be_fulfilled
Identify cryptography enactment change
Consume cryptography progress evidence update
: assess_cryptography_progress_evidence
: identify_cryptography_requirements_change
Identify cryptography requirement change
: Capability
Request for capability update
: determine_storage_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_storage_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 996: Storage Service Dependencies

5.4.2.58 Stores Release

5.4.2.58.1 Role

The role of Stores Release is to select stores for release and initiate their release at the appropriate time and in the appropriate sequence.

5.4.2.58.2 Overview

Control Architecture

Stores Release is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Stores Release will select the appropriate Stores to make up a Release_Package, and devise an associated Release_Schedule, in order to satisfy a Requirement for an operational release or jettison action. This Release_Schedule triggers the appropriate Release_Action_Steps required for the package. The Release_Schedule will be monitored throughout to ensure that it remains feasible.

Examples of Use

This component will be required where:

- The operational release of Stores will be required, e.g. gun rounds, bombs, chaff and flare or deployable sensors.
- Jettison of Stores (including items such as external fuel tanks) may be required to maintain safe operation.

5.4.2.58.3 Service Summary

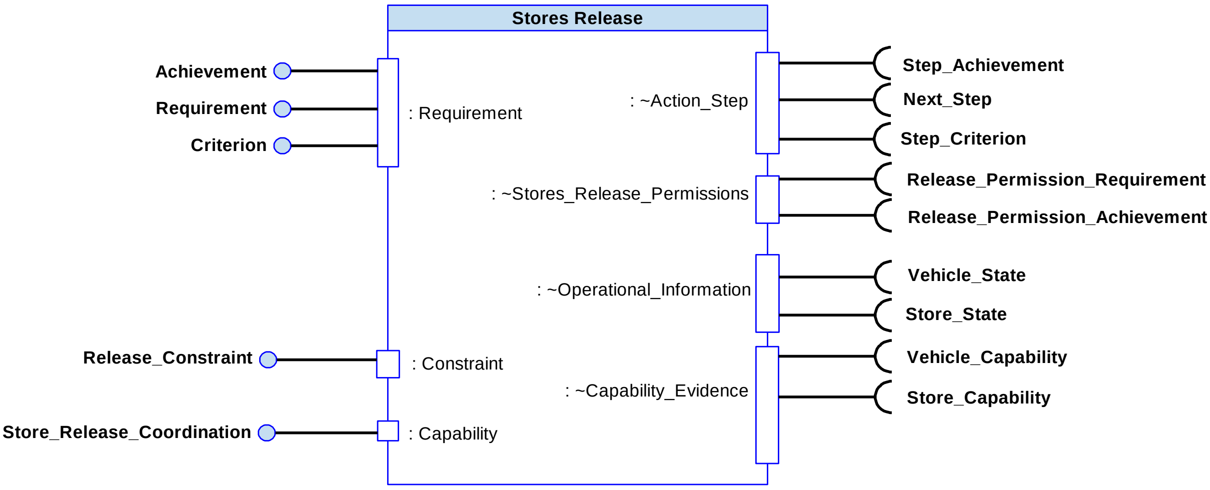


Figure 997: Stores Release Service Summary

5.4.2.58.4 Responsibilities

capture_release_package_requirement

- To capture the Requirements for the release of Stores.

capture_measurement_criteria

- To capture provided Measurement_Criterion which a Release_Schedule will be measured against.

capture_provided_constraints

- To capture provided Constraints for stores release.

determine_release_schedule

- To determine the Release_Schedule for Stores (when part of a Release_Package) with respect to provided Requirements and Constraints.

determine_stores_releasability

- To determine which Stores are releasable.

determine_release_packages

- To determine the Release_Packages (i.e. packages that do not compromise air vehicle safety) in response to a Requirement.

identify_whether_solution_is_feasible

- To identify whether a Release_Schedule in progress remains feasible given current resources.

identify_pre-conditions

- To identify Pre-conditions required to support the Release_Schedule.

execute_release_solution

- To coordinate the execution of a Release_Schedule.

determine_store_release_progress

- To determine the progress against the Release_Schedule (e.g. report whether Stores have hung or have left the platform).

determine_quality_of_solution

- To determine the quality of a proposed Release_Schedule against given Requirements.

assess_stores_release_capability

- To assess the Stores_Release_Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Stores_Release_Capability assessment.

predict_stores_release_capability_progression

- To predict the progression of the Stores_Release_Capability over time and with use.

5.4.2.58.5 Subject Matter Semantics

The subject matter of Stores Release is the activities that result in the operational release or jettison of Stores from the platform.

Exclusions

The subject matter of Stores Release does not include:

- Control of the equipment required to release a Store, only the sequence of the Release_Action_Steps so they occur in the correct order.
- The transmission of mission data or cryptos to a Store prior to its release.
- The validation of mass and balance effects of releasing a Release_Package, only ensuring the Release_Package follows the appropriate rules.
- The preparation of a Store for release (e.g. setting fusing options or verifying targeting data), only the Release_Schedule.

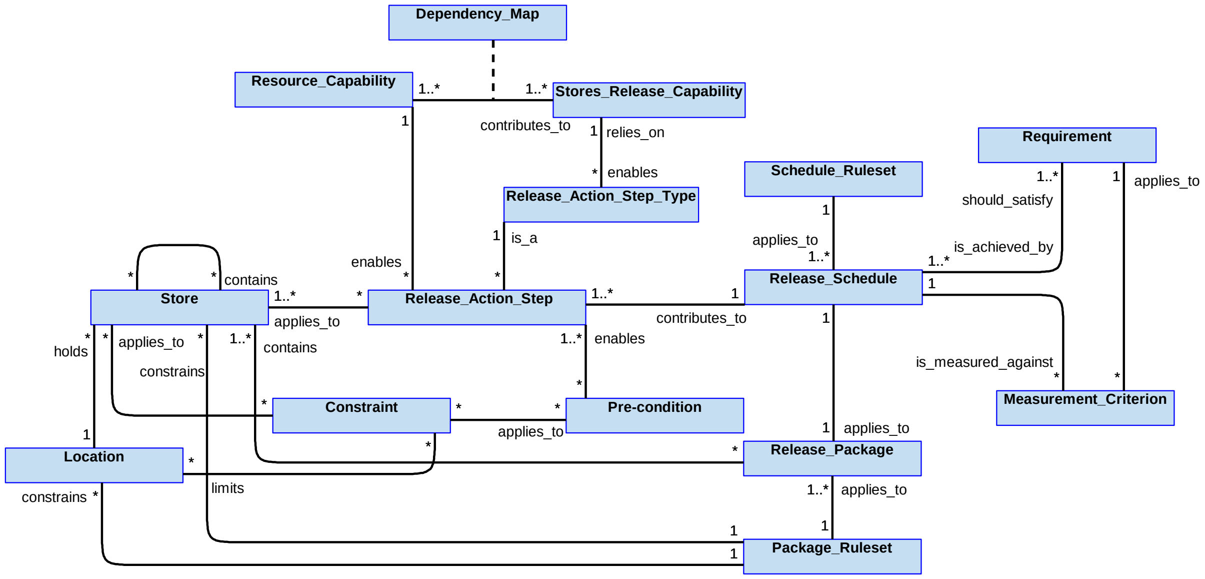


Figure 998: Stores Release Semantics

5.4.2.58.5.1 Entities

Constraint

An externally imposed restriction, e.g. limiting the ability of an Exploiting Platform to open a weapons bay door.

Dependency_Map

A mapping of how the component's Stores_Release_Capability is dependent on the Resource_Capability.

Location

A physical location on the Exploiting Platform that can hold a Store.

Measurement_Criterion

A criterion by which the release operation is measured, e.g. the time required to release a Release_Package.

Package_Ruleset

The rules under which one or more Stores, at given Locations, are allowed or not allowed to be in a Release_Package.

Pre-condition

A condition that must be true before an activity can take place (e.g. undercarriage is raised or the Master Armament Safety Switch is live).

Release_Action_Step_Type

The kinds of action this component knows how to coordinate (e.g. operational store release from a particular station or store jettison from a particular station).

Release_Package

A set of one or more Stores which can be safely released.

Release_Schedule

The order and timing in which actions must be performed in order to release one or more Stores from the Exploiting Platform (e.g. the inter-station schedule, opening doors prior to release and moving rotary launchers).

Requirement

A requirement placed in order to release one or more stores from the Exploiting Platform, e.g. to release stores of type x and the number to be released, or to release a specific nominated store.

Resource_Capability

The capability of the underlying resources to operationally release or jettison a Store.

Schedule_Ruleset

The rules that apply to the scheduling of release for a sequence of Stores, e.g. minimum release intervals.

Store

A specific individual item that can be operationally released or jettisoned from the Exploiting Platform. This can include carriage stores intended to carry other stores and that can be jettisoned (e.g. a multi weapons launcher that carries multiple missiles) or those that give a mission effect (e.g. a bomb or missile, extended range fuel tanks or a sensor pod).

Stores_Release_Capability

The capability to manage the release of Stores from the Exploiting Platform.

Release_Action_Step

An action that, when performed, achieves (or partially achieves) the release of a Store from the Exploiting Platform.

5.4.2.58.6 Design Rationale

5.4.2.58.6.1 Assumptions

- The Exploiting Platform will have appropriate interlocks, such as MASS, to prevent inadvertent release of stores.
- As part of executing the Release_Schedule this component may request interlocks are enabled, doors opened, etc.

5.4.2.58.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Stores Release:

- Data Driving - data driving should be used for the information needed by Stores Release, such as the minimum release interval between different stations based on the fitted store types. In addition, data driving should be used for the Schedule_Ruleset and Package_Ruleset.

Extensions

- It is unlikely that extensions will be appropriate.

Exploitation Considerations

- The rules under which a Store can become part of a Release_Package will be for an Exploiting Programme to define.
- The rules that apply to a Release_Schedule, e.g. minimum release intervals, will be for an Exploiting Programme to define.

5.4.2.58.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Failure of this component could cause Stores to be released in the wrong order (resulting in store to store collision or an out of balance condition), at the wrong time (resulting in store to store collision) or from the wrong stations (resulting in an out of balance condition). In the case of an air vehicle, this could result in uncontrolled flight due to exceedance of the flight envelope or loss of structural integrity (if store impacts air vehicle) leading to an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

5.4.2.58.6.4 Security Considerations

The indicative security classification is O-S.

This component is responsible for selecting a Release_Package, and triggering its release at the appropriate time in accordance with the Release_Schedule. The stores information and rules associated with the release are considered likely to be O-S. The component is one of a group of components involved in the release of stores from the Exploiting Platform, and whilst not responsible for the actual release of the weapon, it does determine the sequence in which the chosen package is released, taking into account the minimal release intervals and station order, etc. As such, its integrity and availability is essential for the safe release of stores, and in coordinating the release of stores when authorised to do so.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to requests for changes to interlocks, package or schedule rules, etc.
- **Maintaining Audit Records** of the packages selected and release actions performed during the mission, including where an authorised release is aborted prior to its execution, in order to support non-repudiation and audit of weapons release events.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.
- Performing **System Status and Monitoring** of the release sequence, unexpected events may indicate the system has been compromised.

The component is expected to at least partially satisfy Security Enforcing Functions by:

- **Verifying Integrity of Data** for the selected package and the release request, ensuring they have come from an authorised source.

5.4.2.58.7 Services

5.4.2.58.7.1 Service Definitions

5.4.2.58.7.1.1 Requirement

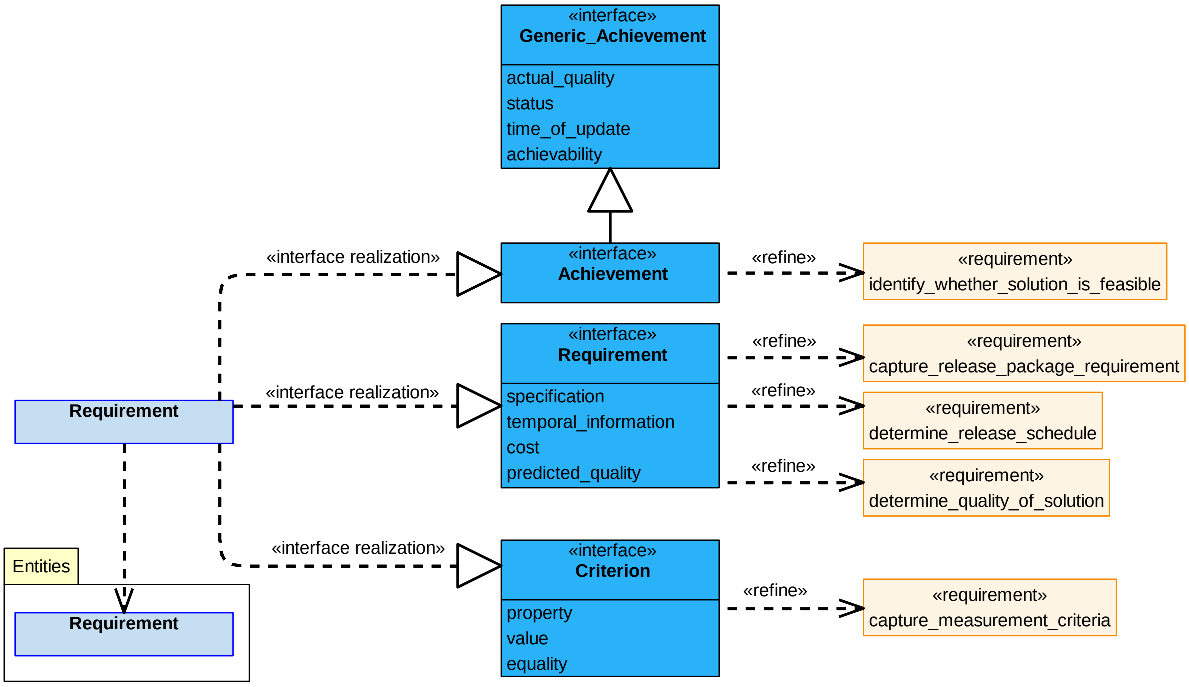
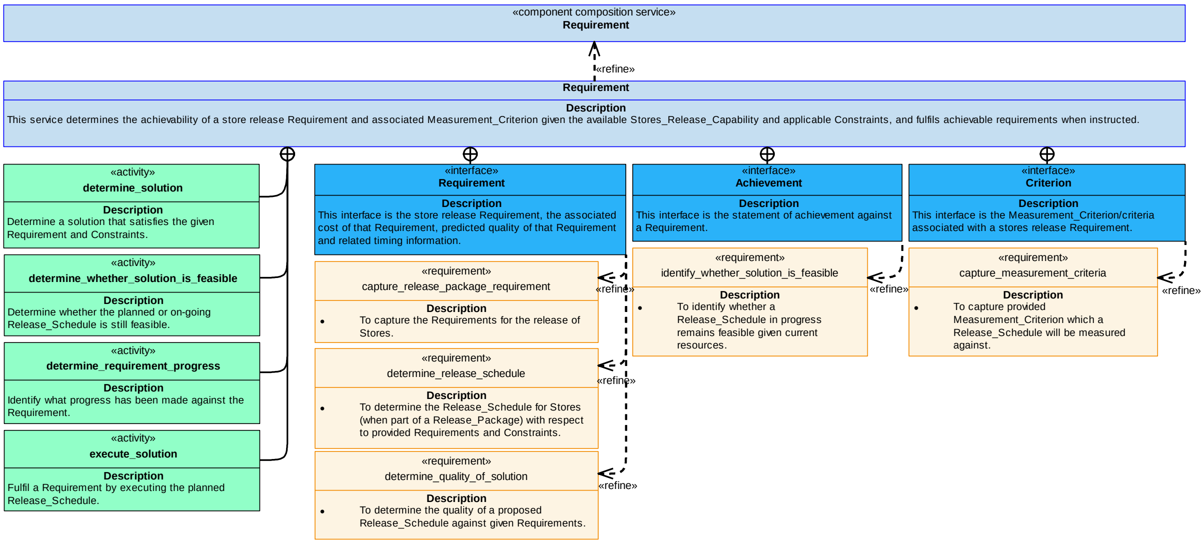


Figure 999: Requirement Service Definition



Requirement

Description
This service determines the achievability of a store release Requirement and associated Measurement_Criterion given the available Stores_Release_Capability and applicable Constraints, and fulfils achievable requirements when instructed.

«activity» determine_solution

Description
Determine a solution that satisfies the given Requirement and Constraints.

«activity» determine_whether_solution_is_feasible

Description
Determine whether the planned or on-going Release_Schedule is still feasible.

«activity» determine_requirement_progress

Description
Identify what progress has been made against the Requirement.

«activity» execute_solution

Description
Fulfil a Requirement by executing the planned Release_Schedule.

«interface» Requirement

Description
This interface is the store release Requirement, the associated cost of that Requirement, predicted quality of that Requirement and related timing information.

«requirement» capture_release_package_requirement

Description
- To capture the Requirements for the release of Stores.

«requirement» determine_release_schedule

Description
- To determine the Release_Schedule for Stores (when part of a Release_Package) with respect to provided Requirements and Constraints.

«requirement» determine_quality_of_solution

Description
- To determine the quality of a proposed Release_Schedule against given Requirements.

«interface» Achievement

Description
This interface is the statement of achievement against a Requirement.

«requirement» identify_whether_solution_is_feasible

Description
- To identify whether a Release_Schedule in progress remains feasible given current resources.

«interface» Criterion

Description
This interface is the Measurement_Criterion/criteria associated with a stores release Requirement.

«requirement» capture_measurement_criteria

Description
- To capture provided Measurement_Criterion which a Release_Schedule will be measured against.

Figure 1000: Requirement Service Policy

Requirement

This service determines the achievability of a store release Requirement and associated Measurement_Criterion given the available Stores_Release_Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Criterion

This interface is the Measurement_Criterion/criteria associated with a stores release Requirement.

Attributes

property	The property to be measured, e.g. quantity of stores remaining.
value	The measured value of the property, e.g. 100.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Achievement

This interface is the statement of achievement against a Requirement.

Requirement

This interface is the store release Requirement, the associated cost of that Requirement, predicted quality of that Requirement and related timing information.

Attributes

specification	The definition of a Requirement, e.g. to select 2x 500 lb bombs for release and initiate their release at the appropriate time and in the appropriate sequence.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used or time taken.
predicted_quality	How well the planned Release_Schedule is predicted to satisfy the requirement.

Activities

execute_solution

Fulfil a Requirement by executing the planned Release_Schedule.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Release_Schedule is still feasible.

determine_solution

Determine a solution that satisfies the given Requirement and Constraints.

determine_requirement_progress

Identify what progress has been made against the Requirement.

5.4.2.58.7.1.2 Action_Step

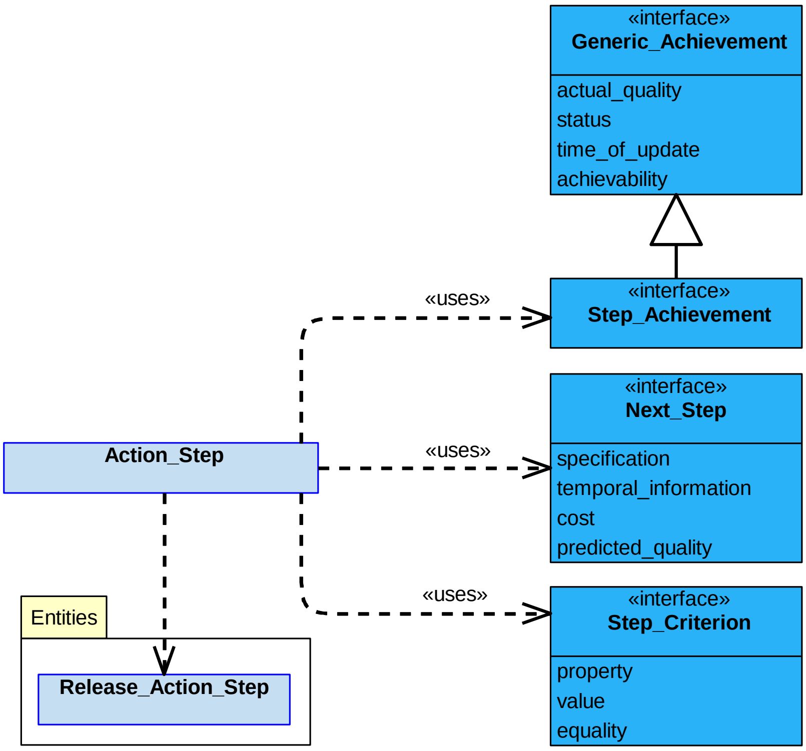


Figure 1001: Action_Step Service Definition

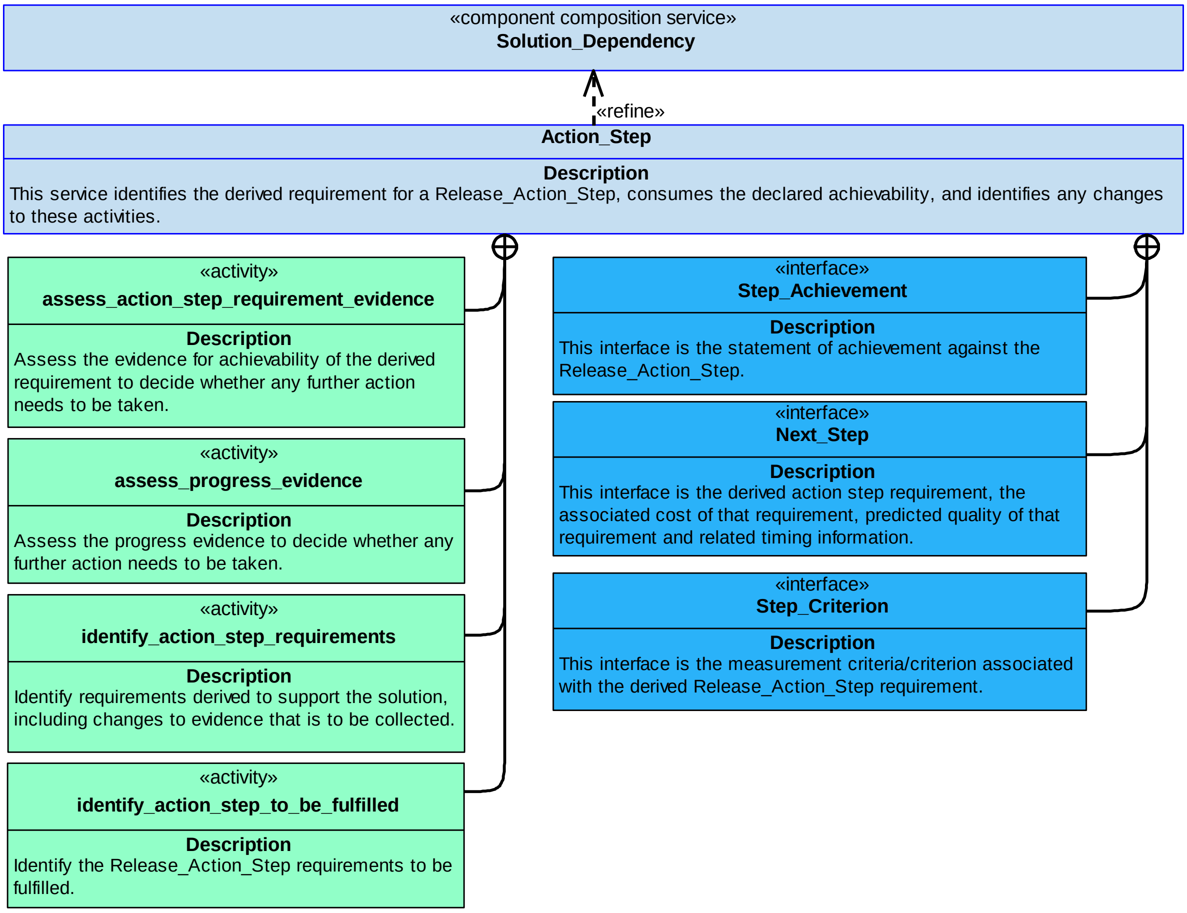

«component composition service»
Solution_Dependency
«refine»
Action_Step
Description
This service identifies the derived requirement for a Release_Action_Step, consumes the declared achievability, and identifies any changes to these activities.
«activity»
assess_action_step_requirement_evidence
Description
Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
identify_action_step_requirements
Description
Identify requirements derived to support the solution, including changes to evidence that is to be collected.
«activity»
identify_action_step_to_be_fulfilled
Description
Identify the Release_Action_Step requirements to be fulfilled.
«interface»
Step_Achievement
Description
This interface is the statement of achievement against the Release_Action_Step.
«interface»
Next_Step
Description
This interface is the derived action step requirement, the associated cost of that requirement, predicted quality of that requirement and related timing information.
«interface»
Step_Criterion
Description
This interface is the measurement criteria/criterion associated with the derived Release_Action_Step requirement.

Figure 1002: Action_Step Service Policy

Action_Step

This service identifies the derived requirement for a Release_Action_Step, consumes the declared achievability, and identifies any changes to these activities.

Interfaces

Next_Step

This interface is the derived action step requirement, the associated cost of that requirement, predicted quality of that requirement and related timing information.

Attributes

specification	The definition of the Release_Action_Step requirement, e.g. prepare the package for jettison.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the Release_Action_Step in a Release_Schedule, e.g. resources used or time taken.
predicted_quality	How well the planned Release_Action_Step is predicted to satisfy the requirement.

Step_Criterion

This interface is the measurement criteria/criterion associated with the derived Release_Action_Step requirement.

Attributes

property	The property to be measured, e.g. quantity of stores released.
value	The measured value of the property, e.g. 5.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Step_Achievement

This interface is the statement of achievement against the Release_Action_Step.

Activities

assess_action_step_requirement_evidence

Assess the evidence for achievability of the derived requirement to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_action_step_requirements

Identify requirements derived to support the solution, including changes to evidence that is to be collected.

identify_action_step_to_be_fulfilled

Identify the Release_Action_Step requirements to be fulfilled.

5.4.2.58.7.1.3 Stores_Release_Permissions

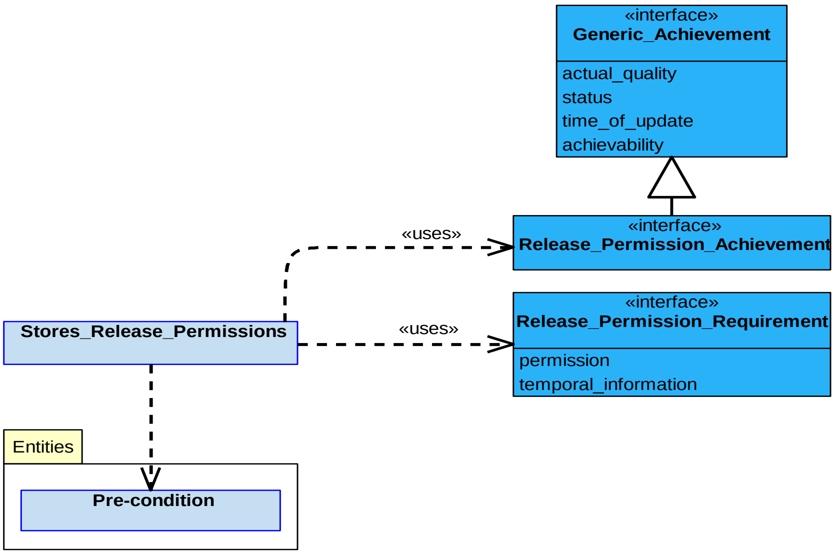


Figure 1003: Stores Release Permissions Service Definition

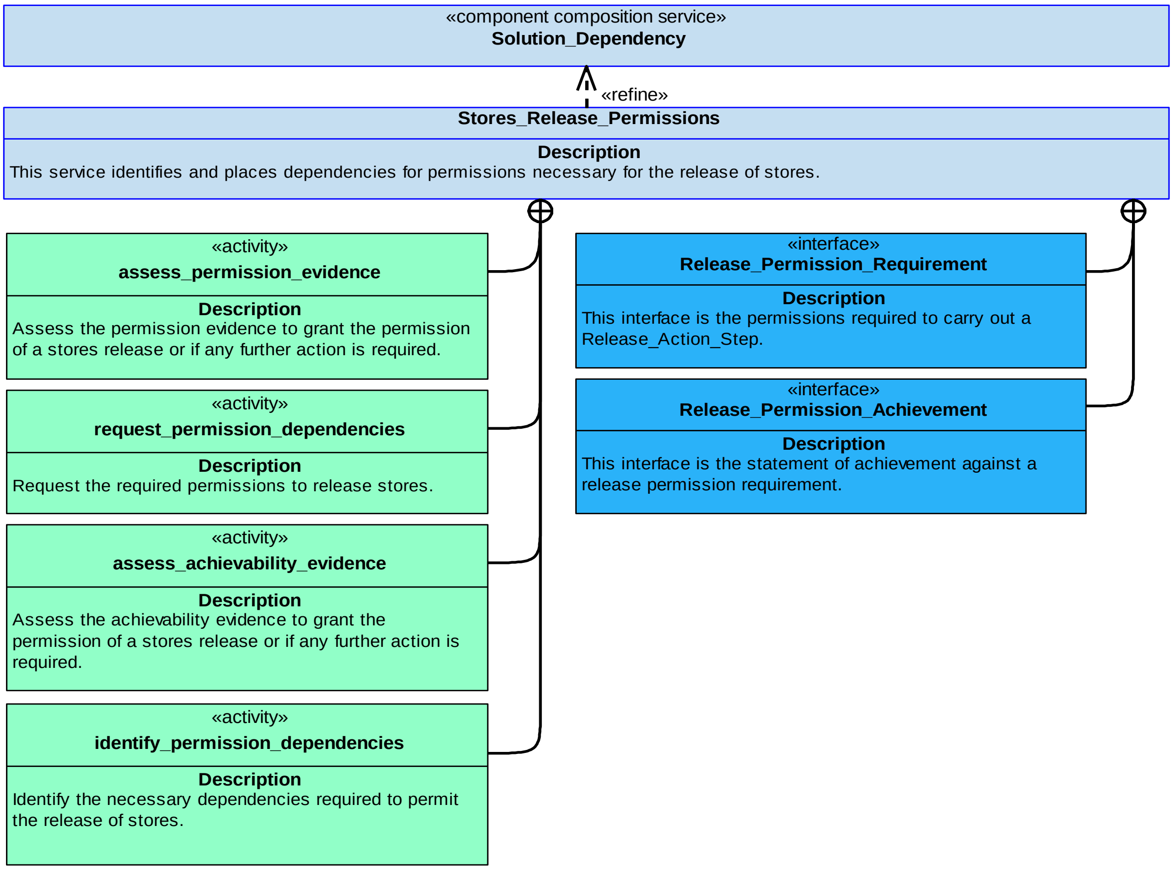

«component composition service»
Solution_Dependency
«refine»
Stores_Release_Permissions
Description
This service identifies and places dependencies for permissions necessary for the release of stores.
«activity»
assess_permission_evidence
Description
Assess the permission evidence to grant the permission of a stores release or if any further action is required.
«activity»
request_permission_dependencies
Description
Request the required permissions to release stores.
«activity»
assess_achievability_evidence
Description
Assess the achievability evidence to grant the permission of a stores release or if any further action is required.
«activity»
identify_permission_dependencies
Description
Identify the necessary dependencies required to permit the release of stores.
«interface»
Release_Permission_Requirement
Description
This interface is the permissions required to carry out a Release_Action_Step.
«interface»
Release_Permission_Achievement
Description
This interface is the statement of achievement against a release permission requirement.

Figure 1004: Stores Release Permissions Service Policy

Stores_Release_Permissions

This service identifies and places dependencies for permissions necessary for the release of stores.

Interfaces

Release_Permission_Requirement

This interface is the permissions required to carry out a Release_Action_Step.

Attributes

permission	The permission required for a store Release_Action_Step.
temporal_information	Information covering timing, such as start and end times.

Release_Permission_Achievement

This interface is the statement of achievement against a release permission requirement.

Activities

assess_permission_evidence

Assess the permission evidence to grant the permission of a stores release or if any further action is required.

request_permission_dependencies

Request the required permissions to release stores.

identify_permission_dependencies

Identify the necessary dependencies required to permit the release of stores.

assess_achievability_evidence

Assess the achievability evidence to grant the permission of a stores release or if any further action is required.

5.4.2.58.7.1.4 Operational_Information

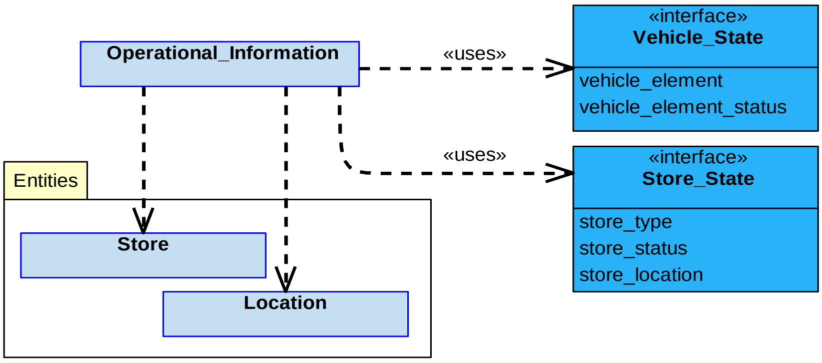


Figure 1005: Operational_Information Service Definition

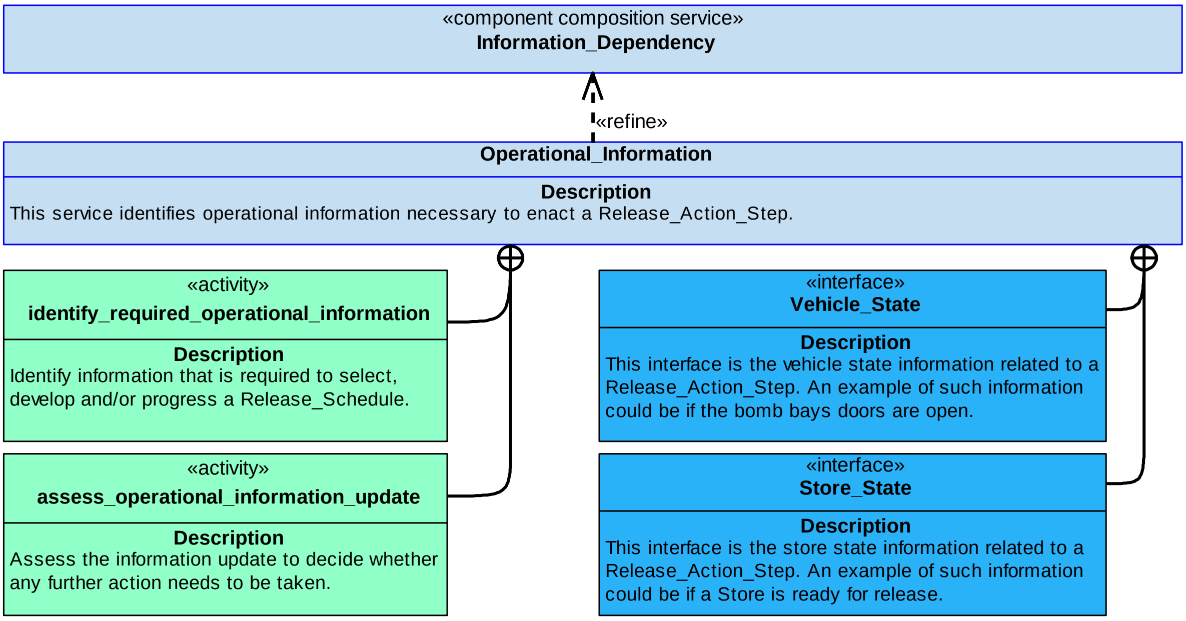

«component composition service»
Information_Dependency
«refine»
Operational_Information
Description
This service identifies operational information necessary to enact a Release_Action_Step.
«activity»
identify_required_operational_information
Description
Identify information that is required to select, develop and/or progress a Release_Schedule.
«activity»
assess_operational_information_update
Description
Assess the information update to decide whether any further action needs to be taken.
«interface»
Vehicle_State
Description
This interface is the vehicle state information related to a Release_Action_Step. An example of such information could be if the bomb bays doors are open.
«interface»
Store_State
Description
This interface is the store state information related to a Release_Action_Step. An example of such information could be if a Store is ready for release.

Figure 1006: Operational_Information Service Policy

Operational_Information

This service identifies operational information necessary to enact a Release_Action_Step.

Interfaces

Store_State

This interface is the store state information related to a Release_Action_Step. An example of such information could be if a Store is ready for release.

Attributes

store_type	The type of the Store.
store_status	The status of a store on an Exploiting Platform. For example, if a Store is ready for release.
store_location	The location of a Store on an Exploiting Platform. For example, a bomb bay or wing pylon.

Vehicle_State

This interface is the vehicle state information related to a Release_Action_Step. An example of such information could be if the bomb bays doors are open.

Attributes

vehicle_element	A specific element on an Exploiting Platform which may affect a release solution. For example, a bomb bay door or a hard point.
vehicle_element_status	The status of the vehicle_element. For example, is the bomb bay door open or a hard point energised.

Activities

identify_required_operational_information

Identify information that is required to select, develop and/or progress a Release_Schedule.

assess_operational_information_update

Assess the information update to decide whether any further action needs to be taken.

5.4.2.58.7.1.5 Constraint

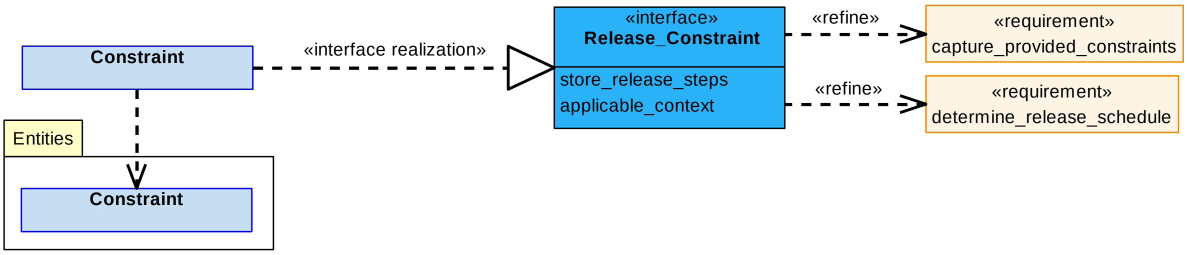


Figure 1007: Constraint Service Definition

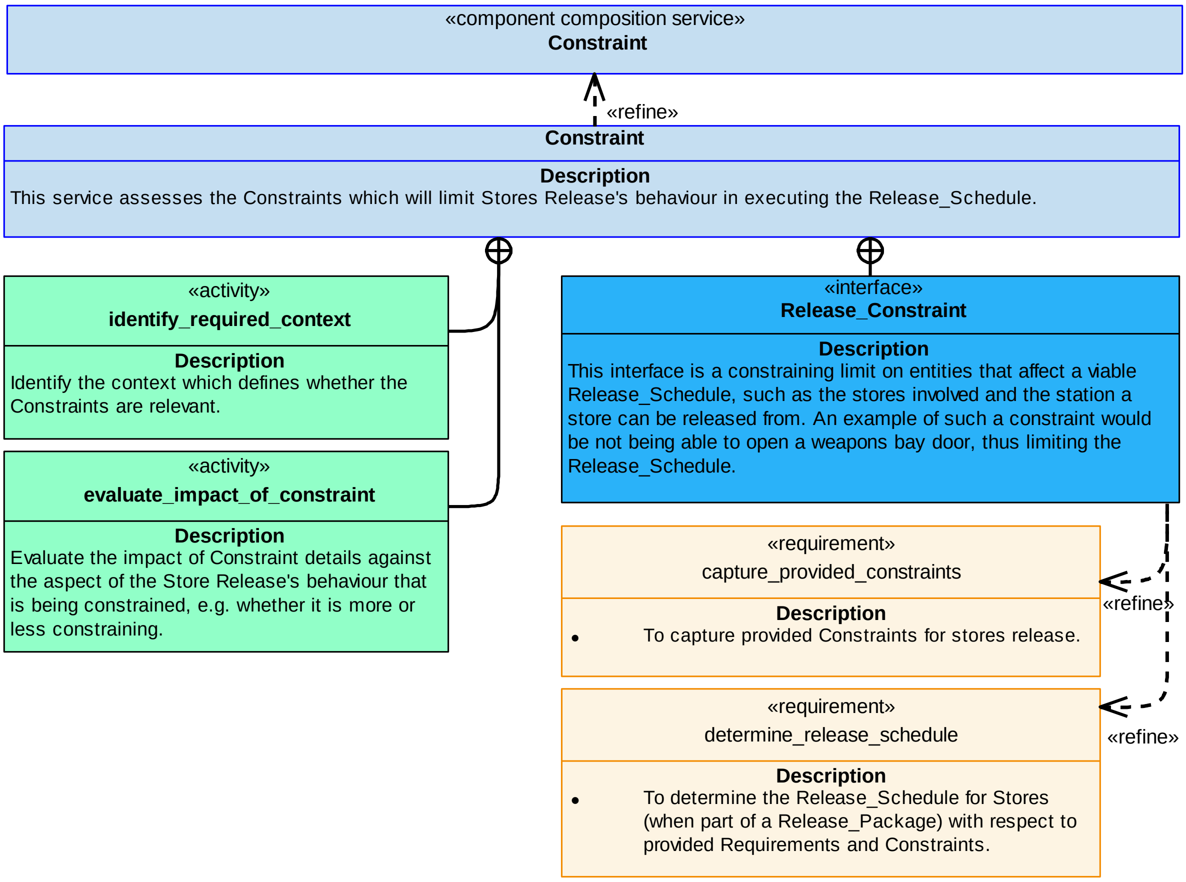

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Constraints which will limit Stores Release's behaviour in executing the Release_Schedule.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of the Store Release's behaviour that is being constrained, e.g. whether it is more or less constraining.
«interface»
Release_Constraint
Description
This interface is a constraining limit on entities that affect a viable Release_Schedule, such as the stores involved and the station a store can be released from. An example of such a constraint would be not being able to open a weapons bay door, thus limiting the Release_Schedule.
«requirement»
capture_provided_constraints
Description
To capture provided Constraints for stores release.
«refine»
«requirement»
determine_release_schedule
Description
To determine the Release_Schedule for Stores (when part of a Release_Package) with respect to provided Requirements and Constraints.
«refine»

Figure 1008: Constraint Service Policy

Constraint

This service assesses the Constraints which will limit Stores Release's behaviour in executing the Release_Schedule.

Interface

Release_Constraint

This interface is a constraining limit on entities that affect a viable Release_Schedule, such as the stores involved and the station a store can be released from. An example of such a constraint would be not being able to open a weapons bay door, thus limiting the Release_Schedule.

Attributes

store_release_steps	The constraining limit on the entities that Stores Release can use during a Release_Schedule.
applicable_context	The context in which the Constraint is applicable.

Activities

identify_required_context

Identify the context which defines whether the Constraints are relevant.

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of the Store Release's behaviour that is being constrained, e.g. whether it is more or less constraining.

5.4.2.58.7.1.6 Capability

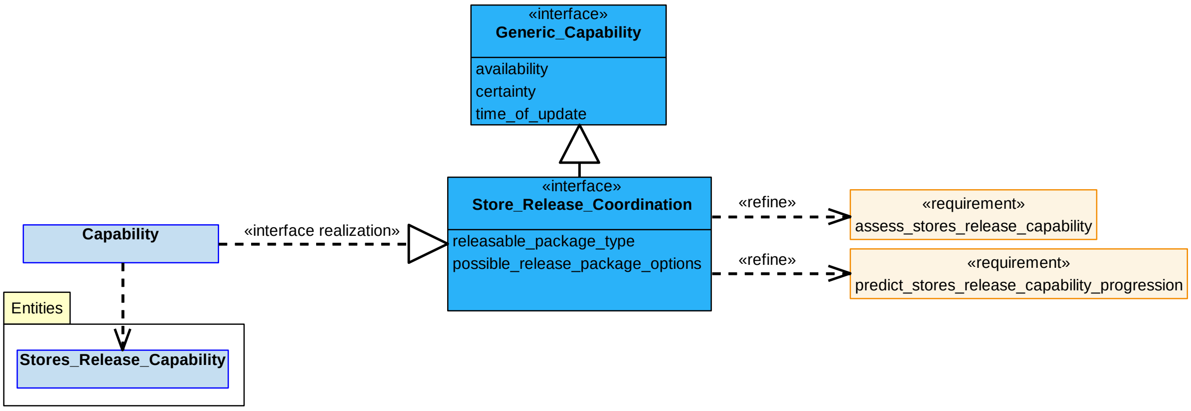


Figure 1009: Capability Service Definition

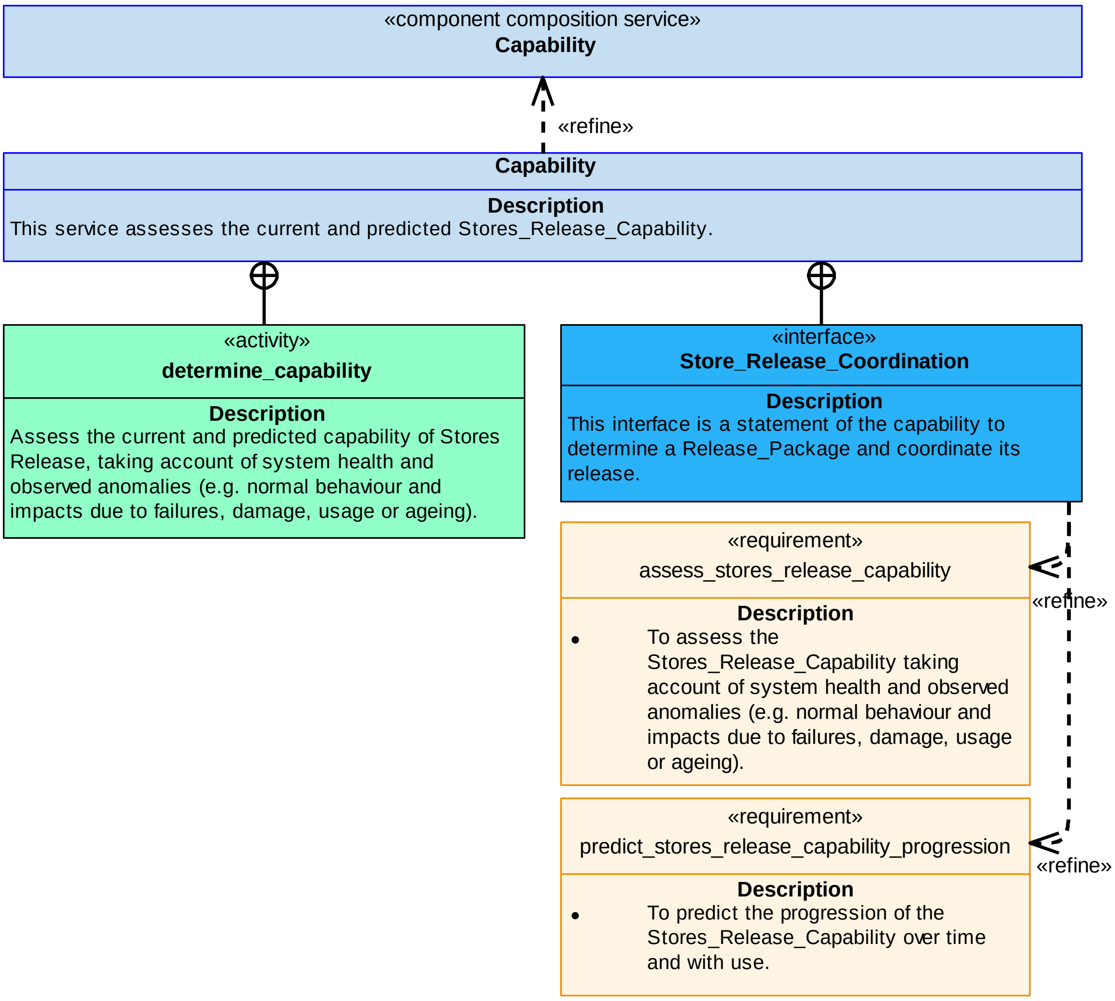

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Stores_Release_Capability.
«activity»
determine_capability
Description
Assess the current and predicted capability of Stores Release, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Store_Release_Coordination
Description
This interface is a statement of the capability to determine a Release_Package and coordinate its release.
«requirement»
assess_stores_release_capability
Description
To assess the Stores_Release_Capability taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_stores_release_capability_progression
Description
To predict the progression of the Stores_Release_Capability over time and with use.
«refine»

Figure 1010: Capability Service Policy

Capability

This service assesses the current and predicted Stores_Release_Capability.

Interface

Store_Release_Coordination

This interface is a statement of the capability to determine a Release_Package and coordinate its release.

Attributes

releasable_package_type	The possible Stores comprising a Release_Package.
possible_release_package_options	The possible release options for a Release_Package.

Activity

determine_capability

Assess the current and predicted capability of Stores Release, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.58.7.1.7 Capability_Evidence

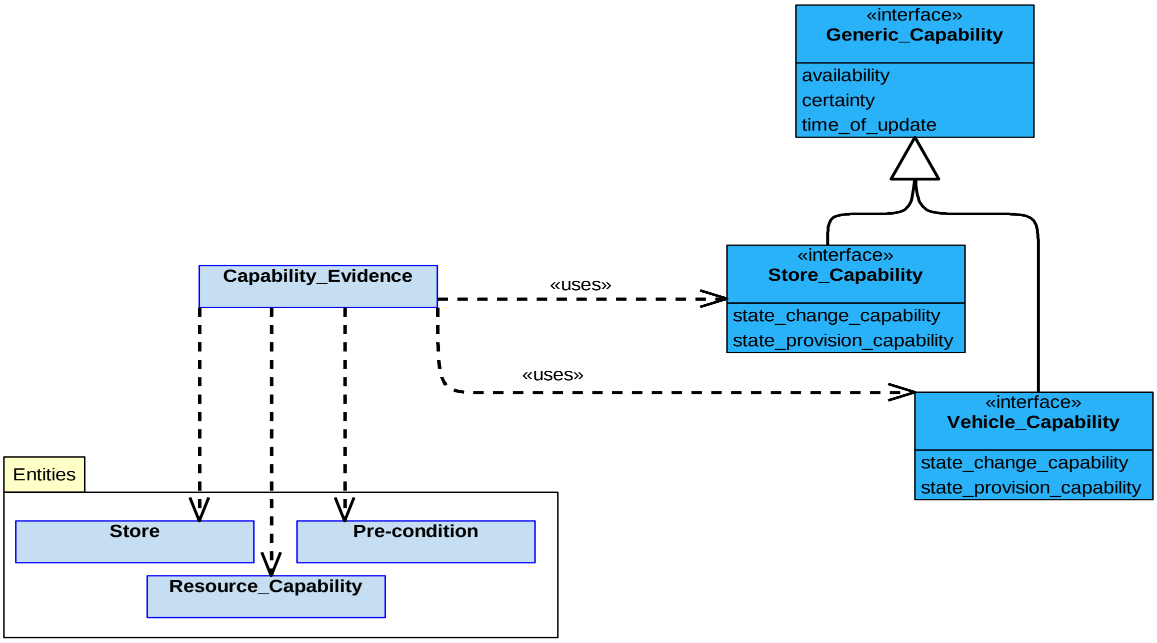


Figure 1011: Capability_Evidence Service Definition

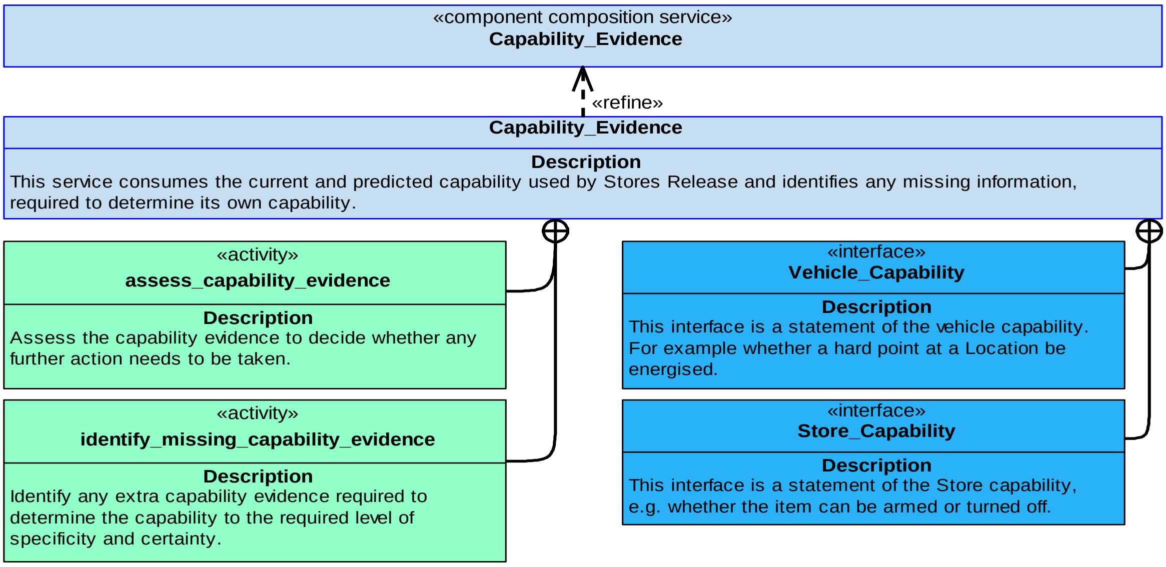

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted capability used by Stores Release and identifies any missing information, required to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.
«interface»
Vehicle_Capability
Description
This interface is a statement of the vehicle capability. For example whether a hard point at a Location be energised.
«interface»
Store_Capability
Description
This interface is a statement of the Store capability, e.g. whether the item can be armed or turned off.

Figure 1012: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capability used by Stores Release and identifies any missing information, required to determine its own capability.

Interfaces

Store_Capability

This interface is a statement of the Store capability, e.g. whether the item can be armed or turned off.

Attributes

state_change_capability	An indication of whether the state of a Store can be changed or not, e.g. whether it can be armed.
state_provision_capability	An indication of whether the state of a Store can be determined.

Vehicle_Capability

This interface is a statement of the vehicle capability. For example whether a hard point at a Location be energised.

Attributes

state_change_capability	An indication of whether the state of the infrastructure needed to enact a Release_Action_Step can be changed or not, e.g. whether bay doors can be opened or not.
state_provision_capability	An indication of whether the state of the infrastructure needed to enact a Release_Action_Step can be determined.

Activities

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

5.4.2.58.7.2 Service Dependencies

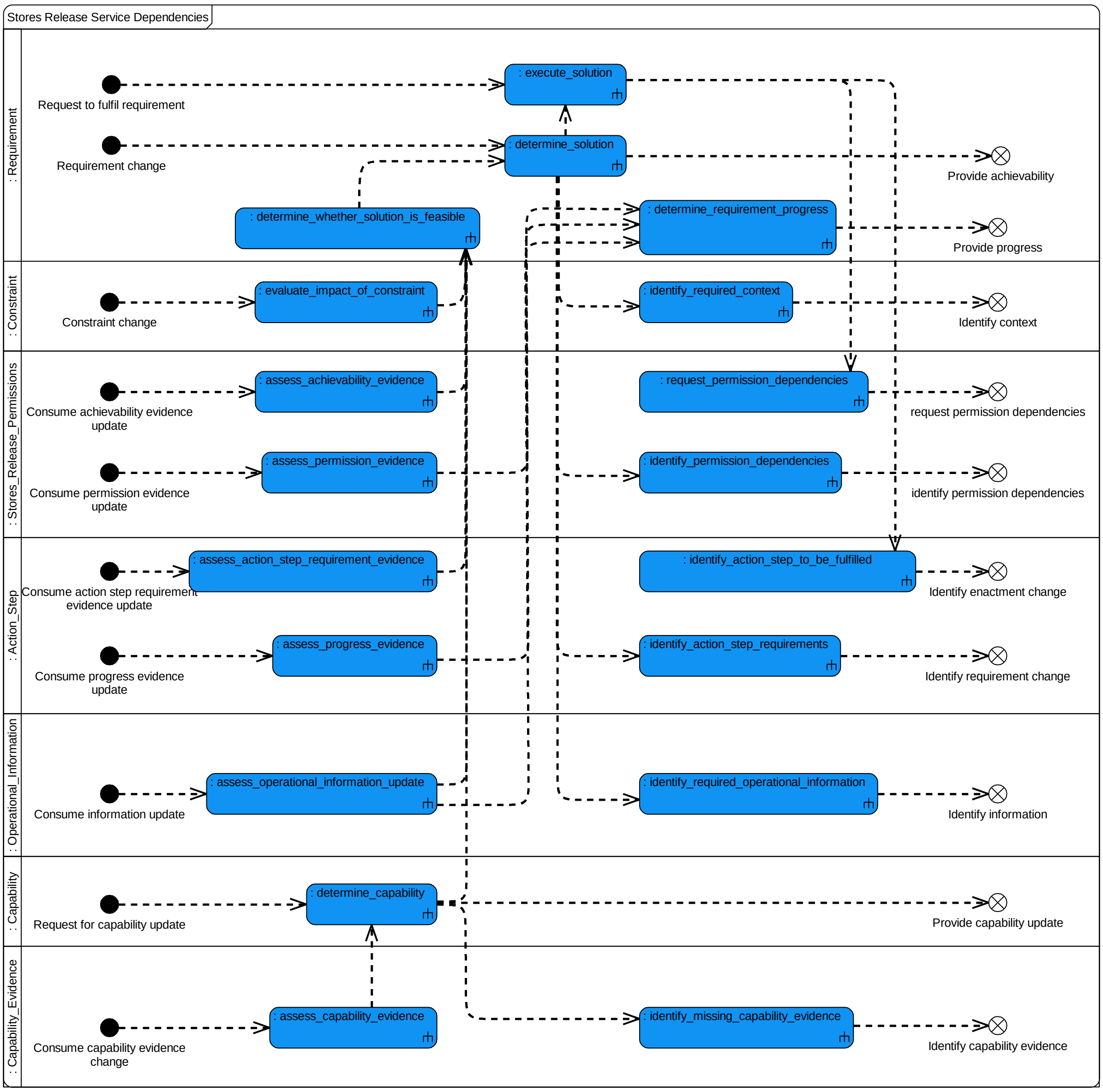

Stores Release Service Dependencies
: Requirement
: Constraint
: Stores_Release_Permissions
: Action_Step
: Operational_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
: execute_solution
: determine_solution
: determine_whether_solution_is_feasible
: determine_requirement_progress
Provide achievability
Provide progress
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume achievability evidence update
: assess_achievability_evidence
: request_permission_dependencies
request permission dependencies
Consume permission evidence update
: assess_permission_evidence
: identify_permission_dependencies
identify permission dependencies
Consume action step requirement evidence update
: assess_action_step_requirement_evidence
: identify_action_step_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_progress_evidence
: identify_action_step_requirements
Identify requirement change
Consume information update
: assess_operational_information_update
: identify_required_operational_information
Identify information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1013: Stores Release Service Dependencies

5.4.2.59 Susceptibility

5.4.2.59.1 Role

The role of Susceptibility is to provide a view of a subject's potential susceptibility to aggressor actions.

5.4.2.59.2 Overview

Control Architecture

Susceptibility is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Susceptibility takes information about the offensive capabilities available to an Aggressor to determine if, for a given Engagement, a Subject's vulnerabilities are open to being exploited. A susceptibility assessment may be performed with ownship as the Aggressor or the Subject, or for two third parties.

Examples of Use

Susceptibility is required when a deployment needs to:

- Determine the best way of exploiting a Vulnerability in a hostile Subject (e.g. enemy aircraft or ground forces) with the available offensive capabilities.
- Determine whether a hostile Aggressor can exploit any vulnerabilities present in ownship or other friendly forces with their offensive capabilities.
- Determine if a subject may be vulnerable to other conditions, including weather or other phenomenon.

5.4.2.59.3 Service Summary

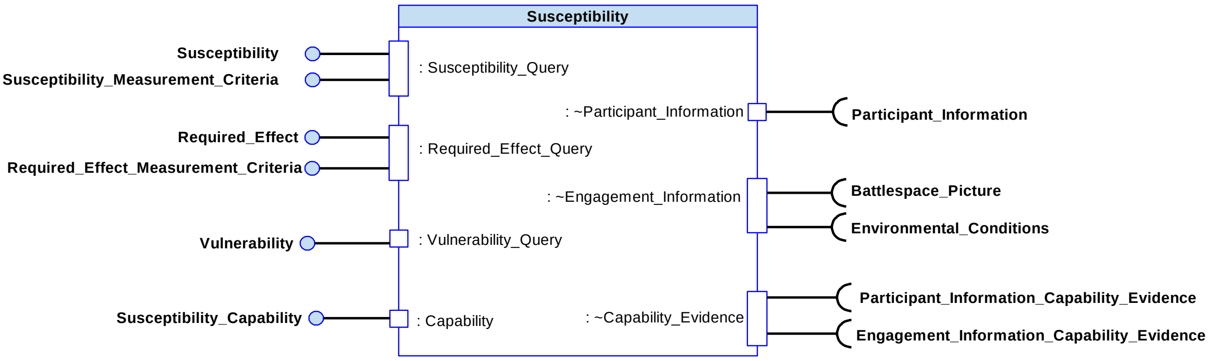

Susceptibility
Susceptibility
Susceptibility_Measurement_Criteria
: Susceptibility_Query
: ~Participant_Information
Participant_Information
Required_Effect
Required_Effect_Measurement_Criteria
: Required_Effect_Query
: ~Engagement_Information
Battlespace_Picture
Environmental_Conditions
Vulnerability
: Vulnerability_Query
Participant_Information_Capability_Evidence
Susceptibility_Capability
: Capability
: ~Capability_Evidence
Engagement_Information_Capability_Evidence

Figure 1014: Susceptibility Service Summary

5.4.2.59.4 Responsibilities

capture_susceptibility_requirements

- To capture requirements for a Susceptibility assessment (i.e. predicted harm and/or required effect) for a given Engagement.

capture_susceptibility_measurement_criteria

- To capture provided Measurement_Criterion for a Susceptibility assessment.

determine_susceptibility

- To determine the Susceptibility of a Subject to the Offensive_Capability of an Aggressor.

identify_subject_vulnerability

- To identify which Subject vulnerabilities an Aggressor could exploit.

determine_required_effect_level

- To determine what level of Offensive_Capability an Aggressor must use to exploit a Subject's Vulnerability through a given medium.

determine_susceptibility_quality

- To determine the quality of the Susceptibility assessment against given Measurement_Criterion/criteria.

assess_susceptibility_capability

- To assess the Capability to determine Susceptibility taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Susceptibility Capability assessment.

predict_capability_progression

- To predict the progression of the Susceptibility Capability over time and with use.

5.4.2.59.5 Subject Matter Semantics

The subject matter of Susceptibility is the sensitivity of a Subject to the Offensive_Capability of an Aggressor. This can include the harm that may be suffered by the Subject, or the level of Offensive_Capability that needs to be applied to exploit a Vulnerability.

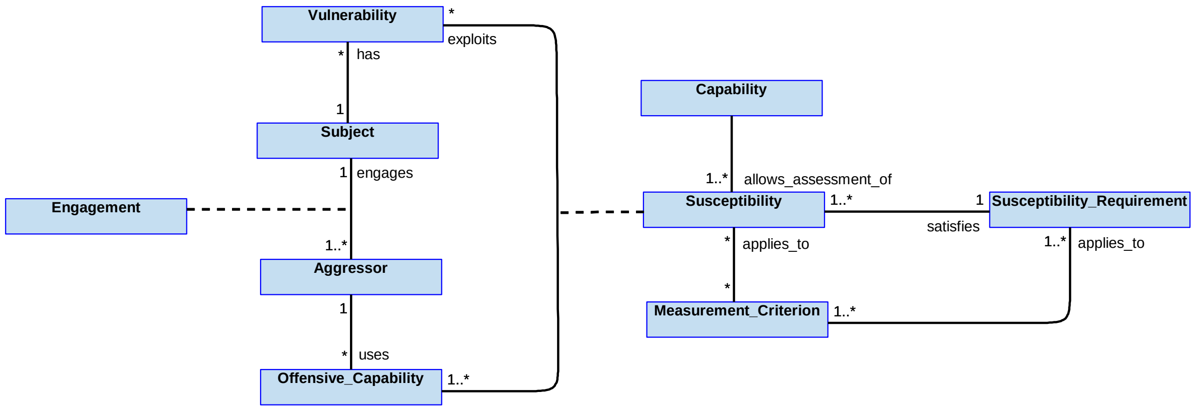


Figure 1015: Susceptibility Semantics

5.4.2.59.5.1 Entities

Aggressor

The offensive party to an Engagement that may cause harm.

Engagement

A specific scenario between an Aggressor and a Subject.

Measurement_Criterion

A criterion by which the quality of the Susceptibility prediction will be measured.

Offensive_Capability

Something that can be used by an Aggressor to exploit a Vulnerability (e.g. long range fuel tanks, long range missile or the potency of a storm).

Susceptibility

A measure of how susceptible a Subject's vulnerabilities are to the offensive capabilities of an Aggressor (e.g. the likelihood that using ASRAAM against a hostile aircraft will cause significant harm).

Subject

The defensive party to an Engagement that may be harmed.

Vulnerability

A weakness of the Subject that may be exploited, e.g. short range fuel tanks.

Capability

The capability to determine the Susceptibility of the Subject.

Susceptibility_Requirement

A requirement to perform a Susceptibility assessment for a given Engagement.

5.4.2.59.6 Design Rationale

5.4.2.59.6.1 Assumptions

- Weather conditions can be considered to be Aggressors to the Subject, as well as more typical threats such as hostile vehicles.

5.4.2.59.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Susceptibility:

- Data Driving - To mitigate the significant variances in the types of Aggressors and their Offensive_Capability, and of Subjects and their vulnerabilities (e.g. types of missiles on an aircraft, or the effects of cumulonimbus clouds).

Extensions

- The Susceptibility component primarily calculates the susceptibility of a Subject to an Aggressor's Offensive_Capability based on the known vulnerabilities of the Subject. The component could use multiple extension components to handle different offensive and defensive profiles or types of vulnerabilities, together with any associated algorithms.

Exploitation Considerations

- This component could be configured to consider the Exploiting Platform to be either the Aggressor or the Subject.
- It may be appropriate for an exploitation to include multiple instances of the Susceptibility component dealing with different types of Engagement (e.g. air-to-air, air-to-ground, or weather effects).
- The assessment of susceptibility could be via a passive implementation that simply provides statically determined lookups, or could have dynamic behaviour based on evolving capabilities of Aggressors and Subjects with algorithmic determination of susceptibility.

5.4.2.59.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- As shown on the Weather interaction view this component determines the susceptibility of the Exploiting Platform to weather and so, in the case of an air vehicle, failure of this component could result in flight in weather conditions that exceed the capability of the air vehicle. Flight in weather conditions that exceed the capability of the air vehicle could result in uncontrolled flight (e.g. if the air vehicle flies into a cumulonimbus cloud) and an uncontrolled crash. This would result in loss of the air vehicle and potentially fatalities.
- No credit has been assumed for the crew controlling the Exploiting Platform directly observing the local weather or its effect on the Exploiting Platform. For Exploiting Programmes where this is possible DAL requirements may be less onerous.

Note: Where instances of this component are used solely to support survival against external physical threats from enemy forces (e.g. missile attack) the DAL requirements are expected to be less onerous as this is normally excluded from safety analysis.

5.4.2.59.6.4 Security Considerations

The indicative security classification is SNEO.

This component determines whether Subjects are susceptible to harm or loss based on available data for the offensive and defensive capabilities of the parties involved. The intelligence data and algorithms for determining susceptibility are likely to be SNEO; in some cases data may possibly be TS. If this is the case, there may be instances in different security domains; these instances may need to communicate with each other to provide a full susceptibility assessment. If so, separation will be handled externally to the component. Any loss of integrity or availability of this component may lead to the Exploiting Platform placing itself in a situation where it may be unknowingly susceptible to harm or inappropriately engage a target (e.g. with a weapon that may cause too great or too little an effect). The confidentiality, integrity and availability requirements of the Exploiting Platform will need to reflect this. Where algorithms are data-driven, the associated configuration data will also carry appropriate confidentiality requirements.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of susceptibility assessments made during the course of a mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.59.7 Services

5.4.2.59.7.1 Service Definitions

5.4.2.59.7.1.1 Susceptibility_Query

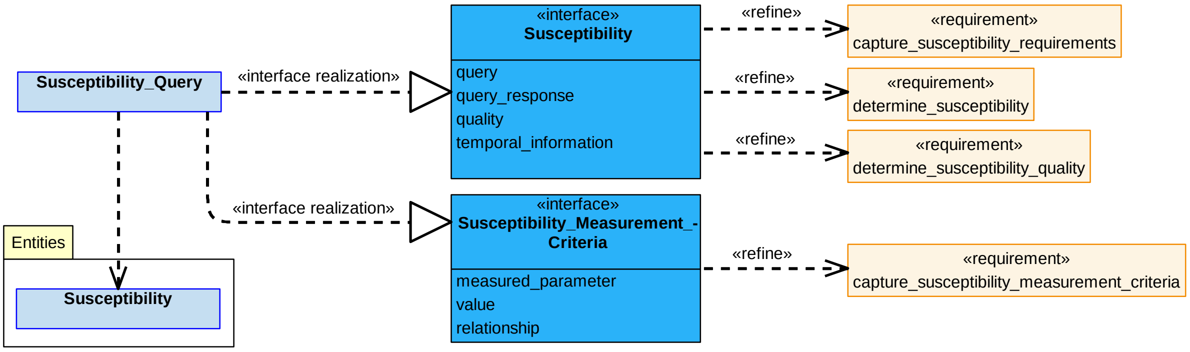


Figure 1016: Susceptibility_Query Service Definition

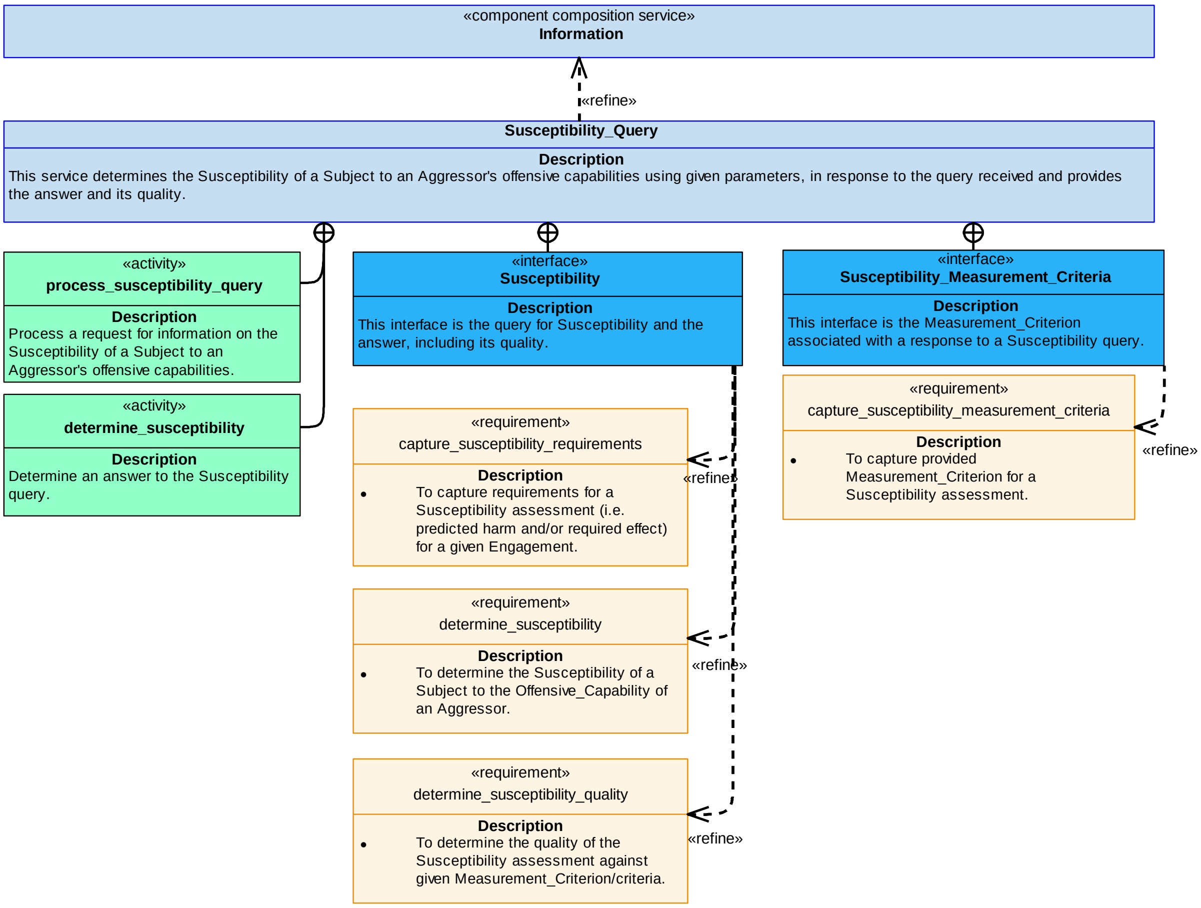

«component composition service»
Information
«refine»
Susceptibility_Query
Description
This service determines the Susceptibility of a Subject to an Aggressor's offensive capabilities using given parameters, in response to the query received and provides the answer and its quality.
«activity»
process_susceptibility_query
Description
Process a request for information on the Susceptibility of a Subject to an Aggressor's offensive capabilities.
«activity»
determine_susceptibility
Description
Determine an answer to the Susceptibility query.
«interface»
Susceptibility
Description
This interface is the query for Susceptibility and the answer, including its quality.
«interface»
Susceptibility_Measurement_Criteria
Description
This interface is the Measurement_Criterion associated with a response to a Susceptibility query.
«requirement»
capture_susceptibility_requirements
Description
To capture requirements for a Susceptibility assessment (i.e. predicted harm and/or required effect) for a given Engagement.
«refine»
«requirement»
determine_susceptibility
Description
To determine the Susceptibility of a Subject to the Offensive_Capability of an Aggressor.
«refine»
«requirement»
determine_susceptibility_quality
Description
To determine the quality of the Susceptibility assessment against given Measurement_Criterion/criteria.
«refine»
«requirement»
capture_susceptibility_measurement_criteria
Description
To capture provided Measurement_Criterion for a Susceptibility assessment.
«refine»

Figure 1017: Susceptibility_Query Service Policy

Susceptibility_Query

This service determines the Susceptibility of a Subject to an Aggressor's offensive capabilities using given parameters, in response to the query received and provides the answer and its quality.

Interfaces

Susceptibility

This interface is the query for Susceptibility and the answer, including its quality.

Attributes

query	A query relating to the Susceptibility of a Subject to a known Aggressor, i.e. how susceptible is the Subject's vulnerabilities to exploitation by an Aggressor's offensive capabilities. For example, whether the potency of a storm can affect the performance of ownship? A susceptibility query can consider ownship as the Aggressor or the Subject, or for two third parties.
query_response	The response to the query, stating the Susceptibility of a Subject to an Aggressor's offensive capabilities for a given Engagement, e.g. an assessment of whether ownship can be influenced or harmed by the potency of the storm.
quality	The quality of a query response against defined Measurement_Criterion.

temporal_information	Information covering timing, such as start and end times and any points in time which define changes in parameters.

Susceptibility_Measurement_Criteria

This interface is the Measurement_Criterion associated with a response to a Susceptibility query.

Attributes

measured_parameter	The parameter the Measurement_Criterion is associated with.
value	An absolute value against which the measured_parameter is to be judged.
relationship	A relationship to a different value against which the measured_parameter is to be judged.

Activities

process_susceptibility_query

Process a request for information on the Susceptibility of a Subject to an Aggressor's offensive capabilities.

determine_susceptibility

Determine an answer to the Susceptibility query.

5.4.2.59.7.1.2 Required_Effect_Query

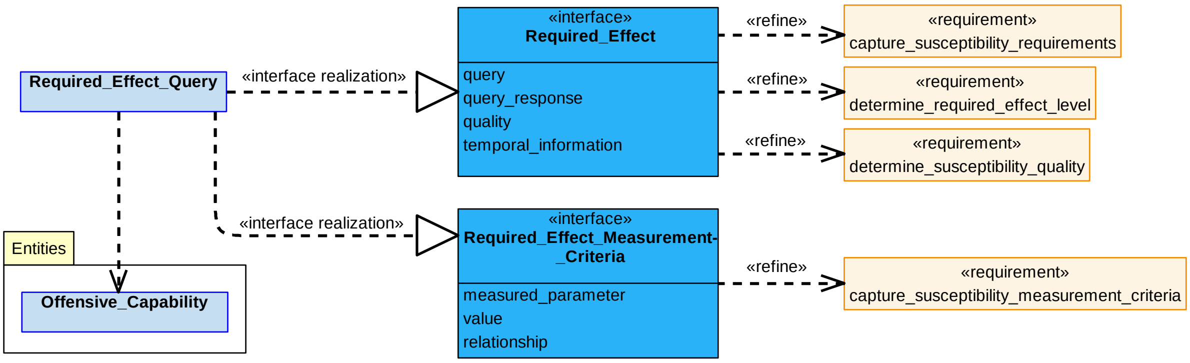


Figure 1018: Required_Effect_Query Service Definition

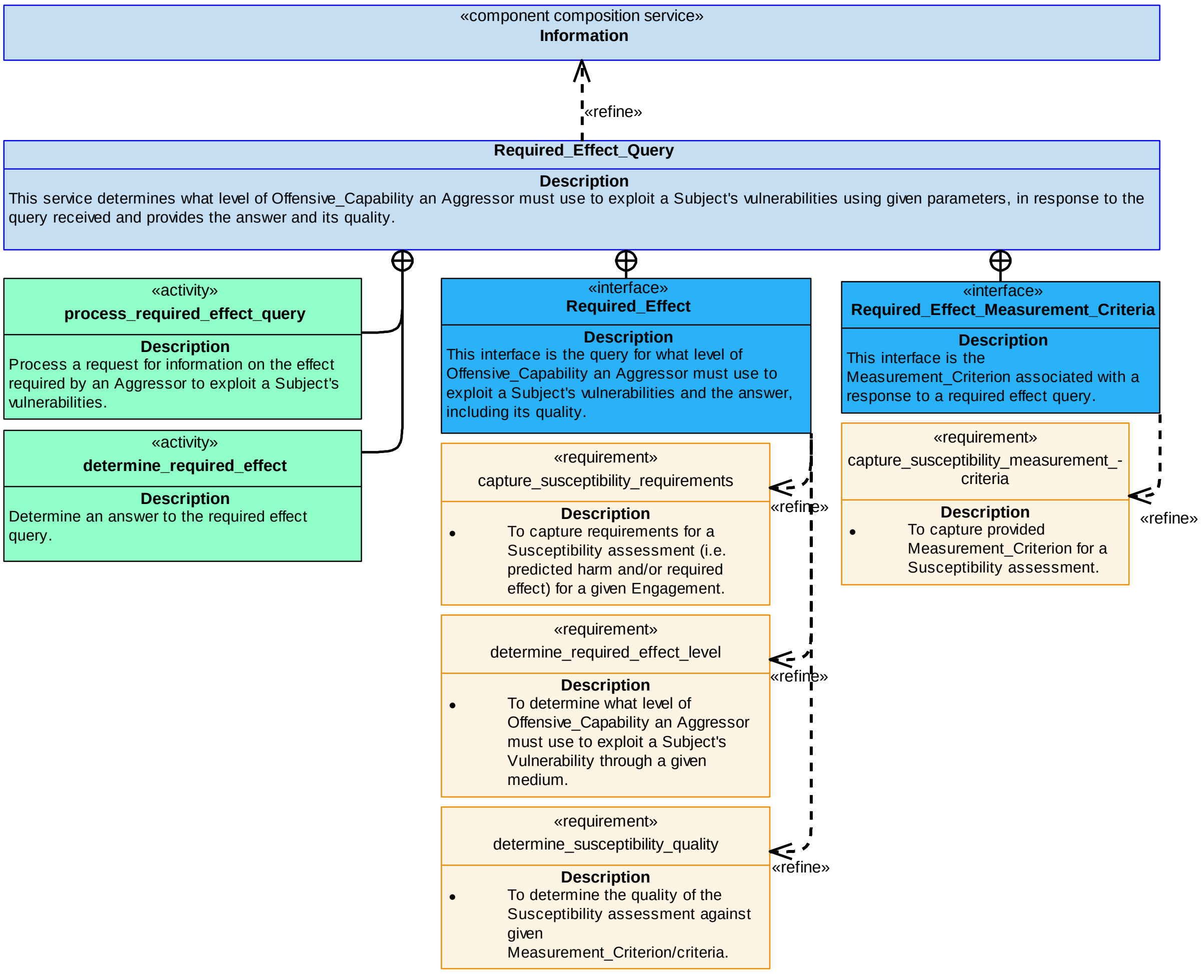

«component composition service»
Information
«refine»
Required_Effect_Query
Description
This service determines what level of Offensive_Capability an Aggressor must use to exploit a Subject's vulnerabilities using given parameters, in response to the query received and provides the answer and its quality.
«activity»
process_required_effect_query
Description
Process a request for information on the effect required by an Aggressor to exploit a Subject's vulnerabilities.
«activity»
determine_required_effect
Description
Determine an answer to the required effect query.
«interface»
Required_Effect
Description
This interface is the query for what level of Offensive_Capability an Aggressor must use to exploit a Subject's vulnerabilities and the answer, including its quality.
«interface»
Required_Effect_Measurement_Criteria
Description
This interface is the Measurement_Criterion associated with a response to a required effect query.
«requirement»
capture_susceptibility_requirements
Description
To capture requirements for a Susceptibility assessment (i.e. predicted harm and/or required effect) for a given Engagement.
«refine»
«requirement»
capture_susceptibility_measurement_-criteria
Description
To capture provided Measurement_Criterion for a Susceptibility assessment.
«refine»
«requirement»
determine_required_effect_level
Description
To determine what level of Offensive_Capability an Aggressor must use to exploit a Subject's Vulnerability through a given medium.
«refine»
«requirement»
determine_susceptibility_quality
Description
To determine the quality of the Susceptibility assessment against given Measurement_Criterion/criteria.
«refine»

Figure 1019: Required_Effect Service Policy

Required_Effect_Query

This service determines what level of Offensive_Capability an Aggressor must use to exploit a Subject's vulnerabilities using given parameters, in response to the query received and provides the answer and its quality.

Interfaces

Required_Effect

This interface is the query for what level of Offensive_Capability an Aggressor must use to exploit a Subject's vulnerabilities and the answer, including its quality.

Attributes

query	A query relating to the offensive capabilities of an Aggressor and what level of Offensive_Capability would be required to exploit a Subject's vulnerabilities. For example, the best way of exploiting a Vulnerability in a Subject with the available Offensive_Capability. A required effect query can consider ownship as the Aggressor or the Subject, or for two third parties.

query_response	The response to the query, stating the offensive capabilities required to exploit a Subject's vulnerabilities for a given Engagement.
quality	The quality of a query response against defined Measurement_Criterion.
temporal_information	Information covering timing, such as start and end times and any points in time which define changes in parameters.

Required_Effect_Measurement_Criteria

This interface is the Measurement_Criterion associated with a response to a required effect query.

Attributes

measured_parameter	The parameter the Measurement_Criterion is associated with.
value	An absolute value against which the measured_parameter is to be judged.
relationship	A relationship to a different value against which the measured_parameter is to be judged.

Activities

process_required_effect_query

Process a request for information on the effect required by an Aggressor to exploit a Subject's vulnerabilities.

determine_required_effect

Determine an answer to the required effect query.

5.4.2.59.7.1.3 Vulnerability_Query

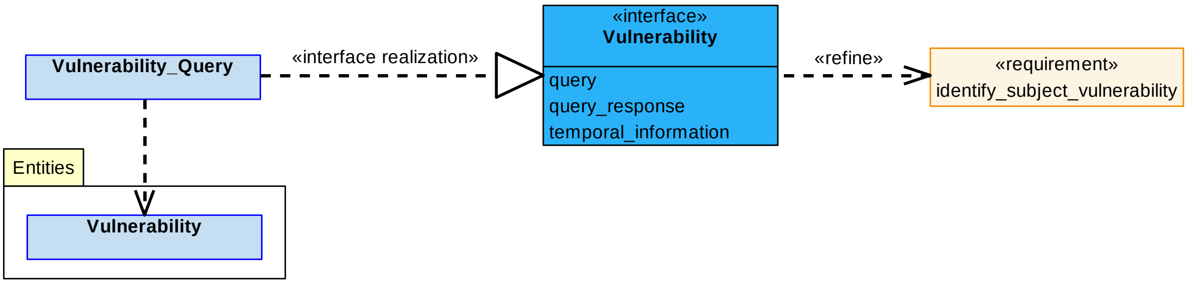


Figure 1020: Vulnerability_Query Service Definition

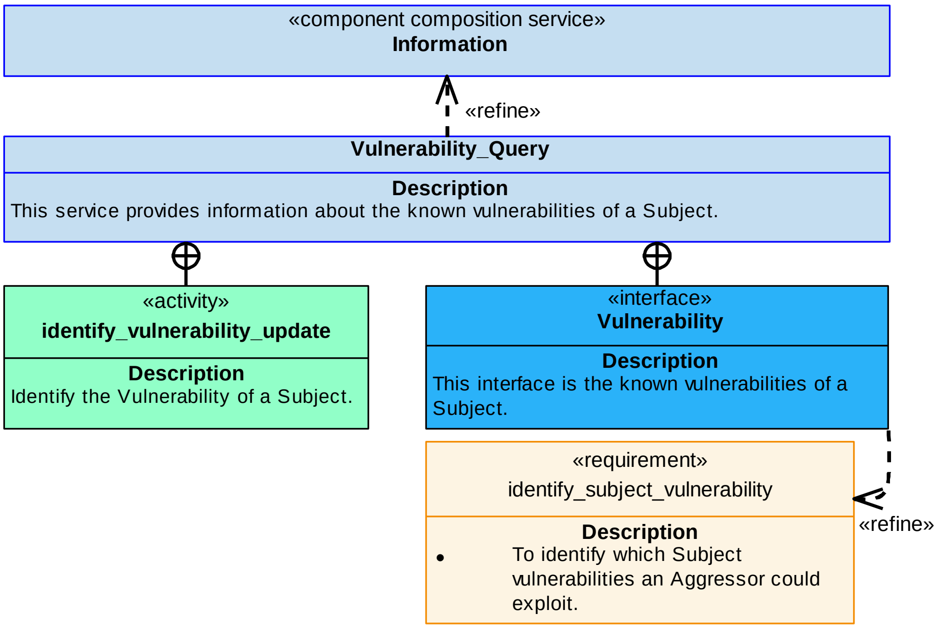

«component composition service»
Information
«refine»
Vulnerability_Query
Description
This service provides information about the known vulnerabilities of a Subject.
«activity»
identify_vulnerability_update
Description
Identify the Vulnerability of a Subject.
«interface»
Vulnerability
Description
This interface is the known vulnerabilities of a Subject.
«requirement»
identify_subject_vulnerability
Description
To identify which Subject vulnerabilities an Aggressor could exploit.
«refine»

Figure 1021: Vulnerability_Query Service Policy

Vulnerability_Query

This service provides information about the known vulnerabilities of a Subject.

Interface

Vulnerability

This interface is the known vulnerabilities of a Subject.

Attributes

query	A query relating to the known vulnerabilities of a Subject.
query_response	The response to the query, stating the vulnerabilities of a Subject, e.g. ownship is vulnerable in cold weather.
temporal_information	Information covering timing, such as start and end times and any points in time which define changes in parameters.

Activity

identify_vulnerability_update

Identify the Vulnerability of a Subject.

5.4.2.59.7.1.4 Participant_Information

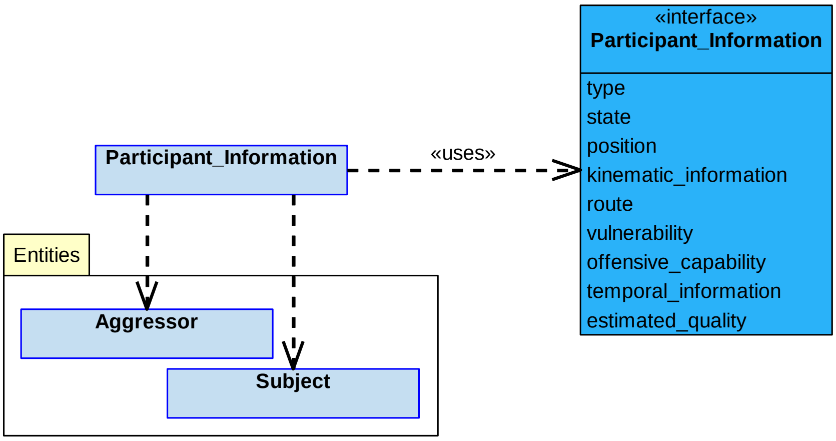


Figure 1022: Participant_Information Service Definition

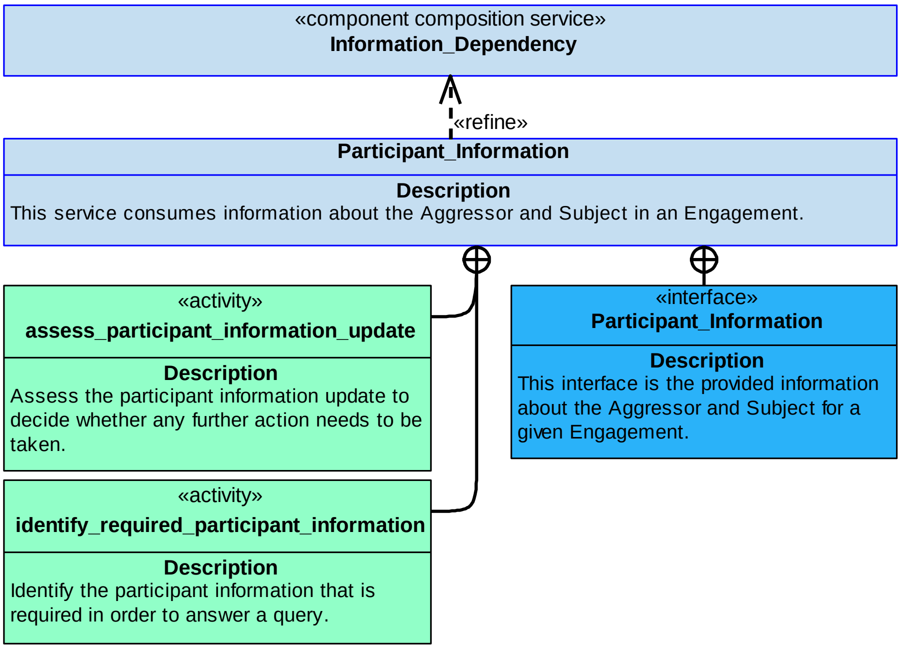

«component composition service»
Information_Dependency
«refine»
Participant_Information
Description
This service consumes information about the Aggressor and Subject in an Engagement.
«activity»
assess_participant_information_update
Description
Assess the participant information update to decide whether any further action needs to be taken.
«interface»
Participant_Information
Description
This interface is the provided information about the Aggressor and Subject for a given Engagement.
«activity»
identify_required_participant_information
Description
Identify the participant information that is required in order to answer a query.

Figure 1023: Participant_Information Service Policy

Participant_Information

This service consumes information about the Aggressor and Subject in an Engagement.

Interface

Participant_Information

This interface is the provided information about the Aggressor and Subject for a given Engagement.

Attributes

type	The type of participant for a given Engagement under consideration, e.g. large turbojet aircraft.

state	The current configuration state and capability of the participant.
position	Where the participant is located, e.g. range/bearing.
kinematic_information	Information relating to the participant's motion which may include course, speed, accelerations (x/y/z), predicted trajectory, etc.
route	The path that the participant is following in the battlespace.
vulnerability	The vulnerabilities that could be exploited by the Aggressor, e.g. low fuel level.
offensive_capability	The characteristic that can be used by an Aggressor to exploit a Vulnerability, e.g. long range fuel tanks, long range missile or the potency of a storm.
temporal_information	Timing information, such as when the kinematic information was captured.
estimated_quality	The quality or confidence of the participant information.

Activities

assess_participant_information_update

Assess the participant information update to decide whether any further action needs to be taken.

identify_required_participant_information

Identify the participant information that is required in order to answer a query.

5.4.2.59.7.1.5 Engagement_Information

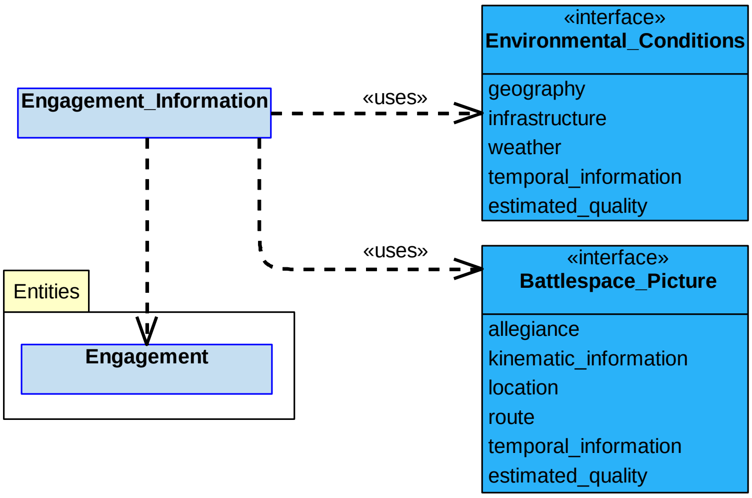
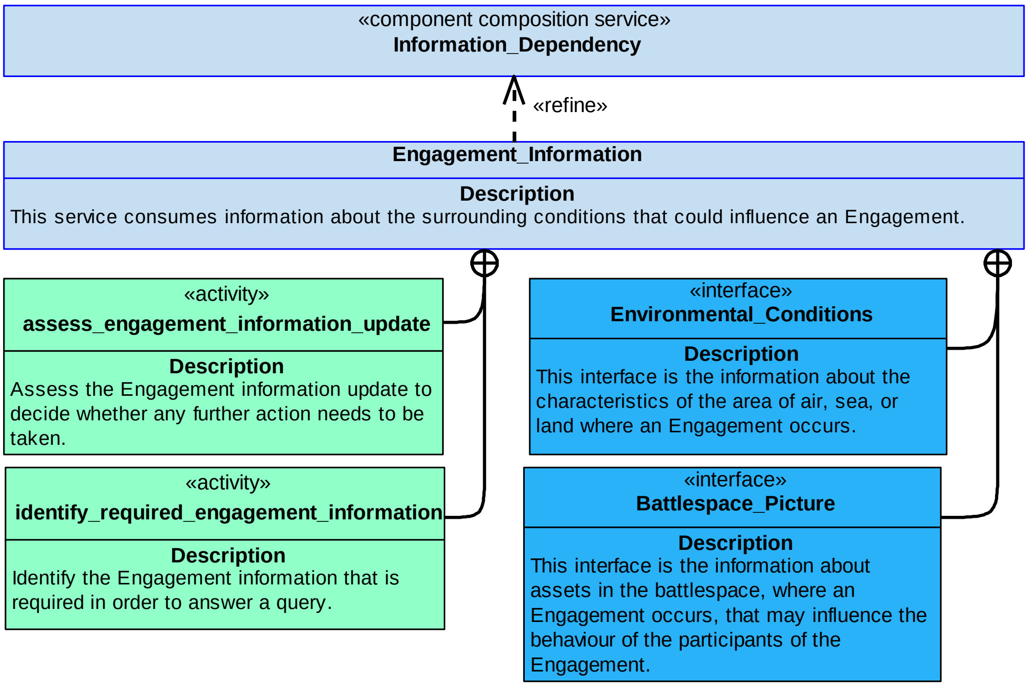


Figure 1024: Engagement_Information Service Definition



«component composition service»
Information_Dependency

«refine»

Engagement_Information

Description
This service consumes information about the surrounding conditions that could influence an Engagement.

«activity»
assess_engagement_information_update

Description
Assess the Engagement information update to decide whether any further action needs to be taken.

«activity»
identify_required_engagement_information

Description
Identify the Engagement information that is required in order to answer a query.

«interface»
Environmental_Conditions

Description
This interface is the information about the characteristics of the area of air, sea, or land where an Engagement occurs.

«interface»
Battlespace_Picture

Description
This interface is the information about assets in the battlespace, where an Engagement occurs, that may influence the behaviour of the participants of the Engagement.

Figure 1025: Engagement_Information Service Policy

Engagement_Information

This service consumes information about the surrounding conditions that could influence an Engagement.

Interfaces

Environmental_Conditions

This interface is the information about the characteristics of the area of air, sea, or land where an Engagement occurs.

Attributes

geography	The geographical features of the area in which an Engagement occurs, e.g. a land border between two countries.
infrastructure	A feature in the operating environment, e.g. a specific airbase.
weather	The meteorological conditions in the area in which an Engagement occurs, e.g. raining with a strong northerly wind.
temporal_information	Information covering timing, such as start and end times and any points in time which define changes in parameters.
estimated_quality	The quality or confidence of information about the environmental conditions.

Battlespace_Picture

This interface is the information about assets in the battlespace, where an Engagement occurs, that may influence the behaviour of the participants of the Engagement.

Attributes

allegiance	Whether an asset in the battlespace is considered a friend, foe or neutral party to the Aggressor or Subject.
kinematic_information	A set of information relating to the motion of assets in the battlespace which may include course, speed, accelerations (x/y/z), predicted trajectory, etc.
location	Where an asset is located in the battlespace, e.g. latitude, longitude, altitude.
route	The path that an asset is following in the battlespace.
temporal_information	Information covering timing, such as start and end times and any points in time which define changes in parameters.
estimated_quality	The quality or confidence of information about the battlespace picture.

Activities

assess_engagement_information_update

Assess the Engagement information update to decide whether any further action needs to be taken.

identify_required_engagement_information

Identify the Engagement information that is required in order to answer a query.

5.4.2.59.7.1.6 Capability

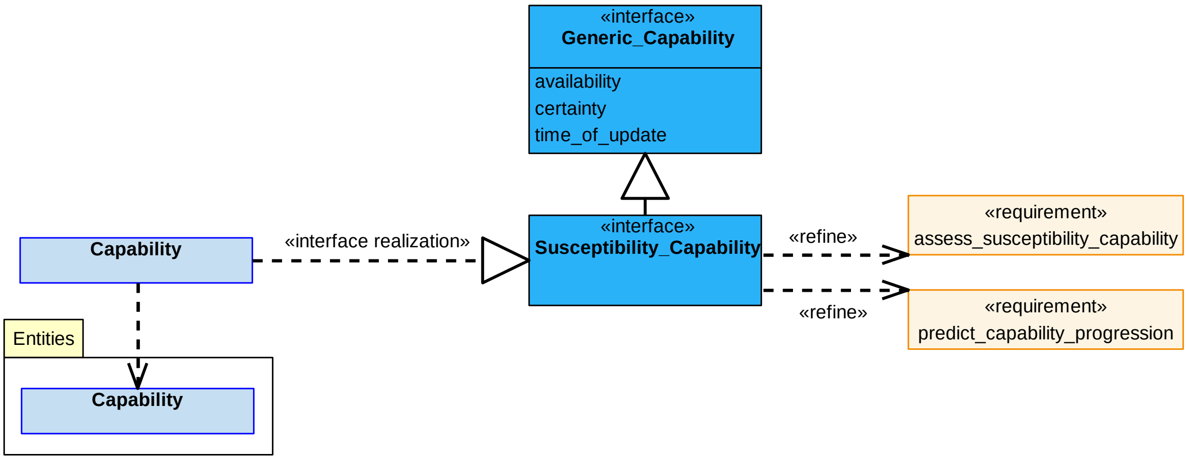


Figure 1026: Capability Service Definition

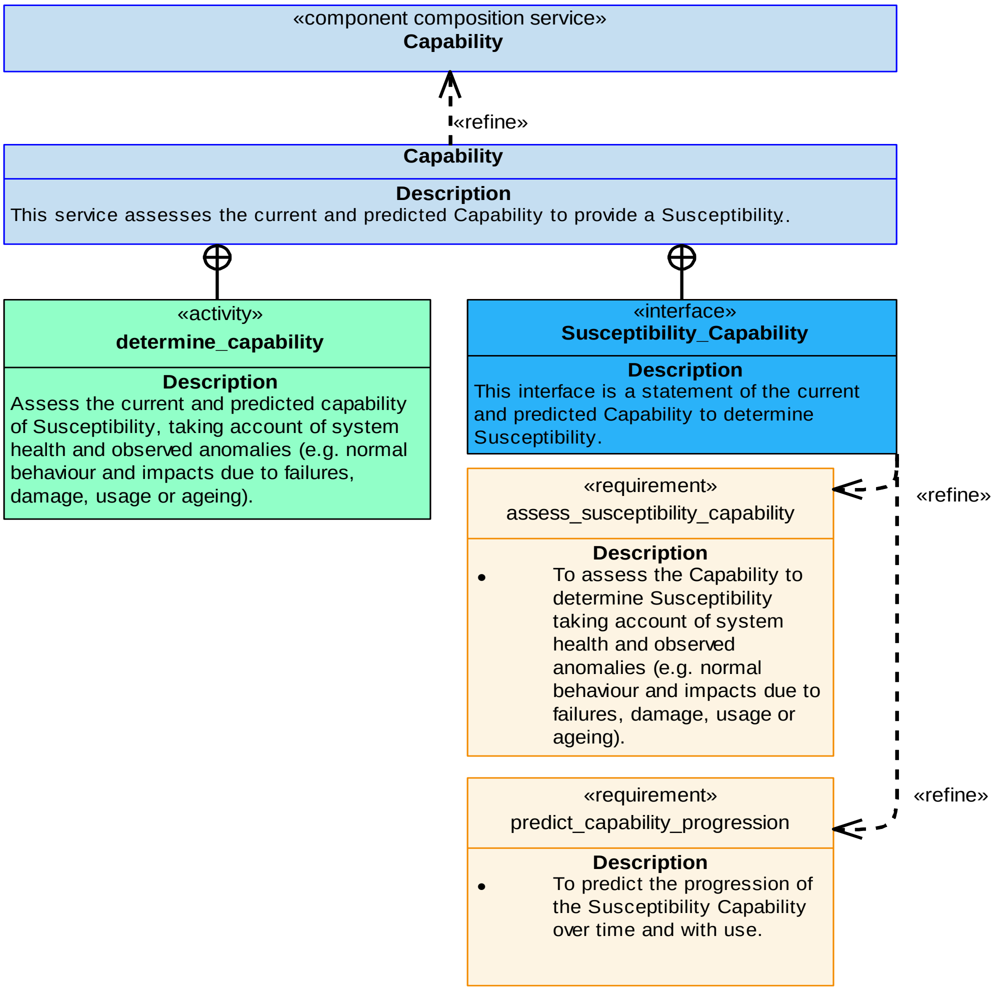

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to provide a Susceptibility.
«activity»
determine_capability
Description
Assess the current and predicted capability of Susceptibility, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Susceptibility_Capability
Description
This interface is a statement of the current and predicted Capability to determine Susceptibility.
«requirement»
assess_susceptibility_capability
Description
To assess the Capability to determine Susceptibility taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the Susceptibility Capability over time and with use.
«refine»

Figure 1027: Capability Service Policy

Capability

This service assesses the current and predicted Capability to provide a Susceptibility assessment.

Interface

Susceptibility_Capability

This interface is a statement of the current and predicted Capability to determine Susceptibility.

Activity

determine_capability

Assess the current and predicted capability of Susceptibility, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.59.7.1.7 Capability_Evidence

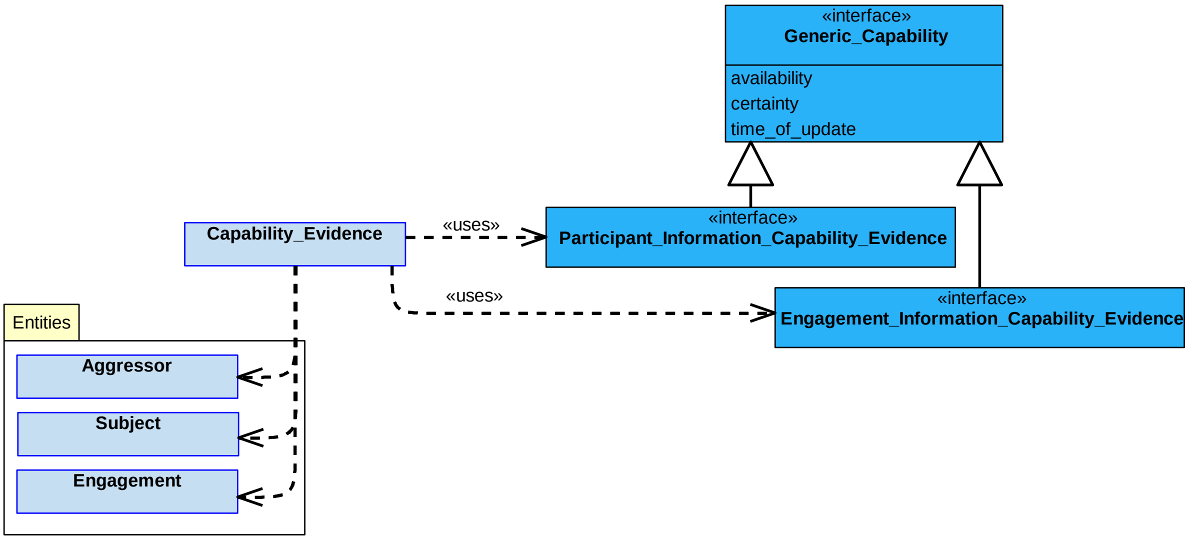


Figure 1028: Capability_Evidence Service Definition

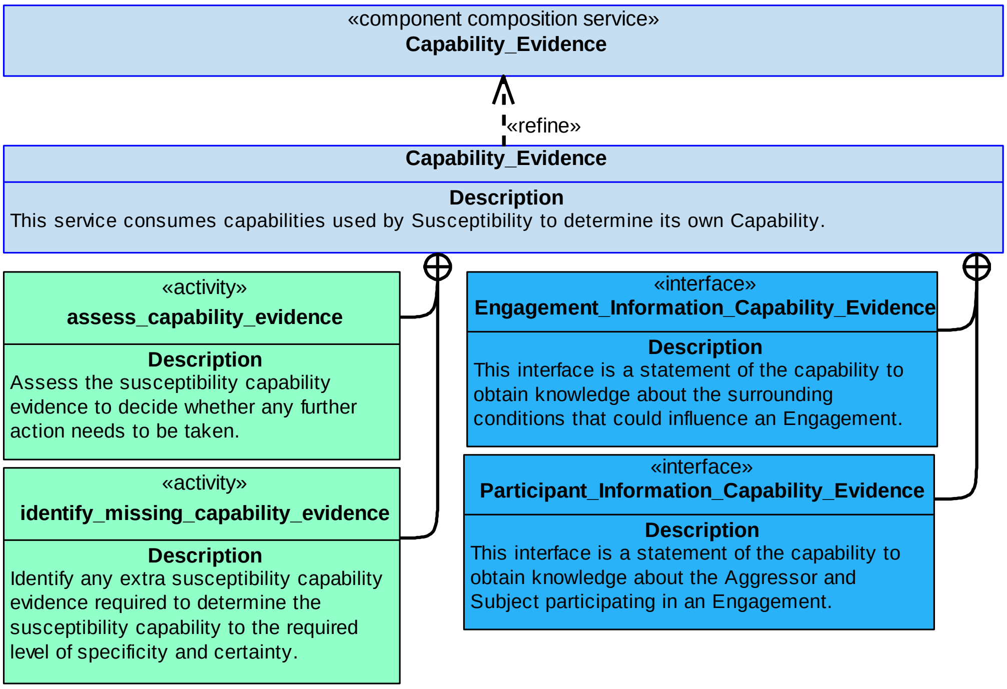

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capabilities used by Susceptibility to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the susceptibility capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra susceptibility capability evidence required to determine the susceptibility capability to the required level of specificity and certainty.
«interface»
Engagement_Information_Capability_Evidence
Description
This interface is a statement of the capability to obtain knowledge about the surrounding conditions that could influence an Engagement.
«interface»
Participant_Information_Capability_Evidence
Description
This interface is a statement of the capability to obtain knowledge about the Aggressor and Subject participating in an Engagement.

Figure 1029: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capabilities used by Susceptibility to determine its own Capability.

Interfaces

Participant_Information_Capability_Evidence

This interface is a statement of the capability to obtain knowledge about the Aggressor and Subject participating in an Engagement.

Engagement_Information_Capability_Evidence

This interface is a statement of the capability to obtain knowledge about the surrounding conditions that could influence an Engagement.

Activities

assess_capability_evidence

Assess the susceptibility capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra susceptibility capability evidence required to determine the susceptibility capability to the required level of specificity and certainty.

5.4.2.59.7.2 Service Dependencies

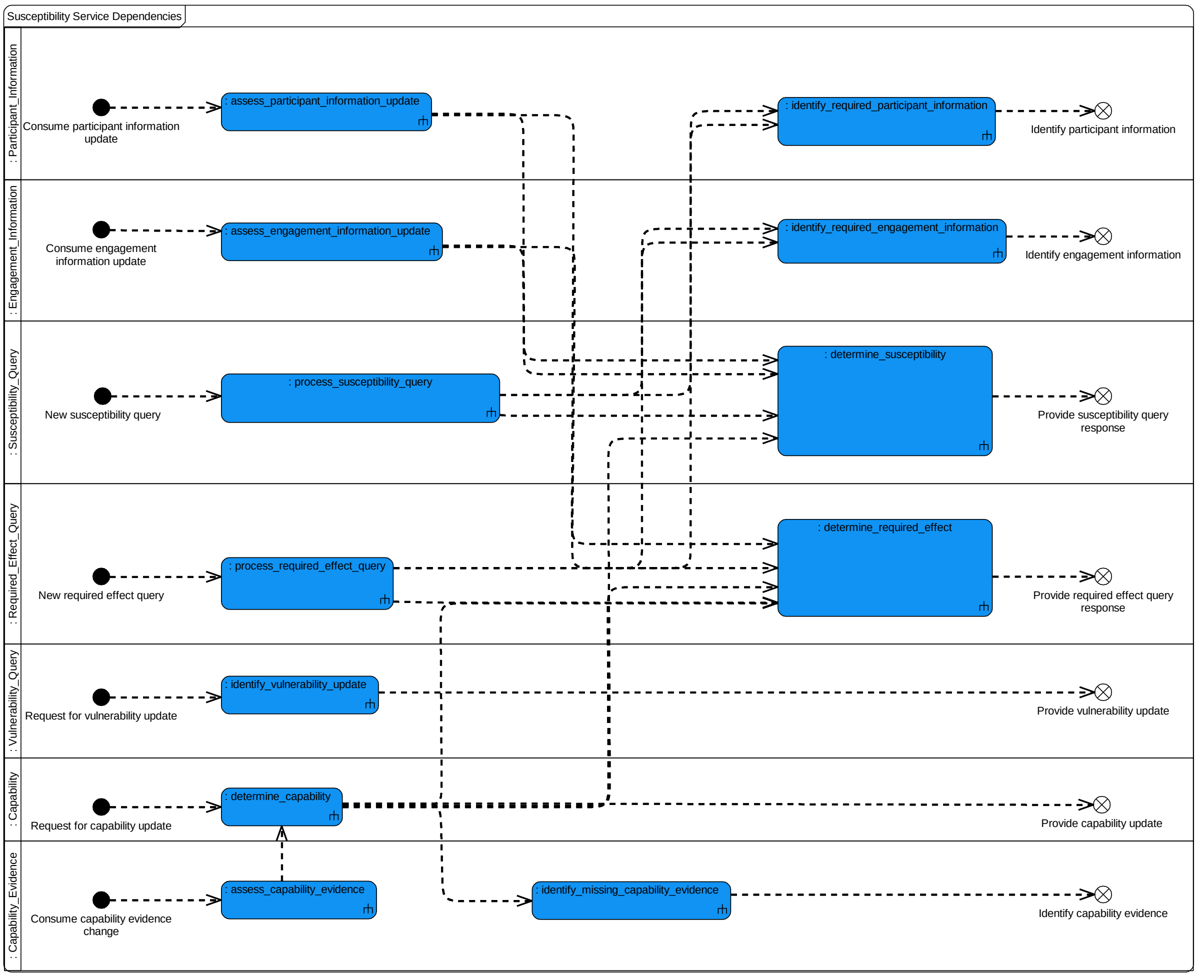

Susceptibility Service Dependencies
: Participant_Information
: Engagement_Information
: Susceptibility_Query
: Required_Effect_Query
: Vulnerability_Query
: Capability
: Capability_Evidence
Consume participant information update
: assess_participant_information_update
: identify_required_participant_information
Identify participant information
Consume engagement information update
: assess_engagement_information_update
: identify_required_engagement_information
Identify engagement information
New susceptibility query
: process_susceptibility_query
: determine_susceptibility
Provide susceptibility query response
New required effect query
: process_required_effect_query
: determine_required_effect
Provide required effect query response
Request for vulnerability update
: identify_vulnerability_update
Provide vulnerability update
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1030: Susceptibility Service Dependencies

5.4.2.60 Tactical Objects

5.4.2.60.1 Role

The role of Tactical Objects is to maintain knowledge of objects which exist, or are assumed to exist, within the battlespace and their relationship to each other.

5.4.2.60.2 Overview

Control Architecture

Tactical Objects is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

The Tactical Objects component is requested to provide and maintain information about tactical objects, including their behaviours, allegiances and associations with other tactical objects in the battlespace. In order to determine this information it obtains source information (such as fused track locations, velocities and object type indications) and requests the generation of this information if it is not available.

Examples of Use

Tactical Objects will be used in a system where it is necessary to:

- Establish the behaviours or states of a tactical object, such as whether it is attempting to track other vehicles.
- Establish the relationships between tactical objects, such as vehicles operating together, or weapons being targeted at another vehicle.
- Establish the identity of a tactical object based on evidence from its determined behaviours and relationships with other objects. This can include reinterpreting the known information about an object based on its behaviours and relationships.

This information can be used to contribute to developing a full ‘picture’ of the battlespace.

5.4.2.60.3 Service Summary

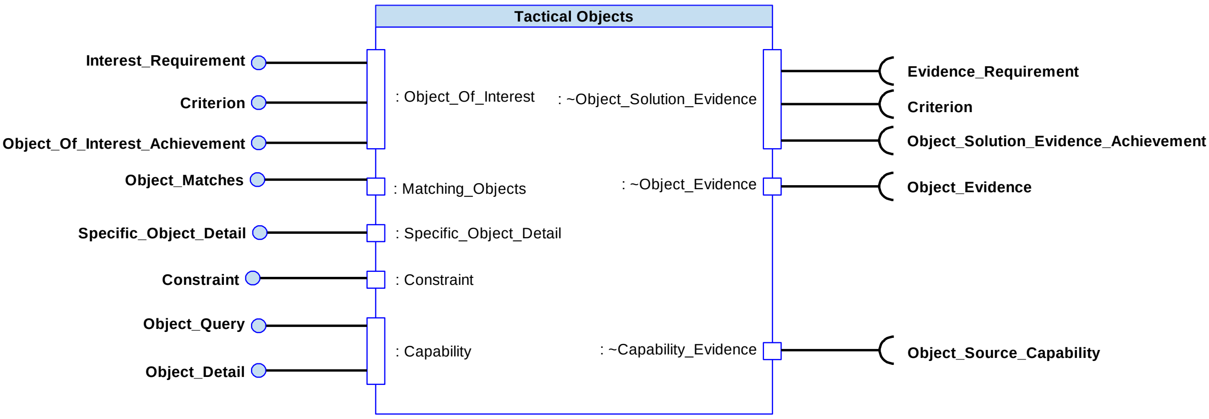


Figure 1031: Tactical Objects Service Summary

5.4.2.60.4 Responsibilities

capture_object_interest_requirement_for_tactical_object

- To capture provided Object_Interest_Requirements (e.g. the scope of the information required and the frequency that it is reported) for Tactical_Objects.

capture_measurement_criteria_for_tactical_object

- To capture provided Measurement_Criterion/criteria for Tactical_Objects.

capture_tactical_objects_constraints

- To capture provided Constraints for Tactical_Objects solutions (e.g. do not process classified information or stop processing information derived from a specified source).

identify_whether_requirement_is_achievable

- To identify whether an Object_Interest_Requirement is still achievable given current or predicted Capability and Constraints.

determine_requirement_solution

- To determine a solution to Object_Interest_Requirements.

determine_potential_objects

- To determine the potential for Tactical_Objects to exist at locations.

determine_object_information_confidence

- To determine a level of confidence in the individual Tactical_Object's characteristics (e.g. behaviour, allegiance and association between tactical objects).

determine_additional_information

- To determine additional information required to satisfy Object_Interest_Requirements, e.g. improved Tactical_Object confidence required.

determine_object_relationships

- To determine the Relationships and dependencies between Tactical_Objects (e.g. if a radar is part of a specific vehicle, or if a specific vehicle is part of this formation and is the flight lead).

estimate_object_behaviour

- To estimate the Behaviour exhibited by Tactical Objects, including individual Behaviour (e.g. that object is loitering) and Behaviour between Tactical Objects (e.g. that Exploiting Platform is tracking that tank, that Exploiting Platform is landing at that airfield). Behaviour also includes the operational state of a Tactical_Object (e.g. ready for operation, operational on ground, operational in air, or non-operational).

identify_progress

- To identify the progress against an Object_Interest_Requirement.

determine_quality_of_tactical_object_of_interest

- To determine the quality of the Tactical_Object_of_Interest provided by Tactical Objects during execution, measured against given Object_Interest_Requirements and Measurement_Criterion/criteria.

capture_object_information

- To capture information about Tactical_Objects, including but not limited to: their allegiance to organisations, kinematic behaviour (e.g. velocity, acceleration or path), location (e.g. at this position, within this region) and classification, including their type (e.g. surface ship or emitter), specific type (e.g. Type-45, Captor radar) and registration (e.g. HMS Daring).

capture_object_probability_densities

- To capture the probability densities of particular Tactical_Objects within locations.

assess_capability

- To assess the Capability to provide Tactical_Object information (e.g. determination and estimation of object characteristics) taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the component's Capability (i.e. determination and estimation of object characteristics) assessment.

predict_capability_progression

- To predict the progression of the Tactical Objects Capability over time and with use.

5.4.2.60.5 Subject Matter Semantics

The subject matter of Tactical Objects is objects of tactical importance which exist, or are assumed to exist within the battlespace, their Relationships between each other, and their Behaviour.

Exclusions

The subject matter of Tactical Objects does not include:

- The detectability of a Tactical_Object.
- The level of risk a Tactical_Object poses to the Exploiting Platform.
- Low level sensor data calculation.
- How the data from the Object_Information_Source is derived.

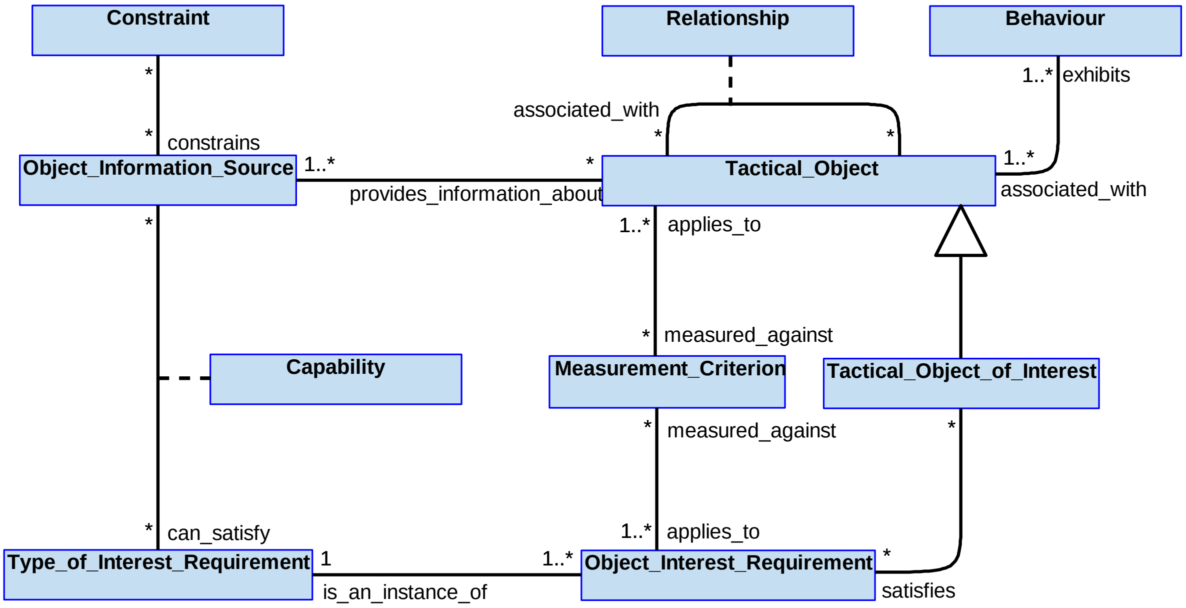


Figure 1032: Tactical Objects Semantics

5.4.2.60.5.1 Entities

Behaviour

The estimated pattern of action of a Tactical_Object (e.g. if the Tactical_Object is loitering) and operational state of a Tactical_Object (e.g. ready for operation, operational on ground, operational in air, non-operational).

Capability

The capability to perform a range of services, including maintaining knowledge of Tactical_Objects, their Behaviour and Relationships, through the available Object_Information_Sources to meet a Type_of_Interest_Requirement.

Constraint

An externally imposed limit on how or when information from various sources may be used.

Measurement_Criterion

A measure against which the predicted or actual quality of the Tactical_Object or Object_Interest_Requirement is assessed, e.g. the confidence in the location of an object.

Object_Interest_Requirement

A requirement to provide information, either continuously or intermittently, about Tactical_Objects.

Relationship

A relationship between different Tactical_Objects. This may be either a general association (e.g. a missile targeting an aircraft) or a hierarchical association (e.g. equipment on a vehicle, or vehicles within a formation).

Tactical_Object

An entity in the battlespace which has relevance to the system (including constituent vehicles, of which one would be ownship). Tactical objects can range from individual equipment (e.g. sensors and weapons) to organised groups of assets (e.g. a formation of aircraft or a platoon of soldiers). A tactical object has associated characteristics (e.g. allegiance, classification, location and kinematics).

Tactical_Object_of_Interest

The deliverable, a particular Tactical_Object which satisfies the Object_Interest_Requirement.

Type_of_Interest_Requirement

A specific type of Object_Interest_Requirement (e.g. a requirement to determine the type of Tactical_Object or any objects in a specific region of space).

Object_Information_Source

A source of information about objects of tactical significance.

5.4.2.60.6 Design Rationale

5.4.2.60.6.1 Assumptions

- The Tactical_Objects this component understands are not solely objects sensed by the Exploiting Platform; it will hold all Tactical_Objects needed for tactical purposes (from local and external sources), e.g. a building that is the target of an attack or a SAM site in a particular location which should be avoided.

5.4.2.60.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Tactical Objects*:*

- Data Driving - The possible Tactical_Object characteristic types (e.g. behaviours, allegiances, or classifications) may be specific to certain object types or the tactical understanding a system requires. While these may be catered for within later issues of the architecture, this component should be developed in a way which allows them to be data-driven as new attribute types.
- Tactical Information - This PYRAMID concept is applicable because Tactical Objects is classified as a tactical information component within this PYRAMID concept.

Extensions

- It is not expected that extension components will be needed.

Other Factors that were Taken into Account

- There is no separate provision for ownship as a tactical object within the PRA.

Exploitation Considerations

Tactical Objects components are expected to be deployed on multiple platforms. Synchronisation across a datalink between Tactical Objects instances is not covered in the PRA however this will need to be catered for in an exploitation. For example, although a service to accept association or behaviour data from another instance of Tactical Objects is not specified in the PRA an exploitation will likely require such a service.

5.4.2.60.6.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component could cause weapons to impact targets not intended by the crew (e.g. if the location of a Tactical_Object was corrupted), resulting in unintended harm to third parties. This drives a DAL B indicative IDAL.

5.4.2.60.6.4 Security Considerations

The indicative security classification is SNEO.

This component fundamentally handles data about Tactical_Objects (including own platform), their identification and their Behaviour, therefore its indicative security classification is SNEO. The confidentiality, integrity and availability of this component’s data and services are considered to be of high mission importance and will need appropriate protection.

The component is expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** relating to high-value data, including its classification, for later forensic examination.
- **Maintaining Audit Records** of shared tactical information for accountability purposes**.**
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.60.7 Services

5.4.2.60.7.1 Service Definitions

5.4.2.60.7.1.1 Object_Of_Interest

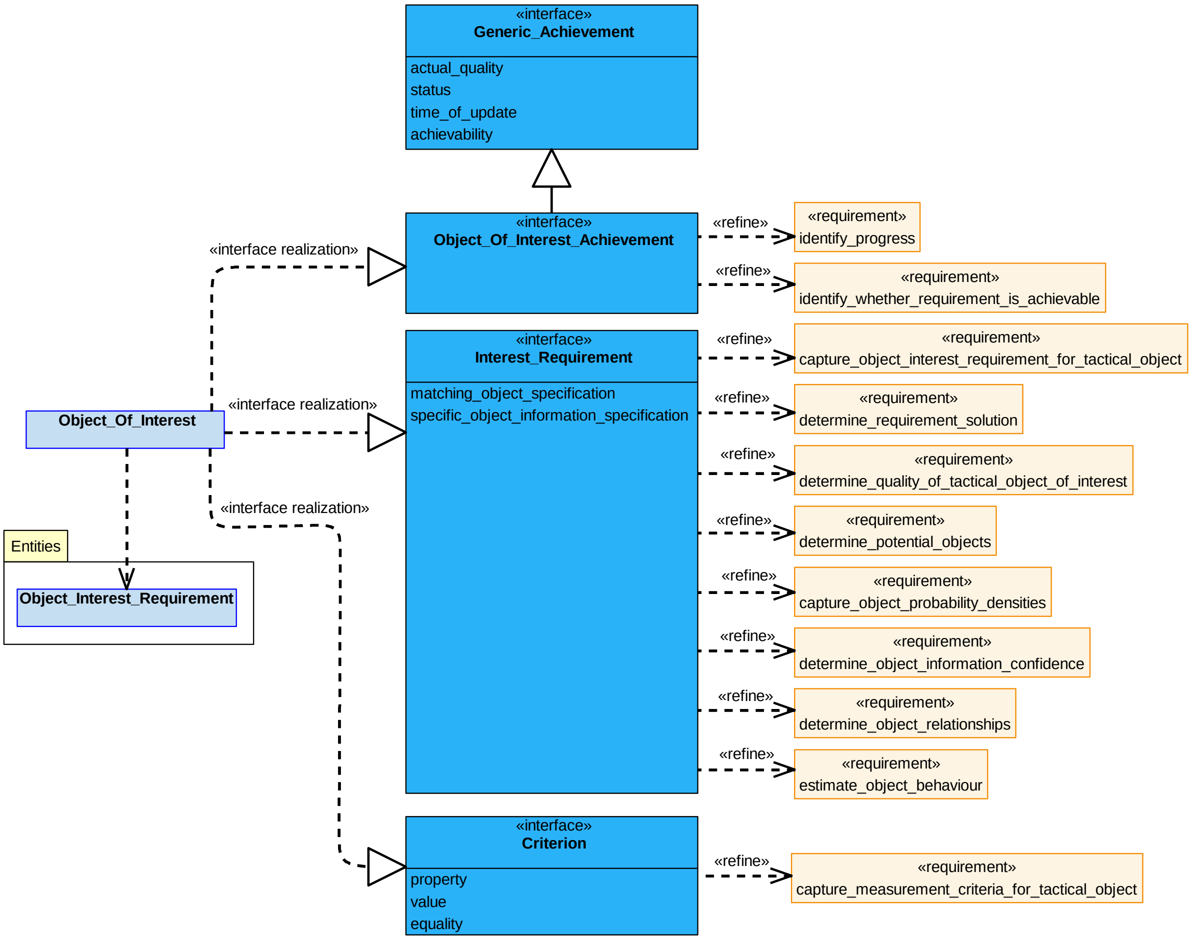

«interface»
Generic_Achievement
actual_quality
status
time_of_update
achievability
«interface»
Object_Of_Interest_Achievement
«interface realization»
«refine»
«requirement»
identify_progress
«refine»
«requirement»
identify_whether_requirement_is_achievable
«interface»
Interest_Requirement
matching_object_specification
specific_object_information_specification
«refine»
«requirement»
capture_object_interest_requirement_for_tactical_object
«refine»
«requirement»
determine_requirement_solution
Object_Of_Interest
«interface realization»
«refine»
«requirement»
determine_quality_of_tactical_object_of_interest
«interface realization»
«refine»
«requirement»
determine_potential_objects
Entities
Object_Interest_Requirement
«refine»
«requirement»
capture_object_probability_densities
«refine»
«requirement»
determine_object_information_confidence
«refine»
«requirement»
determine_object_relationships
«refine»
«requirement»
estimate_object_behaviour
«interface»
Criterion
property
value
equality
«refine»
«requirement»
capture_measurement_criteria_for_tactical_object

Figure 1033: Object_Of_Interest Service Definition

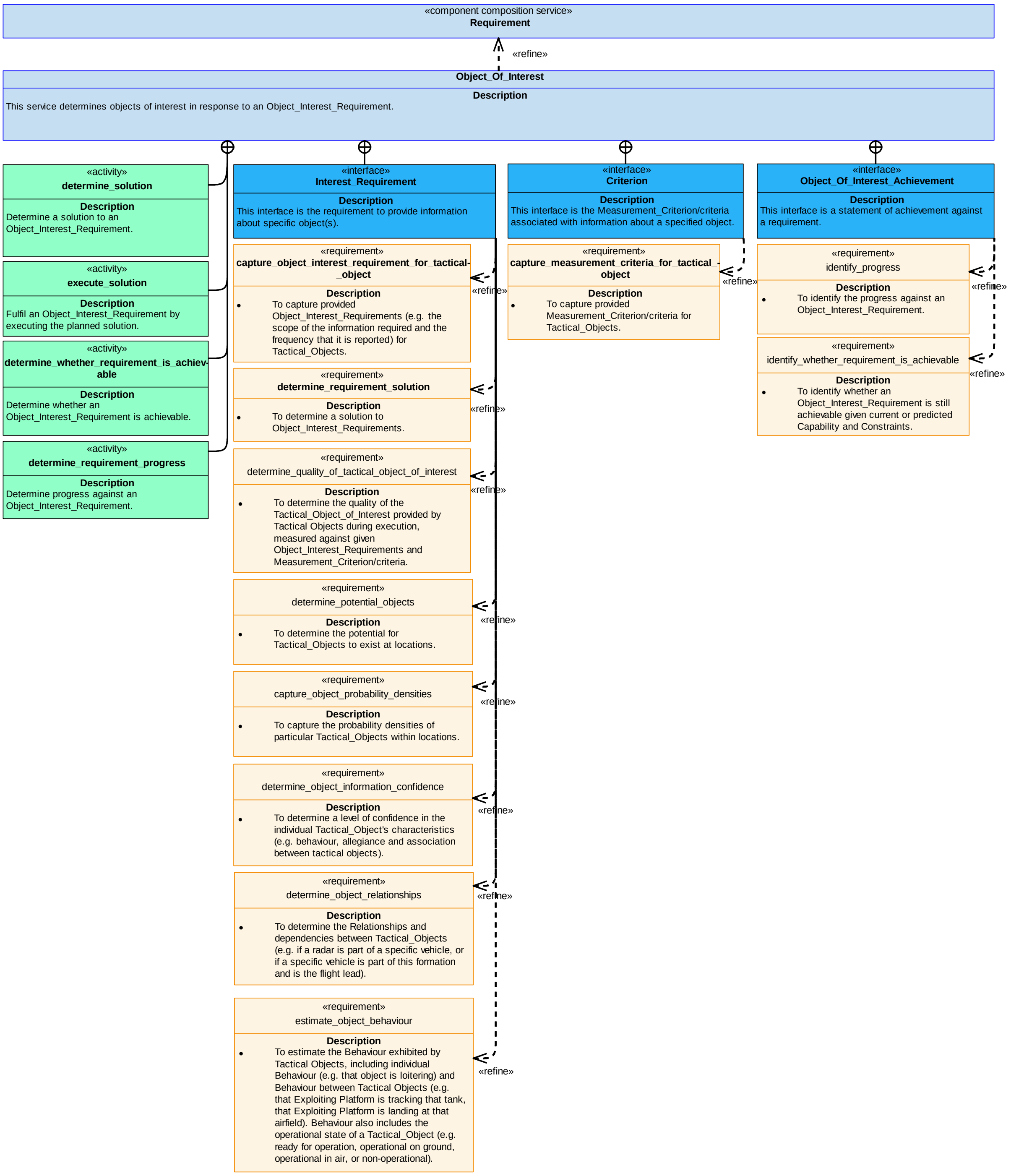

«component composition service»
Requirement
«refine»
Object_Of_Interest
Description
This service determines objects of interest in response to an Object_Interest_Requirement.
«activity»
determine_solution
Description
Determine a solution to an Object_Interest_Requirement.
«activity»
execute_solution
Description
Fulfil an Object_Interest_Requirement by executing the planned solution.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether an Object_Interest_Requirement is achievable.
«activity»
determine_requirement_progress
Description
Determine progress against an Object_Interest_Requirement.
«interface»
Interest_Requirement
Description
This interface is the requirement to provide information about specific object(s).
«interface»
Criterion
Description
This interface is the Measurement_Criterion/criteria associated with information about a specified object.
«interface»
Object_Of_Interest_Achievement
Description
This interface is a statement of achievement against a requirement.
«requirement»
capture_object_interest_requirement_for_tactical_object
Description
To capture provided Object_Interest_Requirements (e.g. the scope of the information required and the frequency that it is reported) for Tactical_Objects.
«refine»
«requirement»
determine_requirement_solution
Description
To determine a solution to Object_Interest_Requirements.
«refine»
«requirement»
determine_quality_of_tactical_object_of_interest
Description
To determine the quality of the Tactical_Object_of_Interest provided by Tactical Objects during execution, measured against given Object_Interest_Requirements and Measurement_Criterion/criteria.
«refine»
«requirement»
determine_potential_objects
Description
To determine the potential for Tactical_Objects to exist at locations.
«refine»
«requirement»
capture_object_probability_densities
Description
To capture the probability densities of particular Tactical_Objects within locations.
«refine»
«requirement»
determine_object_information_confidence
Description
To determine a level of confidence in the individual Tactical_Object's characteristics (e.g. behaviour, allegiance and association between tactical objects).
«refine»
«requirement»
determine_object_relationships
Description
To determine the Relationships and dependencies between Tactical_Objects (e.g. if a radar is part of a specific vehicle, or if a specific vehicle is part of this formation and is the flight lead).
«refine»
«requirement»
estimate_object_behaviour
Description
To estimate the Behaviour exhibited by Tactical Objects, including individual Behaviour (e.g. that object is loitering) and Behaviour between Tactical Objects (e.g. that Exploiting Platform is tracking that tank, that Exploiting Platform is landing at that airfield). Behaviour also includes the operational state of a Tactical_Object (e.g. ready for operation, operational on ground, operational in air, or non-operational).
«refine»
«requirement»
capture_measurement_criteria_for_tactical_object
Description
To capture provided Measurement_Criterion/criteria for Tactical_Objects.
«refine»
«requirement»
identify_progress
Description
To identify the progress against an Object_Interest_Requirement.
«refine»
«requirement»
identify_whether_requirement_is_achievable
Description
To identify whether an Object_Interest_Requirement is still achievable given current or predicted Capability and Constraints.
«refine»

Figure 1034: Object_Of_Interest Service Policy

Object_Of_Interest

This service determines objects of interest in response to an Object_Interest_Requirement.

<u>Interfaces</u>

Interest_Requirement

This interface is the requirement to provide information about specific object(s).

Attributes

matching_object_specification	The criteria against which Tactical_Objects should be assessed (e.g. to enable the reporting of all objects located within a specified volume at or above a known confidence level).
specific_object_information_specification	The definition of the type of information required about a Tactical_Object or Tactical_Objects (e.g. course, speed, altitude, associations, or behaviour).

Criterion

This interface is the Measurement_Criterion/criteria associated with information about a specified object.

Attributes

property	The property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Object_Of_Interest_Achievement

This interface is a statement of achievement against a requirement.

Activities

determine_solution

Determine a solution to an Object_Interest_Requirement.

execute_solution

Fulfil an Object_Interest_Requirement by executing the planned solution.

determine_whether_requirement_is_achievable

Determine whether an Object_Interest_Requirement is achievable.

determine_requirement_progress

Determine progress against an Object_Interest_Requirement.

5.4.2.60.7.1.2 Object_Solution_Evidence

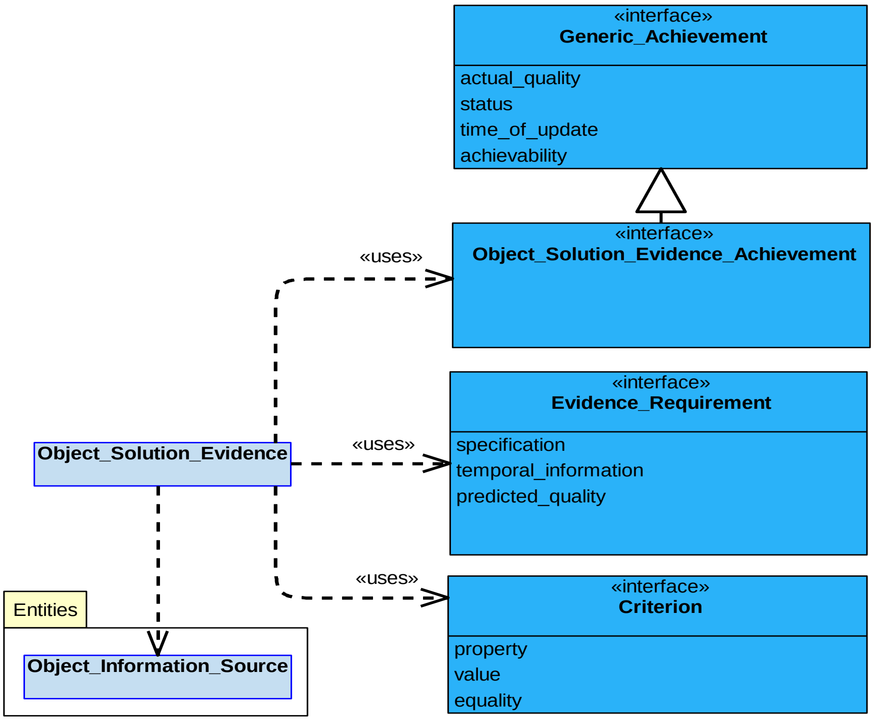
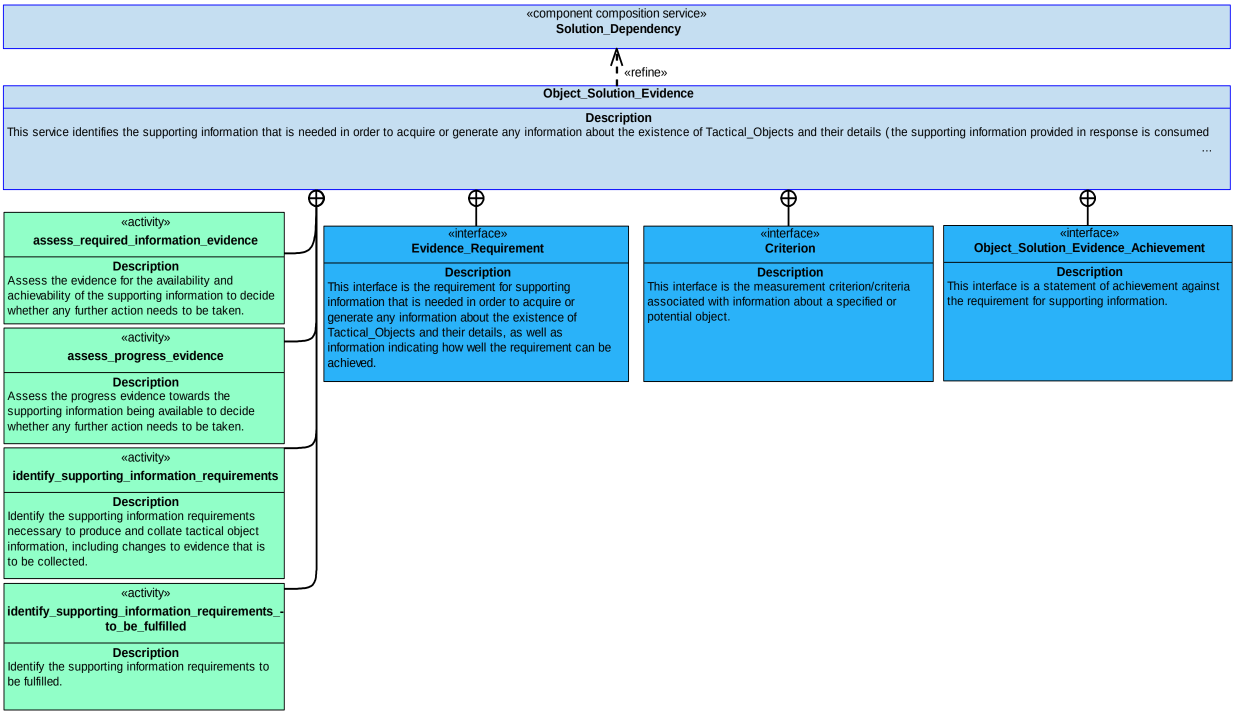


Figure 1035: Object_Solution_Evidence Service Definition


«component composition service»
Solution_Dependency
«refine»
Object_Solution_Evidence
«activity»
assess_required_information_evidence
«activity»
assess_progress_evidence
«activity»
identify_supporting_information_requirements
«activity»
identify_supporting_information_requirements_-to_be_fulfilled
«interface»
Evidence_Requirement
«interface»
Criterion
«interface»
Object_Solution_Evidence_Achievement

Object_Solution_Evidence

Description
This service identifies the supporting information that is needed in order to acquire or generate any information about the existence of Tactical_Objects and their details (the supporting information provided in response is consumed ...

Description
Assess the evidence for the availability and achievability of the supporting information to decide whether any further action needs to be taken.

Description
Assess the progress evidence towards the supporting information being available to decide whether any further action needs to be taken.

Description
Identify the supporting information requirements necessary to produce and collate tactical object information, including changes to evidence that is to be collected.

identify_supporting_information_requirements_-to_be_fulfilled

Description
Identify the supporting information requirements to be fulfilled.

Evidence_Requirement

Description
This interface is the requirement for supporting information that is needed in order to acquire or generate any information about the existence of Tactical_Objects and their details, as well as information indicating how well the requirement can be achieved.

Criterion

Description
This interface is the measurement criterion/criteria associated with information about a specified or potential object.

Object_Solution_Evidence_Achievement

Description
This interface is a statement of achievement against the requirement for supporting information.

Figure 1036: Object_Solution_Evidence Service Policy

Object_Solution_Evidence

This service identifies the supporting information that is needed in order to acquire or generate any information about the existence of Tactical_Objects and their details (the supporting information provided in response is consumed via the Object_Evidence service). It also consumes indications of whether the required information can be provided when required and the progress towards being able to provide it.

Interfaces

Evidence_Requirement

This interface is the requirement for supporting information that is needed in order to acquire or generate any information about the existence of Tactical_Objects and their details, as well as information indicating how well the requirement can be achieved.

Attributes

specification	The definition of the requirement for supporting information about objects, including the regions where greater confidence of the presence or absence of objects is needed, increased quality of information about objects and new information about objects.
temporal_information	The definition of when the object information is required.
predicted_quality	The predicted quality of the object information that can be generated in response to the requirement.

Criterion

This interface is the measurement criterion/criteria associated with information about a specified or potential object.

Attributes

property	The property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Object_Solution_Evidence_Achievement

This interface is a statement of achievement against the requirement for supporting information.

Activities

assess_required_information_evidence

Assess the evidence for the availability and achievability of the supporting information to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the progress evidence towards the supporting information being available to decide whether any further action needs to be taken.

identify_supporting_information_requirements_to_be_fulfilled

Identify the supporting information requirements to be fulfilled.

identify_supporting_information_requirements

Identify the supporting information requirements necessary to produce and collate tactical object information, including changes to evidence that is to be collected.

5.4.2.60.7.1.3 Matching_Objects

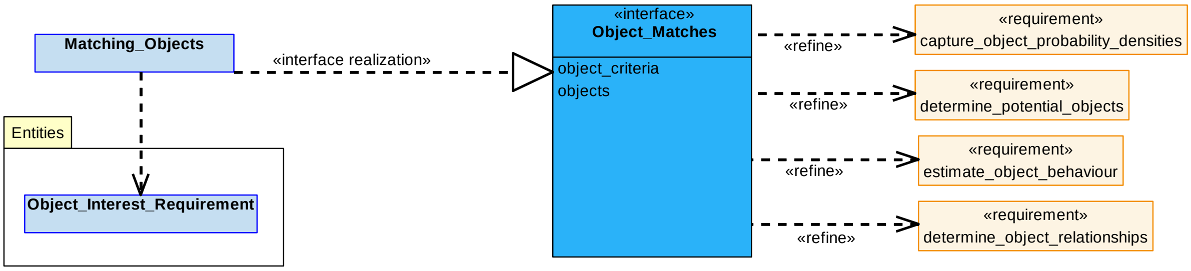


Figure 1037: Matching_Objects Service Definition

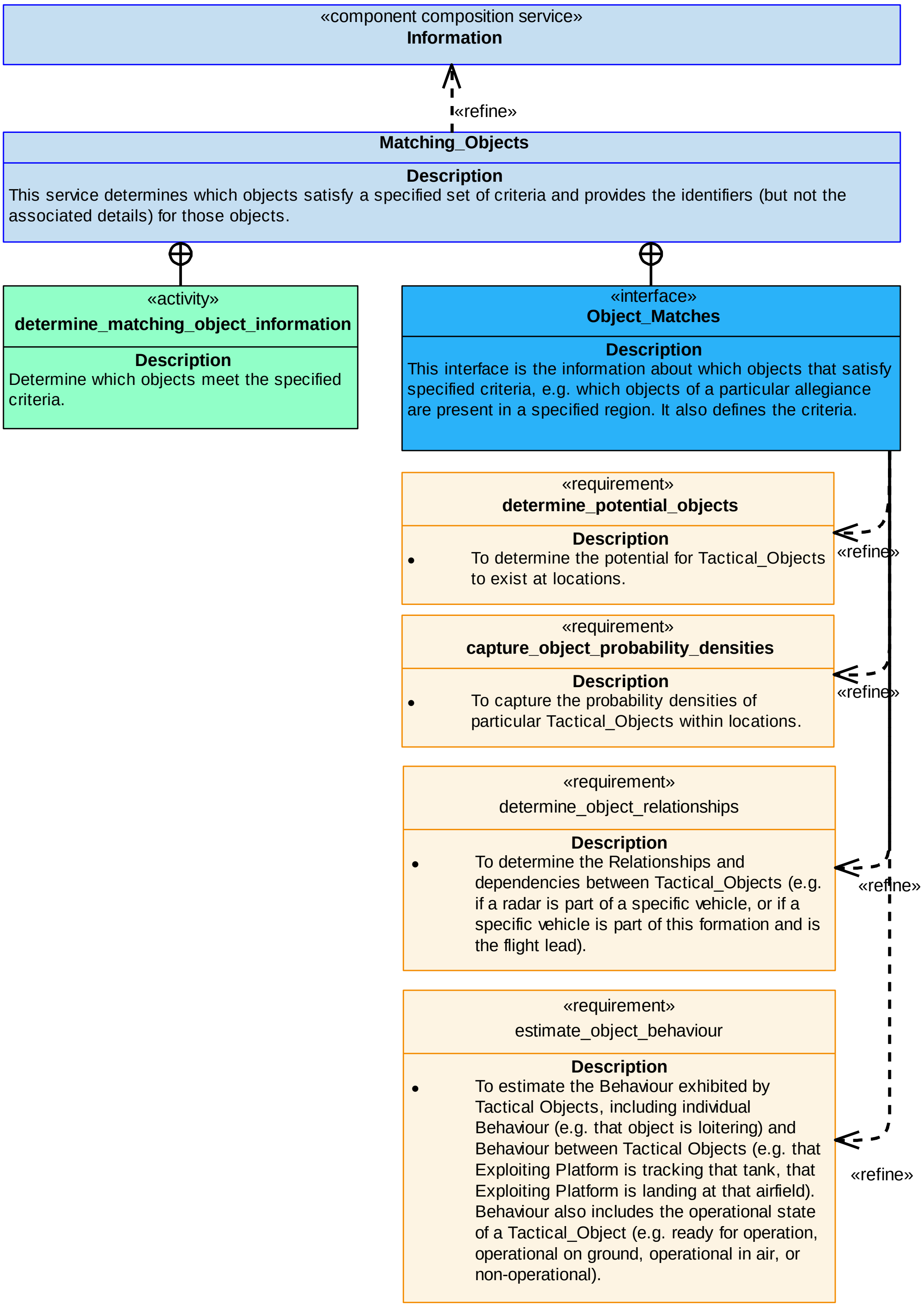

«component composition service»
Information
«refine»
Matching_Objects
Description
This service determines which objects satisfy a specified set of criteria and provides the identifiers (but not the associated details) for those objects.
«activity»
determine_matching_object_information
Description
Determine which objects meet the specified criteria.
«interface»
Object_Matches
Description
This interface is the information about which objects that satisfy specified criteria, e.g. which objects of a particular allegiance are present in a specified region. It also defines the criteria.
«requirement»
determine_potential_objects
Description
To determine the potential for Tactical_Objects to exist at locations.
«refine»
«requirement»
capture_object_probability_densities
Description
To capture the probability densities of particular Tactical_Objects within locations.
«refine»
«requirement»
determine_object_relationships
Description
To determine the Relationships and dependencies between Tactical_Objects (e.g. if a radar is part of a specific vehicle, or if a specific vehicle is part of this formation and is the flight lead).
«refine»
«requirement»
estimate_object_behaviour
Description
To estimate the Behaviour exhibited by Tactical Objects, including individual Behaviour (e.g. that object is loitering) and Behaviour between Tactical Objects (e.g. that Exploiting Platform is tracking that tank, that Exploiting Platform is landing at that airfield). Behaviour also includes the operational state of a Tactical_Object (e.g. ready for operation, operational on ground, operational in air, or non-operational).
«refine»

Figure 1038: Matching_Objects Service Policy

Matching_Objects

This service determines which objects satisfy a specified set of criteria and provides the identifiers (but not the associated details) for those objects.

Interface

Object_Matches

This interface is the information about which objects that satisfy specified criteria, e.g. which objects of a particular allegiance are present in a specified region. It also defines the criteria.

Attributes

object_criteria	The criteria used to determine which objects should be reported, e.g. report all objects located within a specified volume at or above a known confidence level.
objects	The objects that meet the specified criteria. Note that this only includes the object identifiers and not the details associated with the objects.

Activity

determine_matching_object_information

Determine which objects meet the specified criteria.

5.4.2.60.7.1.4 Specific_Object_Detail

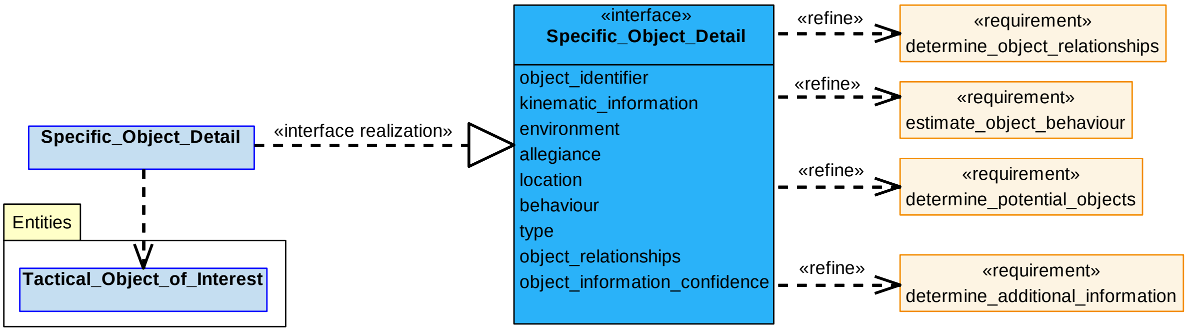


Figure 1039: Specific_Object_Detail Service Definition

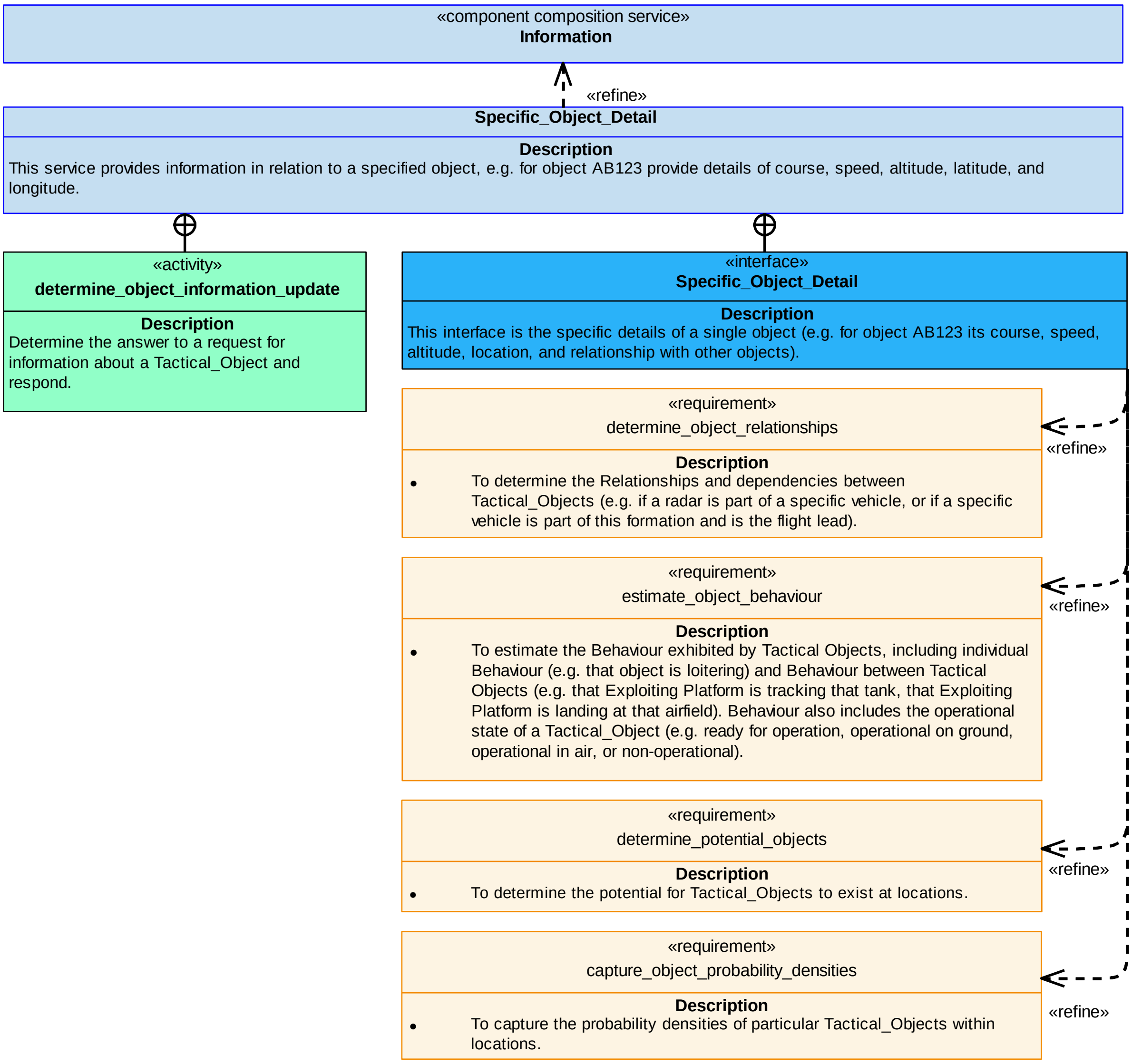

«component composition service»
Information
«refine»
Specific_Object_Detail
Description
This service provides information in relation to a specified object, e.g. for object AB123 provide details of course, speed, altitude, latitude, and longitude.
«activity»
determine_object_information_update
Description
Determine the answer to a request for information about a Tactical_Object and respond.
«interface»
Specific_Object_Detail
Description
This interface is the specific details of a single object (e.g. for object AB123 its course, speed, altitude, location, and relationship with other objects).
«requirement»
determine_object_relationships
Description
To determine the Relationships and dependencies between Tactical_Objects (e.g. if a radar is part of a specific vehicle, or if a specific vehicle is part of this formation and is the flight lead).
«refine»
«requirement»
estimate_object_behaviour
Description
To estimate the Behaviour exhibited by Tactical Objects, including individual Behaviour (e.g. that object is loitering) and Behaviour between Tactical Objects (e.g. that Exploiting Platform is tracking that tank, that Exploiting Platform is landing at that airfield). Behaviour also includes the operational state of a Tactical_Object (e.g. ready for operation, operational on ground, operational in air, or non-operational).
«refine»
«requirement»
determine_potential_objects
Description
To determine the potential for Tactical_Objects to exist at locations.
«refine»
«requirement»
capture_object_probability_densities
Description
To capture the probability densities of particular Tactical_Objects within locations.
«refine»

Figure 1040: Specific_Object_Detail Service Policy

Specific_Object_Detail

This service provides information in relation to a specified object, e.g. for object AB123 provide details of course, speed, altitude, latitude, and longitude.

Interface

Specific_Object_Detail

This interface is the specific details of a single object (e.g. for object AB123 its course, speed, altitude, location, and relationship with other objects).

Attributes

object_identifier	A unique identifier for the object. Note that this typically relates to other system identifiers, e.g. a Link16 track AB123 is related to local track CS227 which is sourced from Radar equipment track 326.

kinematic_information	A set of information relating to the objects motion which may include course, speed, accelerations (x/y/z), altitude, maximum speed, etc.
environment	The environment of the object. For example, surface, air, subsurface, land, or space.
allegiance	The allegiance of the object (e.g. friendly, neutral, hostile, suspect, or unknown).
location	Where the object is located (e.g. latitude / longitude / altitude).
behaviour	The Behaviour of the object (e.g. whether or not it is loitering, whether it is tracking a tank, or its operational state).
type	The classification of the object of being of a particular type, this could include multiple levels of granularity (e.g. amphibious vehicle, fast jet, Typhoon, or HMS Queen Elizabeth).
object_relationships	The Relationships and dependencies between objects.
object_information_confidence	An assessment of the confidence of the information being provided based on knowledge of the information source(s), e.g. the confidence that object AB123 has been correctly identified is 95%.

Activity

determine_object_information_update

Determine the answer to a request for information about a Tactical_Object and respond.

5.4.2.60.7.1.5 Object_Evidence

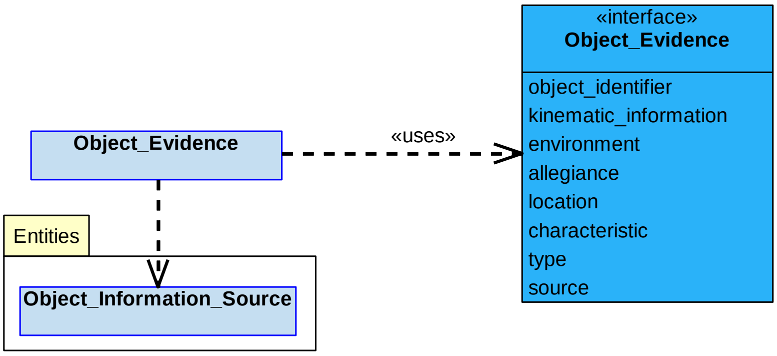


Figure 1041: Object_Evidence Service Definition

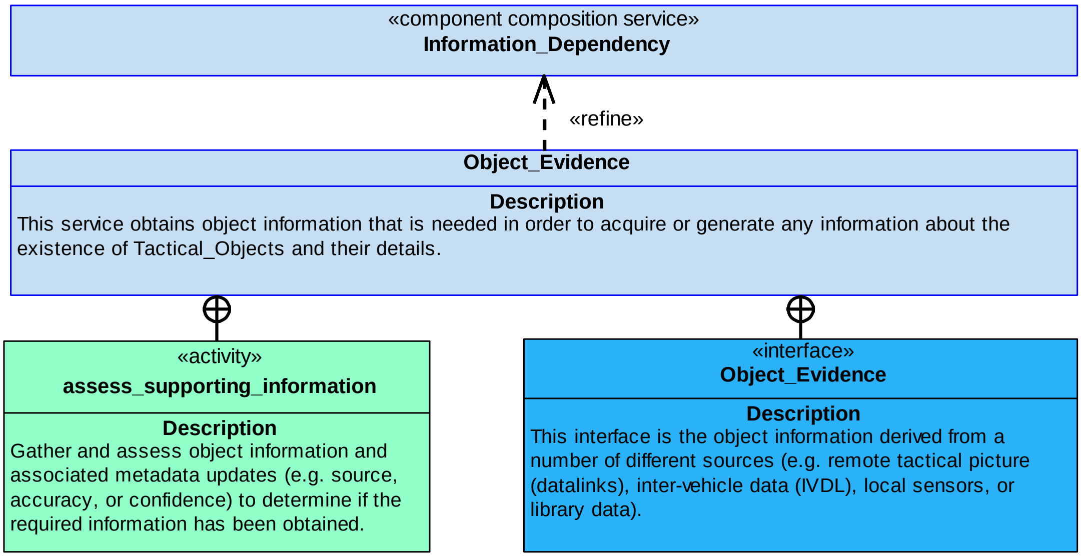

«component composition service»
Information_Dependency
«refine»
Object_Evidence
Description
This service obtains object information that is needed in order to acquire or generate any information about the existence of Tactical_Objects and their details.
«activity»
assess_supporting_information
Description
Gather and assess object information and associated metadata updates (e.g. source, accuracy, or confidence) to determine if the required information has been obtained.
«interface»
Object_Evidence
Description
This interface is the object information derived from a number of different sources (e.g. remote tactical picture (datalinks), inter-vehicle data (IVDL), local sensors, or library data).

Figure 1042: Object_Evidence Service Policy

Object_Evidence

This service obtains object information that is needed in order to acquire or generate any information about the existence of Tactical_Objects and their details.

Interface

Object_Evidence

This interface is the object information derived from a number of different sources (e.g. remote tactical picture (datalinks), inter-vehicle data (IVDL), local sensors, or library data).

Attributes

object_identifier	A unique identifier for the object. Note that this may typically relate to other system identifiers, e.g. a Link16 track AB123 is related to local track CS227 which is sourced from radar equipment track 326.
kinematic_information	A set of information relating to the objects motion which may include course, speed, accelerations (x/y/z), altitude, maximum speed, trajectory, etc.
environment	The environment of the object. For example, surface, air, subsurface, land, or space.
allegiance	The allegiance of the object (e.g. friendly, neutral, hostile, suspect, or unknown).
location	Where the object is located, e.g. latitude / longitude / altitude.
characteristic	A characteristic (e.g. a behaviour, or a confidence measure of other characteristics) specific to the object type.
type	The classification of the object of being of a particular type, this could include multiple levels of granularity, e.g. amphibious vehicle, fast jet, Typhoon, or HMS Queen Elizabeth.
source	The source of the information relating to an object which may include: • A remote tactical picture.

	• The local tactical picture. • An emitter database. • Information from a 'live' intelligence feed. • Operator entered information.

Activity

assess_supporting_information

Gather and assess object information and associated metadata updates (e.g. source, accuracy, or confidence) to determine if the required information has been obtained.

5.4.2.60.7.1.6 Constraint

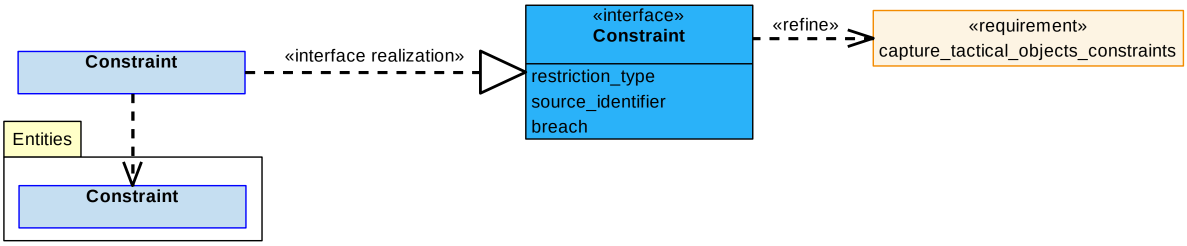
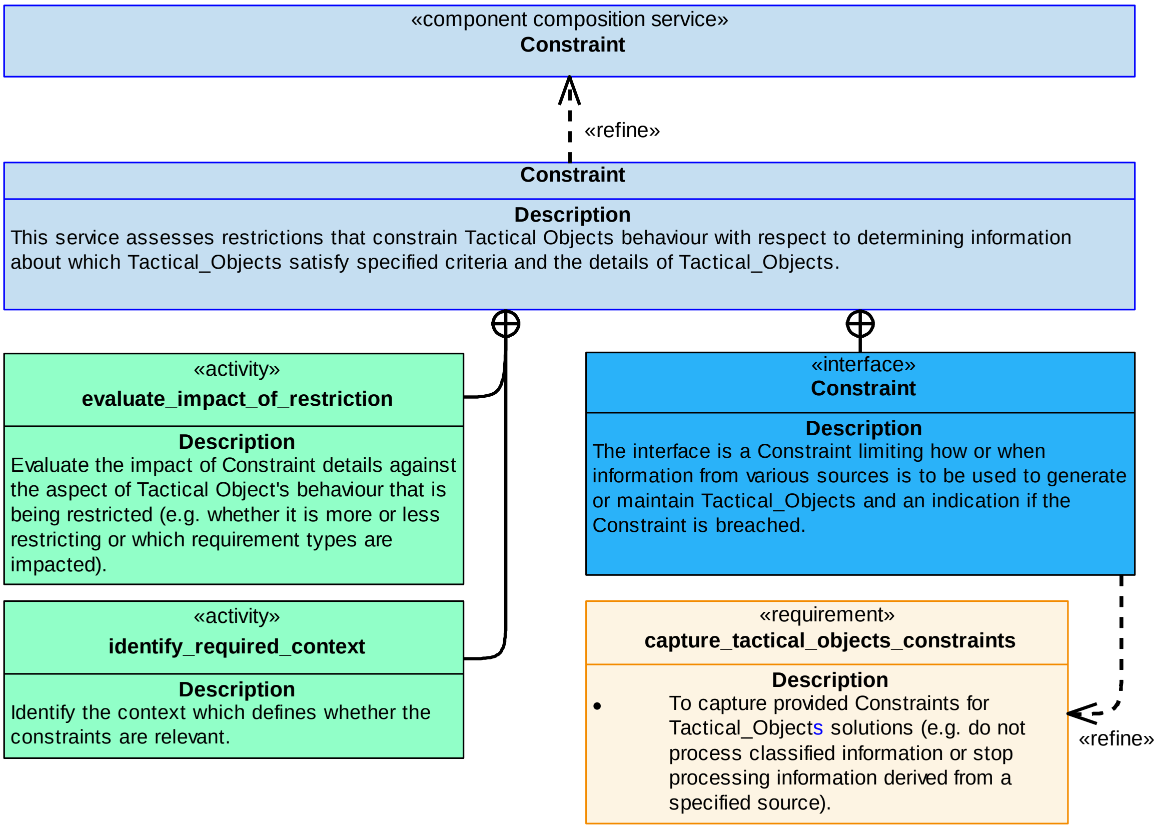


Figure 1043: Constraint Service Definition



This service assesses restrictions that constrain Tactical Objects behaviour with respect to determining information about which Tactical_Objects satisfy specified criteria and the details of Tactical_Objects.

Evaluate the impact of Constraint details against the aspect of Tactical Object's behaviour that is being restricted (e.g. whether it is more or less restricting or which requirement types are impacted).

Identify the context which defines whether the constraints are relevant.

The interface is a Constraint limiting how or when information from various sources is to be used to generate or maintain Tactical_Objects and an indication if the Constraint is breached.

- To capture provided Constraints for Tactical_Objects solutions (e.g. do not process classified information or stop processing information derived from a specified source).

Figure 1044: Constraint Service Policy

Constraint

This service assesses restrictions that constrain Tactical Objects behaviour with respect to determining information about which Tactical_Objects satisfy specified criteria and the details of Tactical_Objects.

Interface

Constraint

The interface is a Constraint limiting how or when information from various sources is to be used to generate or maintain Tactical_Objects and an indication if the Constraint is breached.

Attributes

restriction_type	A type of restriction to be applied against the source. Examples of types of restriction in relation to source restrictions could include: the ignoring of historic information, stopping the use of source information for a period of time, or the ignoring of specific data. Examples of restrictions applying to operation could include: limits to behaviour assessments due to autonomous operation constraints, increased tolerance on sensitive data measurements due to security constraints and changes to weight of evidence needed to assign allegiances depending on rules of engagement constraints.
source_identifier	An identification of the Object_Information_Source to which a restriction type applies.
breach	A statement that the restriction has been breached.

Activities

evaluate_impact_of_restriction

Evaluate the impact of Constraint details against the aspect of Tactical Object's behaviour that is being restricted (e.g. whether it is more or less restricting or which requirement types are impacted).

identify_required_context

Identify the context which defines whether the constraints are relevant.

5.4.2.60.7.1.7 Capability

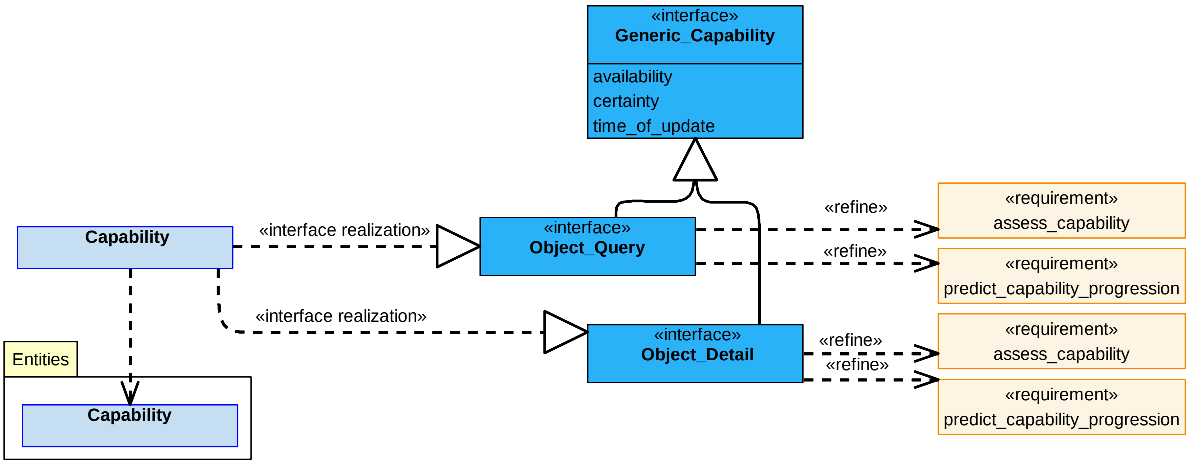
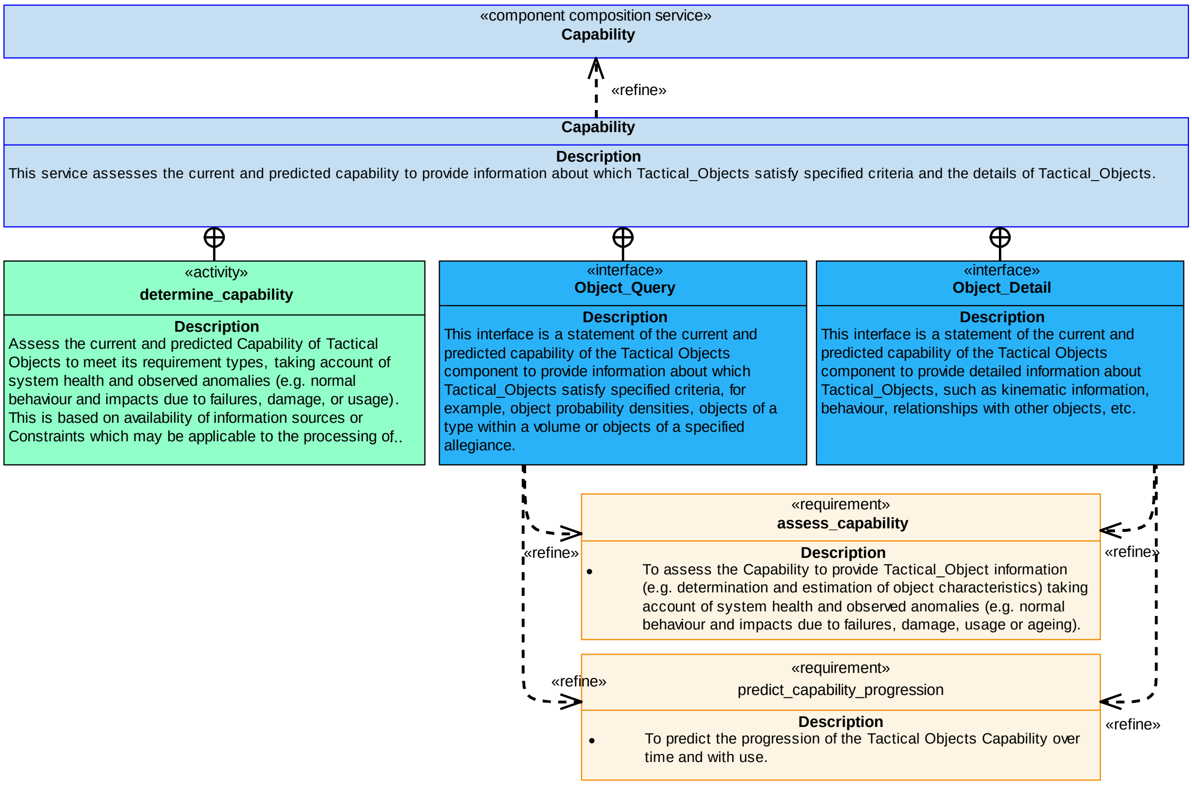


Figure 1045: Capability Service Definition



Description

This service assesses the current and predicted capability to provide information about which Tactical_Objects satisfy specified criteria and the details of Tactical_Objects.

Description

Assess the current and predicted Capability of Tactical Objects to meet its requirement types, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, or usage). This is based on availability of information sources or Constraints which may be applicable to the processing of..

Description

This interface is a statement of the current and predicted capability of the Tactical Objects component to provide information about which Tactical_Objects satisfy specified criteria, for example, object probability densities, objects of a type within a volume or objects of a specified allegiance.

Description

This interface is a statement of the current and predicted capability of the Tactical Objects component to provide detailed information about Tactical_Objects, such as kinematic information, behaviour, relationships with other objects, etc.

Description

- To assess the Capability to provide Tactical_Object information (e.g. determination and estimation of object characteristics) taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Description

- To predict the progression of the Tactical Objects Capability over time and with use.

Figure 1046: Capability Service Policy

Capability

This service assesses the current and predicted capability to provide information about which Tactical_Objects satisfy specified criteria and the details of Tactical_Objects.

Interfaces

Object_Query

This interface is a statement of the current and predicted capability of the Tactical Objects component to provide information about which Tactical_Objects satisfy specified criteria, for example, object probability densities, objects of a type within a volume or objects of a specified allegiance.

Object_Detail

This interface is a statement of the current and predicted capability of the Tactical Objects component to provide detailed information about Tactical_Objects, such as kinematic information, behaviour, relationships with other objects, etc.

Activity

determine_capability

Assess the current and predicted Capability of Tactical Objects to meet its requirement types, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, or usage). This is based on availability of information sources or Constraints which may be applicable to the processing of available information sources.

5.4.2.60.7.1.8 Capability_Evidence

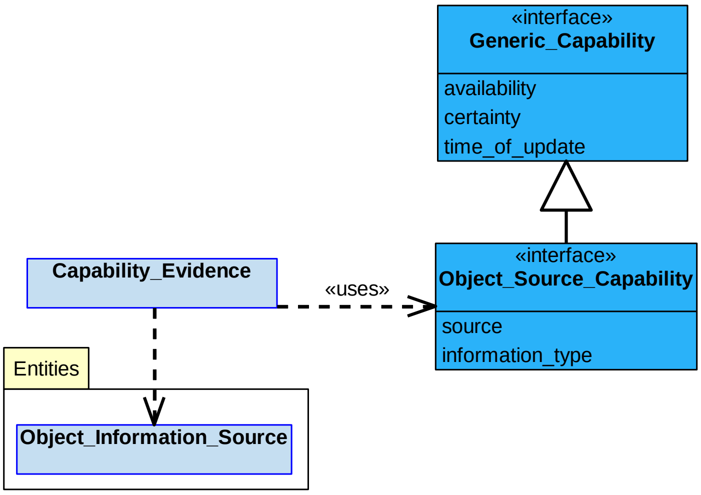


Figure 1047: Capability_Evidence Service Definition

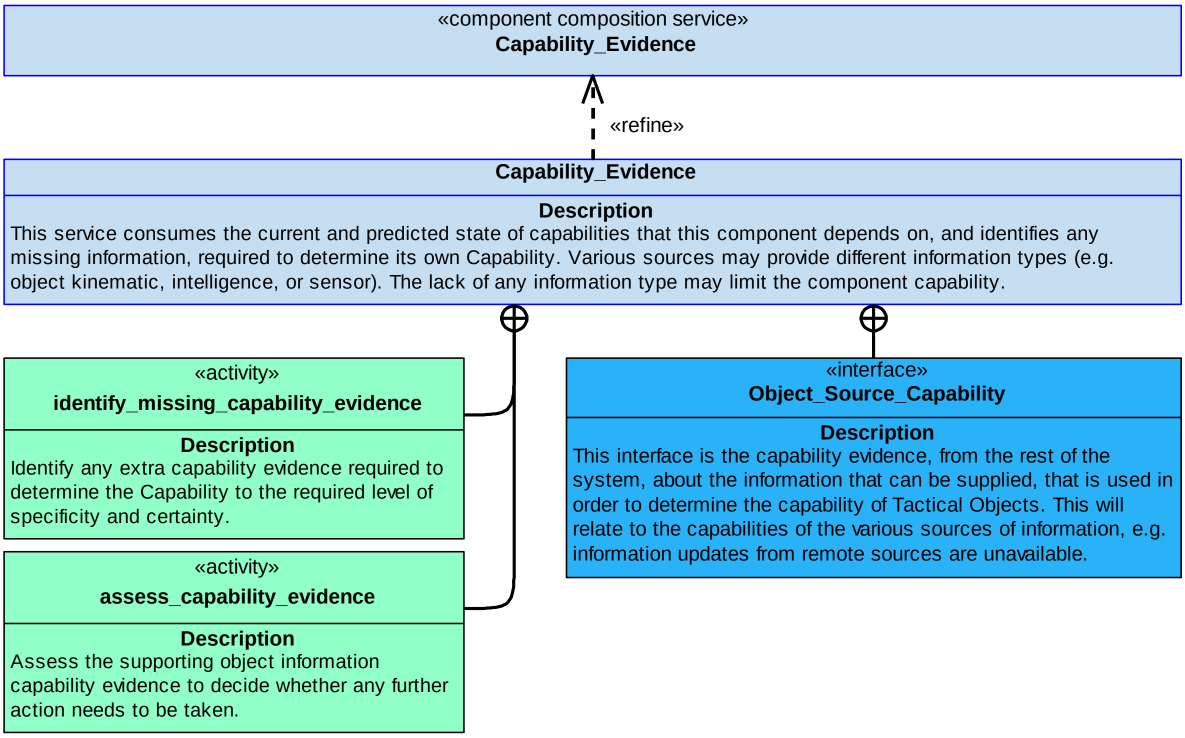

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted state of capabilities that this component depends on, and identifies any missing information, required to determine its own Capability. Various sources may provide different information types (e.g. object kinematic, intelligence, or sensor). The lack of any information type may limit the component capability.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«activity»
assess_capability_evidence
Description
Assess the supporting object information capability evidence to decide whether any further action needs to be taken.
«interface»
Object_Source_Capability
Description
This interface is the capability evidence, from the rest of the system, about the information that can be supplied, that is used in order to determine the capability of Tactical Objects. This will relate to the capabilities of the various sources of information, e.g. information updates from remote sources are unavailable.

Figure 1048: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted state of capabilities that this component depends on, and identifies any missing information, required to determine its own Capability. Various sources may provide different information types (e.g. object kinematic, intelligence, or sensor). The lack of any information type may limit the component capability.

Interface

Object_Source_Capability

This interface is the capability evidence, from the rest of the system, about the information that can be supplied, that is used in order to determine the capability of Tactical Objects. This will relate to the capabilities of the various sources of information, e.g. information updates from remote sources are unavailable.

Attributes

source	The origin of the information that can be supplied.
information_type	The type of information that can be supplied.

Activities

assess_capability_evidence

Assess the supporting object information capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.60.7.2 Service Dependencies

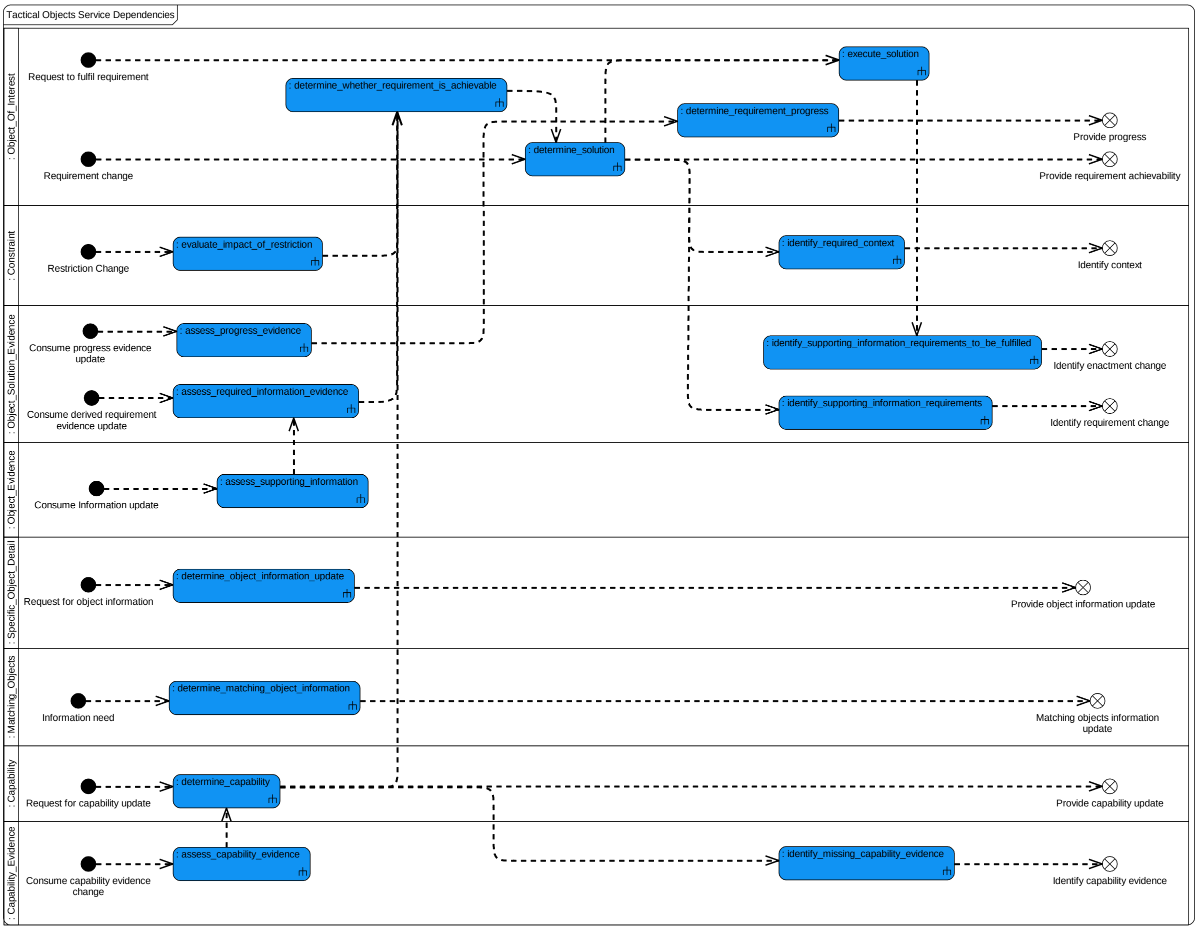

Tactical Objects Service Dependencies
: Object_Of_Interest
Request to fulfil requirement
: determine_whether_requirement_is_achievable
: execute_solution
: determine_requirement_progress
Provide progress
: determine_solution
Requirement change
Provide requirement achievability
: Constraint
Restriction Change
: evaluate_impact_of_restriction
: identify_required_context
Identify context
: Object_Solution_Evidence
Consume progress evidence update
: assess_progress_evidence
: identify_supporting_information_requirements_to_be_fulfilled
Identify enactment change
Consume derived requirement evidence update
: assess_required_information_evidence
: identify_supporting_information_requirements
Identify requirement change
: Object_Evidence
Consume Information update
: assess_supporting_information
: Specific_Object_Detail
Request for object information
: determine_object_information_update
Provide object information update
: Matching_Objects
Information need
: determine_matching_object_information
Matching objects information update
: Capability
Request for capability update
: determine_capability
Provide capability update
: Capability_Evidence
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1049: Tactical Objects Service Dependencies

5.4.2.61 Target Engagement

5.4.2.61.1 Role

The role of Target Engagement is to influence, disrupt, damage, destroy, mark, or deliver physical assets to a target.

5.4.2.61.2 Overview

Control Architecture

Target Engagement is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Target Engagement will determine an Engagement_Solution that most effectively achieves a Requirement for a particular effect, while taking into account Target_Engagement_Resources, Capability and Constraints, and identifying the Pre-conditions that need to be satisfied. The component will enact the Engagement_Solution using Target_Engagement_Resources and monitor the solution to determine the effectiveness until the desired outcome has been achieved.

Examples of Use

Target Engagement will be required to:

- Destroy a target through the use of weapons.
- Temporarily deny a target's capabilities through the use of jamming effectors.
- Threaten a target to influence its actions through the use of focused radar illumination.
- Mark a target to identify it to others or aid weapon guidance through the use of laser illumination.
- Deliver an asset to a location.
- Disable a target's capabilities after making a decision on whether to use destructive, damaging, or temporary denial approaches.
- Coordinate the use of different weapons, assets and effectors that complement each other in an overall solution to achieve a target engagement requirement.

5.4.2.61.3 Service Summary

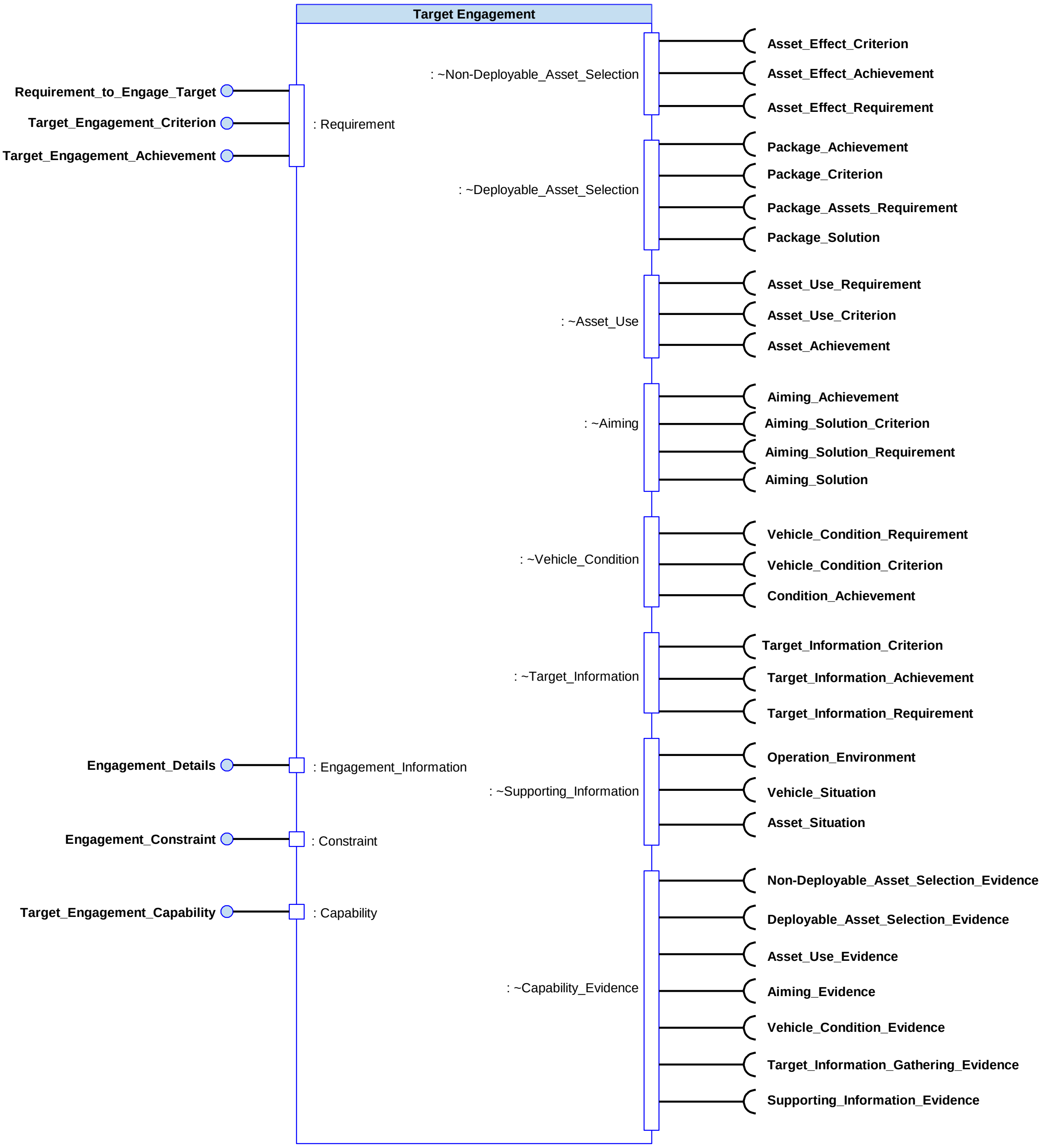

Target Engagement
Requirement_to_Engage_Target
Target_Engagement_Criterion
Target_Engagement_Achievement
: Requirement
Engagement_Details
: Engagement_Information
Engagement_Constraint
: Constraint
Target_Engagement_Capability
: Capability
: ~Non-Deployable_Asset_Selection
Asset_Effect_Criterion
Asset_Effect_Achievement
Asset_Effect_Requirement
: ~Deployable_Asset_Selection
Package_Achievement
Package_Criterion
Package_Assets_Requirement
Package_Solution
: ~Asset_Use
Asset_Use_Requirement
Asset_Use_Criterion
Asset_Achievement
: ~Aiming
Aiming_Achievement
Aiming_Solution_Criterion
Aiming_Solution_Requirement
Aiming_Solution
: ~Vehicle_Condition
Vehicle_Condition_Requirement
Vehicle_Condition_Criterion
Condition_Achievement
: ~Target_Information
Target_Information_Criterion
Target_Information_Achievement
Target_Information_Requirement
: ~Supporting_Information
Operation_Environment
Vehicle_Situation
Asset_Situation
: ~Capability_Evidence
Non-Deployable_Asset_Selection_Evidence
Deployable_Asset_Selection_Evidence
Asset_Use_Evidence
Aiming_Evidence
Vehicle_Condition_Evidence
Target_Information_Gathering_Evidence
Supporting_Information_Evidence

Figure 1050: Target Engagement Service Summary

5.4.2.61.4 Responsibilities

capture_requirements_for_target_engagement

- To capture provided Requirements (e.g. Target, Engagement_Type, timing, and weapon type, if specified) for target engagement.

capture_measurement_criteria_for_target_engagement

- To capture provided Measurement_Criterion/criteria (e.g. effectiveness and precision) that an Engagement_Solution and Engagements will be measured against.

capture_constraints_for_target_engagement

- To capture provided Constraints for target engagement.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current Target_Engagement_Resources and Constraints.

determine_target_engagement_solution

- To determine an Engagement_Solution that meets the given Requirements within provided Constraints using available Target_Engagement_Resources.

determine_predicted_quality_of_target_engagement_solution

- To determine the quality of the proposed Engagement_Solution against given Measurement_Criterion/criteria.

identify_pre-conditions

- To identify Pre-conditions required to support target engagement.

coordinate_target_engagement_solution

- To coordinate the execution of an Engagement_Solution by commanding Target_Engagement_Resources.

identify_progress_of_target_engagement_solution

- To identify the progress of an Engagement_Solution against the Requirements.

determine_actual_quality_of_outcome

- To determine the quality of the Engagement outcomes generated by executing an Engagement_Solution, measured against given Requirements and Measurement_Criterion/criteria.

assess_target_engagement_capability

- To assess the Capability of the component to perform target engagement using available weapons and other Target_Engagement_Resources within given Constraints, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_progression_of_capability

- To predict the progression of the component's Capability to perform Engagement_Solutions over time and with use.

5.4.2.61.5 Subject Matter Semantics

The subject matter of Target Engagement is the solutions for achieving the Engagement of a Target using engagement resources.

Exclusions

The subject matter of Target Engagement does not include:

- The determination of the availability of weapon and own ship capabilities.
- How the capabilities of a weapon or effect can harm or affect a target, however Target Engagement will have an understanding of the expected effects and any interactions or complimentary actions associated with different effects.
- Communication with weapons or effectors.

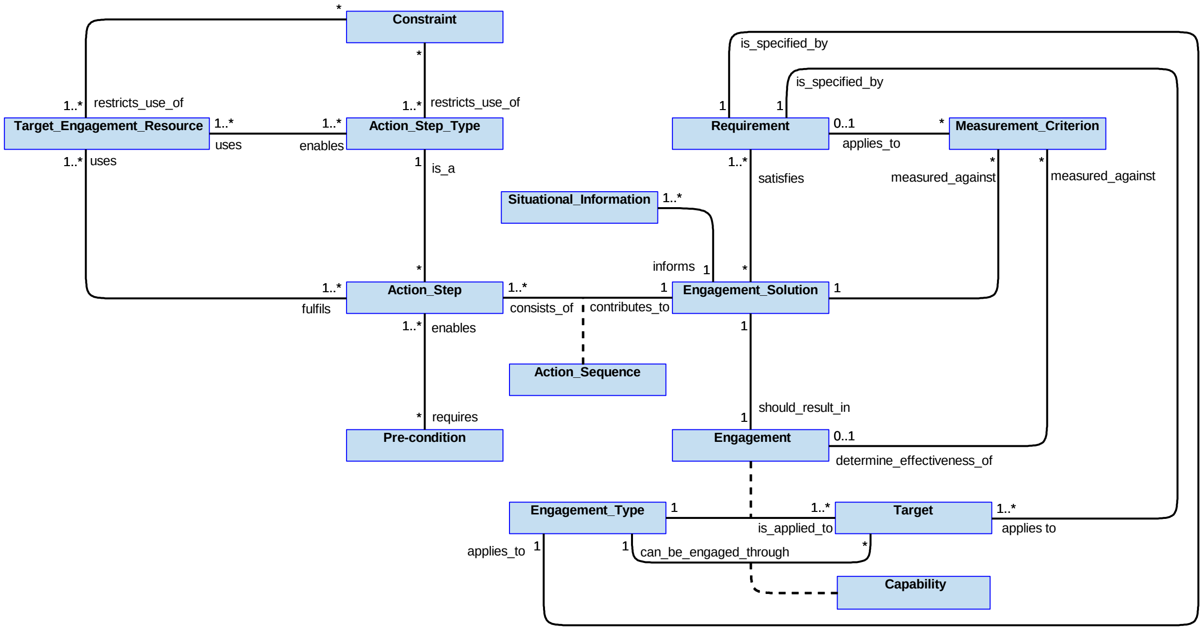


Figure 1051: Target Engagement Semantics

5.4.2.61.5.1 Entities

Action_Sequence

The order in which Action_Steps must be performed to achieve an Engagement_Solution.

Action_Step

An action that contributes to the engagement of a Target (either fully or in part).

Action_Step_Type

A type of action that can contribute to the engagement of a target, either fully or in part (e.g. an action contributing to the use of a weapon or effector, or an action contributing to the delivery of an asset to a target location).

Capability

The range of Target types that can be engaged and what can be done to them (e.g. the capability to destroy tank targets, or the capability to mark fixed artillery targets).

Constraint

An externally imposed restriction, e.g. a restriction on which weapon types are permitted.

Engagement

The act/attempt of influencing, disrupting, damaging, destroying, marking, or delivering physical assets to a tactical target.

Engagement_Solution

A solution to engage a target in a particular tactical way through the use of suitable effects (e.g. harming a target by the coordination of the delivery of a weapon, suppressing a target via the use of jamming, or delivering an asset to the target location).

Engagement_Type

A specific type of engagement (e.g. destroy target, disrupt communications, provoke activity, drop supplies, or position remote sensor).

Measurement_Criterion

A criterion which the quality of an Engagement_Solution and Engagements will be measured against (e.g. precision, or the cost of using Target_Engagement_Resources).

Pre-condition

A condition that must be true (e.g. ownship positioning, sensing support from another component, or authorisation).

Requirement

A requirement to affect a target in some way (e.g. to provoke a desired response from a target, or deliver an asset to the target location).

Target

The subject of a tactical engagement that can be either a location (including an area or a zone), a type of object, a specific object, or a cluster of objects.

Target_Engagement_Resource

A resource which can be instructed to carry out a type of action related to engaging a Target (e.g. equipment such as a missile or active sensor, or Exploiting Platform functionality (which may be supported by any type of component) associated with delivery of a weapon or an effect).

Situational_Information

Situational information to be considered when planning an engagement solution. For example, this could be information about the operating environment.

5.4.2.61.6 Design Rationale

5.4.2.61.6.1 Assumptions

- The types of weapons, effectors, or deployable assets available will be updated rarely (e.g. to represent a new type of weapon like a DEW).
- The supported Engagement_Types and Target types will be updated rarely (e.g. to represent a new type of target such as a satellite).
- Target Engagement will require authorisation before enacting an Engagement_Solution, or a critical stage of the Engagement_Solution.
- Target Engagement may request deployable assets and effectors to be made available for use (e.g. to achieve a ready for release state), or request the release of deployable assets or activation of effectors. It may also provide broad requirements for how they should be used. However, it is unlikely to be involved in the details of how these are achieved.

5.4.2.61.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Target Engagement:

- Data Driving - This PYRAMID concept is applicable to cope with configuring the component for different profiles of available weapons (e.g. AMRAAM or Meteor) or effectors (e.g. jammer or laser), and potential targets (e.g. T90 tank or water tower). This could include the type of weapon, effector, range and modes, or the type of target, max speed and size.
- Multi-Vehicle Coordination - This PYRAMID concept is applicable in scenarios where Target Engagement would need coordination between Exploiting Platforms.
- Recording and Logging and Storage - retention of engagement decisions for audit purposes will be performed in accordance with these PYRAMID concepts.

Extensions

This component could be extended to support different:

- Types of weapon (e.g. air-to-air missile and ballistic bomb).
- Types of effector (e.g. jammer and laser).
- Engagement_Types and/or Target types in relation to weapons or effectors (e.g. air-to-ground, jamming, and denial of service).
- Delivery mechanisms (e.g. ballistic weapon (guided or unguided), steered ejection, fire and forget guided, ballistic gun (fixed or moveable), and on board or deployable effectors).

- Options for bringing together aspects of a more complex target engagement (e.g. where different types of weapon are deployed and jamming is used to suppress intercept capabilities of an adversary).

Exploitation Considerations

- There could be a single or multiple instance(s) of Target Engagement for multiple vehicles (in accordance with the Multi-Vehicle Coordination PYRAMID concept).

5.4.2.61.6.3 Safety Considerations

The indicative IDAL is DAL B*.*

The rationale behind this is:

- Failure of this component could result in weapons impacting locations not intended by the crew and so result in unintended harm to third parties. This drives a DAL B indicative IDAL.

5.4.2.61.6.4 Security Considerations

The indicative security classification is SNEO.

This component is concerned with the means to damage, disrupt or influence a specified Target in some way and will therefore require knowledge of a mission target (or at least, the type of target) and the combat effectiveness of and resources available to the Exploiting Platform in order to match strategies to the target. This information is generally considered SNEO, and its confidentiality should be appropriately protected in order to maintain a tactical advantage.

Loss of integrity and availability for the component will severely limit the capability of the platform to affect the target, and will also need appropriate protection.

The component is expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** of access or interference with these functions to support later forensic examination.
- **Maintaining Audit Records** through retaining engagement decisions and authorisations for accountability and non-repudiation purposes.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected.
- **System Status and Monitoring** that might indicate the chain of events required to prosecute a target has been compromised in some way.

Whilst it is not expected to implement security enforcing functions itself, it is expected to be supported by SEF provided by other components, e.g. through cryptography and secure communications to ensure the integrity of data sources that provide the target information, authorisation, etc. This is especially true where a coordinated multi-vehicle (using wingmen, ground assets, etc.) target engagement activity is planned.

5.4.2.61.7 Services

5.4.2.61.7.1 Service Definitions

5.4.2.61.7.1.1 Requirement

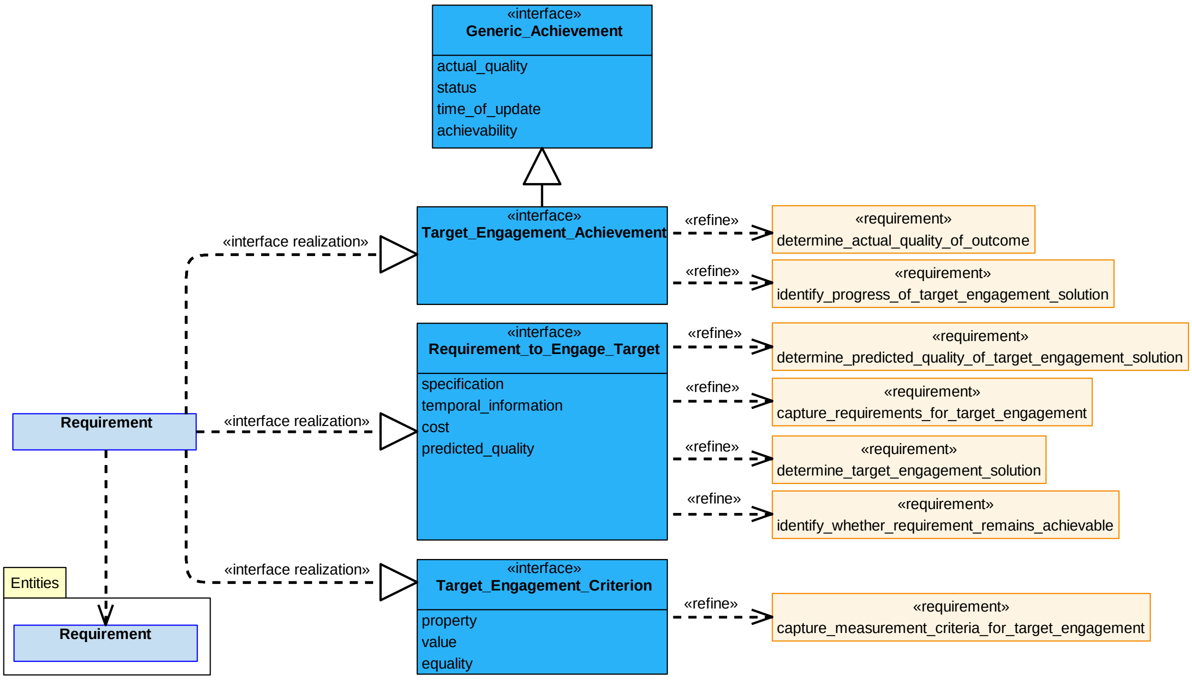


Figure 1052: Requirement Service Definition

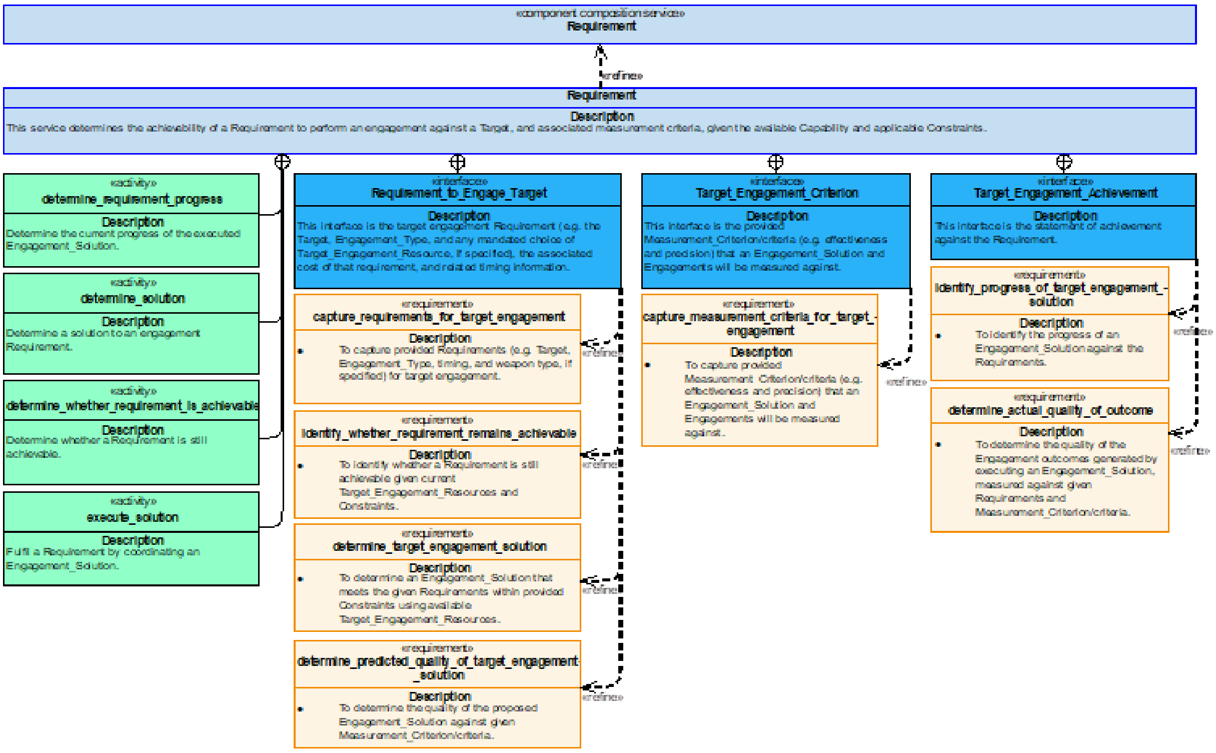

«component composition service»
Requirement
«refine»
Requirement
Description
This service determines the achievability of a Requirement to perform an engagement against a Target, and associated measurement criteria, given the available Capability and applicable Constraints.
«activity»
determine_requirement_progress
Description
Determine the current progress of the executed Engagement_Solution.
«activity»
determine_solution
Description
Determine a solution to an engagement Requirement.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether a Requirement is still achievable.
«activity»
execute_solution
Description
Fulfil a Requirement by coordinating an Engagement_Solution.
«interface»
Requirement_to_Engage_Target
Description
This interface is the target engagement Requirement (e.g. the Target, Engagement_Type, and any mandated choice of Target_Engagement_Resource, if specified), the associated cost of that requirement, and related timing information.
«requirement»
capture_requirements_for_target_engagement
Description
To capture provided Requirements (e.g. Target, Engagement_Type, timing, and weapon type, if specified) for target engagement.
«refine»
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Requirement is still achievable given current Target_Engagement_Resources and Constraints.
«refine»
«requirement»
determine_target_engagement_solution
Description
To determine an Engagement_Solution that meets the given Requirements within provided Constraints using available Target_Engagement_Resources.
«refine»
«requirement»
determine_predicted_quality_of_target_engagement_solution
Description
To determine the quality of the proposed Engagement_Solution against given Measurement_Criterion/criteria.
«refine»
«interface»
Target_Engagement_Criterion
Description
This interface is the provided Measurement_Criterion/criteria (e.g. effectiveness and precision) that an Engagement_Solution and Engagements will be measured against.
«requirement»
capture_measurement_criteria_for_target_engagement
Description
To capture provided Measurement_Criterion/criteria (e.g. effectiveness and precision) that an Engagement_Solution and Engagements will be measured against.
«refine»
«interface»
Target_Engagement_Achievement
Description
This interface is the statement of achievement against the Requirement.
«requirement»
identify_progress_of_target_engagement_solution
Description
To identify the progress of an Engagement_Solution against the Requirements.
«refine»
«requirement»
determine_actual_quality_of_outcome
Description
To determine the quality of the Engagement outcomes generated by executing an Engagement_Solution, measured against given Requirements and Measurement_Criterion/criteria.
«refine»

Figure 1053: Requirement Service Policy

Requirement

This service determines the achievability of a Requirement to perform an engagement against a Target, and associated measurement criteria, given the available Capability and applicable Constraints.

Interfaces

Target_Engagement_Criterion

This interface is the provided Measurement_Criterion/criteria (e.g. effectiveness and precision) that an Engagement_Solution and Engagements will be measured against.

Attributes

property	The property to be measured, e.g. theoretical probability of hitting or destroying a target.
value	The measured value of the property, e.g. 70% probable.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Requirement_to_Engage_Target

This interface is the target engagement Requirement (e.g. the Target, Engagement_Type, and any mandated choice of Target_Engagement_Resource, if specified), the associated cost of that requirement, and related timing information.

Attributes

specification	The definition of the requirement. This includes specification of or reference to the Target(s) and required Engagement_Type, plus any additional context requirements that need to be met.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used and time taken.
predicted_quality	How well the proposed Engagement_Solution is predicted to satisfy the Requirement.

Target_Engagement_Achievement

This interface is the statement of achievement against the Requirement.

Activities

determine_requirement_progress

Determine the current progress of the executed Engagement_Solution.

determine_solution

Determine a solution to an engagement Requirement.

execute_solution

Fulfil a Requirement by coordinating an Engagement_Solution.

determine_whether_requirement_is_achievable

Determine whether a Requirement is still achievable.

5.4.2.61.7.1.2 Non-Deployable_Asset_Selection

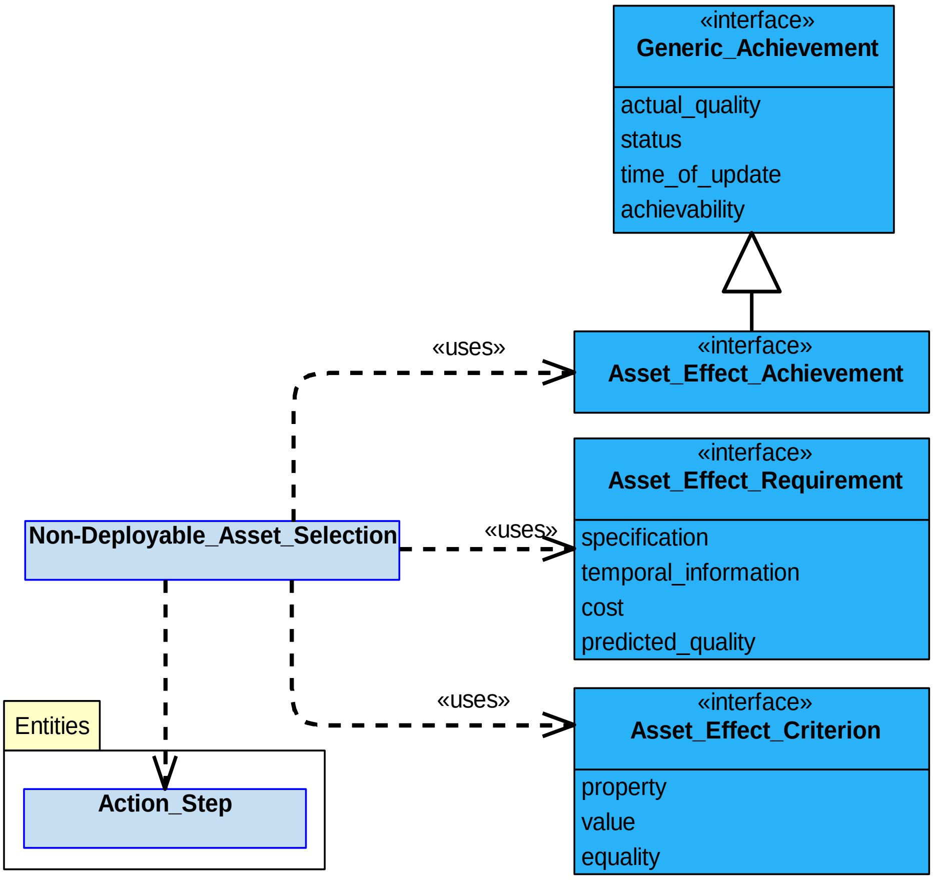


Figure 1054: Non-Deployable_Asset_Selection Service Definition

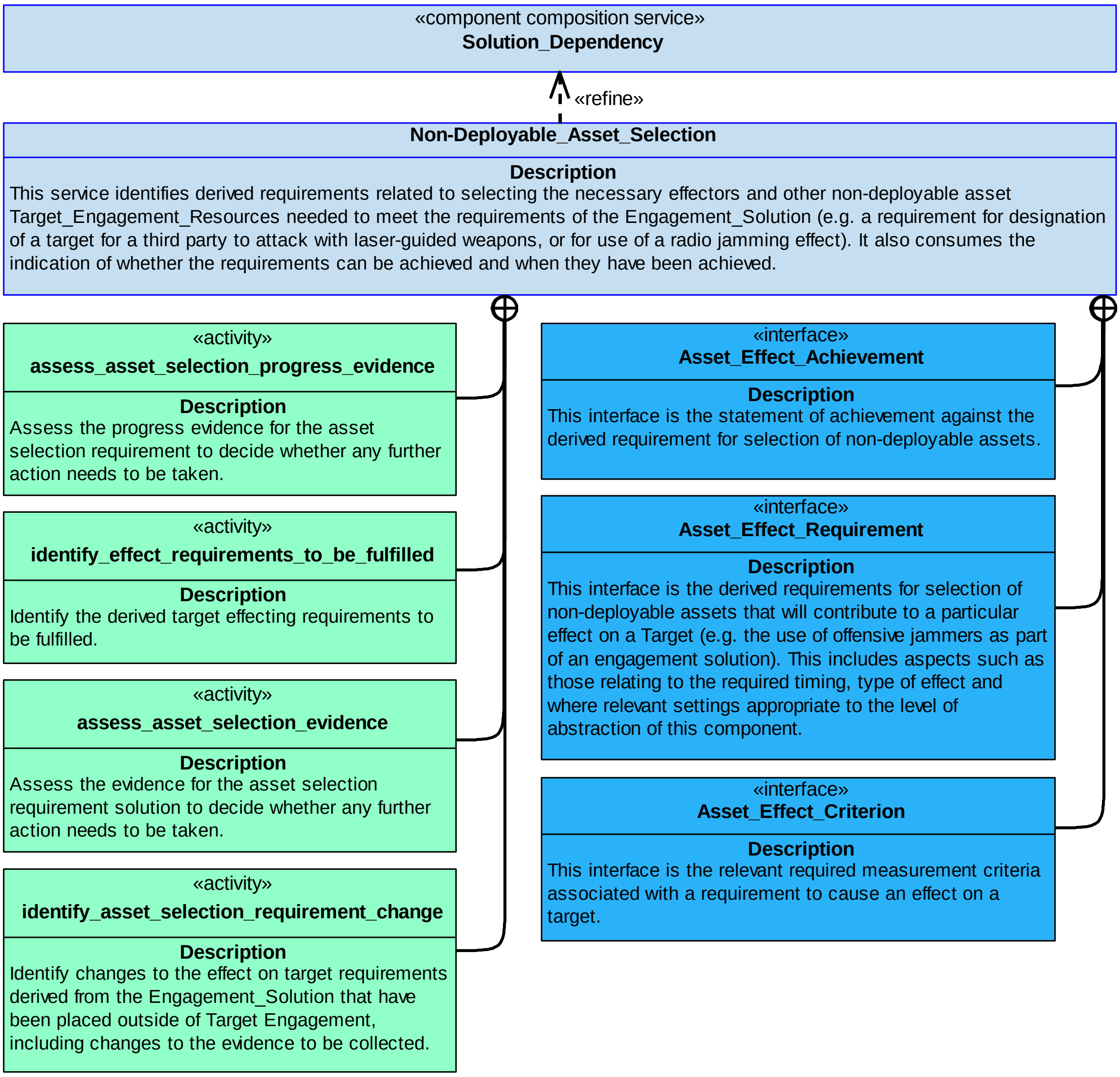

«component composition service»
Solution_Dependency
«refine»
Non-Deployable_Asset_Selection
Description
This service identifies derived requirements related to selecting the necessary effectors and other non-deployable asset Target_Engagement_Resources needed to meet the requirements of the Engagement_Solution (e.g. a requirement for designation of a target for a third party to attack with laser-guided weapons, or for use of a radio jamming effect). It also consumes the indication of whether the requirements can be achieved and when they have been achieved.
«activity»
assess_asset_selection_progress_evidence
Description
Assess the progress evidence for the asset selection requirement to decide whether any further action needs to be taken.
«activity»
identify_effect_requirements_to_be_fulfilled
Description
Identify the derived target effecting requirements to be fulfilled.
«activity»
assess_asset_selection_evidence
Description
Assess the evidence for the asset selection requirement solution to decide whether any further action needs to be taken.
«activity»
identify_asset_selection_requirement_change
Description
Identify changes to the effect on target requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to the evidence to be collected.
«interface»
Asset_Effect_Achievement
Description
This interface is the statement of achievement against the derived requirement for selection of non-deployable assets.
«interface»
Asset_Effect_Requirement
Description
This interface is the derived requirements for selection of non-deployable assets that will contribute to a particular effect on a Target (e.g. the use of offensive jammers as part of an engagement solution). This includes aspects such as those relating to the required timing, type of effect and where relevant settings appropriate to the level of abstraction of this component.
«interface»
Asset_Effect_Criterion
Description
This interface is the relevant required measurement criteria associated with a requirement to cause an effect on a target.

Figure 1055: Non-Deployable_Asset_Selection Service Policy

Non-Deployable_Asset_Selection

This service identifies derived requirements related to selecting the necessary effectors and other non-deployable asset Target_Engagement_Resources needed to meet the requirements of the Engagement_Solution (e.g. a requirement for designation of a target for a third party to attack with laser-guided weapons, or for use of a radio jamming effect). It also consumes the indication of whether the requirements can be achieved and when they have been achieved.

Interfaces

Asset_Effect_Criterion

This interface is the relevant required measurement criteria associated with a requirement to cause an effect on a target.

Attributes

property	The property to be measured, e.g. minimum level of reduction in target capability.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Asset_Effect_Requirement

This interface is the derived requirements for selection of non-deployable assets that will contribute to a particular effect on a Target (e.g. the use of offensive jammers as part of an engagement solution). This includes aspects such as those relating to the required timing, type of effect and where relevant settings appropriate to the level of abstraction of this component.

Attributes

specification	The definition of the derived requirement (e.g. to damage a target with highly focused energy, such as a laser).
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution (e.g. resources used and time taken).
predicted_quality	How well the proposed effect on target solution is predicted to satisfy the requirement.

Asset_Effect_Achievement

This interface is the statement of achievement against the derived requirement for selection of non-deployable assets.

Activities

assess_asset_selection_progress_evidence

Assess the progress evidence for the asset selection requirement to decide whether any further action needs to be taken.

identify_asset_selection_requirement_change

Identify changes to the effect on target requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to the evidence to be collected.

identify_effect_requirements_to_be_fulfilled

Identify the derived target effecting requirements to be fulfilled.

assess_asset_selection_evidence

Assess the evidence for the asset selection requirement solution to decide whether any further action needs to be taken.

5.4.2.61.7.1.3 Deployable_Asset_Selection

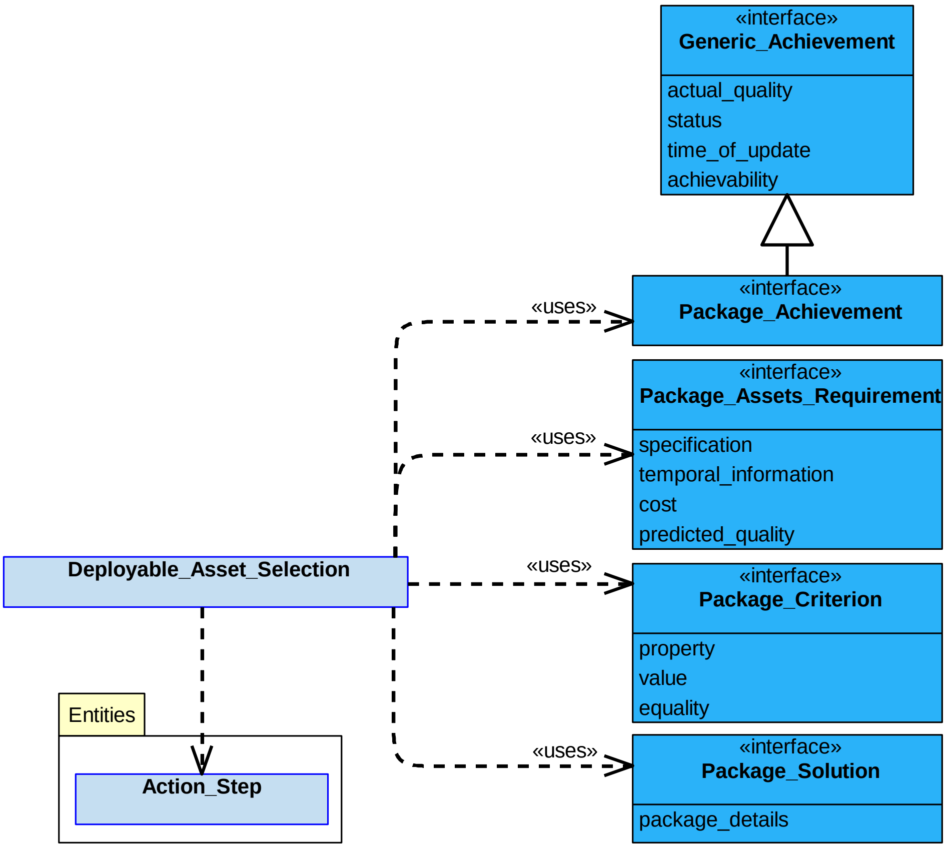


Figure 1056: Deployable_Asset_Package_Creation Service Definition

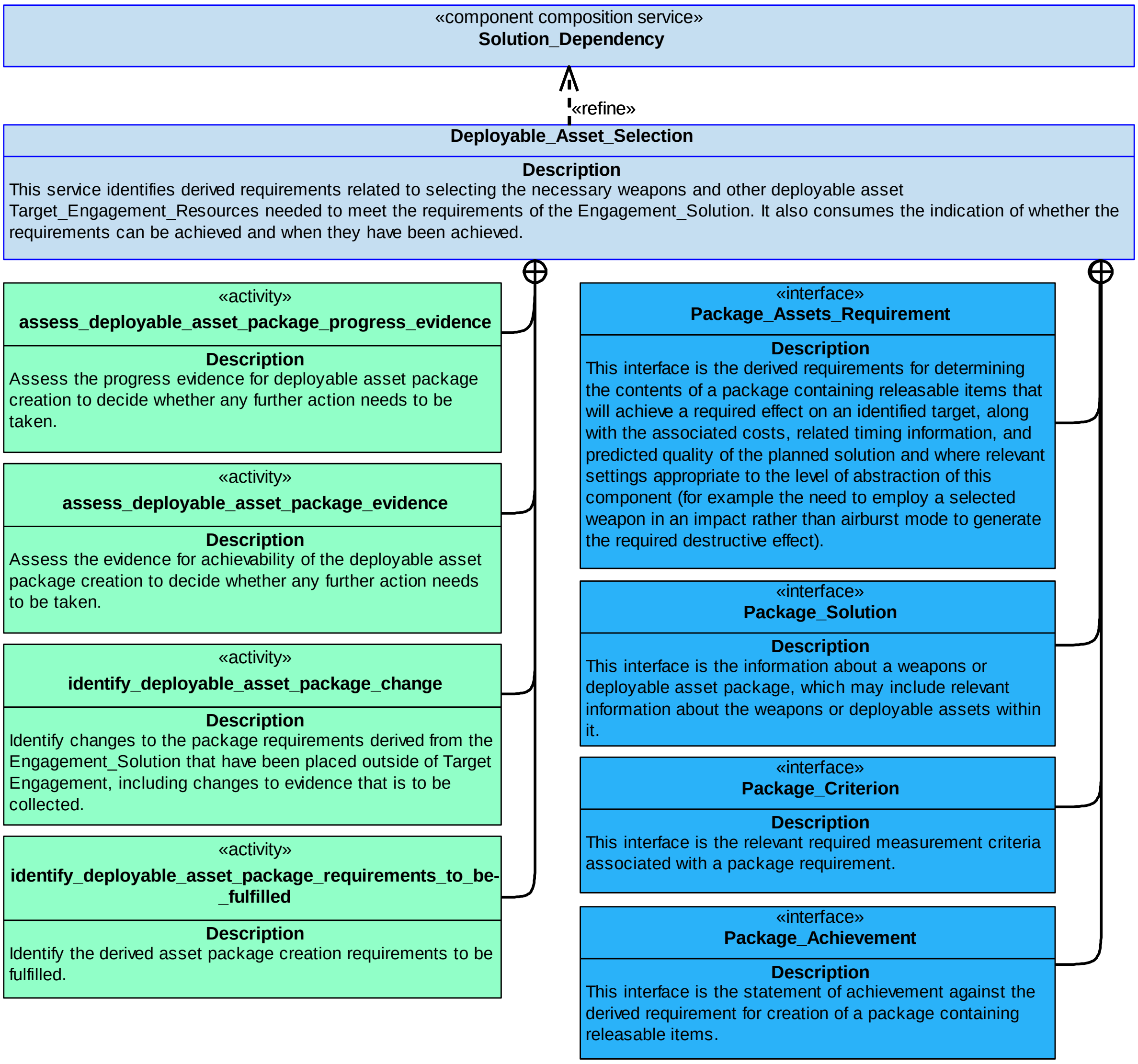

«component composition service»
Solution_Dependency
«refine»
Deployable_Asset_Selection
Description
This service identifies derived requirements related to selecting the necessary weapons and other deployable asset Target_Engagement_Resources needed to meet the requirements of the Engagement_Solution. It also consumes the indication of whether the requirements can be achieved and when they have been achieved.
«activity»
assess_deployable_asset_package_progress_evidence
Description
Assess the progress evidence for deployable asset package creation to decide whether any further action needs to be taken.
«activity»
assess_deployable_asset_package_evidence
Description
Assess the evidence for achievability of the deployable asset package creation to decide whether any further action needs to be taken.
«activity»
identify_deployable_asset_package_change
Description
Identify changes to the package requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to evidence that is to be collected.
«activity»
identify_deployable_asset_package_requirements_to_be-_fulfilled
Description
Identify the derived asset package creation requirements to be fulfilled.
«interface»
Package_Assets_Requirement
Description
This interface is the derived requirements for determining the contents of a package containing releasable items that will achieve a required effect on an identified target, along with the associated costs, related timing information, and predicted quality of the planned solution and where relevant settings appropriate to the level of abstraction of this component (for example the need to employ a selected weapon in an impact rather than airburst mode to generate the required destructive effect).
«interface»
Package_Solution
Description
This interface is the information about a weapons or deployable asset package, which may include relevant information about the weapons or deployable assets within it.
«interface»
Package_Criterion
Description
This interface is the relevant required measurement criteria associated with a package requirement.
«interface»
Package_Achievement
Description
This interface is the statement of achievement against the derived requirement for creation of a package containing releasable items.

Figure 1057: Deployable_Asset_Package_Creation Service Policy

Deployable_Asset_Selection

This service identifies derived requirements related to selecting the necessary weapons and other deployable asset Target_Engagement_Resources needed to meet the requirements of the Engagement_Solution. It also consumes the indication of whether the requirements can be achieved and when they have been achieved.

Interfaces

Package_Criterion

This interface is the relevant required measurement criteria associated with a package requirement.

Attributes

property	The property to be measured, e.g. number of releasable items in a package.
value	The measured value of the property, e.g. 3.

equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Package_Assets_Requirement

This interface is the derived requirements for determining the contents of a package containing releasable items that will achieve a required effect on an identified target, along with the associated costs, related timing information, and predicted quality of the planned solution and where relevant settings appropriate to the level of abstraction of this component (for example the need to employ a selected weapon in an impact rather than airburst mode to generate the required destructive effect).

Attributes

specification	The definition of the derived requirement.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used and time taken.
predicted_quality	How well the proposed asset package solution is predicted to satisfy the requirement.

Package_Solution

This interface is the information about a weapons or deployable asset package, which may include relevant information about the weapons or deployable assets within it.

Attribute

package_details	Information about the package and about the weapons or deployable assets within it. This may include unique references to each item within the package allowing them to be handled separately.

Package_Achievement

This interface is the statement of achievement against the derived requirement for creation of a package containing releasable items.

Activities

assess_deployable_asset_package_progress_evidence

Assess the progress evidence for deployable asset package creation to decide whether any further action needs to be taken.

identify_deployable_asset_package_change

Identify changes to the package requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to evidence that is to be collected.

identify_deployable_asset_package_requirements_to_be_fulfilled

Identify the derived asset package creation requirements to be fulfilled.

assess_deployable_asset_package_evidence

Assess the evidence for achievability of the deployable asset package creation to decide whether any further action needs to be taken.

5.4.2.61.7.1.4 Asset_Use

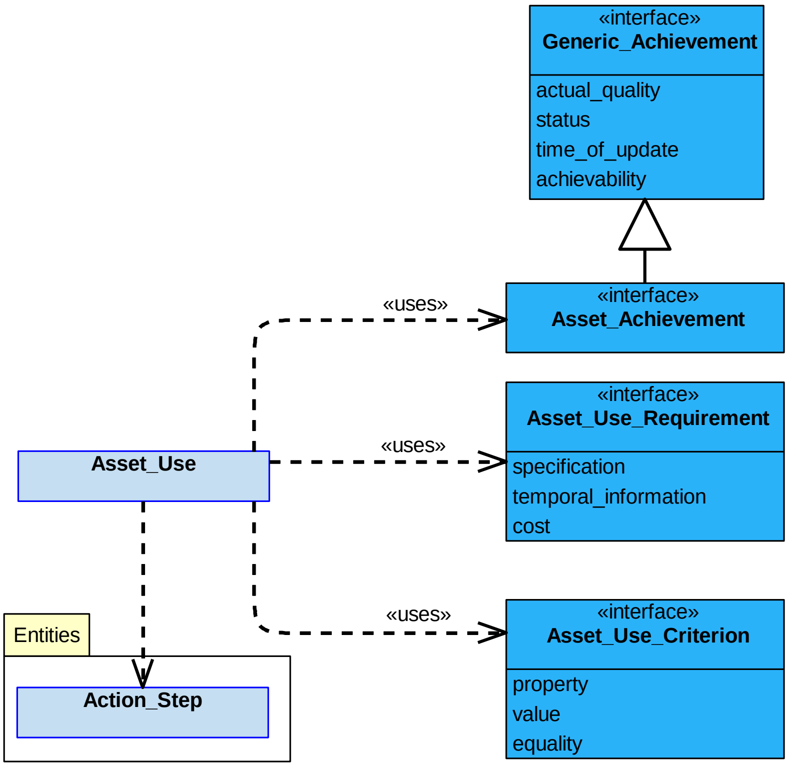


Figure 1058: Asset_Use Service Definition

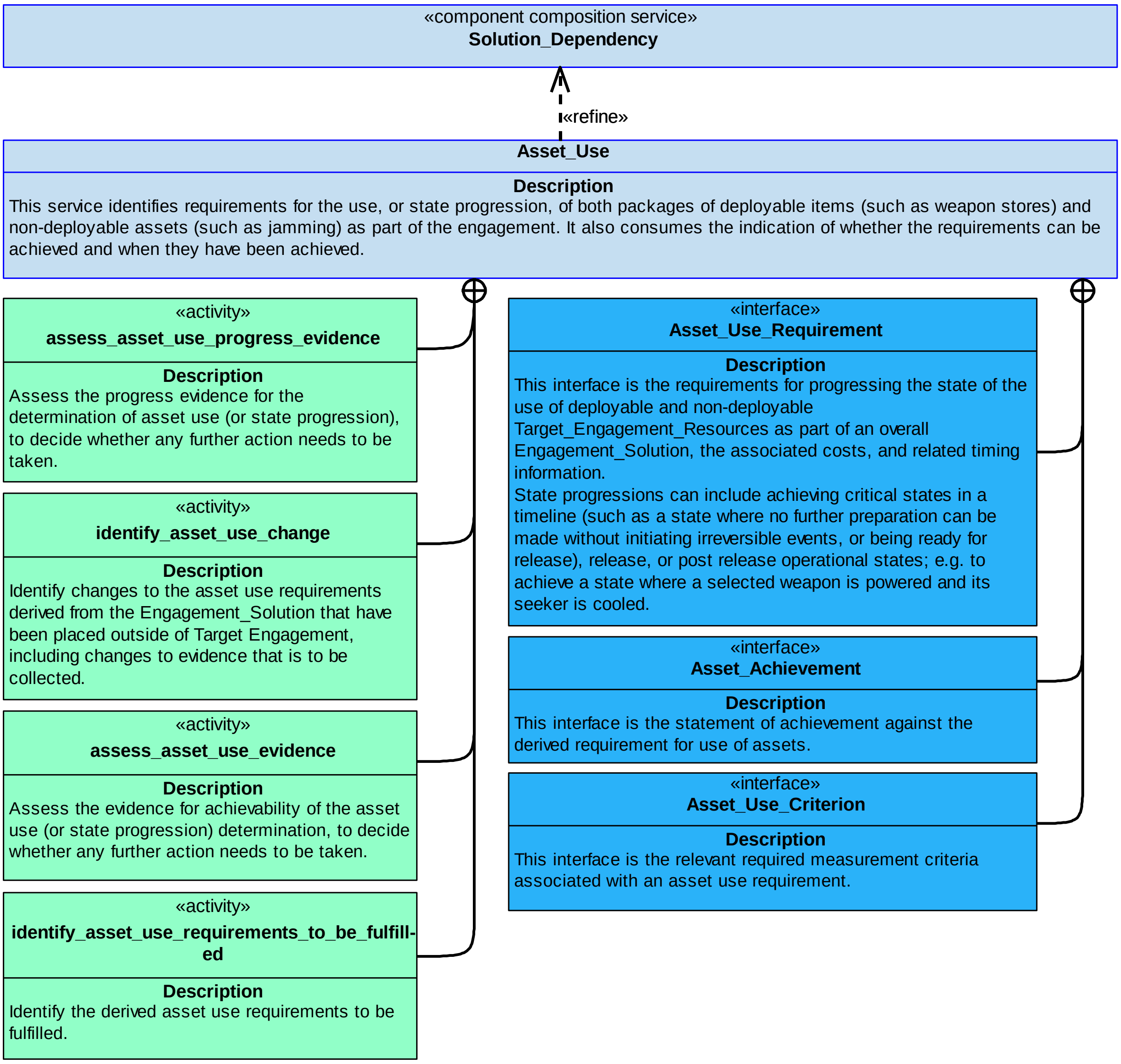

«component composition service»
Solution_Dependency
«refine»
Asset_Use
Description
This service identifies requirements for the use, or state progression, of both packages of deployable items (such as weapon stores) and non-deployable assets (such as jamming) as part of the engagement. It also consumes the indication of whether the requirements can be achieved and when they have been achieved.
«activity»
assess_asset_use_progress_evidence
Description
Assess the progress evidence for the determination of asset use (or state progression), to decide whether any further action needs to be taken.
«activity»
identify_asset_use_change
Description
Identify changes to the asset use requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to evidence that is to be collected.
«activity»
assess_asset_use_evidence
Description
Assess the evidence for achievability of the asset use (or state progression) determination, to decide whether any further action needs to be taken.
«activity»
identify_asset_use_requirements_to_be_fulfill-ed
Description
Identify the derived asset use requirements to be fulfilled.
«interface»
Asset_Use_Requirement
Description
This interface is the requirements for progressing the state of the use of deployable and non-deployable Target_Engagement_Resources as part of an overall Engagement_Solution, the associated costs, and related timing information.
State progressions can include achieving critical states in a timeline (such as a state where no further preparation can be made without initiating irreversible events, or being ready for release), release, or post release operational states; e.g. to achieve a state where a selected weapon is powered and its seeker is cooled.
«interface»
Asset_Achievement
Description
This interface is the statement of achievement against the derived requirement for use of assets.
«interface»
Asset_Use_Criterion
Description
This interface is the relevant required measurement criteria associated with an asset use requirement.

Figure 1059: Asset_Use Service Policy

Asset_Use

This service identifies requirements for the use, or state progression, of both packages of deployable items (such as weapon stores) and non-deployable assets (such as jamming) as part of the engagement. It also consumes the indication of whether the requirements can be achieved and when they have been achieved.

Interfaces

Asset_Use_Criterion

This interface is the relevant required measurement criteria associated with an asset use requirement.

Attributes

property	The property to be measured, e.g. the timeliness of state progression.
value	The measured value of the property.

equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Asset_Use_Requirement

This interface is the requirements for progressing the state of the use of deployable and non-deployable Target_Engagement_Resources as part of an overall Engagement_Solution, the associated costs, and related timing information.

State progressions can include achieving critical states in a timeline (such as a state where no further preparation can be made without initiating irreversible events, or being ready for release), release, or post release operational states; e.g. to achieve a state where a selected weapon is powered and its seeker is cooled.

Attributes

specification	The definition of the derived requirement.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution (e.g. resources used and time taken).

Asset_Achievement

This interface is the statement of achievement against the derived requirement for use of assets.

Activities

assess_asset_use_progress_evidence

Assess the progress evidence for the determination of asset use (or state progression), to decide whether any further action needs to be taken.

identify_asset_use_change

Identify changes to the asset use requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to evidence that is to be collected.

identify_asset_use_requirements_to_be_fulfilled

Identify the derived asset use requirements to be fulfilled.

assess_asset_use_evidence

Assess the evidence for achievability of the asset use (or state progression) determination, to decide whether any further action needs to be taken.

5.4.2.61.7.1.5 Aiming

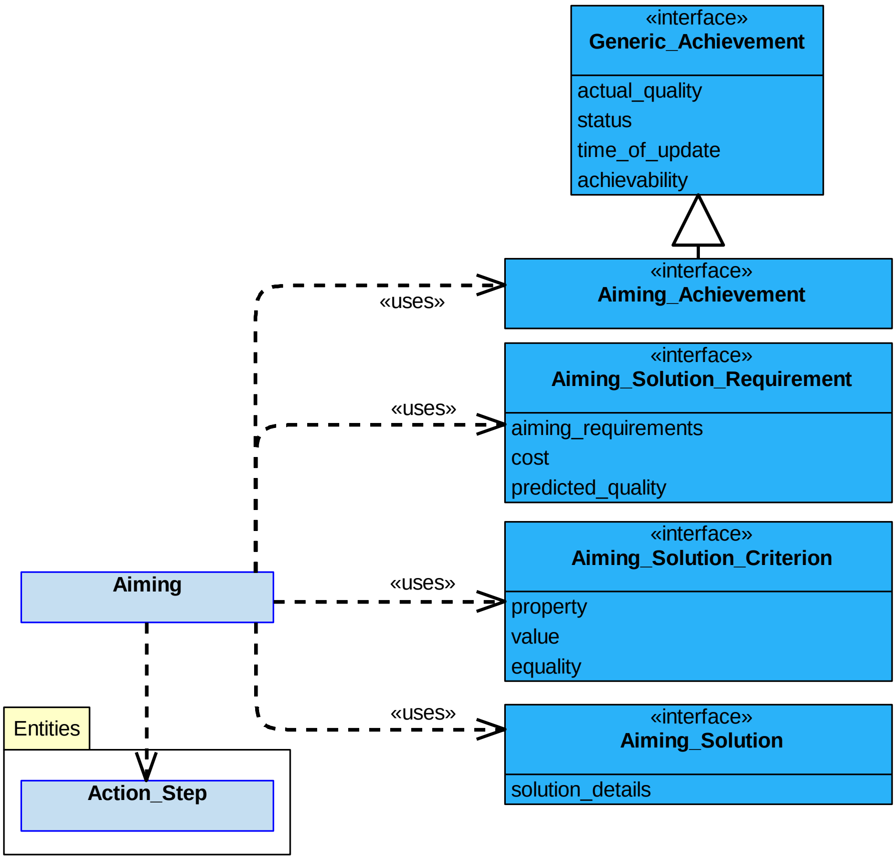


Figure 1060: Aiming Service Definition

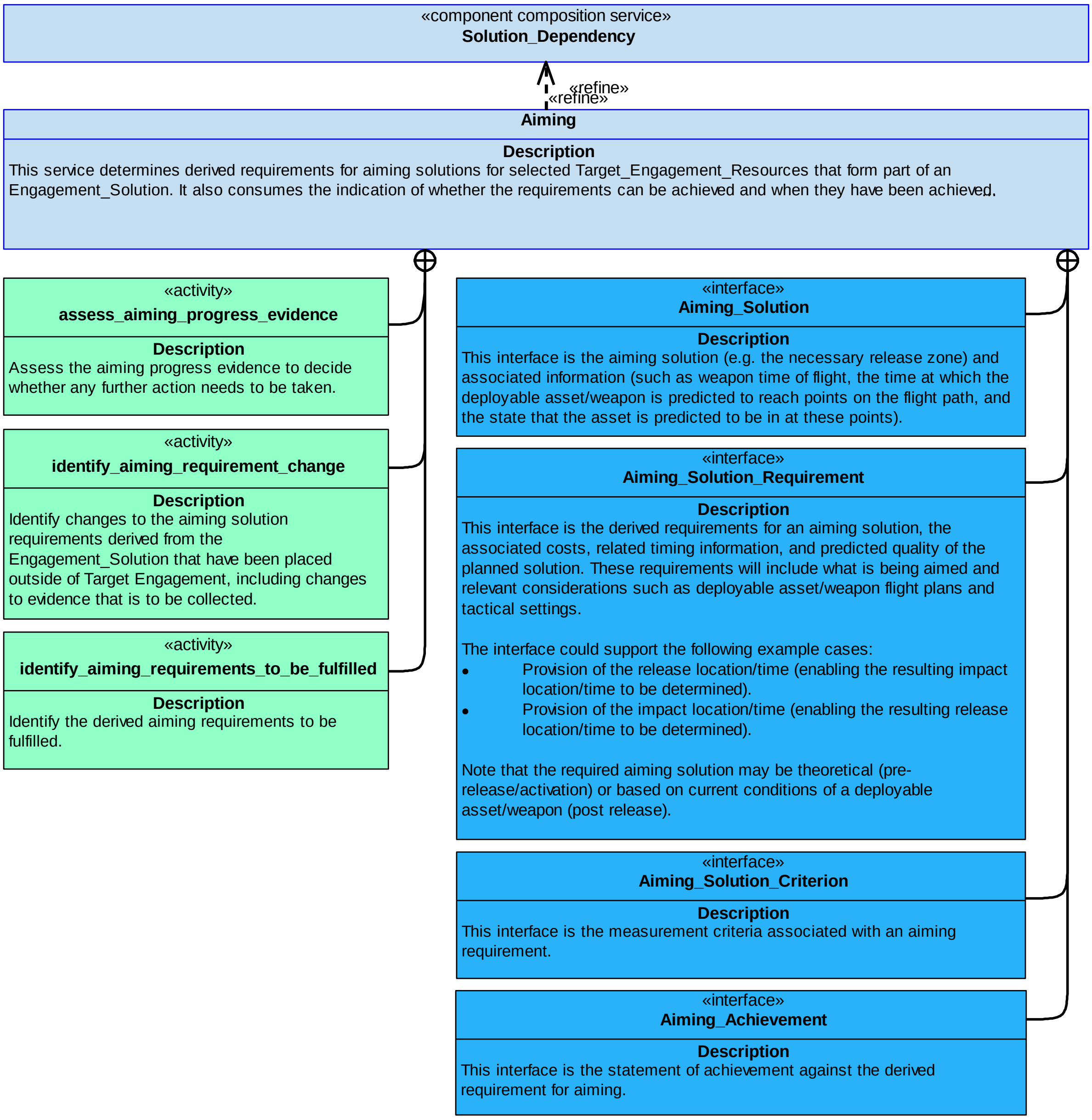

«component composition service»
Solution_Dependency
«refine»
«refine»
Aiming
Description
This service determines derived requirements for aiming solutions for selected Target_Engagement_Resources that form part of an Engagement_Solution. It also consumes the indication of whether the requirements can be achieved and when they have been achieved.
«activity»
assess_aiming_progress_evidence
Description
Assess the aiming progress evidence to decide whether any further action needs to be taken.
«activity»
identify_aiming_requirement_change
Description
Identify changes to the aiming solution requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to evidence that is to be collected.
«activity»
identify_aiming_requirements_to_be_fulfilled
Description
Identify the derived aiming requirements to be fulfilled.
«interface»
Aiming_Solution
Description
This interface is the aiming solution (e.g. the necessary release zone) and associated information (such as weapon time of flight, the time at which the deployable asset/weapon is predicted to reach points on the flight path, and the state that the asset is predicted to be in at these points).
«interface»
Aiming_Solution_Requirement
Description
This interface is the derived requirements for an aiming solution, the associated costs, related timing information, and predicted quality of the planned solution. These requirements will include what is being aimed and relevant considerations such as deployable asset/weapon flight plans and tactical settings.
The interface could support the following example cases:
• Provision of the release location/time (enabling the resulting impact location/time to be determined).
• Provision of the impact location/time (enabling the resulting release location/time to be determined).
Note that the required aiming solution may be theoretical (pre-release/activation) or based on current conditions of a deployable asset/weapon (post release).
«interface»
Aiming_Solution_Criterion
Description
This interface is the measurement criteria associated with an aiming requirement.
«interface»
Aiming_Achievement
Description
This interface is the statement of achievement against the derived requirement for aiming.

Figure 1061: Aiming Service Policy

Aiming

This service determines derived requirements for aiming solutions for selected Target_Engagement_Resources that form part of an Engagement_Solution. It also consumes the indication of whether the requirements can be achieved and when they have been achieved.

Interfaces

Aiming_Solution_Criterion

This interface is the measurement criteria associated with an aiming requirement.

Attributes

property	The property to be measured, e.g. theoretical probability of hitting a target.

value	The measured value of the property, e.g. 90% probable.
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Aiming_Solution_Requirement

This interface is the derived requirements for an aiming solution, the associated costs, related timing information, and predicted quality of the planned solution. These requirements will include what is being aimed and relevant considerations such as deployable asset/weapon flight plans and tactical settings.

The interface could support the following example cases:

- Provision of the release location/time (enabling the resulting impact location/time to be determined).
- Provision of the impact location/time (enabling the resulting release location/time to be determined).

Note that the required aiming solution may be theoretical (pre-release/activation) or based on current conditions of a deployable asset/weapon (post release).

Attributes

aiming_requirements	The definition of the derived aiming requirement.
cost	The cost of executing the solution (e.g. resources used and time taken).
predicted_quality	How well the proposed aiming solution is predicted to satisfy the requirement.

Aiming_Solution

This interface is the aiming solution (e.g. the necessary release zone) and associated information (such as weapon time of flight, the time at which the deployable asset/weapon is predicted to reach points on the flight path, and the state that the asset is predicted to be in at these points).

Attribute

solution_details	Information about the aiming solution.

Aiming_Achievement

This interface is the statement of achievement against the derived requirement for aiming.

Activities

assess_aiming_progress_evidence

Assess the aiming progress evidence to decide whether any further action needs to be taken.

identify_aiming_requirement_change

Identify changes to the aiming solution requirements derived from the Engagement_Solution that have been placed outside of Target Engagement, including changes to evidence that is to be collected.

identify_aiming_requirements_to_be_fulfilled

Identify the derived aiming requirements to be fulfilled.

5.4.2.61.7.1.6 Vehicle_Condition

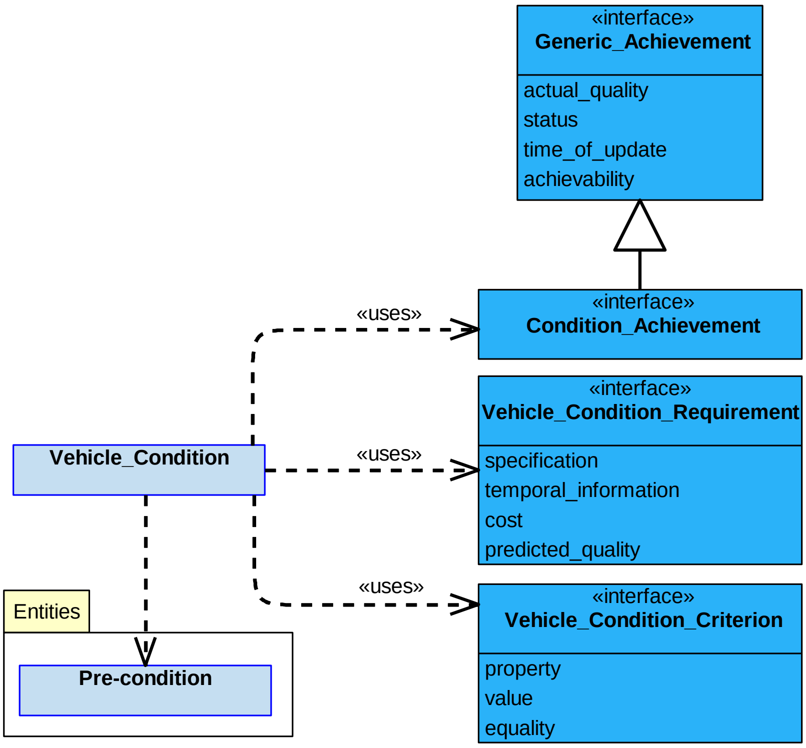


Figure 1062: Vehicle_Condition Service Definition

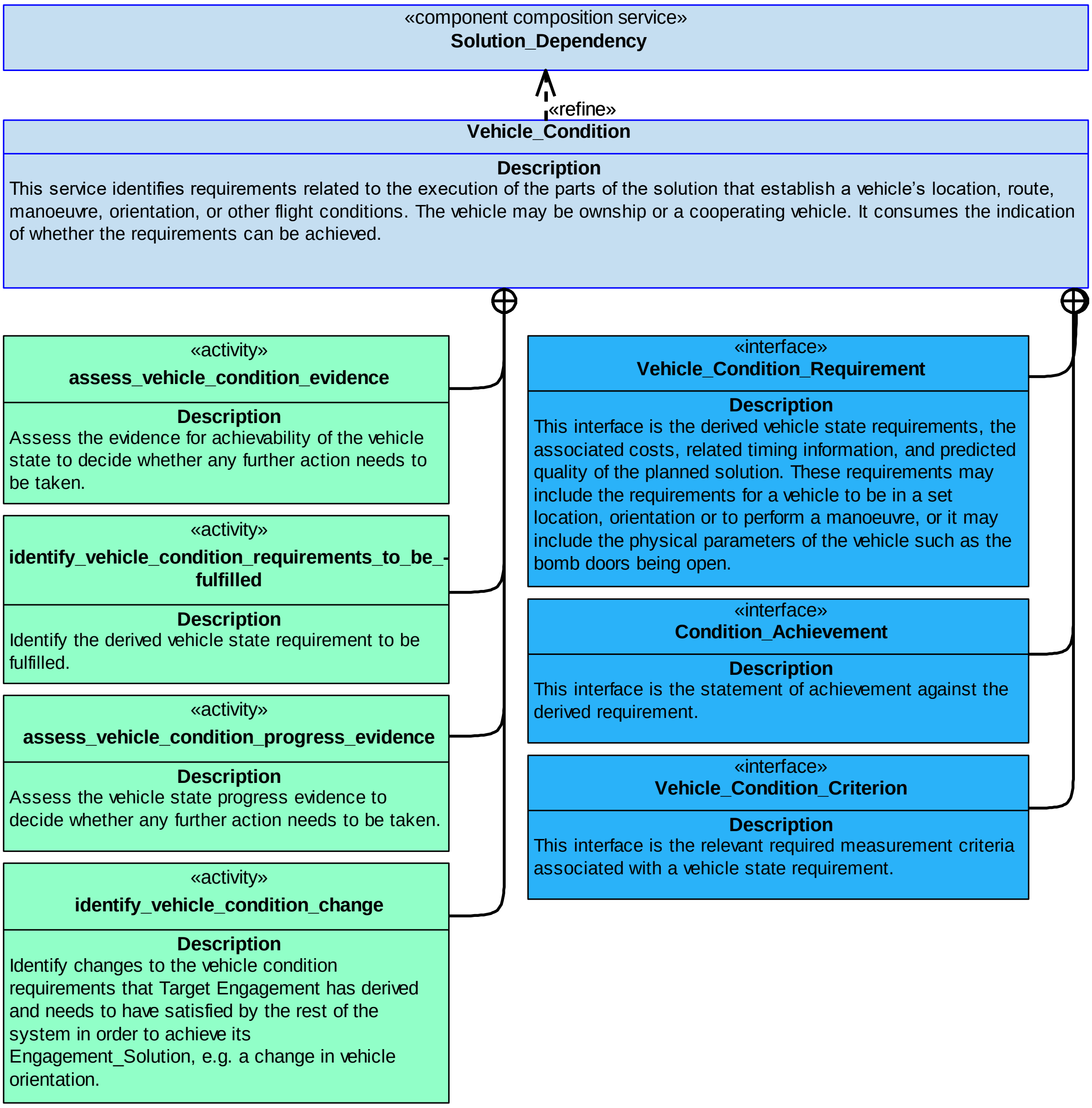

«component composition service»
Solution_Dependency
«refine»
Vehicle_Condition
Description
This service identifies requirements related to the execution of the parts of the solution that establish a vehicle's location, route, manoeuvre, orientation, or other flight conditions. The vehicle may be ownship or a cooperating vehicle. It consumes the indication of whether the requirements can be achieved.
«activity»
assess_vehicle_condition_evidence
Description
Assess the evidence for achievability of the vehicle state to decide whether any further action needs to be taken.
«activity»
identify_vehicle_condition_requirements_to_be_-fulfilled
Description
Identify the derived vehicle state requirement to be fulfilled.
«activity»
assess_vehicle_condition_progress_evidence
Description
Assess the vehicle state progress evidence to decide whether any further action needs to be taken.
«activity»
identify_vehicle_condition_change
Description
Identify changes to the vehicle condition requirements that Target Engagement has derived and needs to have satisfied by the rest of the system in order to achieve its Engagement_Solution, e.g. a change in vehicle orientation.
«interface»
Vehicle_Condition_Requirement
Description
This interface is the derived vehicle state requirements, the associated costs, related timing information, and predicted quality of the planned solution. These requirements may include the requirements for a vehicle to be in a set location, orientation or to perform a manoeuvre, or it may include the physical parameters of the vehicle such as the bomb doors being open.
«interface»
Condition_Achievement
Description
This interface is the statement of achievement against the derived requirement.
«interface»
Vehicle_Condition_Criterion
Description
This interface is the relevant required measurement criteria associated with a vehicle state requirement.

Figure 1063: Vehicle_Condition Service Policy

Vehicle_Condition

This service identifies requirements related to the execution of the parts of the solution that establish a vehicle's location, route, manoeuvre, orientation, or other flight conditions. The vehicle may be ownship or a cooperating vehicle. It consumes the indication of whether the requirements can be achieved.

Interfaces

Vehicle_Condition_Requirement

This interface is the derived vehicle state requirements, the associated costs, related timing information, and predicted quality of the planned solution. These requirements may include the requirements for a vehicle to be in a set location, orientation or to perform a manoeuvre, or it may include the physical parameters of the vehicle such as the bomb doors being open.

Attributes

specification	The definition of the derived requirement.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used and time taken.
predicted_quality	How well the proposed vehicle state solution is predicted to satisfy the requirement.

Vehicle_Condition_Criterion

This interface is the relevant required measurement criteria associated with a vehicle state requirement.

Attributes

property	The property to be measured, e.g. vehicle orientation.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Condition_Achievement

This interface is the statement of achievement against the derived requirement.

Activities

assess_vehicle_condition_evidence

Assess the evidence for achievability of the vehicle state to decide whether any further action needs to be taken.

assess_vehicle_condition_progress_evidence

Assess the vehicle state progress evidence to decide whether any further action needs to be taken.

identify_vehicle_condition_change

Identify changes to the vehicle condition requirements that Target Engagement has derived and needs to have satisfied by the rest of the system in order to achieve its Engagement_Solution, e.g. a change in vehicle orientation.

identify_vehicle_condition_requirements_to_be_fulfilled

Identify the derived vehicle state requirement to be fulfilled.

5.4.2.61.7.1.7 Target_Information

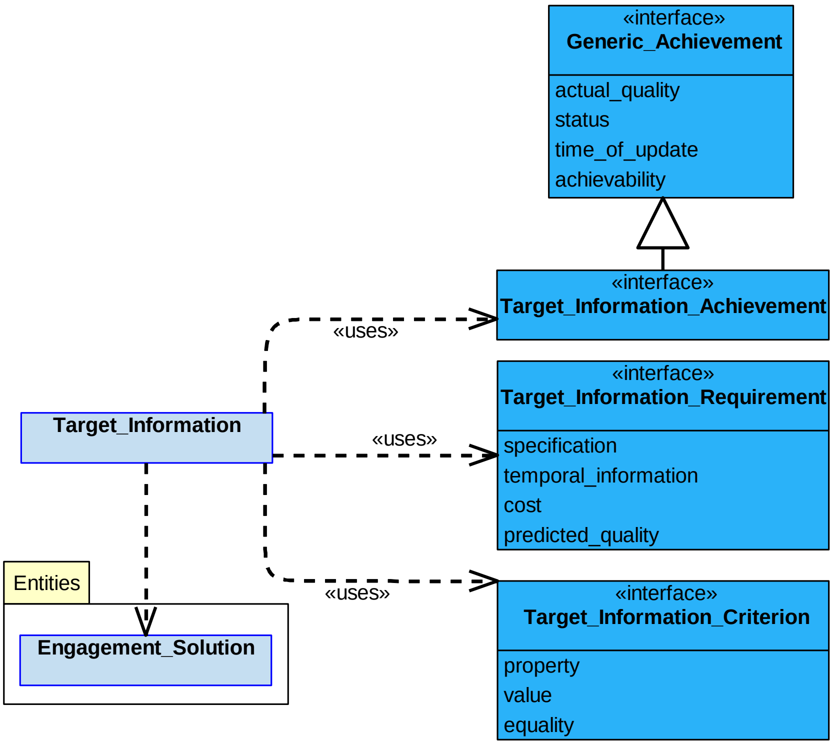


Figure 1064: Target_Information Service Definition

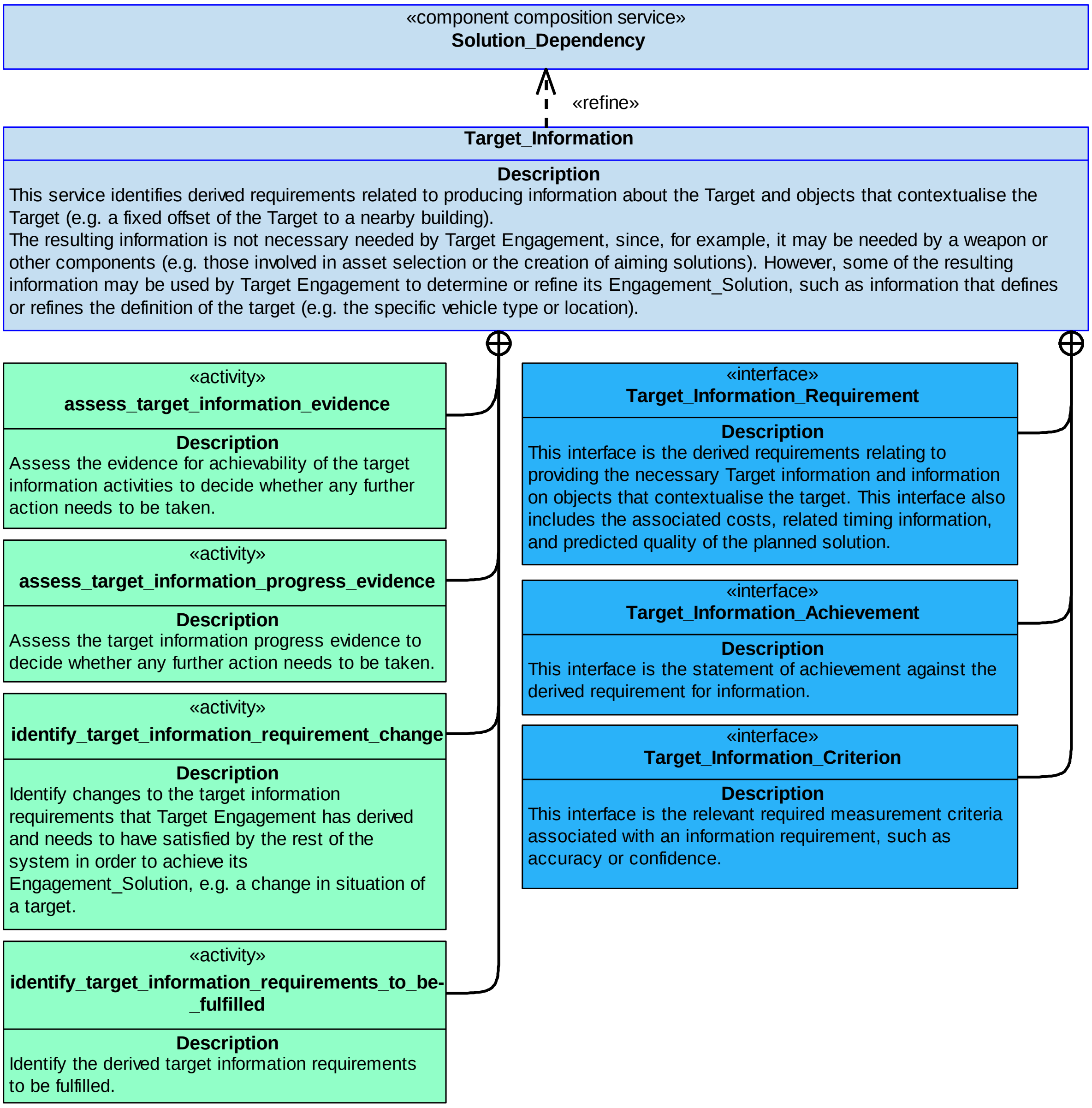

«component composition service»
Solution_Dependency
«refine»
Target_Information
Description
This service identifies derived requirements related to producing information about the Target and objects that contextualise the Target (e.g. a fixed offset of the Target to a nearby building).
The resulting information is not necessary needed by Target Engagement, since, for example, it may be needed by a weapon or other components (e.g. those involved in asset selection or the creation of aiming solutions). However, some of the resulting information may be used by Target Engagement to determine or refine its Engagement_Solution, such as information that defines or refines the definition of the target (e.g. the specific vehicle type or location).
«activity»
assess_target_information_evidence
Description
Assess the evidence for achievability of the target information activities to decide whether any further action needs to be taken.
«activity»
assess_target_information_progress_evidence
Description
Assess the target information progress evidence to decide whether any further action needs to be taken.
«activity»
identify_target_information_requirement_change
Description
Identify changes to the target information requirements that Target Engagement has derived and needs to have satisfied by the rest of the system in order to achieve its Engagement_Solution, e.g. a change in situation of a target.
«activity»
identify_target_information_requirements_to_be_fulfilled
Description
Identify the derived target information requirements to be fulfilled.
«interface»
Target_Information_Requirement
Description
This interface is the derived requirements relating to providing the necessary Target information and information on objects that contextualise the target. This interface also includes the associated costs, related timing information, and predicted quality of the planned solution.
«interface»
Target_Information_Achievement
Description
This interface is the statement of achievement against the derived requirement for information.
«interface»
Target_Information_Criterion
Description
This interface is the relevant required measurement criteria associated with an information requirement, such as accuracy or confidence.

Figure 1065: Target_Information Service Policy

Target_Information

This service identifies derived requirements related to producing information about the Target and objects that contextualise the Target (e.g. a fixed offset of the Target to a nearby building).

The resulting information is not necessary needed by Target Engagement, since, for example, it may be needed by a weapon or other components (e.g. those involved in asset selection or the creation of aiming solutions). However, some of the resulting information may be used by Target Engagement to determine or refine its Engagement_Solution, such as information that defines or refines the definition of the target (e.g. the specific vehicle type or location).

Interfaces

Target_Information_Requirement

This interface is the derived requirements relating to providing the necessary Target information and information on objects that contextualise the target. This interface also includes the associated costs, related timing information, and predicted quality of the planned solution.

Attributes

specification	The definition of the derived requirement.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the solution, for example: resources used and time taken.
predicted_quality	How well the proposed target information provision solution is predicted to satisfy the requirement.

Target_Information_Criterion

This interface is the relevant required measurement criteria associated with an information requirement, such as accuracy or confidence.

Attributes

property	The property to be measured, e.g. the estimated accuracy of the Target's location.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Target_Information_Achievement

This interface is the statement of achievement against the derived requirement for information.

Activities

assess_target_information_evidence

Assess the evidence for achievability of the target information activities to decide whether any further action needs to be taken.

assess_target_information_progress_evidence

Assess the target information progress evidence to decide whether any further action needs to be taken.

identify_target_information_requirement_change

Identify changes to the target information requirements that Target Engagement has derived and needs to have satisfied by the rest of the system in order to achieve its Engagement_Solution, e.g. a change in situation of a target.

identify_target_information_requirements_to_be_fulfilled

Identify the derived target information requirements to be fulfilled.

5.4.2.61.7.1.8 Engagement_Information

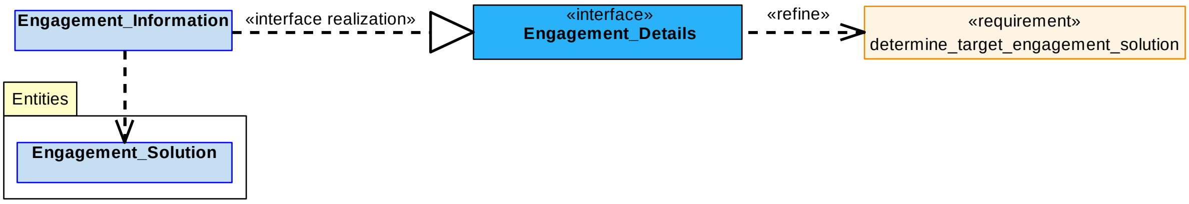


Figure 1066: Engagement_Information Service Definition

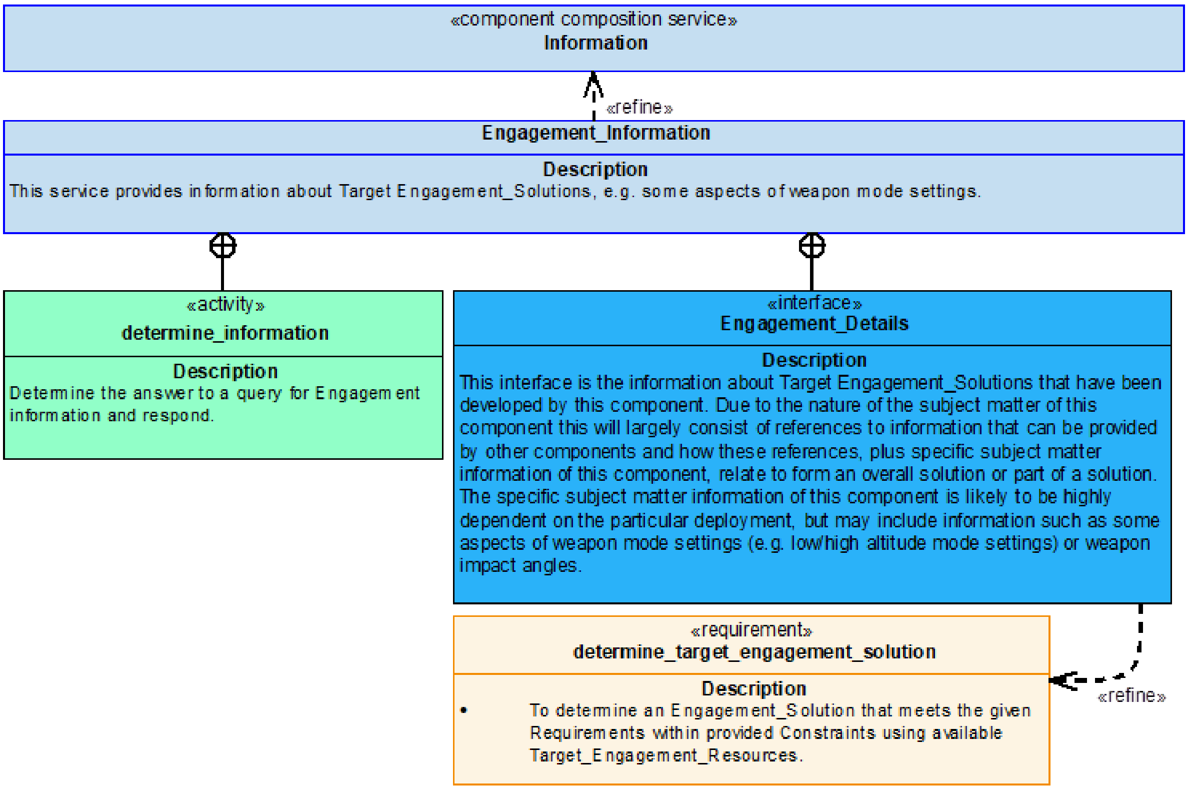

«component composition service»
Information
«refine»
Engagement_Information
Description
This service provides information about Target Engagement_Solutions, e.g. some aspects of weapon mode settings.
«activity»
determine_information
Description
Determine the answer to a query for Engagement information and respond.
«interface»
Engagement_Details
Description
This interface is the information about Target Engagement_Solutions that have been developed by this component. Due to the nature of the subject matter of this component this will largely consist of references to information that can be provided by other components and how these references, plus specific subject matter information of this component, relate to form an overall solution or part of a solution. The specific subject matter information of this component is likely to be highly dependent on the particular deployment, but may include information such as some aspects of weapon mode settings (e.g. low/high altitude mode settings) or weapon impact angles.
«requirement»
determine_target_engagement_solution
Description
• To determine an Engagement_Solution that meets the given Requirements within provided Constraints using available Target_Engagement_Resources.
«refine»

Figure 1067: Engagement_Information Service Policy

Engagement_Information

This service provides information about Target Engagement_Solutions, e.g. some aspects of weapon mode settings.

Interface

Engagement_Details

This interface is the information about Target Engagement_Solutions that have been developed by this component. Due to the nature of the subject matter of this component this will largely consist of references to information that can be provided by other components and how these references, plus specific subject matter information of this component, relate to form an overall solution or part of a solution.

The specific subject matter information of this component is likely to be highly dependent on the particular deployment, but may include information such as some aspects of weapon mode settings (e.g. low/high altitude mode settings) or weapon impact angles.

Activity

determine_information

Determine the answer to a query for Engagement information and respond.

5.4.2.61.7.1.9 Supporting_Information

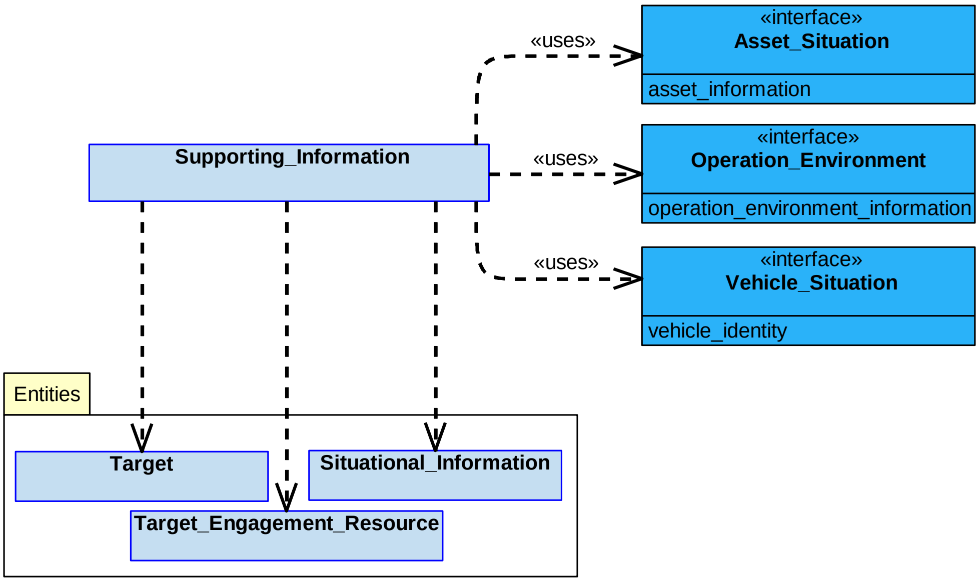


Figure 1068: Supporting_Information Service Definition

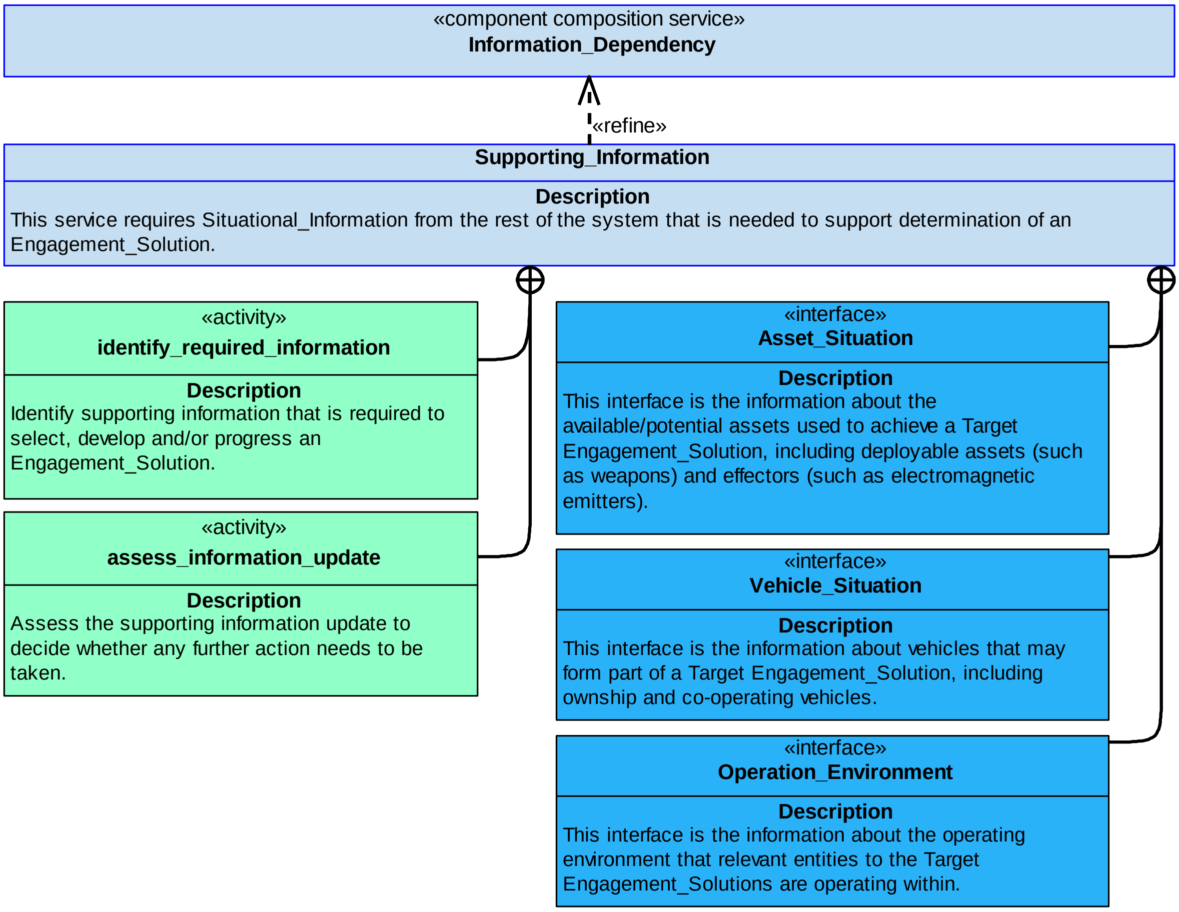

«component composition service»
Information_Dependency
«refine»
Supporting_Information
Description
This service requires Situational_Information from the rest of the system that is needed to support determination of an Engagement_Solution.
«activity»
identify_required_information
Description
Identify supporting information that is required to select, develop and/or progress an Engagement_Solution.
«activity»
assess_information_update
Description
Assess the supporting information update to decide whether any further action needs to be taken.
«interface»
Asset_Situation
Description
This interface is the information about the available/potential assets used to achieve a Target Engagement_Solution, including deployable assets (such as weapons) and effectors (such as electromagnetic emitters).
«interface»
Vehicle_Situation
Description
This interface is the information about vehicles that may form part of a Target Engagement_Solution, including ownship and co-operating vehicles.
«interface»
Operation_Environment
Description
This interface is the information about the operating environment that relevant entities to the Target Engagement_Solutions are operating within.

Figure 1069: Supporting_Information Service Policy

Supporting_Information

This service requires Situational_Information from the rest of the system that is needed to support determination of an Engagement_Solution.

Interfaces

Asset_Situation

This interface is the information about the available/potential assets used to achieve a Target Engagement_Solution, including deployable assets (such as weapons) and effectors (such as electromagnetic emitters).

Attribute

asset_information	Information about assets, e.g. ID, type (as typed by the capability that they provide to this component), quantity (for clusters of assists such as gun rounds), location, and operating state.

Operation_Environment

This interface is the information about the operating environment that relevant entities to the Target Engagement_Solutions are operating within.

Attribute

operation_environment_information	Information about the operating environment. This could include information such as time of day, environment type, weather conditions, or theatre context information.

Vehicle_Situation

This interface is the information about vehicles that may form part of a Target Engagement_Solution, including ownship and co-operating vehicles.

Attributes

vehicle_identity	The type of vehicle and specific ID.
vehicle_position	The vehicle location, velocity and orientation.
vehicle_state	Specific information about the vehicle's state and capability, such as its current weapons inventory.

Activities

identify_required_information

Identify supporting information that is required to select, develop and/or progress an Engagement_Solution.

assess_information_update

Assess the supporting information update to decide whether any further action needs to be taken.

5.4.2.61.7.1.10 Constraint

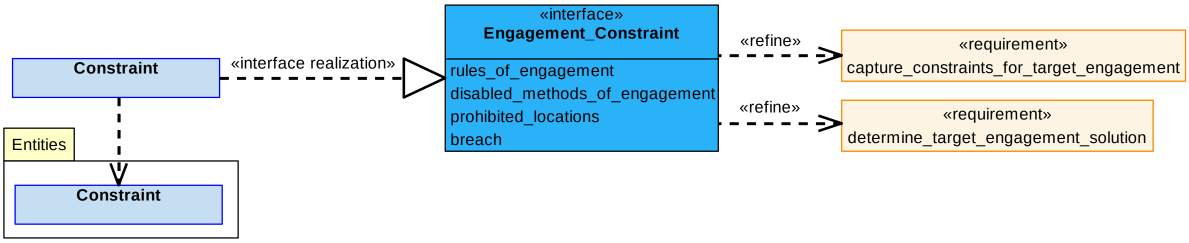


Figure 1070: Constraint Service Definition

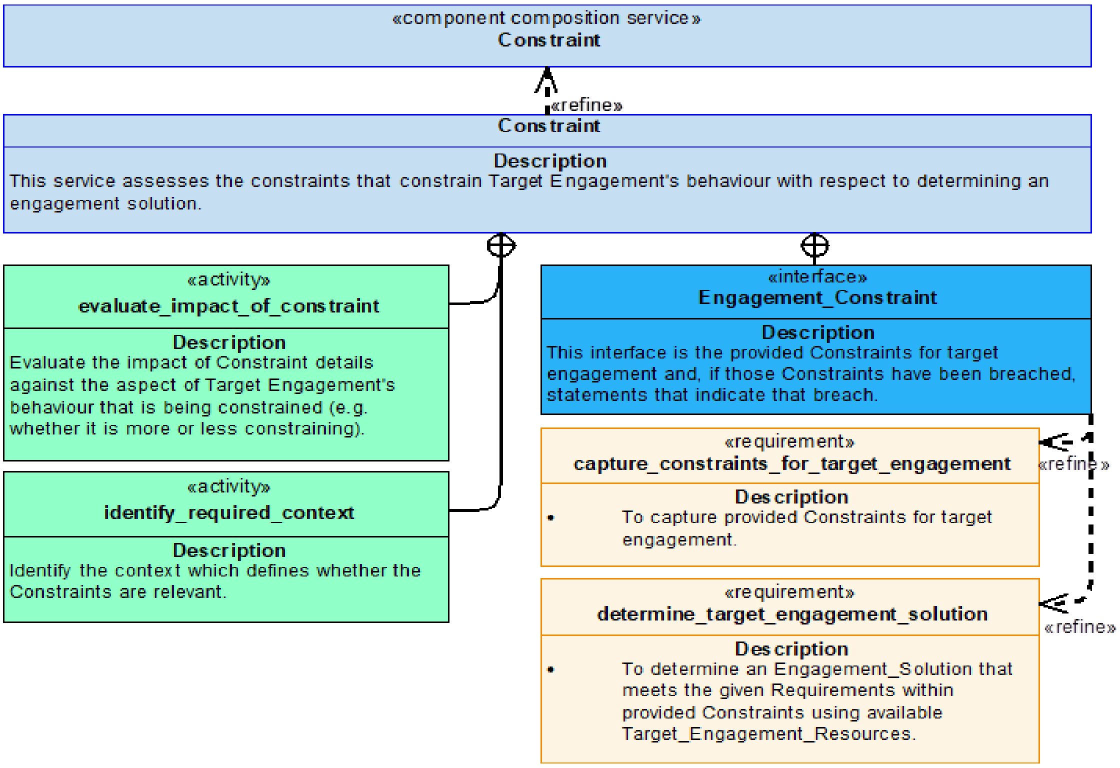

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the constraints that constrain Target Engagement's behaviour with respect to determining an engagement solution.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of Target Engagement's behaviour that is being constrained (e.g. whether it is more or less constraining).
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Engagement_Constraint
Description
This interface is the provided Constraints for target engagement and, if those Constraints have been breached, statements that indicate that breach.
«requirement»
capture_constraints_for_target_engagement
Description
To capture provided Constraints for target engagement.
«refine»
«requirement»
determine_target_engagement_solution
Description
To determine an Engagement_Solution that meets the given Requirements within provided Constraints using available Target_Engagement_Resources.
«refine»

Figure 1071: Constraint Service Policy

Constraint

This service assesses the constraints that constrain Target Engagement's behaviour with respect to determining an engagement solution.

Interface

Engagement_Constraint

This interface is the provided Constraints for target engagement and, if those Constraints have been breached, statements that indicate that breach.

Attributes

rules_of_engagement	Rules of engagement as relevant to the abstraction of target engagement (such as the minimum level of target identification needed before a target can be attacked).
disabled_methods_of_engagement	Methods of engagement (e.g. destructive effects) that are currently not allowed for particular target types in particular contexts.
prohibited_locations	References to area or zones where activity is prohibited (e.g. weapon no fly or no impact zones).
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of Target Engagement's behaviour that is being constrained (e.g. whether it is more or less constraining).

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.61.7.1.11 Capability

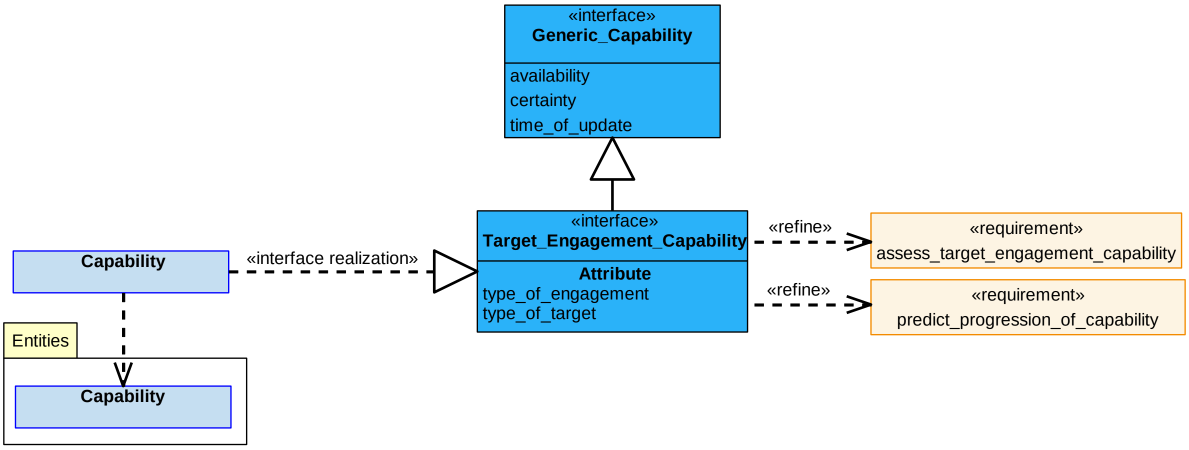


Figure 1072: Capability Service Definition

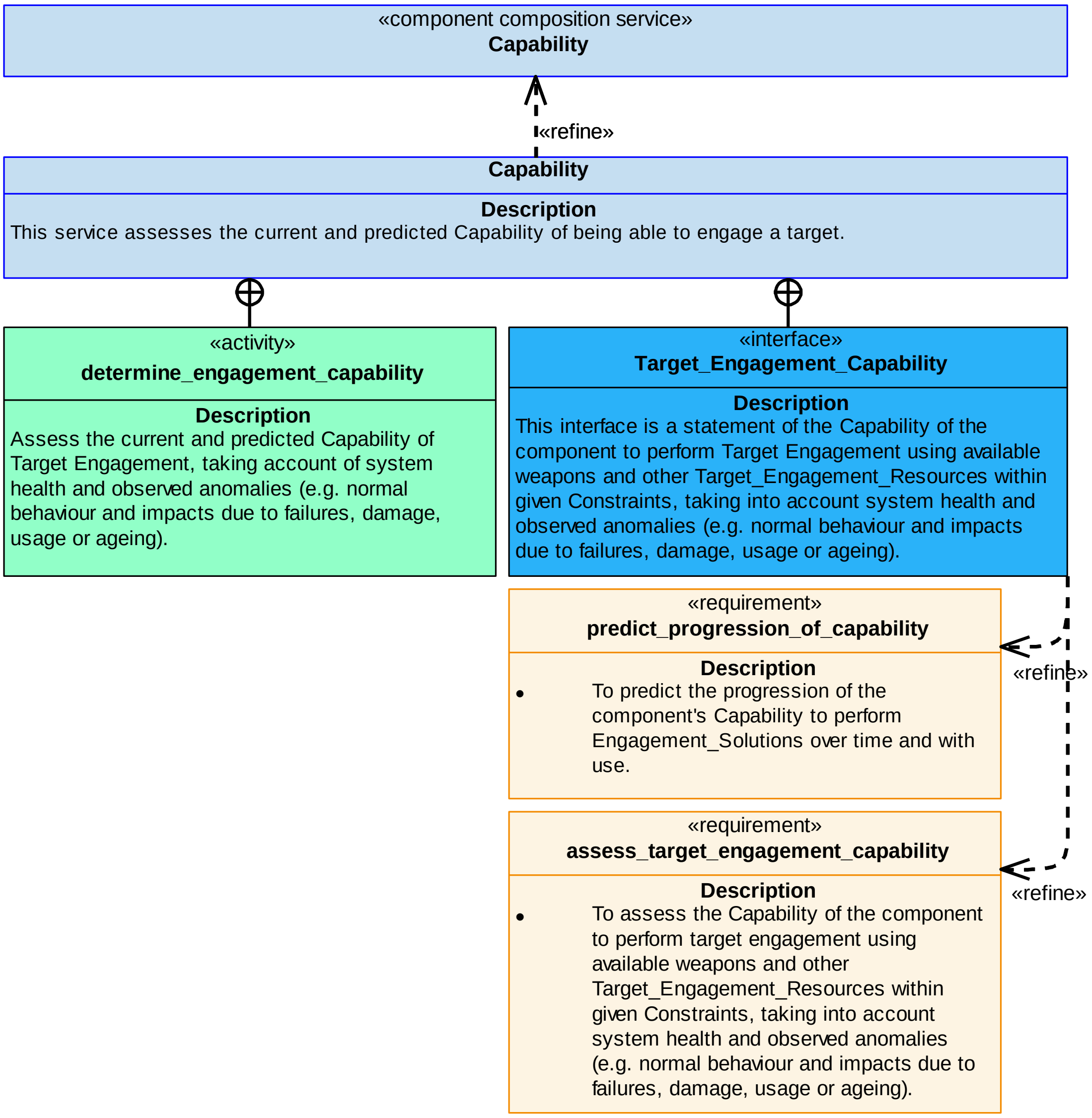

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability of being able to engage a target.
«activity»
determine_engagement_capability
Description
Assess the current and predicted Capability of Target Engagement, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Target_Engagement_Capability
Description
This interface is a statement of the Capability of the component to perform Target Engagement using available weapons and other Target_Engagement_Resources within given Constraints, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«requirement»
predict_progression_of_capability
Description
To predict the progression of the component's Capability to perform Engagement_Solutions over time and with use.
«refine»
«requirement»
assess_target_engagement_capability
Description
To assess the Capability of the component to perform target engagement using available weapons and other Target_Engagement_Resources within given Constraints, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 1073: Capability Service Policy

Capability

This service assesses the current and predicted Capability of being able to engage a target.

<u>Interface</u>

Target_Engagement_Capability

This interface is a statement of the Capability of the component to perform Target Engagement using available weapons and other Target_Engagement_Resources within given Constraints, taking into account system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

<u>Attributes</u>

type_of_engagement	A specific type of engagement (e.g. destroy target, disrupt communications, provoke activity, drop supplies, or position a remote sensor).

type_of_target	A type of target (e.g. a location (including an area or zone), a type of object, a specific object, or a cluster of objects).

Activity

determine_engagement_capability

Assess the current and predicted Capability of Target Engagement, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.61.7.1.12 Capability_Evidence

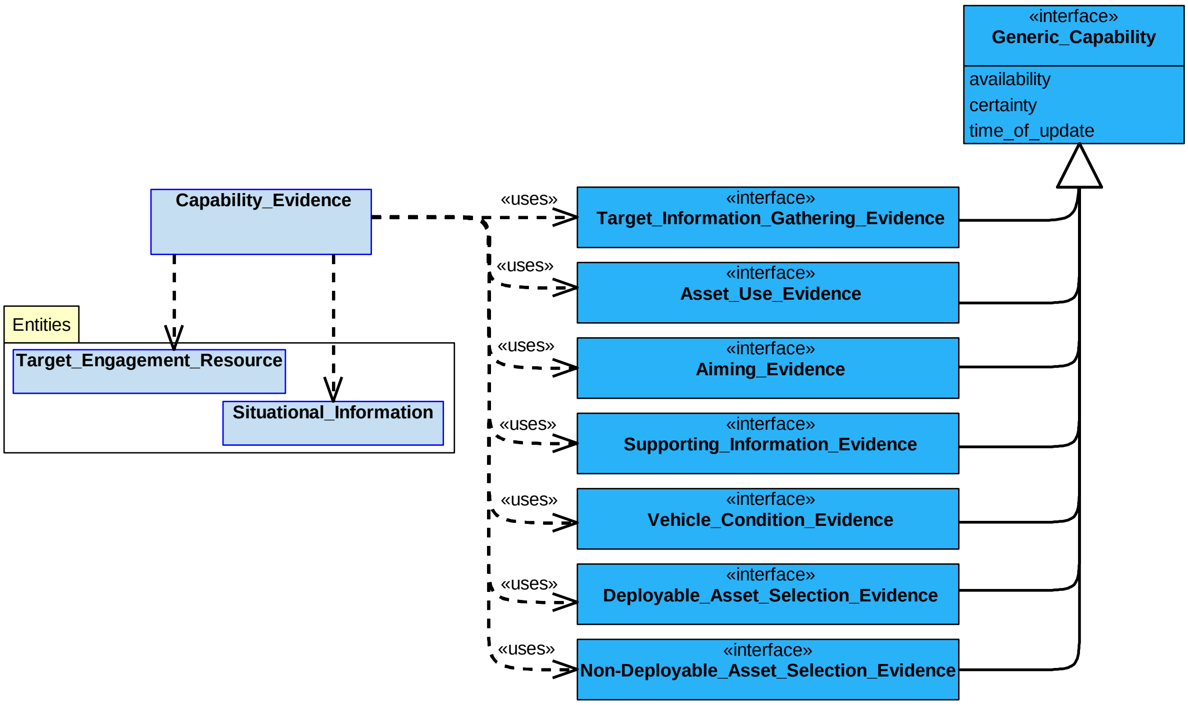


Figure 1074: Capability_Evidence Service Definition

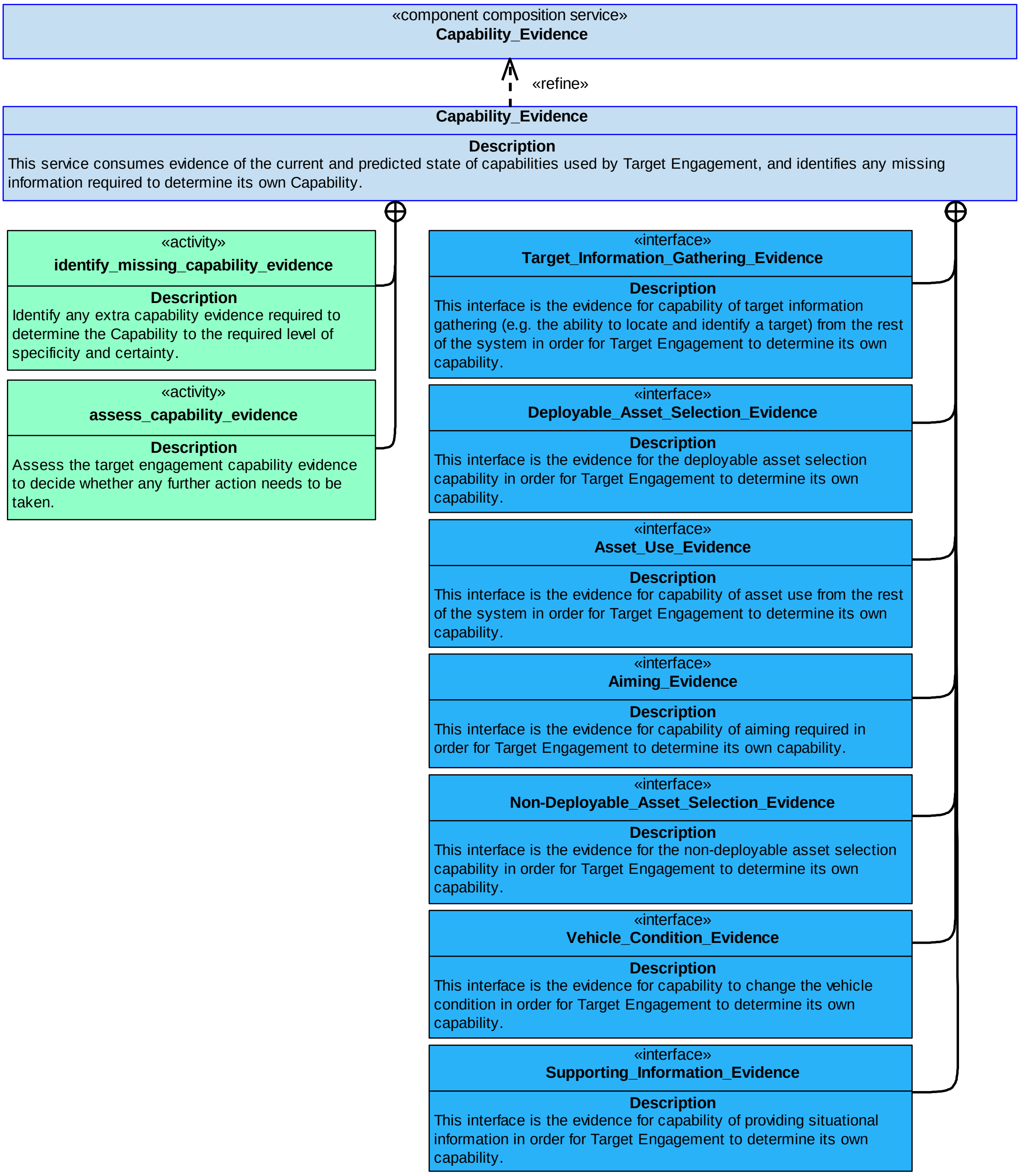

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes evidence of the current and predicted state of capabilities used by Target Engagement, and identifies any missing information required to determine its own Capability.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«activity»
assess_capability_evidence
Description
Assess the target engagement capability evidence to decide whether any further action needs to be taken.
«interface»
Target_Information_Gathering_Evidence
Description
This interface is the evidence for capability of target information gathering (e.g. the ability to locate and identify a target) from the rest of the system in order for Target Engagement to determine its own capability.
«interface»
Deployable_Asset_Selection_Evidence
Description
This interface is the evidence for the deployable asset selection capability in order for Target Engagement to determine its own capability.
«interface»
Asset_Use_Evidence
Description
This interface is the evidence for capability of asset use from the rest of the system in order for Target Engagement to determine its own capability.
«interface»
Aiming_Evidence
Description
This interface is the evidence for capability of aiming required in order for Target Engagement to determine its own capability.
«interface»
Non-Deployable_Asset_Selection_Evidence
Description
This interface is the evidence for the non-deployable asset selection capability in order for Target Engagement to determine its own capability.
«interface»
Vehicle_Condition_Evidence
Description
This interface is the evidence for capability to change the vehicle condition in order for Target Engagement to determine its own capability.
«interface»
Supporting_Information_Evidence
Description
This interface is the evidence for capability of providing situational information in order for Target Engagement to determine its own capability.

Figure 1075: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence of the current and predicted state of capabilities used by Target Engagement, and identifies any missing information required to determine its own Capability.

Interfaces

Non-Deployable_Asset_Selection_Evidence

This interface is the evidence for the non-deployable asset selection capability in order for Target Engagement to determine its own capability.

Deployable_Asset_Selection_Evidence

This interface is the evidence for the deployable asset selection capability in order for Target Engagement to determine its own capability.

Asset_Use_Evidence

This interface is the evidence for capability of asset use from the rest of the system in order for Target Engagement to determine its own capability.

Aiming_Evidence

This interface is the evidence for capability of aiming required in order for Target Engagement to determine its own capability.

Vehicle_Condition_Evidence

This interface is the evidence for capability to change the vehicle condition in order for Target Engagement to determine its own capability.

Target_Information_Gathering_Evidence

This interface is the evidence for capability of target information gathering (e.g. the ability to locate and identify a target) from the rest of the system in order for Target Engagement to determine its own capability.

Supporting_Information_Evidence

This interface is the evidence for capability of providing situational information in order for Target Engagement to determine its own capability.

Activities

assess_capability_evidence

Assess the target engagement capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.61.7.2 Service Dependencies

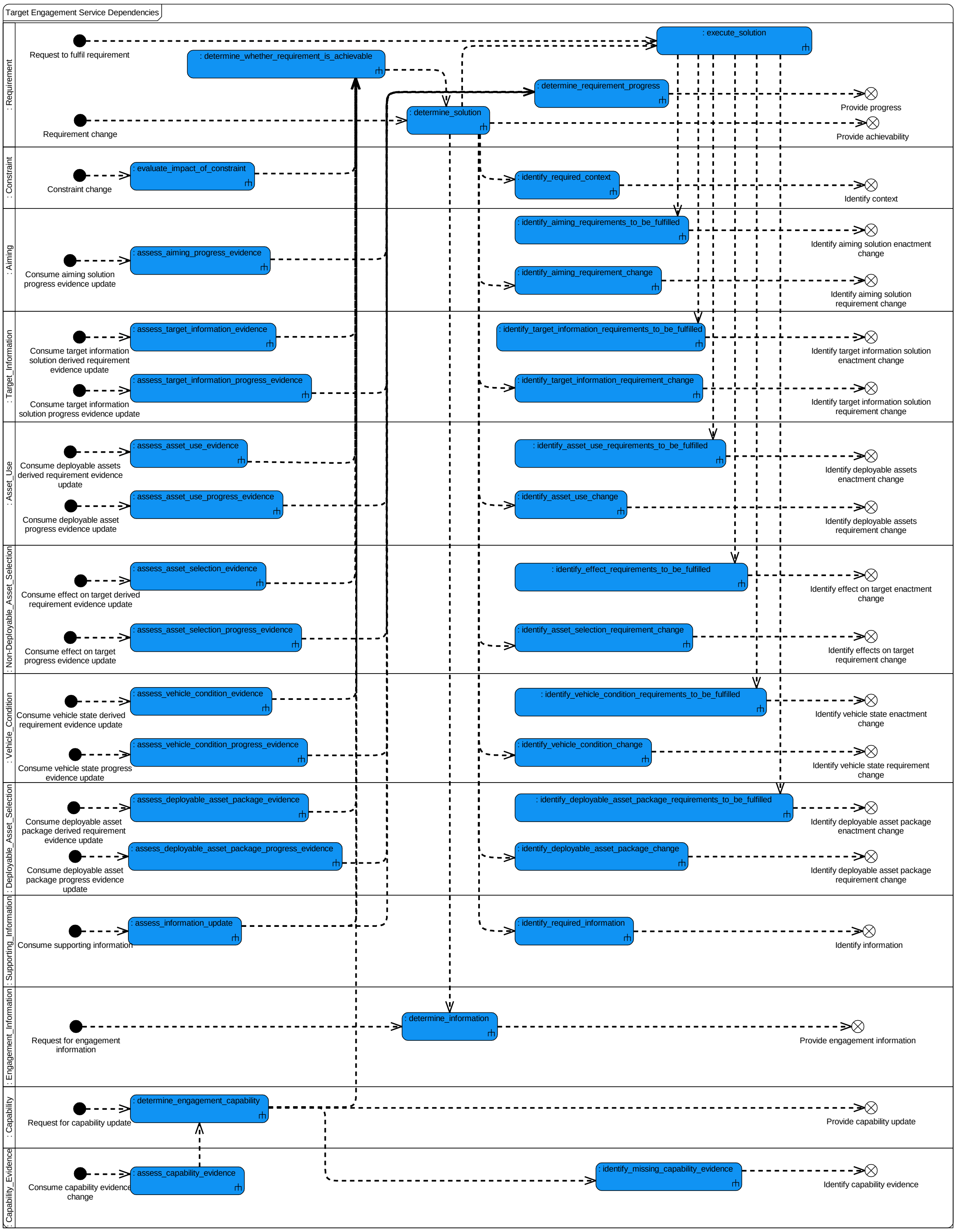

Target Engagement Service Dependencies
: Requirement
Request to fulfil requirement
: determine_whether_requirement_is_achievable
: execute_solution
: determine_requirement_progress
determine_solution
Requirement change
Provide progress
Provide achievability
: Constraint
: evaluate_impact_of_constraint
Constraint change
: identify_required_context
Identify context
: Aiming
: identify_aiming_requirements_to_be_fulfilled
: assess_aiming_progress_evidence
Consume aiming solution progress evidence update
: identify_aiming_requirement_change
Identify aiming solution enactment change
Identify aiming solution requirement change
: Target_Information
: assess_target_information_evidence
Consume target information solution derived requirement evidence update
: identify_target_information_requirements_to_be_fulfilled
Identify target information solution enactment change
: assess_target_information_progress_evidence
Consume target information solution progress evidence update
: identify_target_information_requirement_change
Identify target information solution requirement change
: Asset_Use
: assess_asset_use_evidence
Consume deployable assets derived requirement evidence update
: identify_asset_use_requirements_to_be_fulfilled
Identify deployable assets enactment change
: assess_asset_use_progress_evidence
Consume deployable asset progress evidence update
: identify_asset_use_change
Identify deployable assets requirement change
: Non-Deployable_Asset_Selection
: assess_asset_selection_evidence
Consume effect on target derived requirement evidence update
: identify_effect_requirements_to_be_fulfilled
Identify effect on target enactment change
: assess_asset_selection_progress_evidence
Consume effect on target progress evidence update
: identify_asset_selection_requirement_change
Identify effects on target requirement change
: Vehicle_Condition
: assess_vehicle_condition_evidence
Consume vehicle state derived requirement evidence update
: identify_vehicle_condition_requirements_to_be_fulfilled
Identify vehicle state enactment change
: assess_vehicle_condition_progress_evidence
Consume vehicle state progress evidence update
: identify_vehicle_condition_change
Identify vehicle state requirement change
: Deployable_Asset_Selection
: assess_deployable_asset_package_evidence
Consume deployable asset package derived requirement evidence update
: identify_deployable_asset_package_requirements_to_be_fulfilled
Identify deployable asset package enactment change
: assess_deployable_asset_package_progress_evidence
Consume deployable asset package progress evidence update
: identify_deployable_asset_package_change
Identify deployable asset package requirement change
: Supporting_Information
: assess_information_update
Consume supporting information
: identify_required_information
Identify information
: Engagement_Information
: determine_information
Request for engagement information
Provide engagement information
: Capability
: determine_engagement_capability
Request for capability update
Provide capability update
: Capability_Evidence
: assess_capability_evidence
Consume capability evidence change
: identify_missing_capability_evidence
Identify capability evidence

Figure 1076: Target Engagement Service Dependencies

5.4.2.62 Tasks

5.4.2.62.1 Role

The role of Tasks is to coordinate the activities required to respond to demands placed on an aircraft system.

5.4.2.62.2 Overview

Control Architecture

Tasks is the only component in the Task Layer, as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

The need for a task to be carried out is either identified by Tasks, such as when it identifies the need to respond to an unplanned situation, or provided to Tasks as a requirement. Tasks will determine a solution (a Sequence of Derived_Needs), which is a coordinated set of Actions, Information_Needs or Behavioural_Constraints, taking into account any System_Constraints. Tasks will manage the execution of the solution and, if this solution becomes unachievable (for example due to a hardware failure), Tasks will re-plan.

Examples of Use

Tasks will be required where the management and use of a Composite_Capability to respond to a System_Stimulus is required. For example:

- Where an aircraft system has been tasked with the search of a specific area for hostile assets, requiring the system to determine a routing solution to aviate to the area of interest, then to provide the search pattern and then the initiation of the sensing action.
- Where an aircraft system is tasked to patrol a region, requiring a suitable route to aviate be made available, such as a combat air patrol, while requiring suitable surveillance capabilities to be available at the necessary times along the patrol route.

5.4.2.62.3 Service Summary

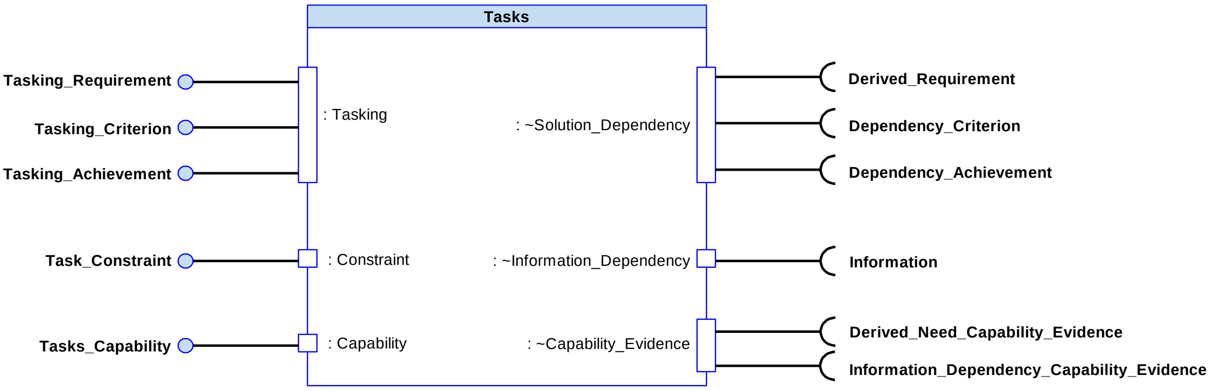


Figure 1077: Tasks Service Summary

5.4.2.62.4 Responsibilities

capture_task_requirements

- To capture given Tasking requirements (e.g. task type, timing, importance of success, or risk profile).

capture_measurement_criteria

- To capture given optimisation criteria for task solutions (e.g. quality, risk, or robustness).

capture_system_constraints

- To capture given System_Constraints (e.g. budgets or operating restrictions).

identify_whether_requirement_remains_achievable

- To identify whether a Tasking requirement is still achievable given current or predicted Composite_Capability and System_Constraints.

determine_implementation_solution

- To determine a Sequence of Derived_Needs against given optimisation criteria.

satisfy_dependencies_between_derived_needs

- To satisfy the dependencies between Derived_Needs by arranging their Sequence.

coordinate_solution_enactment

- To coordinate the enactment of a response to a System_Stimulus via the execution of Derived_Needs in a Sequence.

identify_progress_of_solution

- To identify the progress of a Sequence of Derived_Needs.

evaluate_solution_quality

- To evaluate the quality of a Sequence of Derived_Needs against given optimisation criteria.

determine_implementation_solution_cost

- To determine the cost of implementing a Sequence of Derived_Needs using the available composite capabilities.

capture_contingency_definitions

- To capture given contingency definitions (e.g. the rules that constitute a Contingency_Situation and responses to that Contingency_Situation).

assess_task_capability

- To assess the ability to respond to a System_Stimulus based upon available composite capabilities and observed anomalies.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the assessment of the ability to respond to a System_Stimulus.

predict_capability_progression

- To predict the progression of the ability to respond to a System_Stimulus over time and with use.

5.4.2.62.5 Subject Matter Semantics

The subject matter of Tasks is the use of a Composite_Capability or capabilities to meet the high level needs of an aircraft system. This includes knowledge of how to balance conflicting drivers for different solution qualities.

Exclusions:

The direct actioning of low level system capabilities and resources (i.e. the capabilities making up a Composite_Capability).

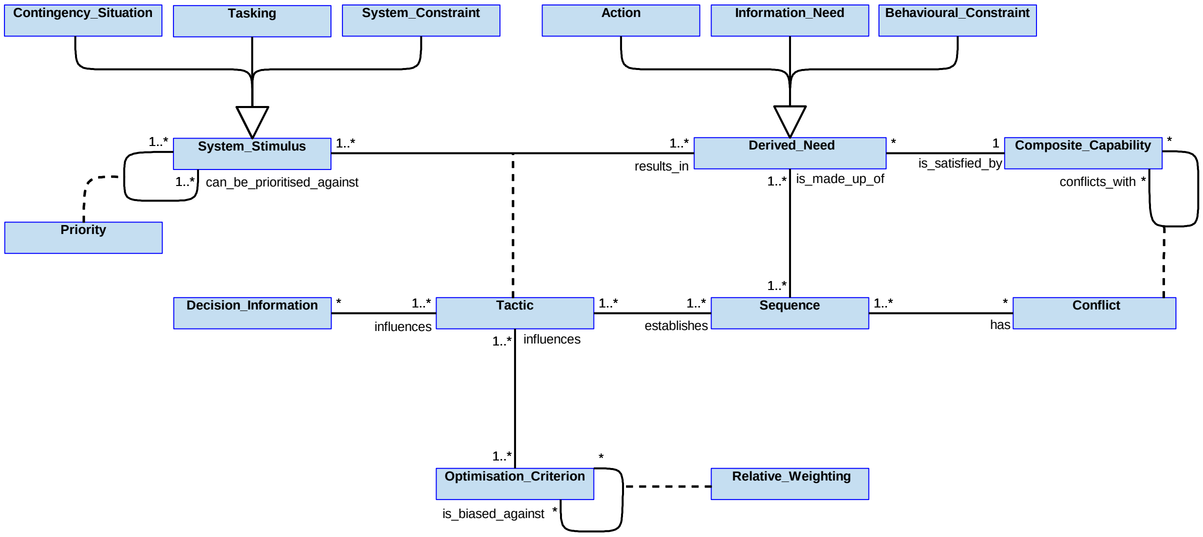


Figure 1078: Tasks Semantics

5.4.2.62.5.1 Entities

Action

A discrete action that requires the specific use of a Composite_Capability.

Behavioural_Constraint

A derived restriction on a Composite_Capability or capabilities of an aircraft system.

Composite_Capability

The ability of a composite element of the aircraft system to meet aircraft system needs, either currently or in the future.

Conflict

A contradiction preventing or limiting the use of a Composite_Capability based on either the demand on that capability or the use of another capability (e.g. the ability to engage a target is diminished when refuelling is taking place).

Contingency_Situation

An unplanned event or situation requiring an aircraft system response (e.g. an unacceptably high risk hostile engagement that needs to be mitigated, or a potentially catastrophic failure requiring diversion to an emergency airfield).

Decision_Information

Information that supports decision making. This may be information on the way an aircraft system's needs can be met (e.g. information about available composite capabilities) or the context of the aircraft system's use (e.g. information about the situation around the aircraft system). This could include factors about the state of the aircraft system, the environment external to the aircraft system, or the approval to act.

Derived_Need

A demand required to be placed on a Composite_Capability of an aircraft system to achieve a desired outcome. The Derived_Need can affect both the current and future behaviour.

Optimisation_Criterion

A level of importance and/or acceptable limits (positive or negative) of a variable property of a solution in satisfying aircraft system needs. For example properties that fall under the categories of risk (including survivability), retained or established redundancy/contingency, and performance/effectiveness.

Priority

The relative level of importance of a System_Stimulus when compared to another.

Information_Need

A need for information to be generated.

Relative_Weighting

The level of bias of one Optimisation_Criterion in relation to another.

Sequence

The relative order and timing of activities.

System_Constraint

A given restriction on the behaviour of an aircraft system.

System_Stimulus

An external demand or occurrence that affects the behaviour of an aircraft system. It can be planned (e.g. tasking to meet a mission objective) or unplanned (e.g. discovery of a threat indicating a hazardous situation that the aircraft system will need to be removed from) and can affect the current or predicted future behaviour of the system.

Tactic

A strategy to determine Derived_Needs in response to a System_Stimulus.

Tasking

A given requirement that specifies a goal to be achieved using the capability of an aircraft system.

5.4.2.62.6 Design Rationale

5.4.2.62.6.1 Assumptions

- Types of tasks may vary occasionally, but are unlikely to vary during a mission.
- Types of System_Constraint may vary occasionally, but are unlikely to vary during a mission.
- New types of optimisation criteria may be defined from time to time, but not within a mission.
- Information_Needs may be achieved in response to Actions.

5.4.2.62.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Tasks:

- Component Extensions - See below.
- Constraint Management - Tasks must generate Derived_Needs within the bounds of the given System_Constraints.
- Data Driving - Types of Tasking, types of System_Constraint, Tactics and types of optimisation criteria could be data-driven.
- Dependency Management - Describes the process for achieving Derived_Needs that are generated by the component.
- Multi-Vehicle Coordination - Different Tasks components operating within different aircraft systems will need to cooperate and coordinate with each other. A Tasks component on one air vehicle may also be ultimately responsible for other air vehicles, such as deployable assets, within the same aircraft system, or where it has been delegated control over specific aspects of the capability of other aircraft systems.
- Resource Management - Where resources can be shared or used to achieve Derived_Needs, Tasks may need to resolve Conflicts to achieve those Derived_Needs in line with mission priorities.

Extensions

- The breakdown of a particular type of task into Derived_Needs can be achieved by the application of specialised tactics capabilities (which may involve a single Tactic or a set of Tactics). These could be implemented through the use of extension components to aid development and to allow additional, specialised, or improved tactics to be developed without impacting other tactics or the parent Tasks component. The Component Extensions PYRAMID concept identifies example Tactics Extensions of this aspect of the Tasks component, each corresponding to a particular type of task. However, additional or alternative extensions may be developed as part of a deployment, potentially by different developers.

Exploitation Considerations

- In a multi-node system, there may be an instance of Tasks on each node. This could include separate instances of Tasks on separately developed units, such as weapons. Each node would be responsible for carrying out its own tasks, as allocated by the Objectives component, with coordination between them as required.

5.4.2.62.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- As shown on the Weather IV this component determines the mitigation strategy to avoid unsuitable weather and so failure of this component could result in flight in weather conditions that exceed the capability of the air vehicle. Flight in weather conditions that exceed the capability of the air vehicle would result in uncontrolled flight and an uncontrolled crash. This would result in loss of the air vehicle and potentially fatalities.
- No credit has been assumed for the crew controlling the air vehicle directly observing the local weather or its effect on the air vehicle. For Exploiting Programmes where this is possible DAL requirements may be less onerous.
- Where instances of this component contribute to hazards that are less severe (or are normally excluded from safety analysis such as direct enemy attack), then the Exploiting Platform may require a less onerous DAL.

5.4.2.62.6.4 Security Considerations

The indicative security classification is SNEO.

This component receives tasking orders, and contains the Tactics, techniques and procedures (potentially as extensions) required to plan tasks and coordinate the Derived_Needs required to achieve them. The classification of tasks will vary, from those involved with transit through civil airspace during peace time to those relating to the mission objectives. Where concerned with the latter, the indicative security classification is considered SNEO within the context of the security analysis. However, it is possible there will be declassified instances of the component in different security domains to deal with lower classifications of tasks, and communication between instances in these domains is considered probable.

Due to the central role in conducting mission activities, enhanced measures to ensure ongoing confidentiality, integrity, availability and authenticity are considered appropriate.

The component is expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data** of authorisation success/failures, access and changes to high-value data, etc. for later forensic examination.
- **Maintaining Audit Records** to support non-repudiation of decision-making and events performed in the fulfilment of a task. This component is at the core of the tasks being performed during the mission, and will be central to the mission audit process.

- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **Supporting Secure Remote Operation** through the secure planning of tasks that can be performed autonomously.
- **System Status and Monitoring** through the monitoring of the tasks set and progress against them. Unexpected deviation from the task (e.g. to perform an unsafe action) may indicate a cyber adversary has infiltrated the control architecture.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.62.7 Services

5.4.2.62.7.1 Service Definitions

5.4.2.62.7.1.1 Tasking

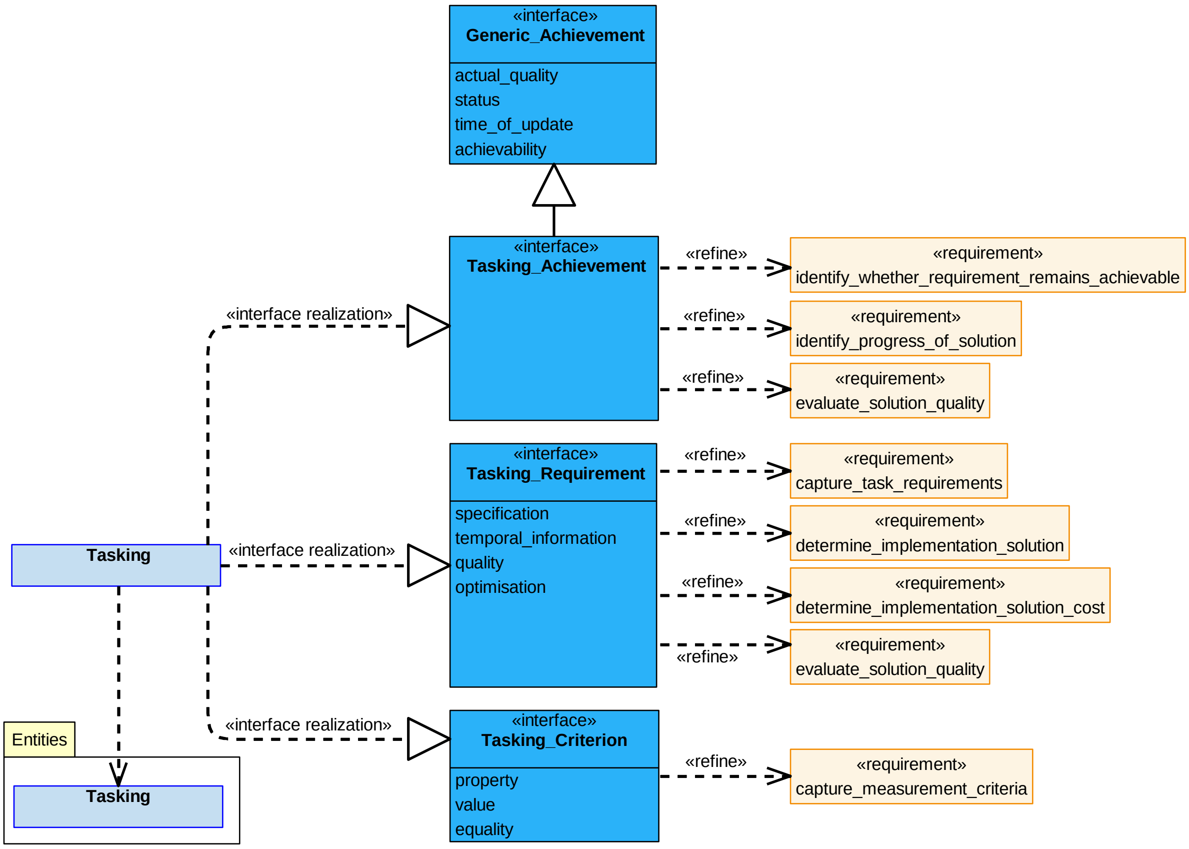


Figure 1079: Tasking Service Definition

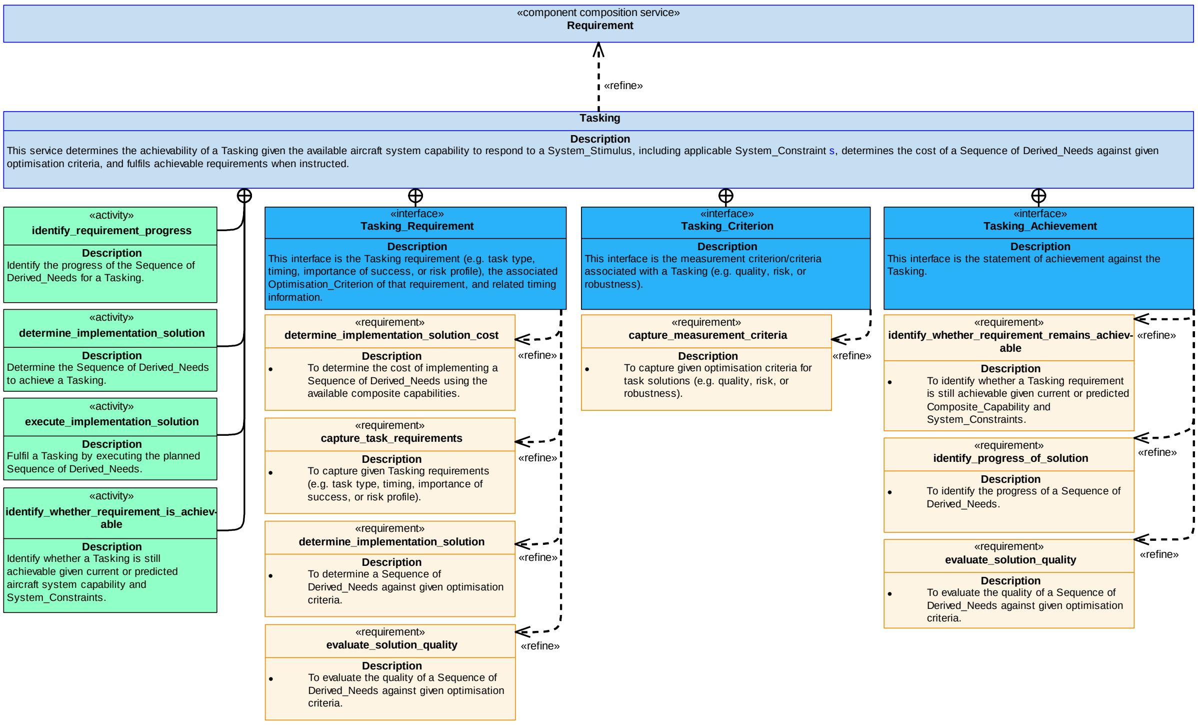

«component composition service»
Requirement
«refine»
Tasking
Description
This service determines the achievability of a Tasking given the available aircraft system capability to respond to a System_Stimulus, including applicable System_Constraint s, determines the cost of a Sequence of Derived_Needs against given optimisation criteria, and fulfils achievable requirements when instructed.
«activity»
identify_requirement_progress
Description
Identify the progress of the Sequence of Derived_Needs for a Tasking.
«activity»
determine_implementation_solution
Description
Determine the Sequence of Derived_Needs to achieve a Tasking.
«activity»
execute_implementation_solution
Description
Fulfil a Tasking by executing the planned Sequence of Derived_Needs.
«activity»
identify_whether_requirement_is_achievable
Description
Identify whether a Tasking is still achievable given current or predicted aircraft system capability and System_Constraints.
«interface»
Tasking_Requirement
Description
This interface is the Tasking requirement (e.g. task type, timing, importance of success, or risk profile), the associated Optimisation_Criterion of that requirement, and related timing information.
«requirement»
determine_implementation_solution_cost
Description
To determine the cost of implementing a Sequence of Derived_Needs using the available composite capabilities.
«refine»
«requirement»
capture_task_requirements
Description
To capture given Tasking requirements (e.g. task type, timing, importance of success, or risk profile).
«refine»
«requirement»
determine_implementation_solution
Description
To determine a Sequence of Derived_Needs against given optimisation criteria.
«refine»
«requirement»
evaluate_solution_quality
Description
To evaluate the quality of a Sequence of Derived_Needs against given optimisation criteria.
«refine»
«interface»
Tasking_Criterion
Description
This interface is the measurement criterion/criteria associated with a Tasking (e.g. quality, risk, or robustness).
«requirement»
capture_measurement_criteria
Description
To capture given optimisation criteria for task solutions (e.g. quality, risk, or robustness).
«refine»
«interface»
Tasking_Achievement
Description
This interface is the statement of achievement against the Tasking.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Tasking requirement is still achievable given current or predicted Composite_Capability and System_Constraints.
«refine»
«requirement»
identify_progress_of_solution
Description
To identify the progress of a Sequence of Derived_Needs.
«refine»
«requirement»
evaluate_solution_quality
Description
To evaluate the quality of a Sequence of Derived_Needs against given optimisation criteria.
«refine»

Figure 1080: Tasking Service Policy

Tasking

This service determines the achievability of a Tasking given the available aircraft system capability to respond to a System_Stimulus, including applicable System_Constraints, determines the cost of a Sequence of Derived_Needs against given optimisation criteria, and fulfils achievable requirements when instructed.

Interfaces

Tasking_Criterion

This interface is the measurement criterion/criteria associated with a Tasking (e.g. quality, risk, or robustness).

Attributes

property	The property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Tasking_Requirement

This interface is the Tasking requirement (e.g. task type, timing, importance of success, or risk profile), the associated Optimisation_Criterion of that requirement, and related timing information.

Attributes

specification	The definition of a Tasking requirement, for example, a task to transit to a region and search for a target of interest.

temporal_information	Information covering timing, such as start and end times.
quality	How well the Sequence of Derived_Needs is predicted to satisfy the Tasking.
optimisation	The Optimisation_Criterion of the Derived_Need.

Tasking_Achievement

This interface is the statement of achievement against the Tasking.

Activities

identify_requirement_progress

Identify the progress of the Sequence of Derived_Needs for a Tasking.

determine_implementation_solution

Determine the Sequence of Derived_Needs to achieve a Tasking.

execute_implementation_solution

Fulfil a Tasking by executing the planned Sequence of Derived_Needs.

identify_whether_requirement_is_achievable

Identify whether a Tasking is still achievable given current or predicted aircraft system capability and System_Constraints.

5.4.2.62.7.1.2 Solution_Dependency

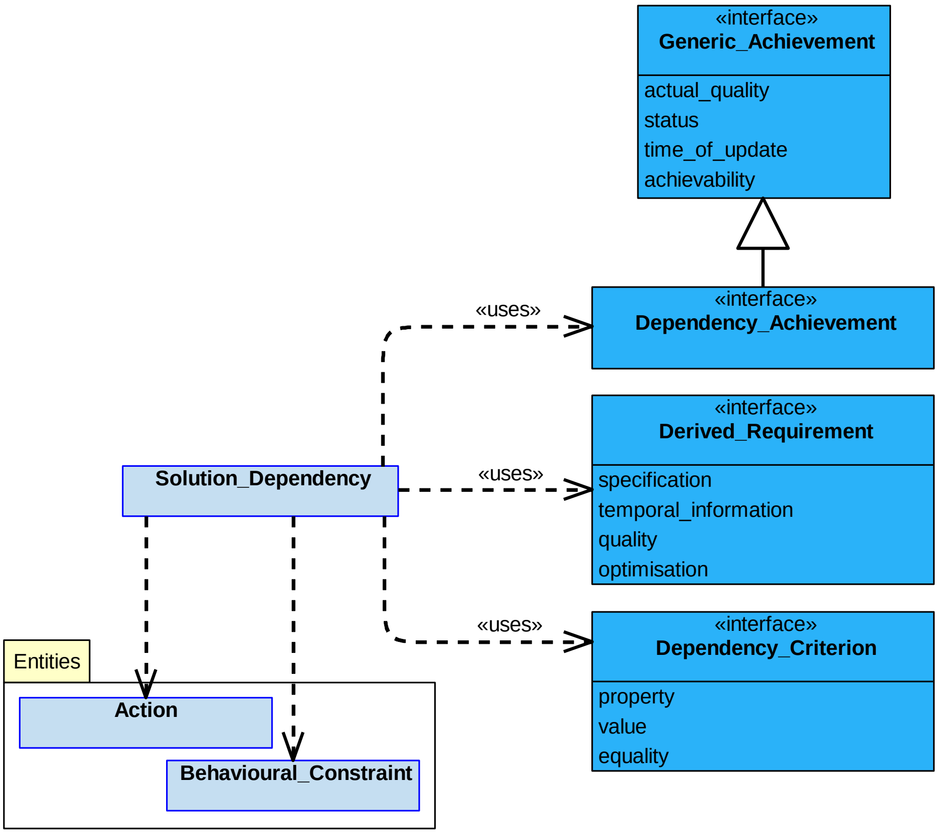


Figure 1081: Solution_Dependency Service Definition

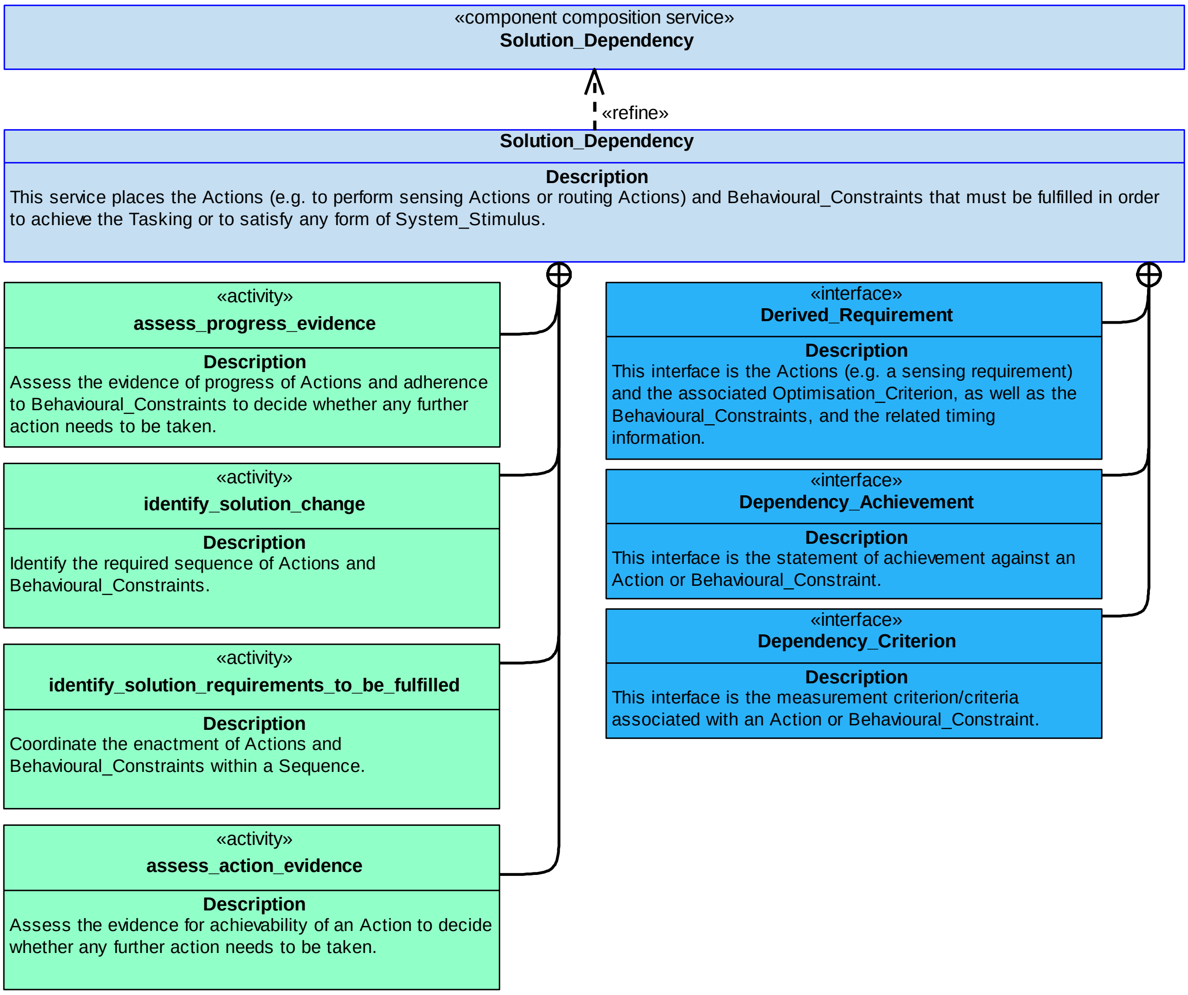

«component composition service»
Solution_Dependency
«refine»
Solution_Dependency
Description
This service places the Actions (e.g. to perform sensing Actions or routing Actions) and Behavioural_Constraints that must be fulfilled in order to achieve the Tasking or to satisfy any form of System_Stimulus.
«activity»
assess_progress_evidence
Description
Assess the evidence of progress of Actions and adherence to Behavioural_Constraints to decide whether any further action needs to be taken.
«activity»
identify_solution_change
Description
Identify the required sequence of Actions and Behavioural_Constraints.
«activity»
identify_solution_requirements_to_be_fulfilled
Description
Coordinate the enactment of Actions and Behavioural_Constraints within a Sequence.
«activity»
assess_action_evidence
Description
Assess the evidence for achievability of an Action to decide whether any further action needs to be taken.
«interface»
Derived_Requirement
Description
This interface is the Actions (e.g. a sensing requirement) and the associated Optimisation_Criterion, as well as the Behavioural_Constraints, and the related timing information.
«interface»
Dependency_Achievement
Description
This interface is the statement of achievement against an Action or Behavioural_Constraint.
«interface»
Dependency_Criterion
Description
This interface is the measurement criterion/criteria associated with an Action or Behavioural_Constraint.

Figure 1082: Solution_Dependency Service Policy

Solution_Dependency

This service places the Actions (e.g. to perform sensing Actions or routing Actions) and Behavioural_Constraints that must be fulfilled in order to achieve the Tasking or to satisfy any form of System_Stimulus.

Interfaces

Derived_Requirement

This interface is the Actions (e.g. a sensing requirement) and the associated Optimisation_Criterion, as well as the Behavioural_Constraints, and the related timing information.

Attributes

specification	The definition of the Action or Behavioural_Constraint.
temporal_information	Information covering timing, such as start and end times.
quality	How well the Action is predicted to be satisfied.
optimisation	The Optimisation_Criterion of the Action.

Dependency_Criterion

This interface is the measurement criterion/criteria associated with an Action or Behavioural_Constraint.

Attributes

property	The property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Dependency_Achievement

This interface is the statement of achievement against an Action or Behavioural_Constraint.

Activities

assess_action_evidence

Assess the evidence for achievability of an Action to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the evidence of progress of Actions and adherence to Behavioural_Constraints to decide whether any further action needs to be taken.

identify_solution_change

Identify the required sequence of Actions and Behavioural_Constraints.

identify_solution_requirements_to_be_fulfilled

Coordinate the enactment of Actions and Behavioural_Constraints within a Sequence.

5.4.2.62.7.1.3 Information_Dependency

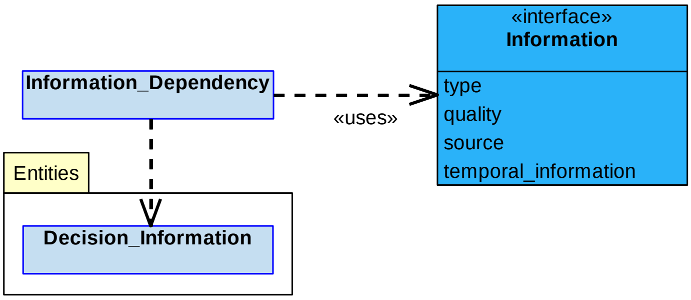


Figure 1083: Information_Dependency Service Definition

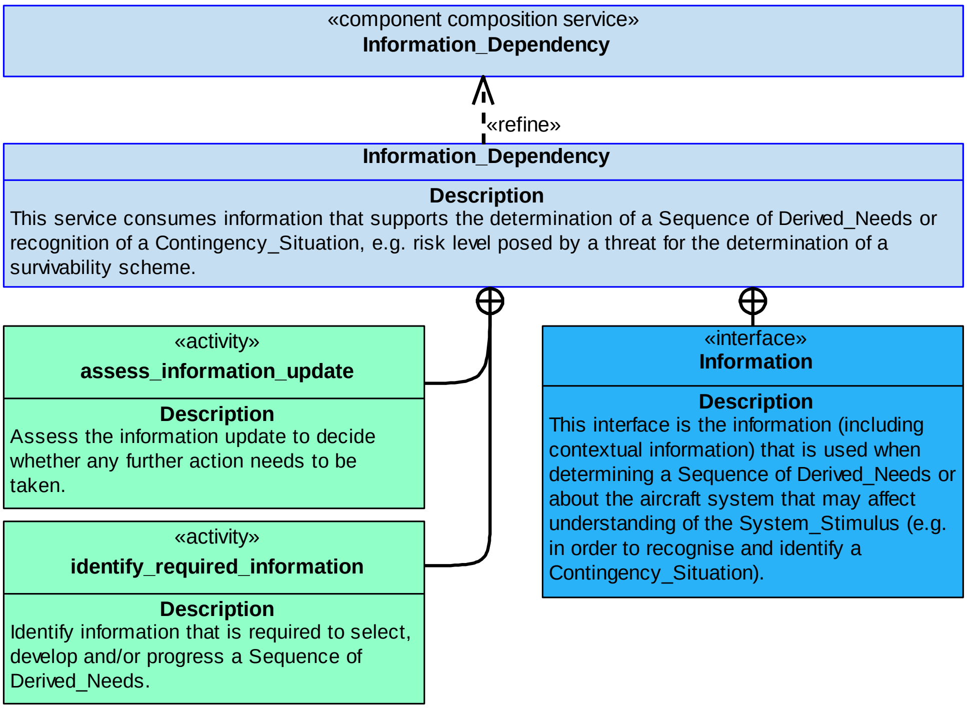

«component composition service»
Information_Dependency
«refine»
Information_Dependency
Description
This service consumes information that supports the determination of a Sequence of Derived_Needs or recognition of a Contingency_Situation, e.g. risk level posed by a threat for the determination of a survivability scheme.
«activity»
assess_information_update
Description
Assess the information update to decide whether any further action needs to be taken.
«activity»
identify_required_information
Description
Identify information that is required to select, develop and/or progress a Sequence of Derived_Needs.
«interface»
Information
Description
This interface is the information (including contextual information) that is used when determining a Sequence of Derived_Needs or about the aircraft system that may affect understanding of the System_Stimulus (e.g. in order to recognise and identify a Contingency_Situation).

Figure 1084: Information_Dependency Service Policy

Information_Dependency

This service consumes information that supports the determination of a Sequence of Derived_Needs or recognition of a Contingency_Situation, e.g. risk level posed by a threat for the determination of a survivability scheme.

Interface

Information

This interface is the information (including contextual information) that is used when determining a Sequence of Derived_Needs or about the aircraft system that may affect understanding of the System_Stimulus (e.g. in order to recognise and identify a Contingency_Situation).

Attributes

type	The type of information.
quality	The quality of the information received.
source	The source of the information.
temporal_information	Information covering timing, such as start and end timing.

Activities

assess_information_update

Assess the information update to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to select, develop and/or progress a Sequence of Derived_Needs.

5.4.2.62.7.1.4 Constraint

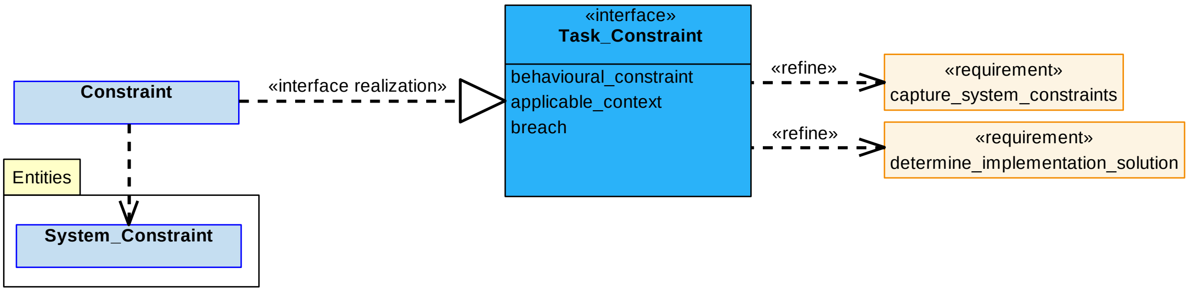


Figure 1085: Constraint Service Definition

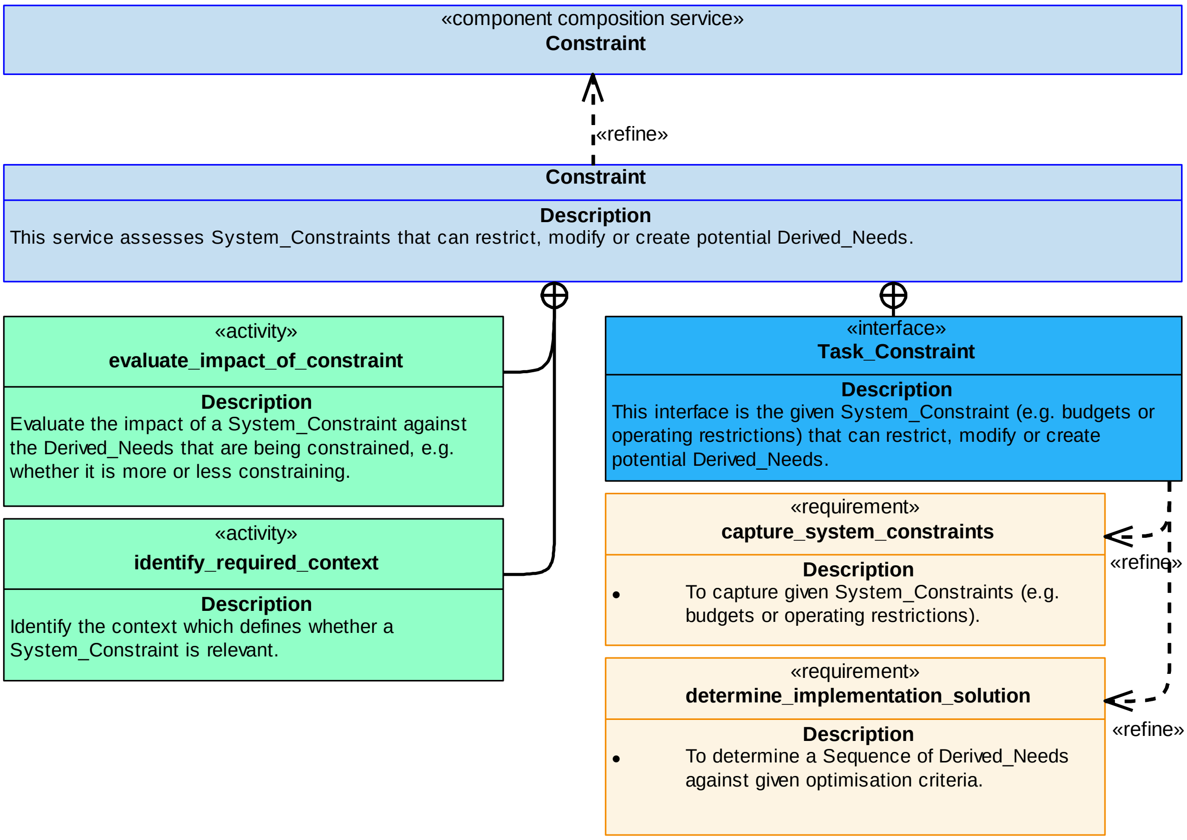

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses System_Constraints that can restrict, modify or create potential Derived_Needs.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of a System_Constraint against the Derived_Needs that are being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether a System_Constraint is relevant.
«interface»
Task_Constraint
Description
This interface is the given System_Constraint (e.g. budgets or operating restrictions) that can restrict, modify or create potential Derived_Needs.
«requirement»
capture_system_constraints
Description
To capture given System_Constraints (e.g. budgets or operating restrictions).
«refine»
«requirement»
determine_implementation_solution
Description
To determine a Sequence of Derived_Needs against given optimisation criteria.
«refine»

Figure 1086: Constraint Service Policy

Constraint

This service assesses System_Constraints that can restrict, modify or create potential Derived_Needs.

Interface

Task_Constraint

This interface is the given System_Constraint (e.g. budgets or operating restrictions) that can restrict, modify or create potential Derived_Needs.

Attributes

behavioural_constraint	A System_Constraint that can restrict or modify the potential behaviour of Tasks, such as a category of Action that is prohibited from being performed.
applicable_context	The context in which the System_Constraint is applicable.
breach	A statement that the System_Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of a System_Constraint against the Derived_Needs that are being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether a System_Constraint is relevant.

5.4.2.62.7.1.5 Capability

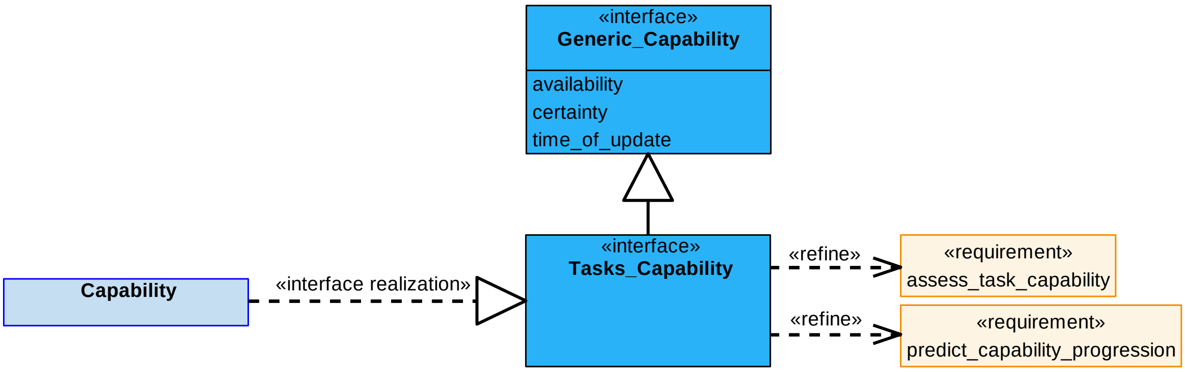


Figure 1087: Capability Service Definition

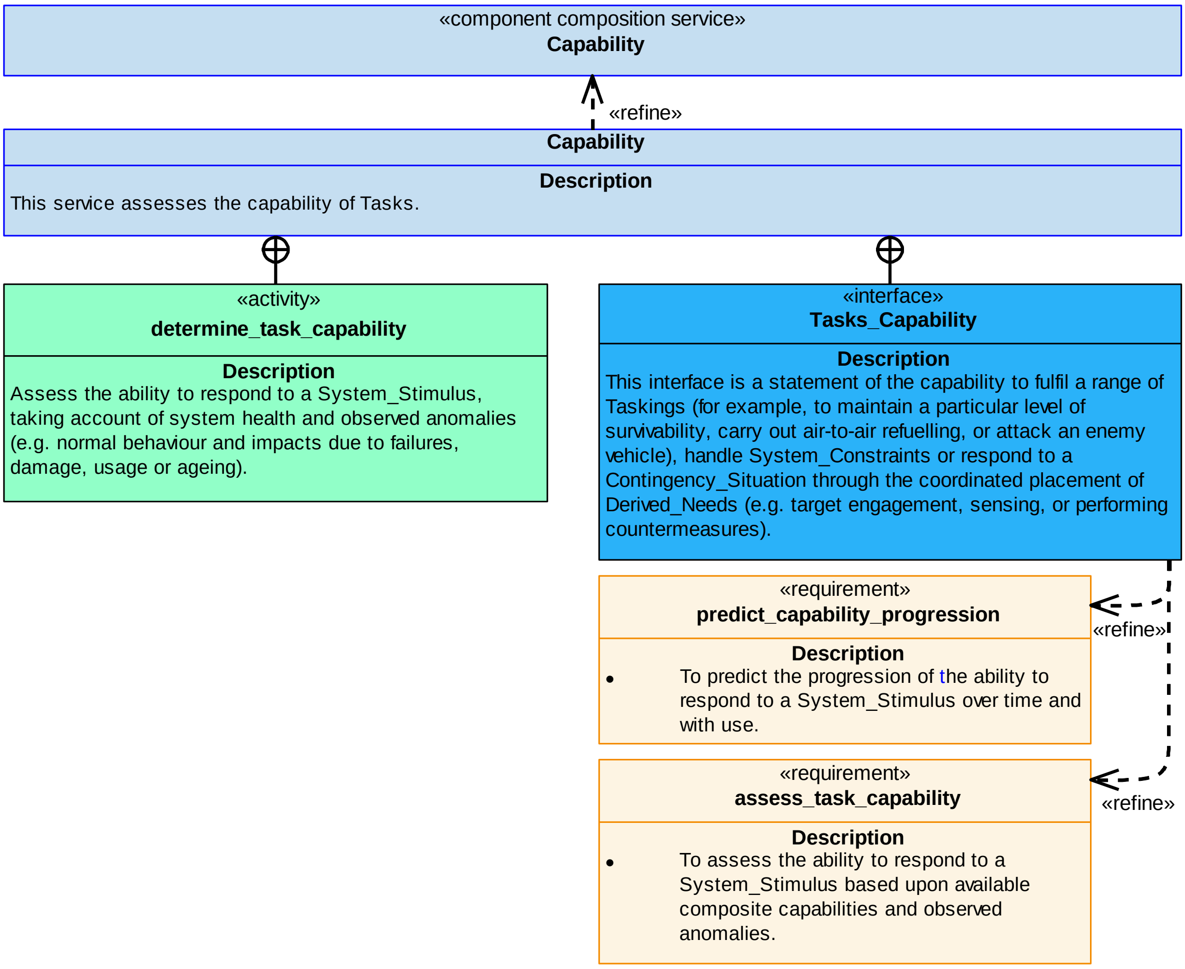

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the capability of Tasks.
«activity»
determine_task_capability
Description
Assess the ability to respond to a System_Stimulus, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Tasks_Capability
Description
This interface is a statement of the capability to fulfil a range of Taskings (for example, to maintain a particular level of survivability, carry out air-to-air refuelling, or attack an enemy vehicle), handle System_Constraints or respond to a Contingency_Situation through the coordinated placement of Derived_Needs (e.g. target engagement, sensing, or performing countermeasures).
«requirement»
predict_capability_progression
Description
To predict the progression of the ability to respond to a System_Stimulus over time and with use.
«refine»
«requirement»
assess_task_capability
Description
To assess the ability to respond to a System_Stimulus based upon available composite capabilities and observed anomalies.
«refine»

Figure 1088: Capability Service Policy

Capability

This service assesses the capability of Tasks.

Interface

Tasks_Capability

This interface is a statement of the capability to fulfil a range of Taskings (for example, to maintain a particular level of survivability, carry out air-to-air refuelling, or attack an enemy vehicle), handle System_Constraints or respond to a Contingency_Situation through the coordinated placement of Derived_Needs (e.g. target engagement, sensing, or performing countermeasures).

Activity

determine_task_capability

Assess the ability to respond to a System_Stimulus, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.62.7.1.6 Capability_Evidence

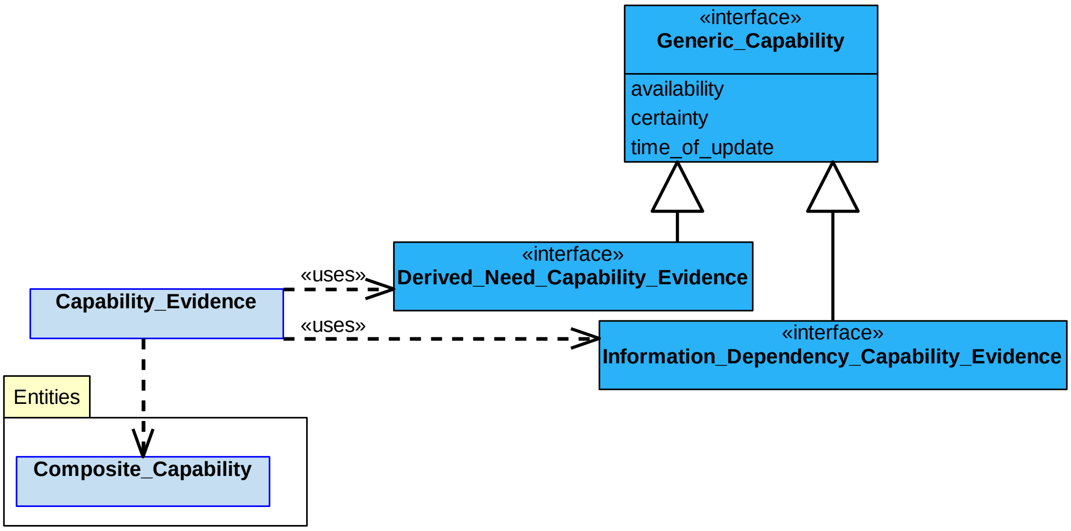
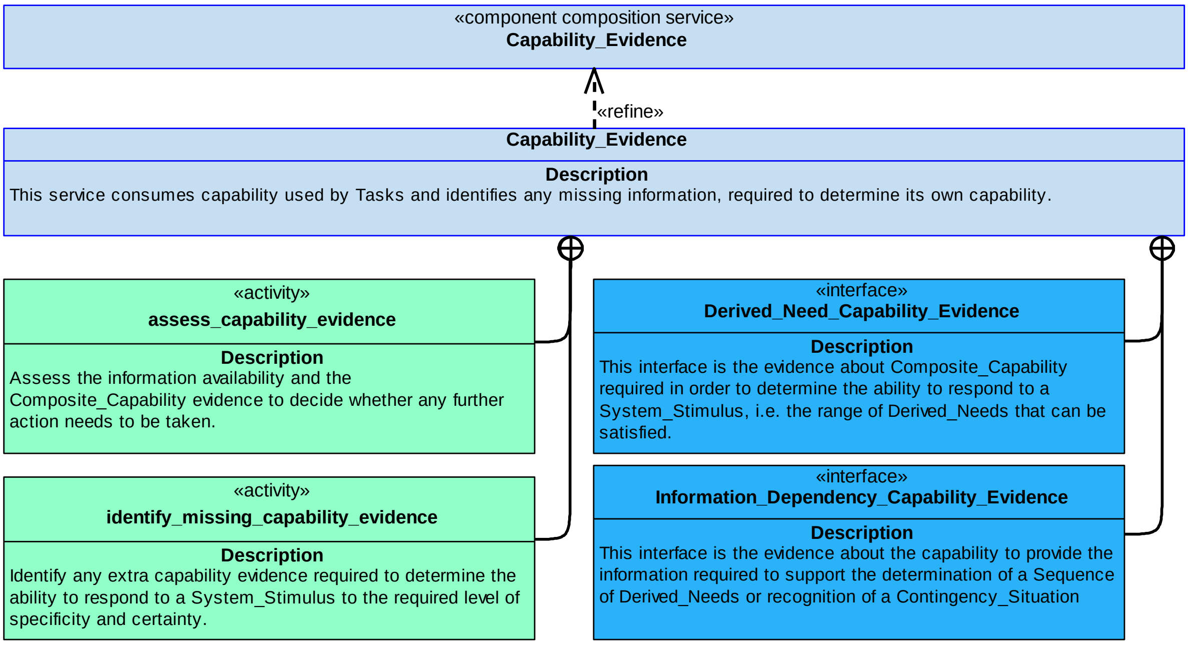


Figure 1089: Capability_Evidence Service Definition


«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
«activity»
assess_capability_evidence
Description
«interface»
Derived_Need_Capability_Evidence
Description
«activity»
identify_missing_capability_evidence
Description
«interface»
Information_Dependency_Capability_Evidence
Description

This service consumes capability used by Tasks and identifies any missing information, required to determine its own capability.

Assess the information availability and the Composite_Capability evidence to decide whether any further action needs to be taken.

This interface is the evidence about Composite_Capability required in order to determine the ability to respond to a System_Stimulus, i.e. the range of Derived_Needs that can be satisfied.

Identify any extra capability evidence required to determine the ability to respond to a System_Stimulus to the required level of specificity and certainty.

This interface is the evidence about the capability to provide the information required to support the determination of a Sequence of Derived_Needs or recognition of a Contingency_Situation

Figure 1090: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability used by Tasks and identifies any missing information, required to determine its own capability.

Interfaces

Derived_Need_Capability_Evidence

This interface is the evidence about Composite_Capability required in order to determine the ability to respond to a System_Stimulus, i.e. the range of Derived_Needs that can be satisfied.

Information_Dependency_Capability_Evidence

This interface is the evidence about the capability to provide the information required to support the determination of a Sequence of Derived_Needs or recognition of a Contingency_Situation

Activities

assess_capability_evidence

Assess the information availability and the Composite_Capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the ability to respond to a System_Stimulus to the required level of specificity and certainty.

5.4.2.62.7.2 Service Dependencies

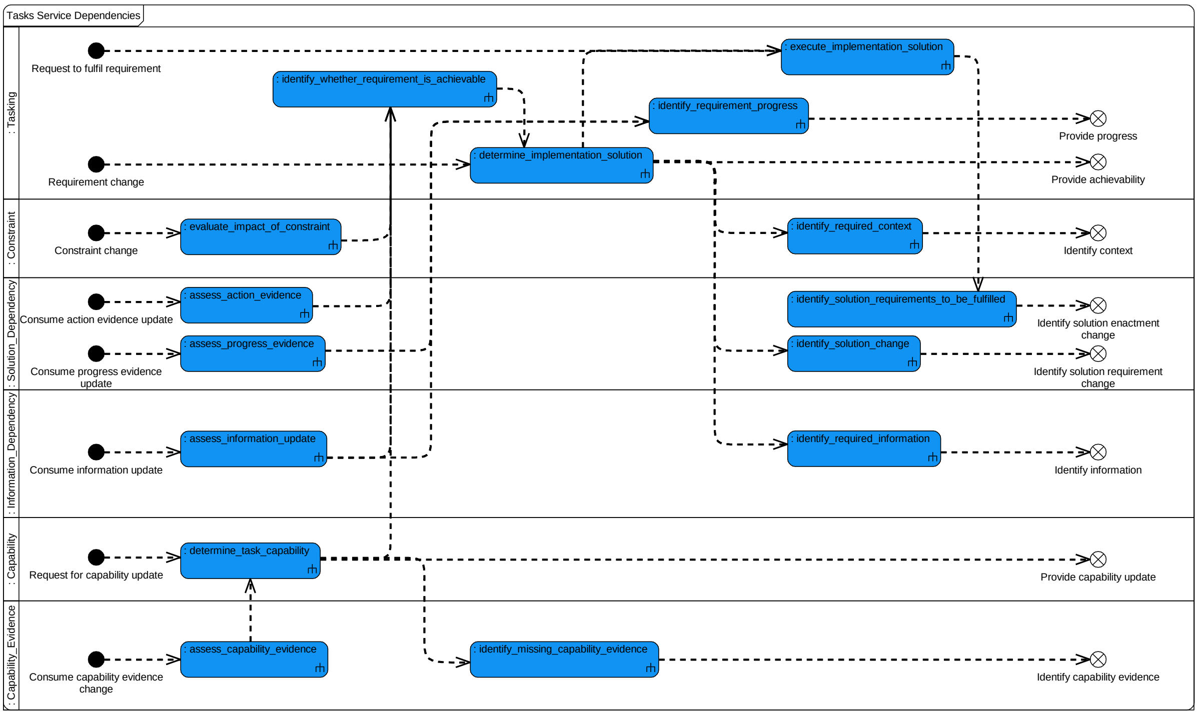

Tasks Service Dependencies
: Tasking
: Constraint
: Solution_Dependency
: Information_Dependency
: Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
Constraint change
Consume action evidence update
Consume progress evidence update
Consume information update
Request for capability update
Consume capability evidence change
: execute_implementation_solution
: identify_whether_requirement_is_achievable
: identify_requirement_progress
: determine_implementation_solution
: evaluate_impact_of_constraint
: identify_required_context
: assess_action_evidence
: identify_solution_requirements_to_be_fulfilled
: assess_progress_evidence
: identify_solution_change
: assess_information_update
: identify_required_information
: determine_task_capability
: assess_capability_evidence
: identify_missing_capability_evidence
Provide progress
Provide achievability
Identify context
Identify solution enactment change
Identify solution requirement change
Identify information
Provide capability update
Identify capability evidence

Figure 1091: Tasks Service Dependencies

5.4.2.63 Test

5.4.2.63.1 Role

The role of Test is to coordinate tests by managing the resources available to it.

5.4.2.63.2 Overview

Control Architecture

Test is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a Test_Requirement is received, Test will propose a Test_Strategy with associated Pre-conditions (e.g. required system configuration, power requirements, or resource moding). If Pre-conditions are met, the associated Test_Actions can then be executed, which will involve coordinated control of Test_Resources.

Examples of Use

Test can be used for:

- Determining the possible capability limits prior to use.
- Achieving awareness of a loss of capability or anomaly.
- Effectively determining likely causes of a recognised loss of capability.

5.4.2.63.3 Service Summary

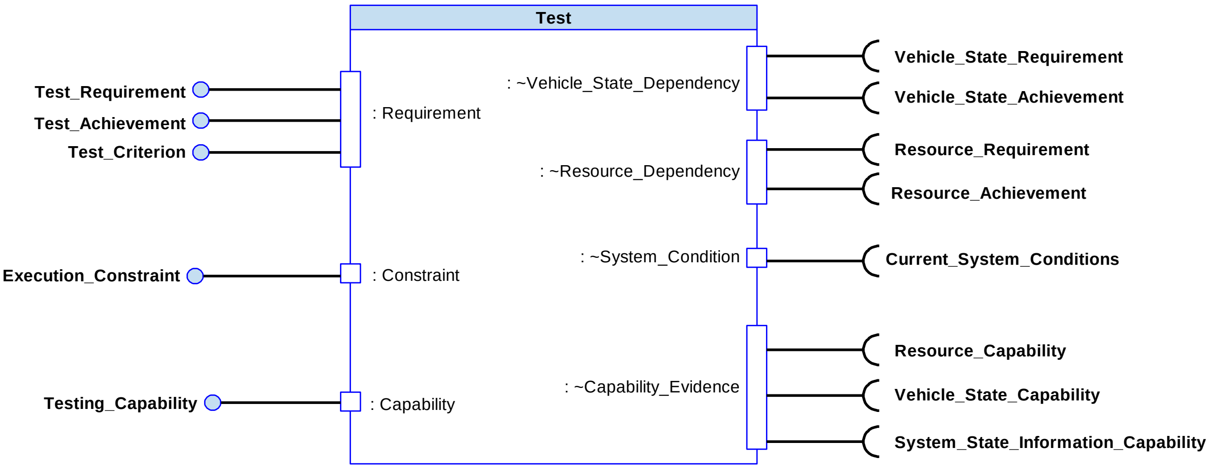


Figure 1092: Test Service Summary

5.4.2.63.4 Responsibilities

capture_test_requirements

- To capture Test_Requirements (e.g. system characteristics to test or a test methodology).

capture_measurement_criteria

- To capture given Measurement_Criterion for the Test_Strategy.

capture_test_constraints

- To capture test Constraints (e.g. operational limits or safety limits).

identify_whether_requirement_remains_achievable

- To identify whether a Test_Requirement is still achievable given current or predicted Capability and Constraints.

determine_test_which_meets_requirements

- To determine a Test_Strategy that meets the given Test_Requirement and Constraints using available resources.

identify_test_pre-conditions

- To identify Pre-conditions required for a given Test_Strategy (e.g. required system configuration, power requirements, or resource moding).

enact_test

- To carry out a Test_Strategy by executing individual Test_Actions (e.g. stimulating a component or requesting an item of equipment to initialise Built In Test (BIT)).

determine_test_progress

- To determine the progress of a Test_Strategy against the Test_Requirement (e.g. percentage complete, time remaining, or phase of test).

determine_test_outcome

- To determine the Test_Outcome of an enacted Test_Strategy.

determine_cost_of_test_solution

- To determine the cost of a Test_Strategy (e.g. affected capabilities or cost of using resources).

assess_capability

- To determine the test Capability (i.e. available tests) using available resources within given constraints, taking into account system health and observed anomalies.

identify_missing_information

- To identify missing information that could improve the certainty or specificity of the test Capability assessment.

predict_capability_progression

- To predict the progression of test Capability over time and with use.

5.4.2.63.5 Subject Matter Semantics

The subject matter of Test is tests that can be run to establish the condition of part of a system.

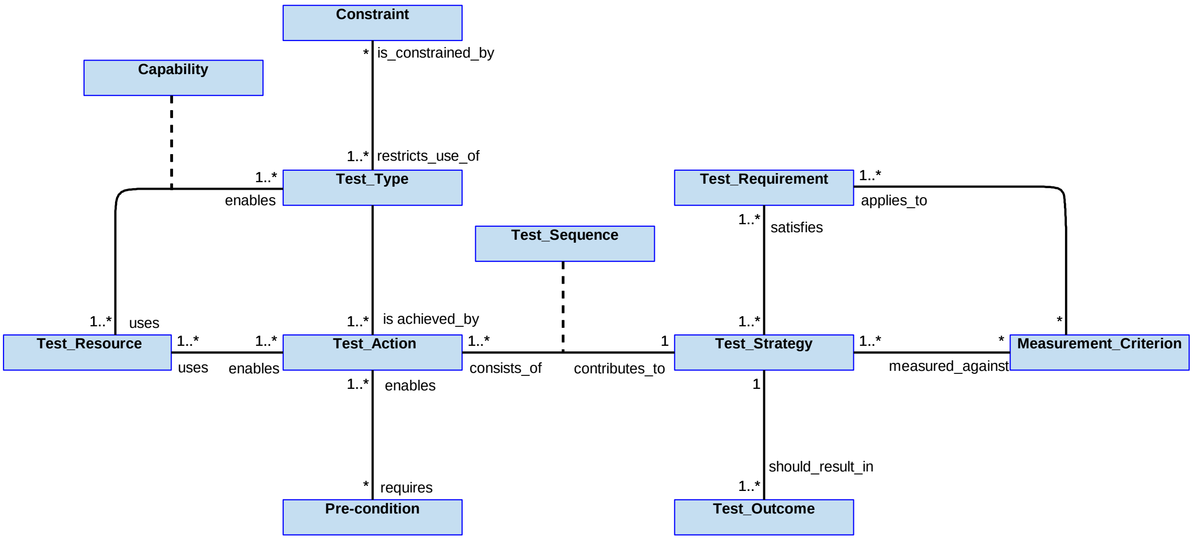


Figure 1093: Test Semantics

5.4.2.63.5.1 Entities

Capability

The range of Test_Types that the component is able to perform with its available Test_Resources.

Constraint

An externally imposed restriction that limits the tests that Test is able to run/use.

Measurement_Criterion

A criterion which a Test_Strategy will be measured against, e.g. the cost of using Test_Resources.

Pre-condition

A condition that must be true before a Test_Action can take place (e.g. required system configuration, power requirements, or resource moding).

Test_Requirement

A requirement to enact the necessary tests to generate information about the condition of (part of) the system: its capability and its potential state.

Test_Action

An activity, defined in terms of what needs to be done, to achieve the test.

Test_Outcome

The status of an enacted test, e.g. not started, in progress, or complete. This may also include the test result, e.g. pass or fail.

Test_Resource

A resource that can be instructed to carry out a Test_Action.

Test_Sequence

The order in which Test_Actions must be performed to achieve a Test_Strategy.

Test_Strategy

The strategy identified to address the Test_Requirements.

Test_Type

A specific type of test that can be managed by the component, e.g. a type of BIT or calibration test.

5.4.2.63.6 Design Rationale

5.4.2.63.6.1 Assumptions

- The component is responsible for commanding resources to initiate a built in test but is not responsible for conducting the test.
- This component will trigger the test and monitor progress. It may receive the test result (e.g. pass or fail) as this may influence how the component conducts subsequent tests. However, in most cases it will not receive any details about the test results, only the Test_Outcome. Where the component receives any information about the result of a test, it does not pass this information on.
- Additional tests may be commanded based upon a Test_Outcome, e.g. failure of a simple test may result in the need to perform a more in-depth test.
- Extensions to the component may introduce additional security functions.
- Not all tests need to be managed by this component: only those that need to be coordinated (with each other or with other resource users). Start up tests would not ordinarily be managed by the Test component.

5.4.2.63.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Test:

- Test - This PYRAMID concept explains how testing is coordinated, Test is a fundamental part of this.
- Recording and Logging - Logging which tests have been enacted will be performed in accordance with this PYRAMID concept.

Extensions

- Extensions to this component could be used to accommodate different types of test and specifics of the system-under-test's behaviour.

5.4.2.63.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component controls the tests performed by air systems. This could include initiating BIT.
- Performing intrusive testing on some systems at the wrong time could have catastrophic consequences. For example, performing a start-up BIT on a flight control system or inertial navigation system during flight would result in uncontrolled flight, an uncontrolled crash, loss of air vehicle and potentially fatalities.

5.4.2.63.6.4 Security Considerations

The indicative security classification is O-S, however the component(s) with which it is associated will be a significant factor.

It is expected that there may be multiple instances of this component, each of which will reside in a security domain that reflects the hardware under test. Whilst not expected to be of an inherently high classification itself, the component is responsible for triggering and monitoring testing, for which some potentially SNEO configuration, performance or simulation data might be handled by the component, with an associated increase in the component's classification.

The authenticity of test requests is necessary in order to prevent needless tests being enacted; these would likely lead to a temporary loss of functionality and represent a denial of service.

The component may be expected to at least partially satisfy security related functions relating to:

- **Logging of Security Data**, testing may assist subsequent forensic examination of anomalies, which might then point to the presence of a cyber attack or other breach.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- **System Status and Monitoring** during testing might indicate or improve the specificity of a problem with integrity or availability of functions.

The component may be expected to at least partially satisfy security enforcing functions relating to:

- **HW Authentication** through the verification of hardware under test, supporting the continuing chain of trust from the hardware and up through the infrastructure.

5.4.2.63.7 Services

5.4.2.63.7.1 Service Definitions

5.4.2.63.7.1.1 Requirement

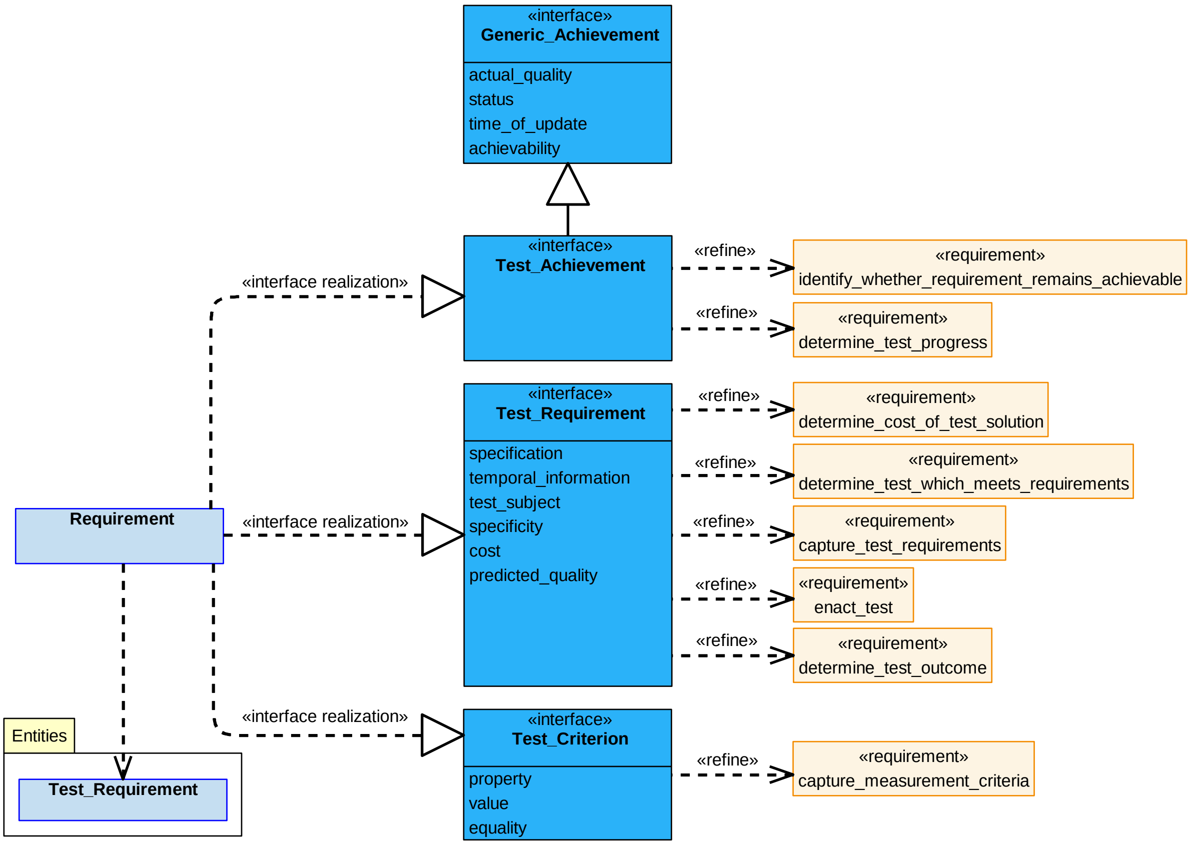


Figure 1094: Requirement Service Definition

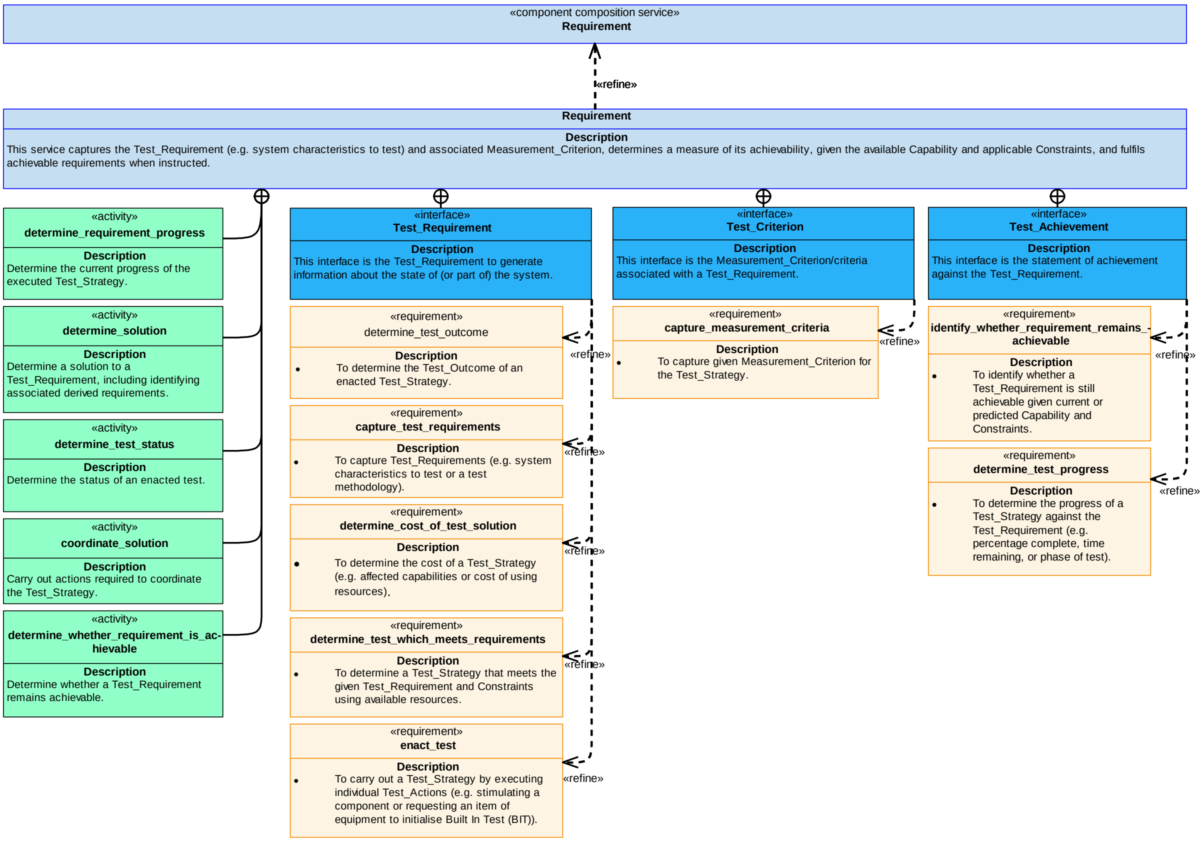

«component composition service»
Requirement
«refine»
Requirement
Description
This service captures the Test_Requirement (e.g. system characteristics to test) and associated Measurement_Criterion, determines a measure of its achievability, given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_requirement_progress
Description
Determine the current progress of the executed Test_Strategy.
«activity»
determine_solution
Description
Determine a solution to a Test_Requirement, including identifying associated derived requirements.
«activity»
determine_test_status
Description
Determine the status of an enacted test.
«activity»
coordinate_solution
Description
Carry out actions required to coordinate the Test_Strategy.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether a Test_Requirement remains achievable.
«interface»
Test_Requirement
Description
This interface is the Test_Requirement to generate information about the state of (or part of) the system.
«requirement»
determine_test_outcome
Description
To determine the Test_Outcome of an enacted Test_Strategy.
«requirement»
capture_test_requirements
Description
To capture Test_Requirements (e.g. system characteristics to test or a test methodology).
«requirement»
determine_cost_of_test_solution
Description
To determine the cost of a Test_Strategy (e.g. affected capabilities or cost of using resources).
«requirement»
determine_test_which_meets_requirements
Description
To determine a Test_Strategy that meets the given Test_Requirement and Constraints using available resources.
«requirement»
enact_test
Description
To carry out a Test_Strategy by executing individual Test_Actions (e.g. stimulating a component or requesting an item of equipment to initialise Built In Test (BIT)).
«interface»
Test_Criterion
Description
This interface is the Measurement_Criterion/criteria associated with a Test_Requirement.
«requirement»
capture_measurement_criteria
Description
To capture given Measurement_Criterion for the Test_Strategy.
«interface»
Test_Achievement
Description
This interface is the statement of achievement against the Test_Requirement.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Test_Requirement is still achievable given current or predicted Capability and Constraints.
«requirement»
determine_test_progress
Description
To determine the progress of a Test_Strategy against the Test_Requirement (e.g. percentage complete, time remaining, or phase of test).
«refine»

Figure 1095: Requirement Service Policy

Requirement

This service captures the Test_Requirement (e.g. system characteristics to test) and associated Measurement_Criterion, determines a measure of its achievability, given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Test_Requirement

This interface is the Test_Requirement to generate information about the state of (or part of) the system.

Attributes

specification	The information that other components require (e.g. to determine the state of an actuator or determine if an actuator will move in response to a command).
temporal_information	Information covering timing, such as test duration and start and end times.
test_subject	A group of one or more elements to which the Test_Requirement applies.
specificity	The required granularity of test results.
cost	The cost of executing the solution (e.g. resources used or time taken).
predicted_quality	How well the proposed Test_Strategy is predicted to satisfy the requirement.

Test_Criterion

This interface is the Measurement_Criterion/criteria associated with a Test_Requirement.

Attributes

property	The property to be measured, e.g. time taken for test.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Test_Achievement

This interface is the statement of achievement against the Test_Requirement.

Activities

determine_requirement_progress

Determine the current progress of the executed Test_Strategy.

determine_solution

Determine a solution to a Test_Requirement, including identifying associated derived requirements.

coordinate_solution

Carry out actions required to coordinate the Test_Strategy.

determine_whether_requirement_is_achievable

Determine whether a Test_Requirement remains achievable.

determine_test_status

Determine the status of an enacted test.

5.4.2.63.7.1.2 Vehicle_State_Dependency

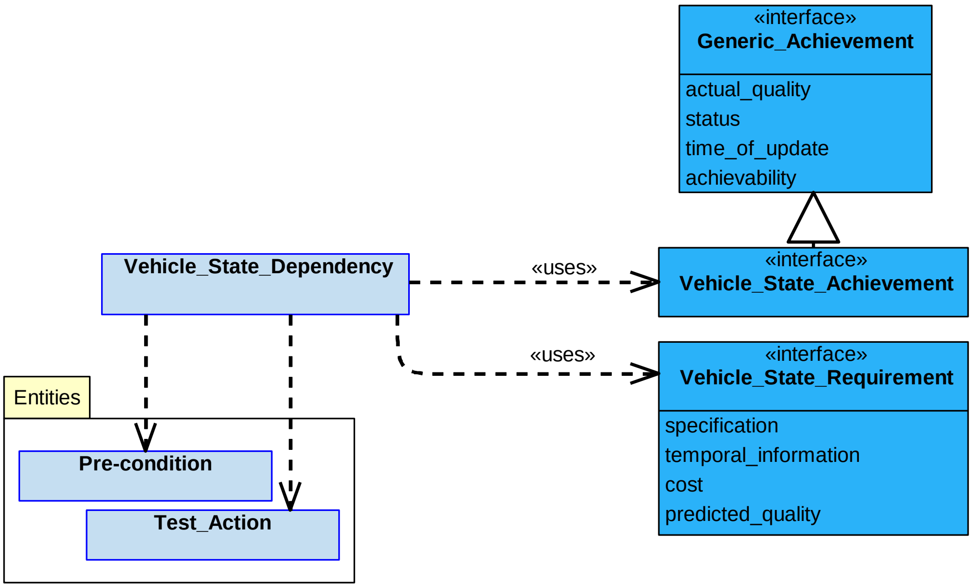
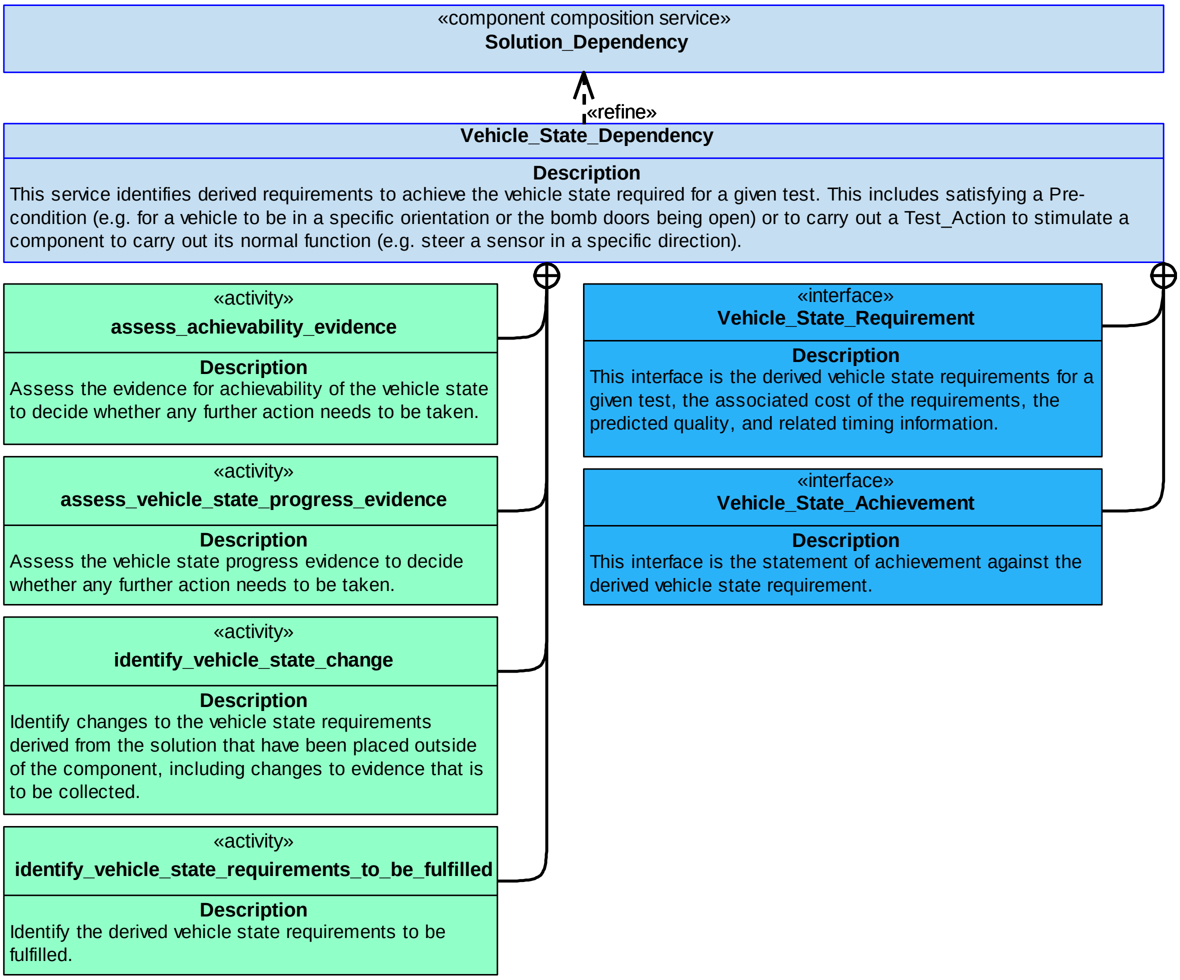


Figure 1096: Vehicle_State_Dependency Service Definition


«component composition service»
Solution_Dependency
«refine»
Vehicle_State_Dependency
Description
«activity»
assess_achievability_evidence
Description
«activity»
assess_vehicle_state_progress_evidence
Description
«activity»
identify_vehicle_state_change
Description
«activity»
identify_vehicle_state_requirements_to_be_fulfilled
Description
«interface»
Vehicle_State_Requirement
Description
«interface»
Vehicle_State_Achievement
Description

This service identifies derived requirements to achieve the vehicle state required for a given test. This includes satisfying a Pre-condition (e.g. for a vehicle to be in a specific orientation or the bomb doors being open) or to carry out a Test_Action to stimulate a component to carry out its normal function (e.g. steer a sensor in a specific direction).

Assess the evidence for achievability of the vehicle state to decide whether any further action needs to be taken.

Assess the vehicle state progress evidence to decide whether any further action needs to be taken.

Identify changes to the vehicle state requirements derived from the solution that have been placed outside of the component, including changes to evidence that is to be collected.

Identify the derived vehicle state requirements to be fulfilled.

This interface is the derived vehicle state requirements for a given test, the associated cost of the requirements, the predicted quality, and related timing information.

This interface is the statement of achievement against the derived vehicle state requirement.

Figure 1097: Vehicle_State_Dependency Service Policy

Vehicle_State_Dependency

This service identifies derived requirements to achieve the vehicle state required for a given test. This includes satisfying a Pre-condition (e.g. for a vehicle to be in a specific orientation or the bomb doors being open) or to carry out a Test_Action to stimulate a component to carry out its normal function (e.g. steer a sensor in a specific direction).

Interfaces

Vehicle_State_Requirement

This interface is the derived vehicle state requirements for a given test, the associated cost of the requirements, the predicted quality, and related timing information.

Attributes

specification	The required vehicle state(s), including the environment in which testing is to be performed.
temporal_information	Information covering timing, such as start and end times of the derived requirement.
cost	The cost of executing the solution (e.g. resources used or time taken).
predicted_quality	How well the proposed vehicle state solution is predicted to satisfy the requirement.

Vehicle_State_Achievement

This interface is the statement of achievement against the derived vehicle state requirement.

Activities

assess_achievability_evidence

Assess the evidence for achievability of the vehicle state to decide whether any further action needs to be taken.

assess_vehicle_state_progress_evidence

Assess the vehicle state progress evidence to decide whether any further action needs to be taken.

identify_vehicle_state_change

Identify changes to the vehicle state requirements derived from the solution that have been placed outside of the component, including changes to evidence that is to be collected.

identify_vehicle_state_requirements_to_be_fulfilled

Identify the derived vehicle state requirements to be fulfilled.

5.4.2.63.7.1.3 Resource_Dependency

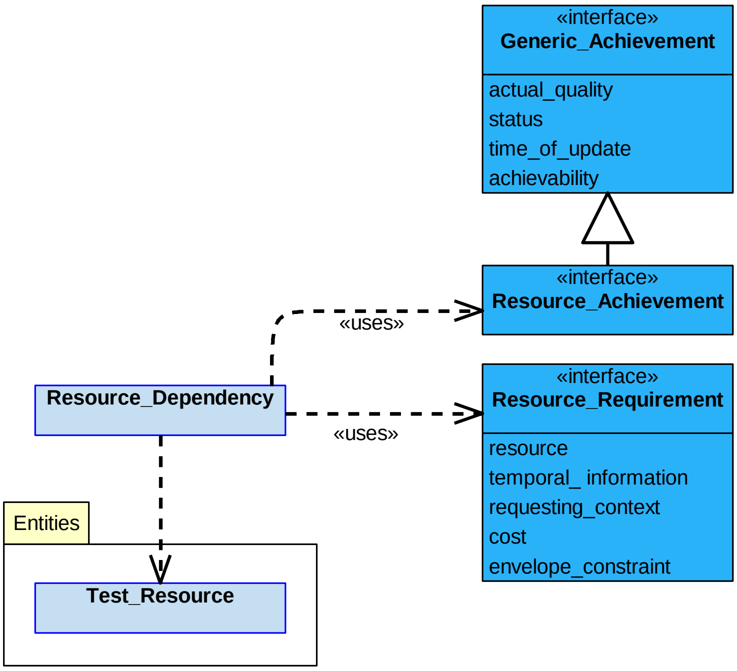


Figure 1098: Resource_Dependency Service Definition

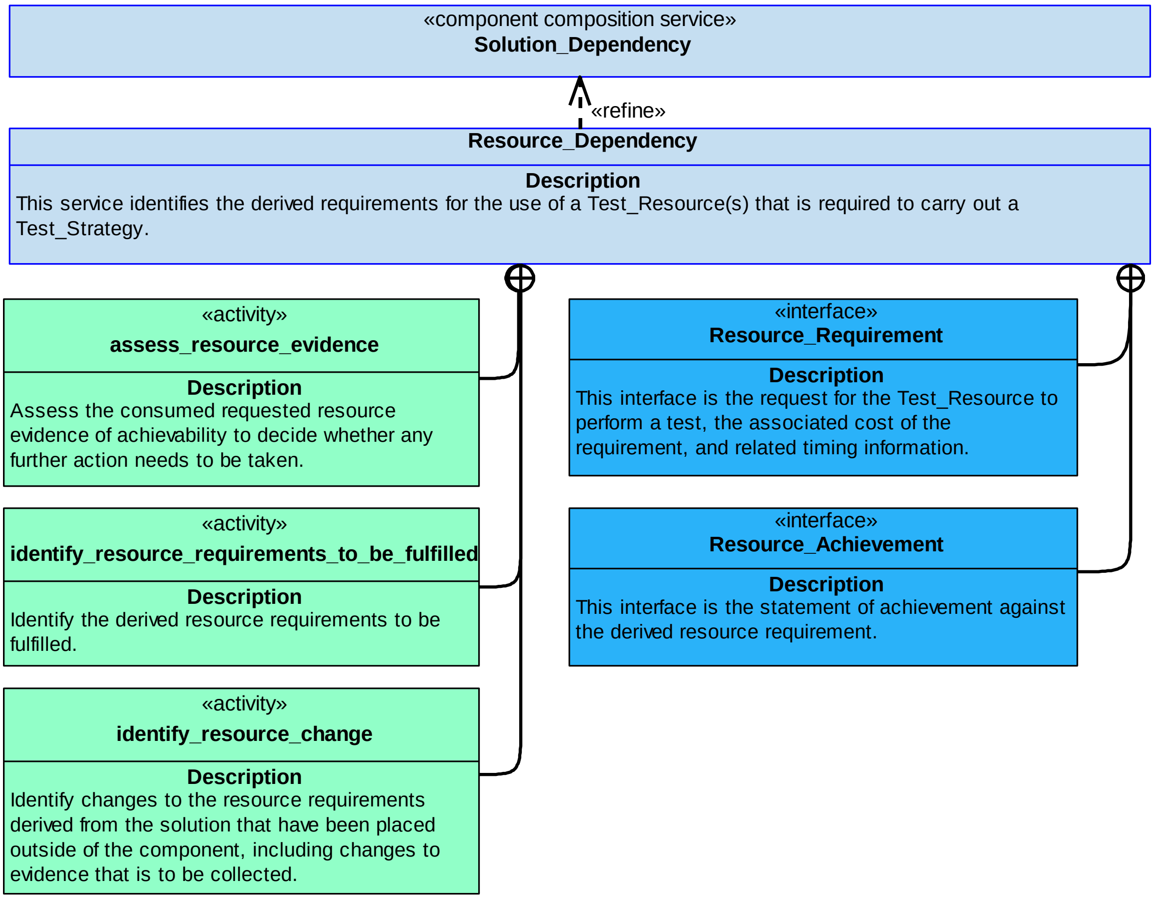

«component composition service»
Solution_Dependency
«refine»
Resource_Dependency
Description
This service identifies the derived requirements for the use of a Test_Resource(s) that is required to carry out a Test_Strategy.
«activity»
assess_resource_evidence
Description
Assess the consumed requested resource evidence of achievability to decide whether any further action needs to be taken.
«interface»
Resource_Requirement
Description
This interface is the request for the Test_Resource to perform a test, the associated cost of the requirement, and related timing information.
«activity»
identify_resource_requirements_to_be_fulfilled
Description
Identify the derived resource requirements to be fulfilled.
«interface»
Resource_Achievement
Description
This interface is the statement of achievement against the derived resource requirement.
«activity»
identify_resource_change
Description
Identify changes to the resource requirements derived from the solution that have been placed outside of the component, including changes to evidence that is to be collected.

Figure 1099: Resource_Dependency Service Policy

Resource_Dependency

This service identifies the derived requirements for the use of a Test_Resource(s) that is required to carry out a Test_Strategy.

Interfaces

Resource_Requirement

This interface is the request for the Test_Resource to perform a test, the associated cost of the requirement, and related timing information.

Attributes

resource	The Test_Resource being requested.
temporal_ information	Information covering timing for the requested Test_Resource, such as start and end times. This may include segments of a requested time window that must not be interrupted, etc.
requesting_context	The information that identifies the source or reason for the request.
cost	The cost of executing the solution (e.g. time taken).
envelope_constraint	The limits placed on a resource when achieving a test requirement, e.g. limit actuator to 2%.

Resource_Achievement

This interface is the statement of achievement against the derived resource requirement.

Activities

assess_resource_evidence

Assess the consumed requested resource evidence of achievability to decide whether any further action needs to be taken.

identify_resource_requirements_to_be_fulfilled

Identify the derived resource requirements to be fulfilled.

identify_resource_change

Identify changes to the resource requirements derived from the solution that have been placed outside of the component, including changes to evidence that is to be collected.

5.4.2.63.7.1.4 System_Condition

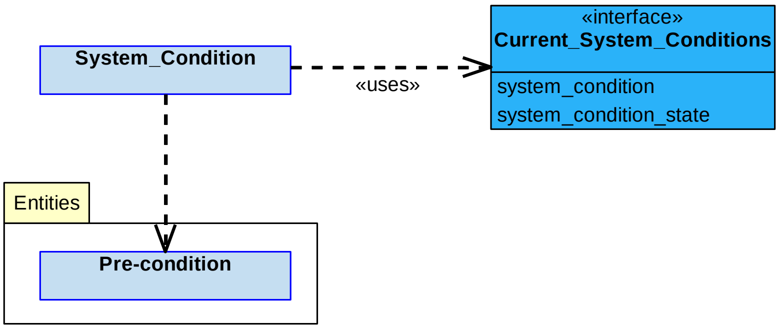


Figure 1100: System_Condition Service Definition

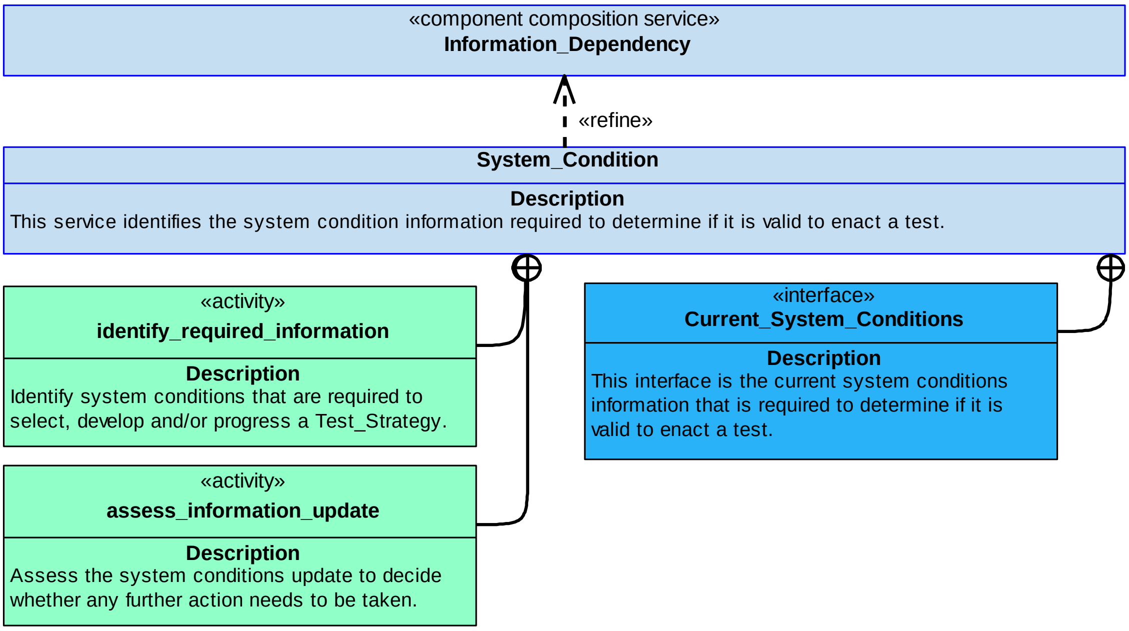

«component composition service»
Information_Dependency
«refine»
System_Condition
Description
This service identifies the system condition information required to determine if it is valid to enact a test.
«activity»
identify_required_information
Description
Identify system conditions that are required to select, develop and/or progress a Test_Strategy.
«activity»
assess_information_update
Description
Assess the system conditions update to decide whether any further action needs to be taken.
«interface»
Current_System_Conditions
Description
This interface is the current system conditions information that is required to determine if it is valid to enact a test.

Figure 1101: System_Condition Service Policy

System_Condition

This service identifies the system condition information required to determine if it is valid to enact a test.

Interface

Current_System_Conditions

This interface is the current system conditions information that is required to determine if it is valid to enact a test.

Attributes

system_condition	A specific type of system condition of an Exploiting Platform which may affect whether or how the component can enact a test, e.g. the operating state of a Test_Resource or a state of the vehicle or part of the vehicle (such whether or not is it airborne or the acceleration force on the vehicle).

system_condition_state	The current state of a system condition. For example, a specific Test_Resource is ready to perform a test, the aircraft is airborne, or the current g on the airframe.

Activities

assess_information_update

Assess the system conditions update to decide whether any further action needs to be taken.

identify_required_information

Identify system conditions that are required to select, develop and/or progress a Test_Strategy.

5.4.2.63.7.1.5 Constraint

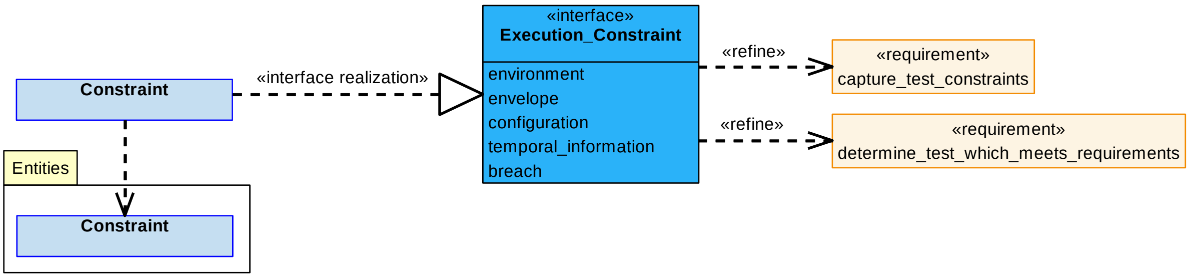


Figure 1102: Constraint Service Definition

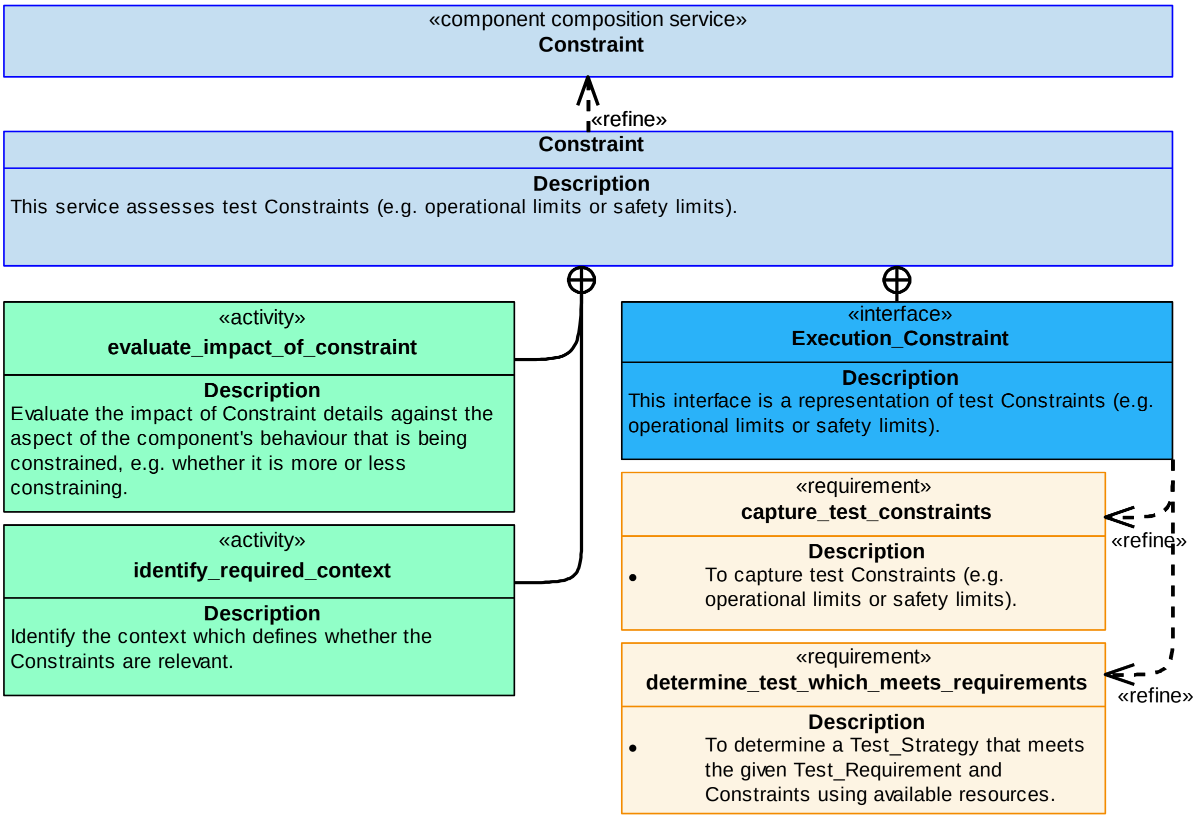

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses test Constraints (e.g. operational limits or safety limits).
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of the component's behaviour that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Execution_Constraint
Description
This interface is a representation of test Constraints (e.g. operational limits or safety limits).
«requirement»
capture_test_constraints
Description
• To capture test Constraints (e.g. operational limits or safety limits).
«refine»
«requirement»
determine_test_which_meets_requirements
Description
• To determine a Test_Strategy that meets the given Test_Requirement and Constraints using available resources.
«refine»

Figure 1103: Constraint Service Policy

Constraint

This service assesses test Constraints (e.g. operational limits or safety limits).

Interface

Execution_Constraint

This interface is a representation of test Constraints (e.g. operational limits or safety limits).

Attributes

environment	The locations within which testing cannot be performed (e.g. not on the ground or over a built up area).
envelope	The parts of the test envelope within which testing cannot be performed (e.g. above a certain g limit or below a set temperature/pressure level).
configuration	The vehicle states in which testing cannot be performed (e.g. with landing gear extended).
temporal_information	A timing limit associated with a Constraint.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of the component's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.63.7.1.6 Capability

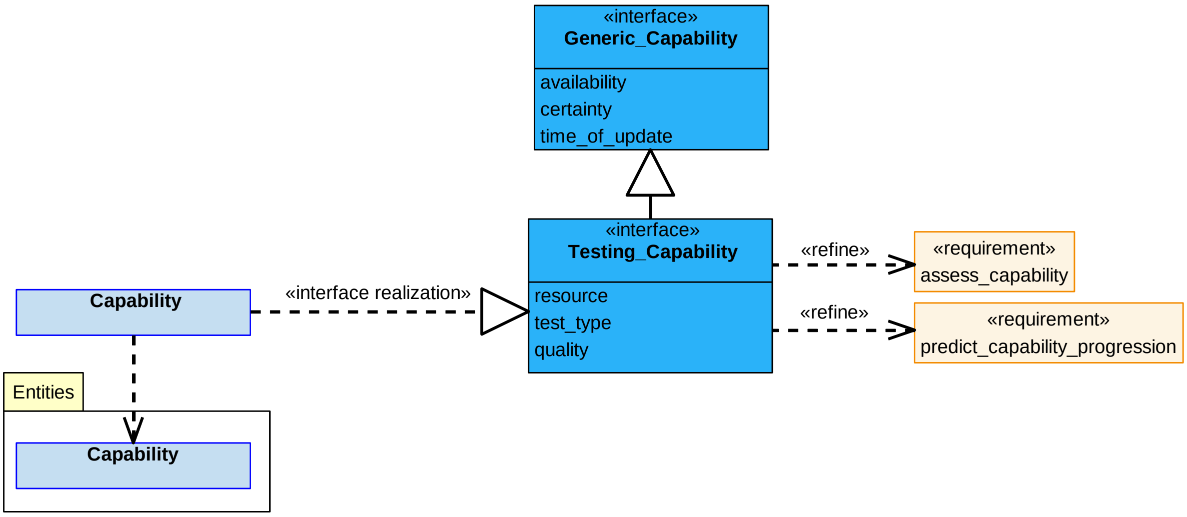


Figure 1104: Capability Service Definition

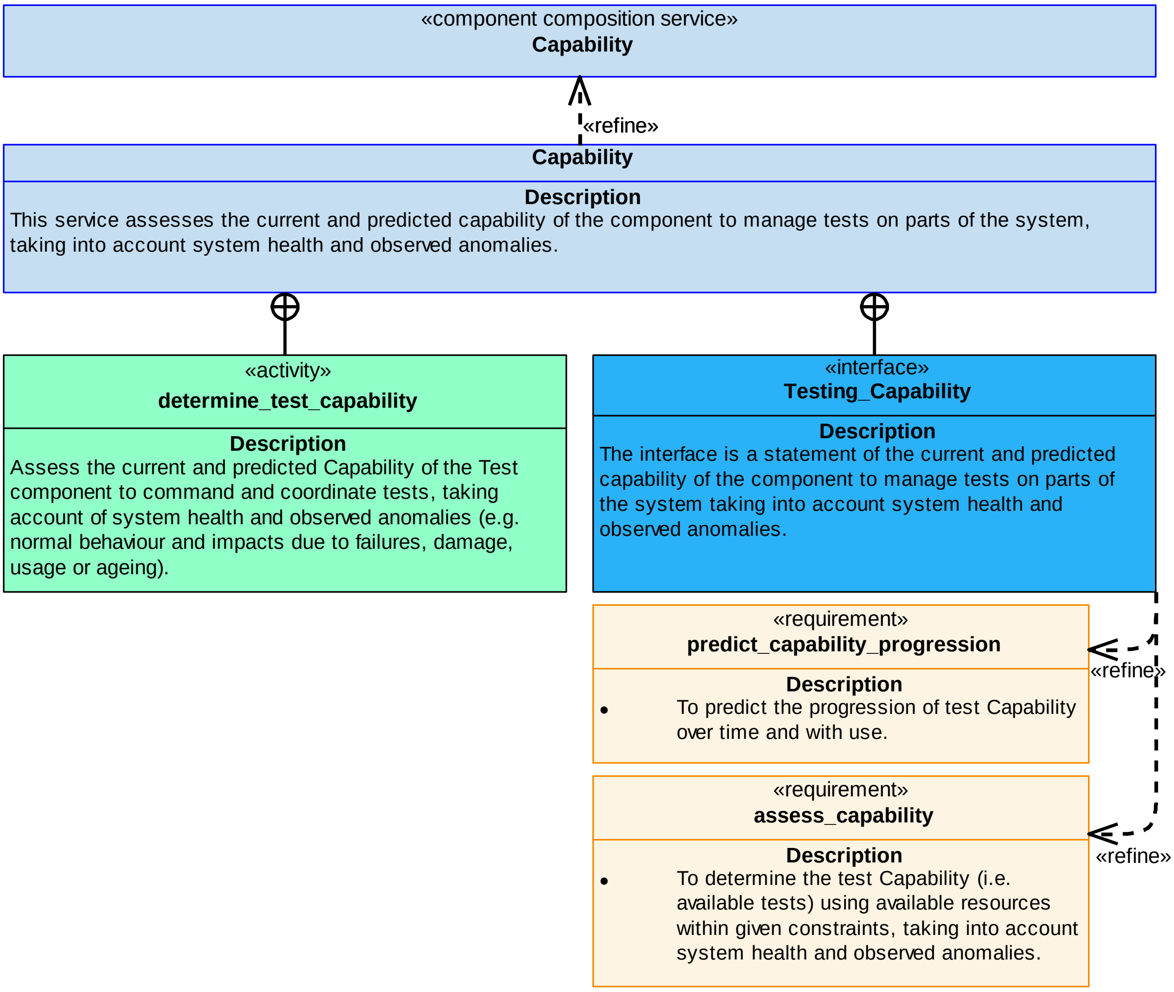

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted capability of the component to manage tests on parts of the system, taking into account system health and observed anomalies.
«activity»
determine_test_capability
Description
Assess the current and predicted Capability of the Test component to command and coordinate tests, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Testing_Capability
Description
The interface is a statement of the current and predicted capability of the component to manage tests on parts of the system taking into account system health and observed anomalies.
«requirement»
predict_capability_progression
Description
To predict the progression of test Capability over time and with use.
«refine»
«requirement»
assess_capability
Description
To determine the test Capability (i.e. available tests) using available resources within given constraints, taking into account system health and observed anomalies.
«refine»

Figure 1105: Capability Service Policy

Capability

This service assesses the current and predicted capability of the component to manage tests on parts of the system, taking into account system health and observed anomalies.

Interface

Testing_Capability

The interface is a statement of the current and predicted capability of the component to manage tests on parts of the system taking into account system health and observed anomalies.

Attributes

resource	The resources that can be tested.
test_type	The type of test that can be triggered (e.g. IBIT or capacity test).
quality	The measure of confidence in the test result.

Activity

determine_test_capability

Assess the current and predicted Capability of the Test component to command and coordinate tests, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.63.7.1.7 Capability_Evidence

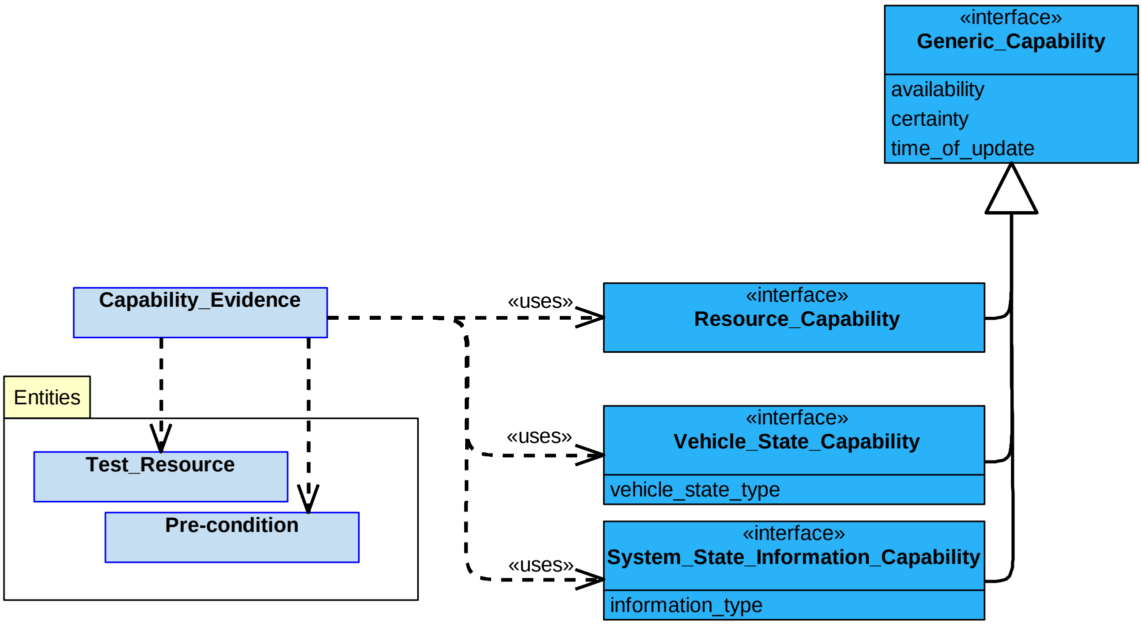


Figure 1106: Capability_Evidence Service Definition

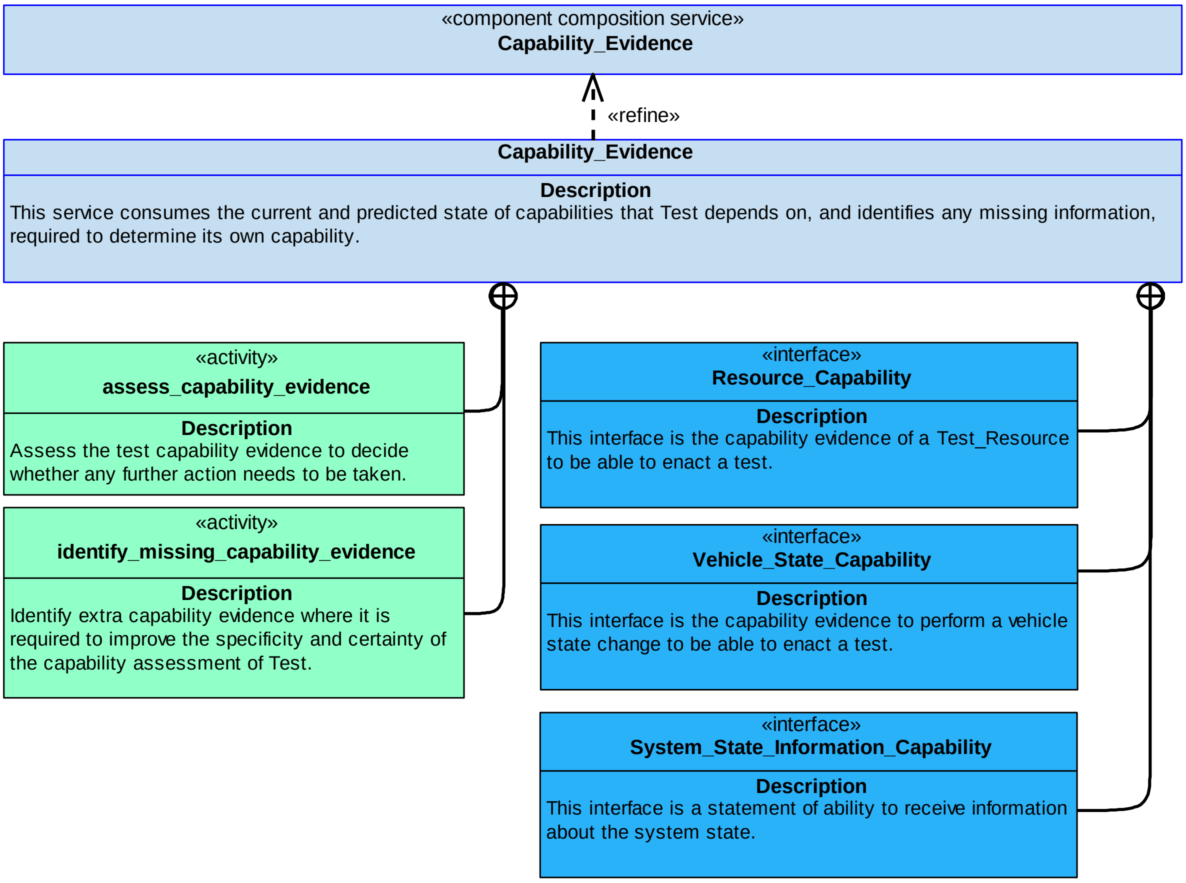

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the current and predicted state of capabilities that Test depends on, and identifies any missing information, required to determine its own capability.
«activity»
assess_capability_evidence
Description
Assess the test capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify extra capability evidence where it is required to improve the specificity and certainty of the capability assessment of Test.
«interface»
Resource_Capability
Description
This interface is the capability evidence of a Test_Resource to be able to enact a test.
«interface»
Vehicle_State_Capability
Description
This interface is the capability evidence to perform a vehicle state change to be able to enact a test.
«interface»
System_State_Information_Capability
Description
This interface is a statement of ability to receive information about the system state.

Figure 1107: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted state of capabilities that Test depends on, and identifies any missing information, required to determine its own capability.

Interfaces

Resource_Capability

This interface is the capability evidence of a Test_Resource to be able to enact a test.

System_State_Information_Capability

This interface is a statement of ability to receive information about the system state.

Attribute

information_type	The type of system state information required, e.g. weight on wheels status.

Vehicle_State_Capability

This interface is the capability evidence to perform a vehicle state change to be able to enact a test.

Attribute

vehicle_state_type	The type of vehicle state change required to enact a test, e.g. within a certain flight envelope.

Activities

assess_capability_evidence

Assess the test capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify extra capability evidence where it is required to improve the specificity and certainty of the capability assessment of Test.

5.4.2.63.7.2 Service Dependencies

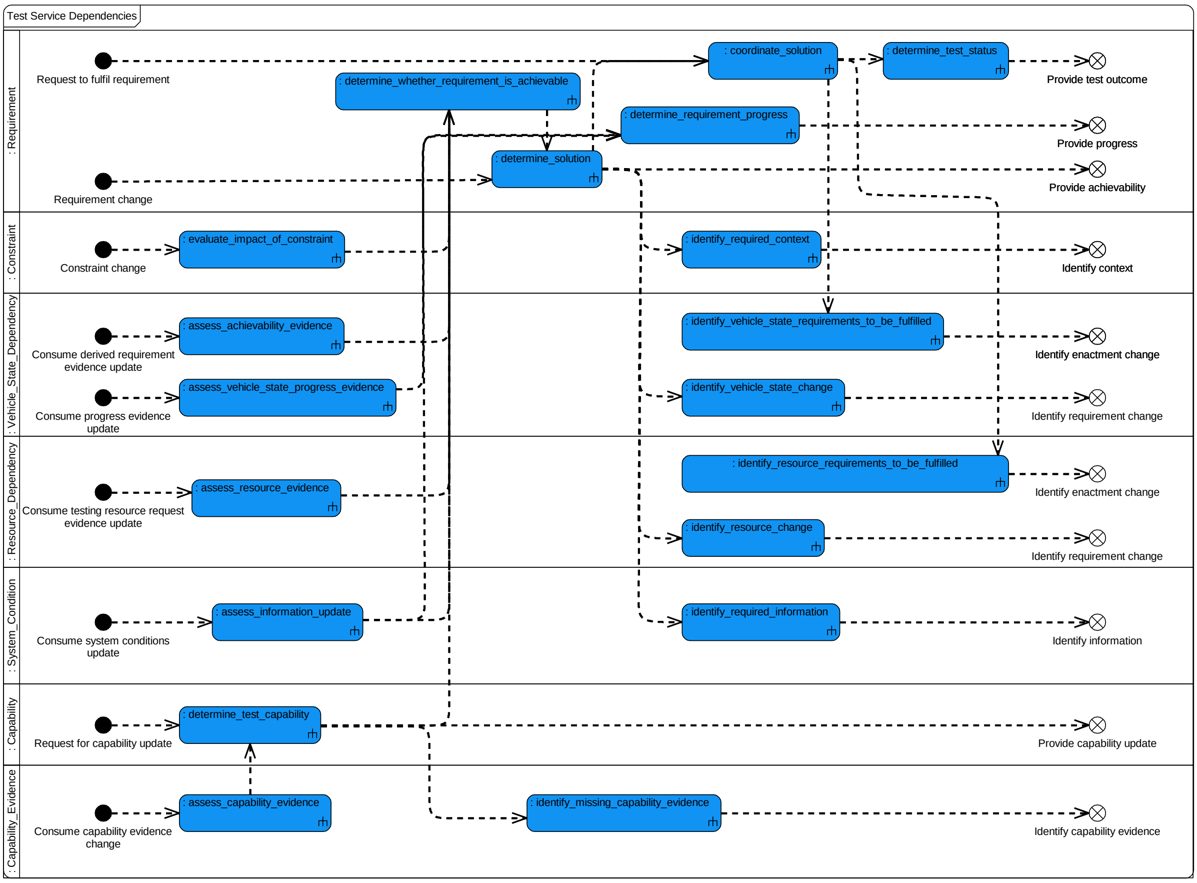

Test Service Dependencies
: Requirement
: Constraint
: Vehicle_State_Dependency
: Resource_Dependency
: System_Condition
: Capability
: Capability_Evidence
Request to fulfil requirement
: coordinate_solution
: determine_test_status
Provide test outcome
: determine_whether_requirement_is_achievable
: determine_requirement_progress
Provide progress
: determine_solution
Provide achievability
Requirement change
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume derived requirement evidence update
: assess_achievability_evidence
: identify_vehicle_state_requirements_to_be_fulfilled
Identify enactment change
Consume progress evidence update
: assess_vehicle_state_progress_evidence
: identify_vehicle_state_change
Identify requirement change
: identify_resource_requirements_to_be_fulfilled
Identify enactment change
Consume testing resource request evidence update
: assess_resource_evidence
: identify_resource_change
Identify requirement change
Consume system conditions update
: assess_information_update
: identify_required_information
Identify information
Request for capability update
: determine_test_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1108: Test Service Dependencies

5.4.2.64 Threats

5.4.2.64.1 Role

The role of Threats is to determine the level of risk posed by a tactical object to another tactical object.

5.4.2.64.2 Overview

Control Architecture

Threats is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a Threat entity is detected, or is predicted to exist within the mission space at the mission planning stage, then the Threats component is called upon to generate a continuous Threat_Assessment of the Threat entity against a Threatenable_Object using all available information sources such as sensor and intelligence data. It provides information about the Threat that exists in the battlespace, and provides a notification if the Threat exceeds a threat-level threshold.

Examples of Use

Threats will be needed in situations where hostile entities may exist, or are known to exist, and there is a need to generate a Threat_Assessment to assess the risks those entities pose as currently or potentially threatening, for example:

- Where a primary asset is being escorted by the vehicle(s).
- Where there is a direct risk of attack by hostile forces targeted at the vehicle(s).
- Where there may be no immediate risk of attack, but a Threat_Assessment is still required, e.g. during border patrol.

5.4.2.64.3 Service Summary

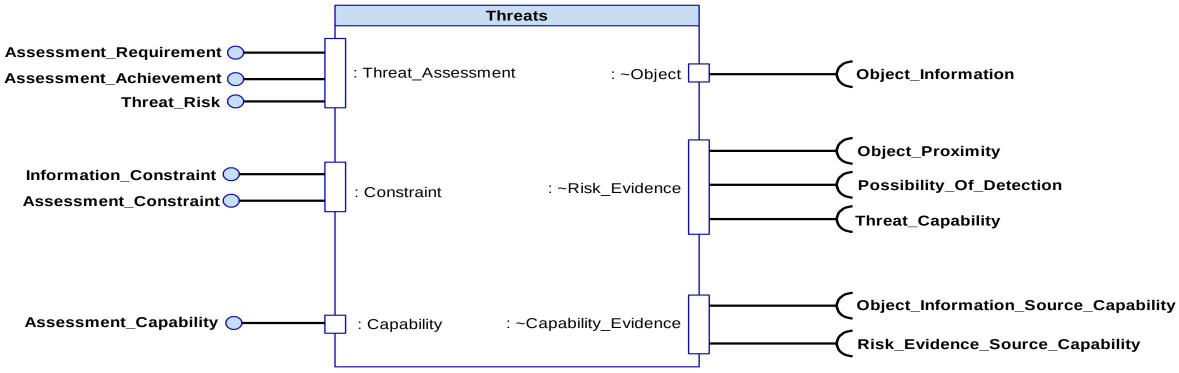


Figure 1109: Threats Service Summary

5.4.2.64.4 Responsibilities

capture_threat_assessment_requirement

- To capture the provided requirements for Threat_Assessments (e.g. the subject of the assessment (Threatenable_Object), region under consideration or frequency of assessments).

capture_threat_assessment_constraints

- To capture provided Constraints for generating Threat_Assessments (e.g. limitations on threat types to consider or restrictions on information considered).

identify_whether_requirement_remains_achievable

- To identify whether a Threat_Assessment is still achievable given current Capability and dependencies.

determine_threat_assessment_solution

- To determine a solution (taking account of what information should be considered, the priority of Threatenable_Objects to assess and the trigger for assessment) which meets given requirements and Constraints for generating Threat_Assessments using available tactical information.

assess_threats

- To generate Threat_Assessments for Threatenable_Objects based on the determined solution.

assess_threat_assessment_capability

- To assess the Capability to generate Threat_Assessments, for particular Threatenable_Objects and Threat_Types (based on, for example, the available tactical information services and assessments).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Threat_Assessment generation Capability, i.e. capability to determine tactical information (e.g. observability or behaviour).

predict_capability_progression

- To predict the progression of Threat_Assessment generation Capability over time and with use.

5.4.2.64.5 Subject Matter Semantics

The subject matter of Threats is entities in the external environment that have the potential to disrupt, degrade, damage or destroy.

Exclusions

The subject matter of Threats does not include:

- Identification, e.g. associated platform identification.
- Classification of entities, e.g. hostile / friendly / neutral.
- Determining the risk of the Threatenable_Object being observed.
- Determining the vulnerability of the Threatenable_Object.

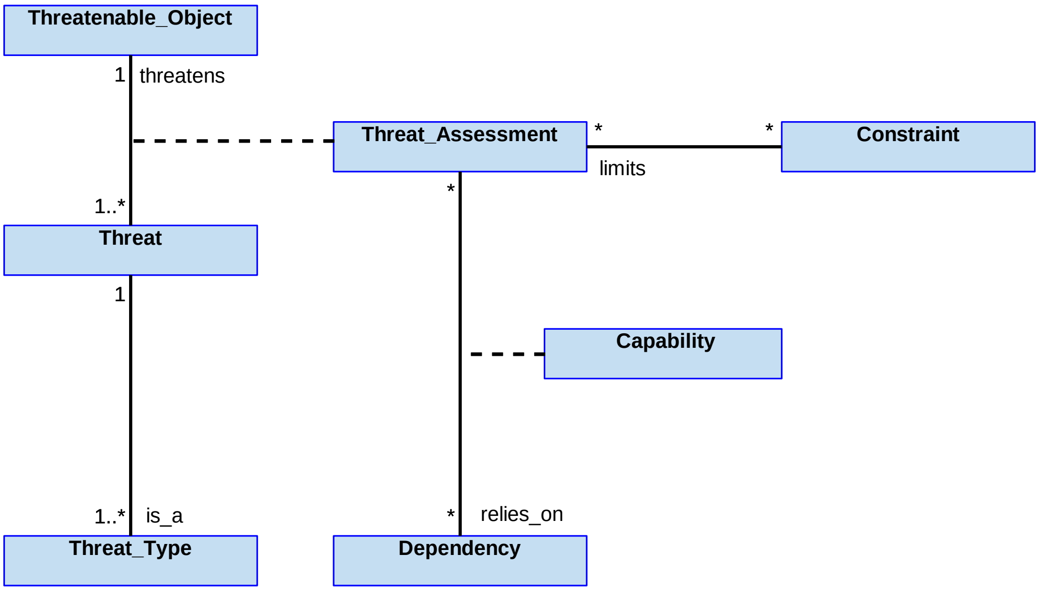


Figure 1110: Threats Semantics

5.4.2.64.5.1 Entities

Capability

The range of Threat_Assessments that the component is able to perform with its available dependencies.

Constraint

An externally imposed restriction which limits when or how a Threat_Assessment can be generated.

Dependency

Something which the component relies on in order to generate Threat_Assessments, e.g. tactical information.

Threat

A man-made entity in the external environment which has the potential to disrupt, degrade, damage or destroy, regardless of whether it is actually threatening at any particular instant.

Threat_Assessment

The determination of Threats and the level of risk they pose towards a Threatenable_Object, conditions under which this risk applies and confidence in this risk.

Threatenable_Object

A tactical object (e.g. own air vehicle, flight member or formation) which can be threatened by a Threat.

Threat_Type

A type of Threat (e.g. search radar, tracking radar, laser or missile).

5.4.2.64.6 Design Rationale

5.4.2.64.6.1 Assumptions

- The component will have access to information about tactical objects, their location and their capabilities against the Threatenable_Object.

5.4.2.64.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Threats:

- Tactical Information - To be flexible enough to integrate with other tactical information components.
- Data Driving - To cope with the particular types of tactical objects considered during Threat_Assessments, including Threats and Threatenable_Objects, varying per mission.

Extensions

- Extensions could also be used to extend for particular threat assessments (see Component Extensions PYRAMID concept).

Exploitation Considerations

- The complexity of Threat_Assessments, in particular the tactical information about threats and threatenable objects assessed, is expected to vary based on the requirements for the Exploiting Programme. For simpler deployments, the type of threat and its distance to the air vehicle could be considered. For more complex deployments, the component may reason about lots of different aspects of tactical information (e.g. whether the object is able to observe ownship or whether the object is part of a network).
- The determination of Threats is based on the following information. Note that some of this information is sourced from other components, and is not determined by the Threats component itself.
    - The probability of a Threatenable_Object encountering a Threat under defined circumstances.
    - The Threat's capability to locate, identify and engage the Threatenable_Object in an operational environment.

- The extent to how susceptible (i.e. physical damage/destruction or functional degradation/disruption) a Threatenable_Object is to a Threat.
- The Threat's capability to avoid detection.
- The Threat's capability to relocate.
- The Threat's associated capabilities (e.g. the Threat - search radar, is associated with a separately located weapon system).

5.4.2.64.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

Failure of this component could result in:

- Incorrectly identifying an object as a threat when it is not a threat. This could result in countermeasures being deployed against it when not appropriate. However, it is expected that other components (e.g. Interlocks or Authorisation) will be relied upon to prevent countermeasures being enacted when not safe, independently of this component. In the case where weapons are deployed to defeat a threat it is expected that humans would have confirmed the threat using raw sensor data, for example. In the case of expendables (e.g. chaff and flare) release, where deployment is time critical it is expected that a human would have pre-authorised release only in locations where it is considered an acceptable risk. Therefore, DAL C is appropriate for this component as the likelihood of any catastrophic accidents is reduced to an acceptable level by other components and/or humans.
- Failure to defeat a threat due to incorrectly identifying a threat (so an ineffective countermeasure is enacted) or determining a genuine threat is not hostile. Whilst this could result in loss of the air vehicle or crew fatality, failure to defeat external physical threats are not considered within the scope of safety analysis.

5.4.2.64.6.4 Security Considerations

The indicative security classification is SCEO.

This component determines whether an entity poses a threat to the Exploiting Platform or another entity, as a minimum based on the type of entity and its distance from the subject, but potentially based on complex tactical information. Basic threat information is considered likely to be SCEO as they may be shared with coalition forces, however some algorithms and intelligence information may carry a higher classification. If this is the case, there may be instances in different security domains; these instances may need to communicate with each other to provide a full threat assessment. If so, separation will be handled externally to the component. Any loss of integrity or availability of this component may lead to the Exploiting Platform placing itself at risk. The confidentiality, integrity and availability requirements of the Exploiting Platform will need to reflect this. Where algorithms are data-driven, the associated configuration data will also carry appropriate confidentiality requirements.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of threat assessments made during the course of a mission.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.64.7 Services

5.4.2.64.7.1 Service Definitions

5.4.2.64.7.1.1 Threat_Assessment

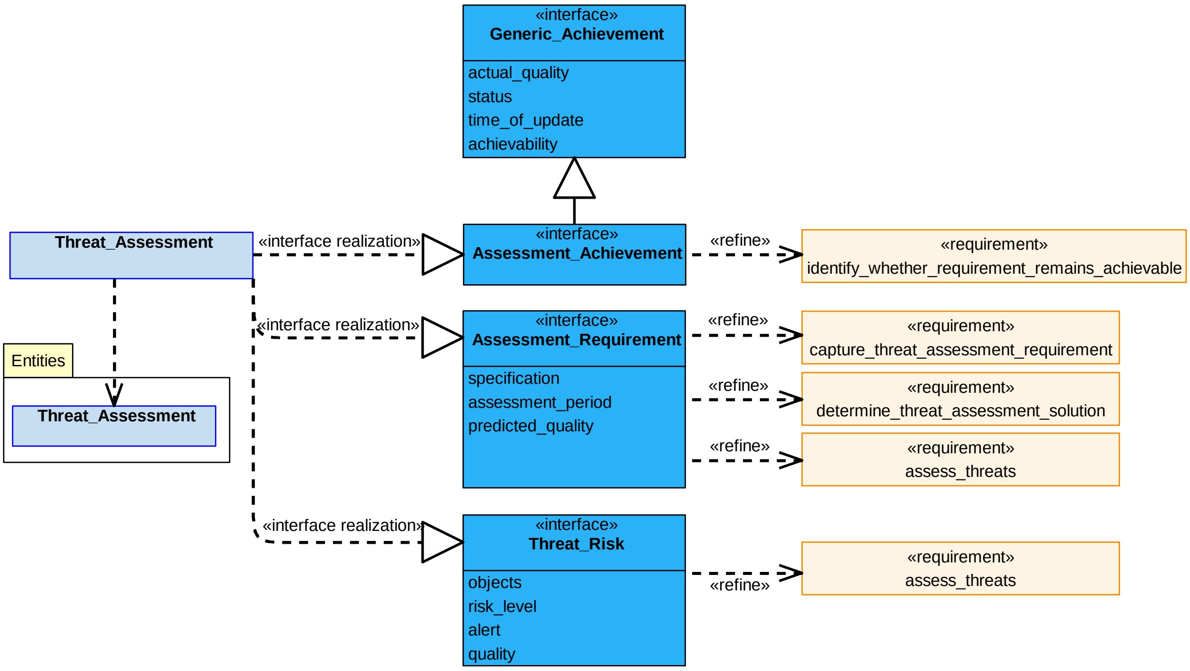


Figure 1111: Threat_Assessment Service Definition

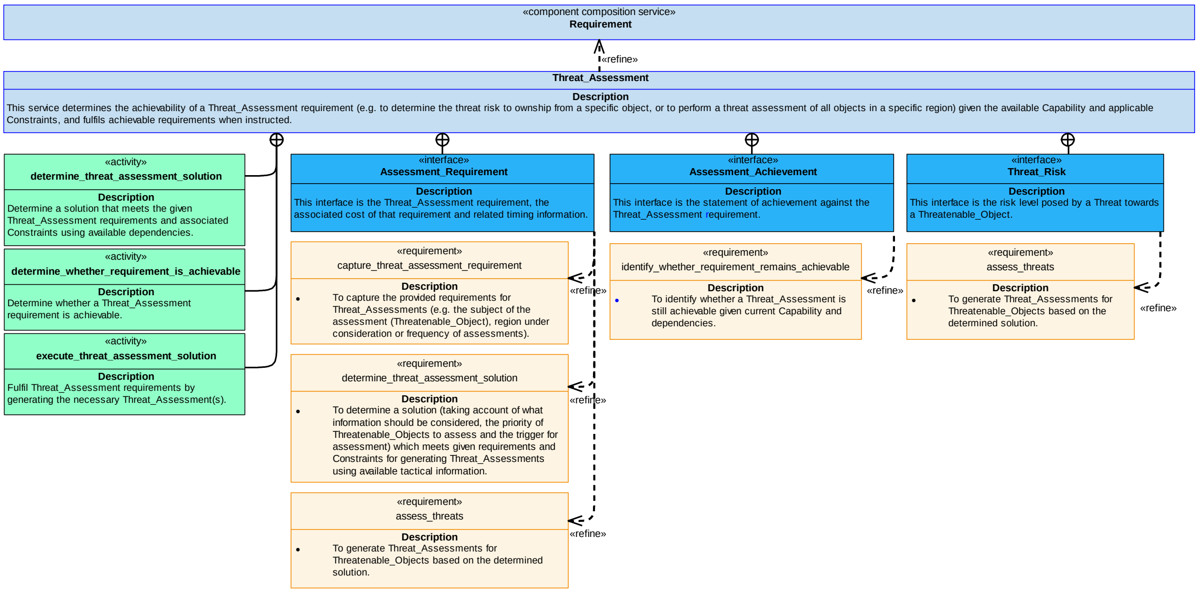

«component composition service»
Requirement
«refine»
Threat_Assessment
Description
This service determines the achievability of a Threat_Assessment requirement (e.g. to determine the threat risk to ownship from a specific object, or to perform a threat assessment of all objects in a specific region) given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_threat_assessment_solution
Description
Determine a solution that meets the given Threat_Assessment requirements and associated Constraints using available dependencies.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether a Threat_Assessment requirement is achievable.
«activity»
execute_threat_assessment_solution
Description
Fulfil Threat_Assessment requirements by generating the necessary Threat_Assessment(s).
«interface»
Assessment_Requirement
Description
This interface is the Threat_Assessment requirement, the associated cost of that requirement and related timing information.
«requirement»
capture_threat_assessment_requirement
Description
To capture the provided requirements for Threat_Assessments (e.g. the subject of the assessment (Threatenable_Object), region under consideration or frequency of assessments).
«refine»
«requirement»
determine_threat_assessment_solution
Description
To determine a solution (taking account of what information should be considered, the priority of Threatenable_Objects to assess and the trigger for assessment) which meets given requirements and Constraints for generating Threat_Assessments using available tactical information.
«refine»
«requirement»
assess_threats
Description
To generate Threat_Assessments for Threatenable_Objects based on the determined solution.
«refine»
«interface»
Assessment_Achievement
Description
This interface is the statement of achievement against the Threat_Assessment requirement.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Threat_Assessment is still achievable given current Capability and dependencies.
«refine»
«interface»
Threat_Risk
Description
This interface is the risk level posed by a Threat towards a Threatenable_Object.
«requirement»
assess_threats
Description
To generate Threat_Assessments for Threatenable_Objects based on the determined solution.
«refine»

Figure 1112: Threat_Assessment Service Policy

Threat_Assessment

This service determines the achievability of a Threat_Assessment requirement (e.g. to determine the threat risk to ownship from a specific object, or to perform a threat assessment of all objects in a specific region) given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

Interfaces

Assessment_Achievement

This interface is the statement of achievement against the Threat_Assessment requirement.

Assessment_Requirement

This interface is the Threat_Assessment requirement, the associated cost of that requirement and related timing information.

Attributes

specification	The definition of the Threat_Assessment requirement (e.g. determine the threat risk to one aircraft from another or determine the threat risk to ownship within a specific region).
assessment_period	Timing information over which a Threat_Assessment is to be determined.
predicted_quality	The predicted quality of the threat risk information resulting from a Threat_Assessment.

Threat_Risk

This interface is the risk level posed by a Threat towards a Threatenable_Object.

Attributes

objects	The two objects that are the subject of a Threat_Assessment.
risk_level	The level of risk posed by a Threat towards a Threatenable_Object.
alert	An indication that a threat risk threshold has been passed.
quality	The quality of the level of risk information in a Threat_Assessment.

Activities

determine_threat_assessment_solution

Determine a solution that meets the given Threat_Assessment requirements and associated Constraints using available dependencies.

determine_whether_requirement_is_achievable

Determine whether a Threat_Assessment requirement is achievable.

execute_threat_assessment_solution

Fulfil Threat_Assessment requirements by generating the necessary Threat_Assessment(s).

5.4.2.64.7.1.2 Object

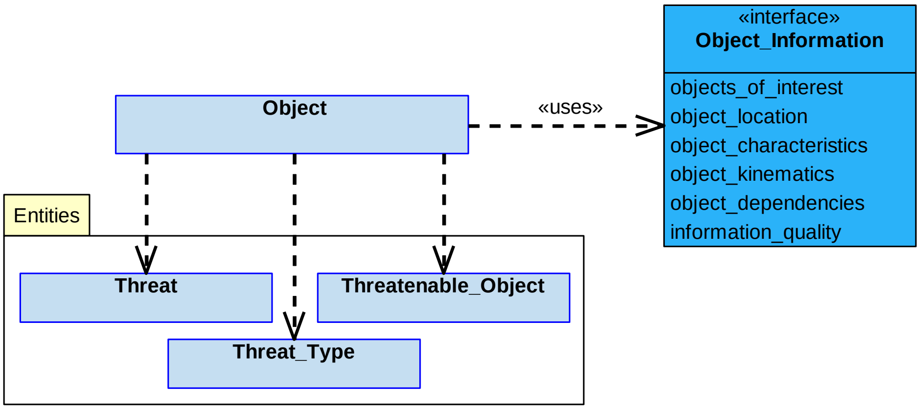


Figure 1113: Object Service Definition

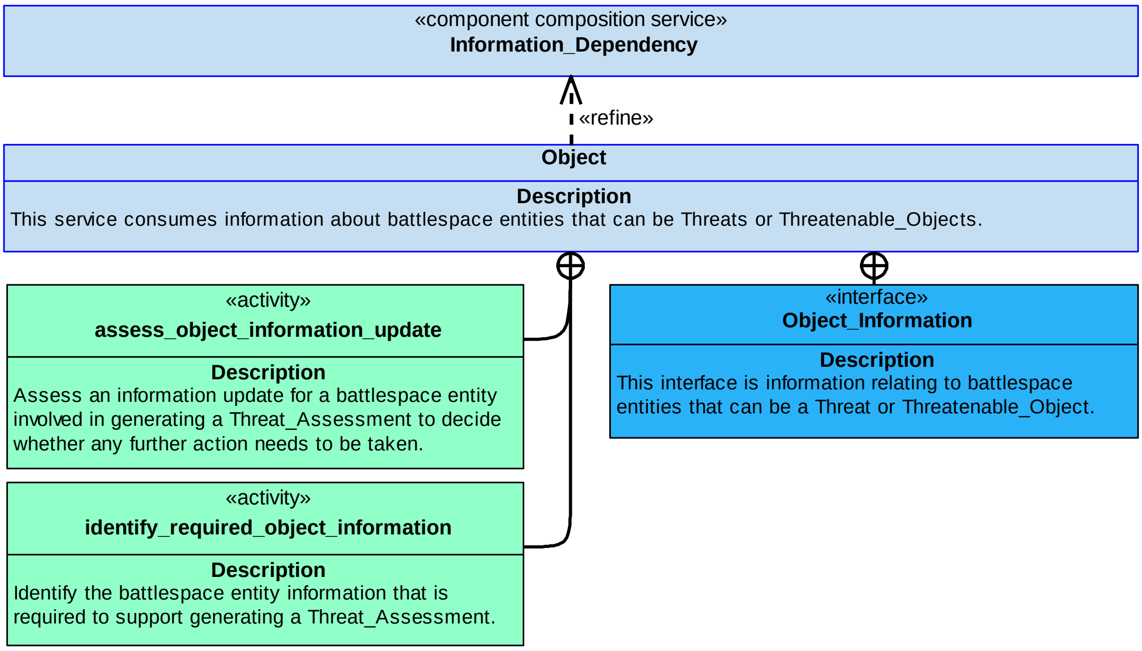

«component composition service»
Information_Dependency
«refine»
Object
Description
This service consumes information about battlespace entities that can be Threats or Threatenable_Objects.
«activity»
assess_object_information_update
Description
Assess an information update for a battlespace entity involved in generating a Threat_Assessment to decide whether any further action needs to be taken.
«activity»
identify_required_object_information
Description
Identify the battlespace entity information that is required to support generating a Threat_Assessment.
«interface»
Object_Information
Description
This interface is information relating to battlespace entities that can be a Threat or Threatenable_Object.

Figure 1114: Object Service Policy

Object

This service consumes information about battlespace entities that can be Threats or Threatenable_Objects.

<u>Interface</u>

Object_Information

This interface is information relating to battlespace entities that can be a Threat or Threatenable_Object.

<u>Attributes</u>

objects_of_interest	The objects that are relevant to a Threat_Assessment requirement.

object_location	Where an object is located (e.g. latitude / longitude / altitude).
object_characteristics	The characteristics of an object (e.g. type, behaviour or allegiance).
object_kinematics	Information relating to an objects motion which may include trajectory, speed, accelerations (x/y/z), altitude, maximum speed, etc.
object_dependencies	The dependencies between objects that can affect their behaviour, e.g. the dependence between a missile launcher on an associated tracking radar.
information_quality	The quality of object information.

Activities

assess_object_information_update

Assess an information update for a battlespace entity involved in generating a Threat_Assessment to decide whether any further action needs to be taken.

identify_required_object_information

Identify the battlespace entity information that is required to support generating a Threat_Assessment.

5.4.2.64.7.1.3 Risk_Evidence

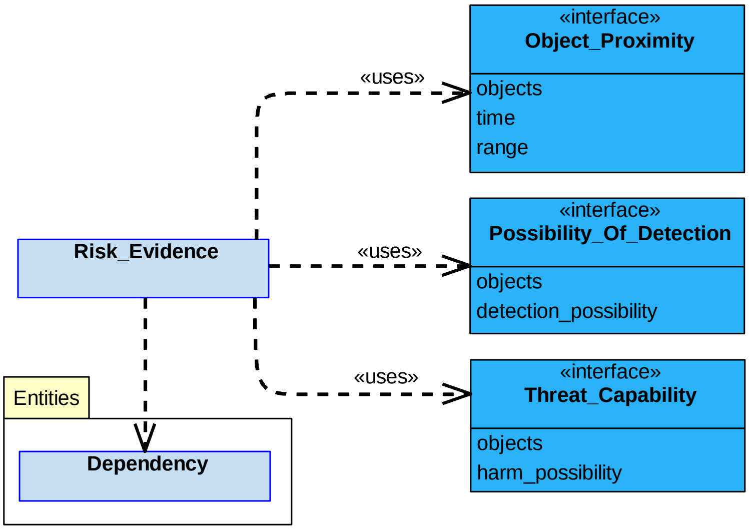


Figure 1115: Risk_Evidence Service Definition

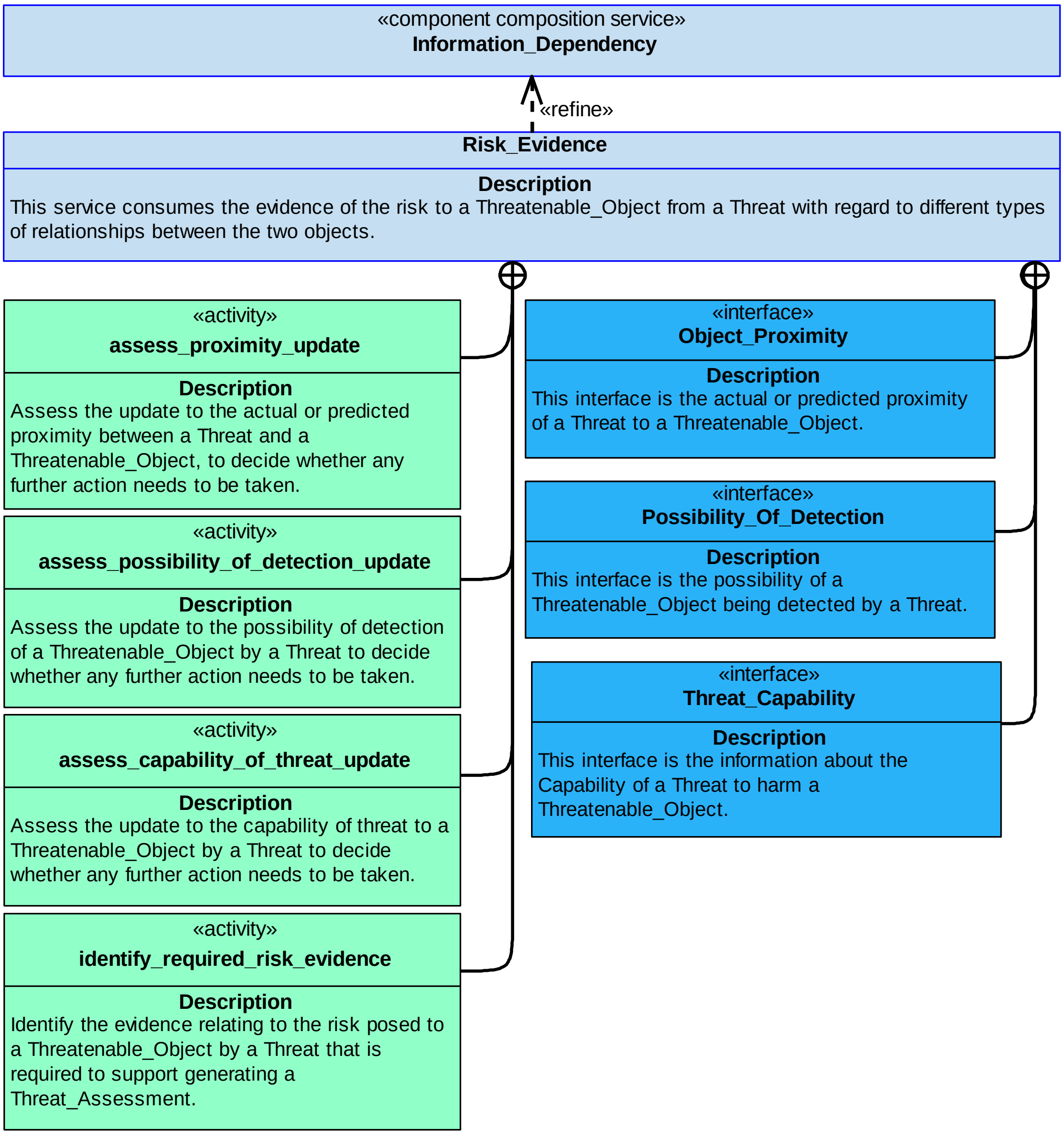

«component composition service»
Information_Dependency
«refine»
Risk_Evidence
Description
This service consumes the evidence of the risk to a Threatenable_Object from a Threat with regard to different types of relationships between the two objects.
«activity»
assess_proximity_update
Description
Assess the update to the actual or predicted proximity between a Threat and a Threatenable_Object, to decide whether any further action needs to be taken.
«activity»
assess_possibility_of_detection_update
Description
Assess the update to the possibility of detection of a Threatenable_Object by a Threat to decide whether any further action needs to be taken.
«activity»
assess_capability_of_threat_update
Description
Assess the update to the capability of threat to a Threatenable_Object by a Threat to decide whether any further action needs to be taken.
«activity»
identify_required_risk_evidence
Description
Identify the evidence relating to the risk posed to a Threatenable_Object by a Threat that is required to support generating a Threat_Assessment.
«interface»
Object_Proximity
Description
This interface is the actual or predicted proximity of a Threat to a Threatenable_Object.
«interface»
Possibility_Of_Detection
Description
This interface is the possibility of a Threatenable_Object being detected by a Threat.
«interface»
Threat_Capability
Description
This interface is the information about the Capability of a Threat to harm a Threatenable_Object.

Figure 1116: Risk_Evidence Service Policy

Risk_Evidence

This service consumes the evidence of the risk to a Threatenable_Object from a Threat with regard to different types of relationships between the two objects.

Interfaces

Object_Proximity

This interface is the actual or predicted proximity of a Threat to a Threatenable_Object.

Attributes

objects	The objects that are the subject of a proximity assessment.
time	The time period over which the proximity assessment is valid.

range	The current range between two objects or the predicted range between two objects, based on their trajectories over a specified time period.

Possibility_Of_Detection

This interface is the possibility of a Threatenable_Object being detected by a Threat.

Attributes

objects	The two objects that are the subject of a possibility of detection assessment.
detection_possibility	The possibility that an object will be detected by another object.

Threat_Capability

This interface is the information about the Capability of a Threat to harm a Threatenable_Object.

Attributes

objects	The two objects that are the subject of a possibility of harm assessment.
harm_possibility	The possibility that an object will be harmed by another object.

Activities

assess_proximity_update

Assess the update to the actual or predicted proximity between a Threat and a Threatenable_Object, to decide whether any further action needs to be taken.

assess_possibility_of_detection_update

Assess the update to the possibility of detection of a Threatenable_Object by a Threat to decide whether any further action needs to be taken.

assess_capability_of_threat_update

Assess the update to the capability of threat to a Threatenable_Object by a Threat to decide whether any further action needs to be taken.

identify_required_risk_evidence

Identify the evidence relating to the risk posed to a Threatenable_Object by a Threat that is required to support generating a Threat_Assessment.

5.4.2.64.7.1.4 Constraint

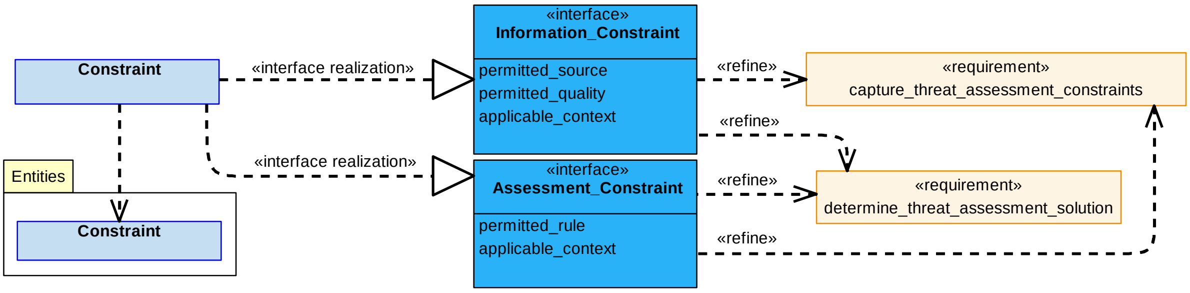


Figure 1117: Constraint Service Definition

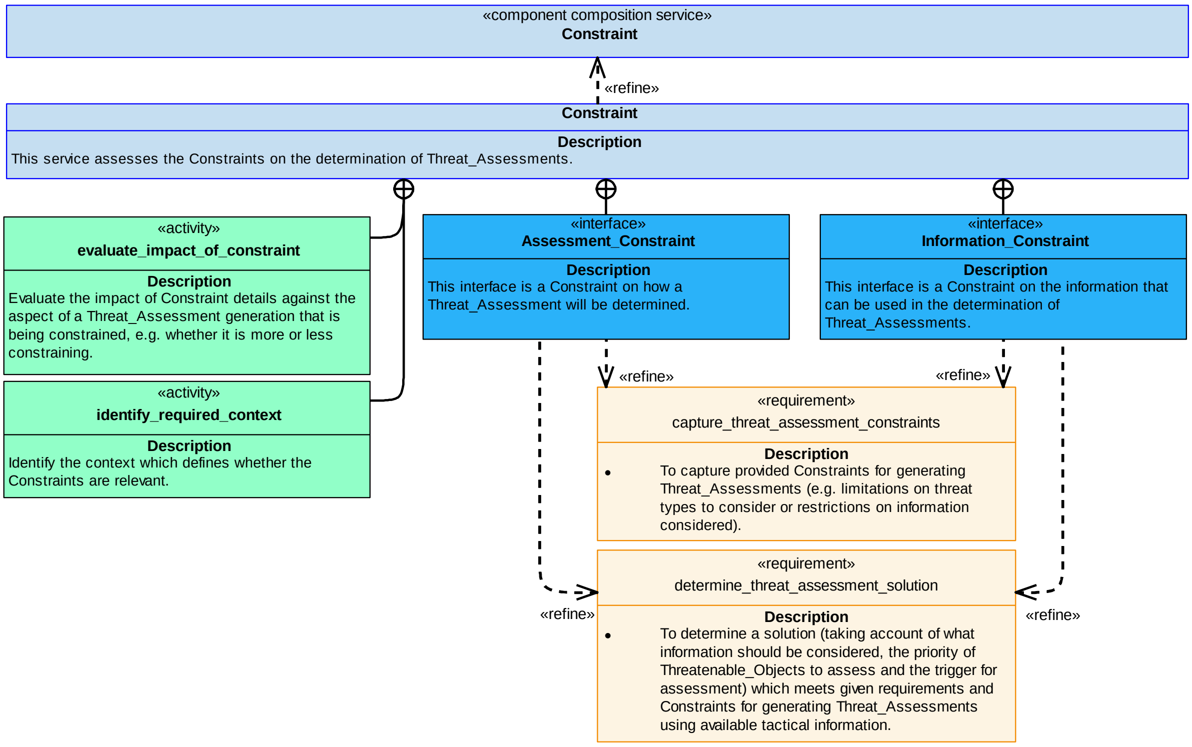

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Constraints on the determination of Threat_Assessments.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of a Threat_Assessment generation that is being constrained, e.g. whether it is more or less constraining.
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Assessment_Constraint
Description
This interface is a Constraint on how a Threat_Assessment will be determined.
«interface»
Information_Constraint
Description
This interface is a Constraint on the information that can be used in the determination of Threat_Assessments.
«refine»
«refine»
«requirement»
capture_threat_assessment_constraints
Description
To capture provided Constraints for generating Threat_Assessments (e.g. limitations on threat types to consider or restrictions on information considered).
«requirement»
determine_threat_assessment_solution
Description
To determine a solution (taking account of what information should be considered, the priority of Threatenable_Objects to assess and the trigger for assessment) which meets given requirements and Constraints for generating Threat_Assessments using available tactical information.
«refine»
«refine»

Figure 1118: Constraint Service Policy

Constraint

This service assesses the Constraints on the determination of Threat_Assessments.

Interfaces

Assessment_Constraint

This interface is a Constraint on how a Threat_Assessment will be determined.

Attributes

permitted_rule	The rules that are permitted to be used in generating a Threat_Assessment.
applicable_context	The context in which the Constraint is applicable.

Information_Constraint

This interface is a Constraint on the information that can be used in the determination of Threat_Assessments.

Attributes

permitted_source	The restricted information sources that are permitted to be used in generating a Threat_Assessment.
permitted_quality	The quality of information required to permit the information to be used in performing a Threat_Assessment.
applicable_context	The context in which the Constraint is applicable.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of a Threat_Assessment generation that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.64.7.1.5 Capability

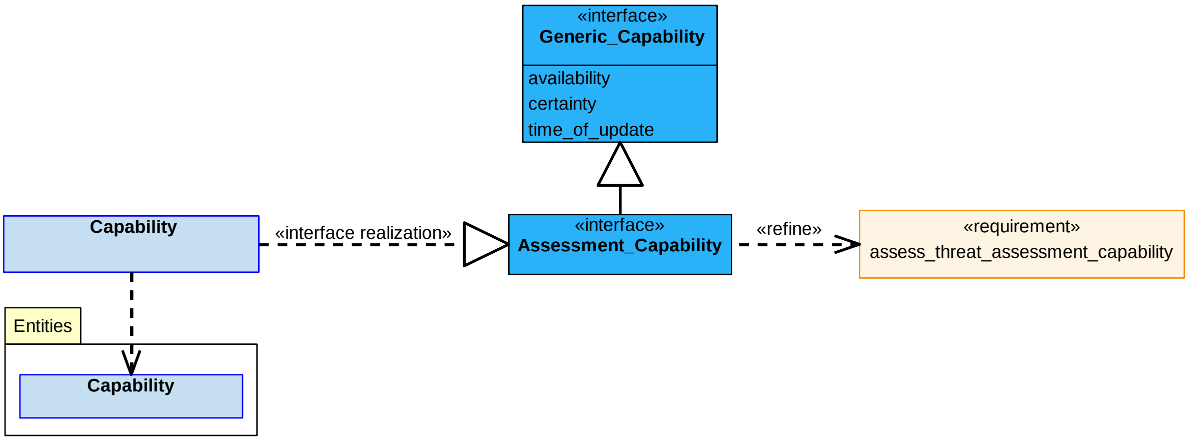
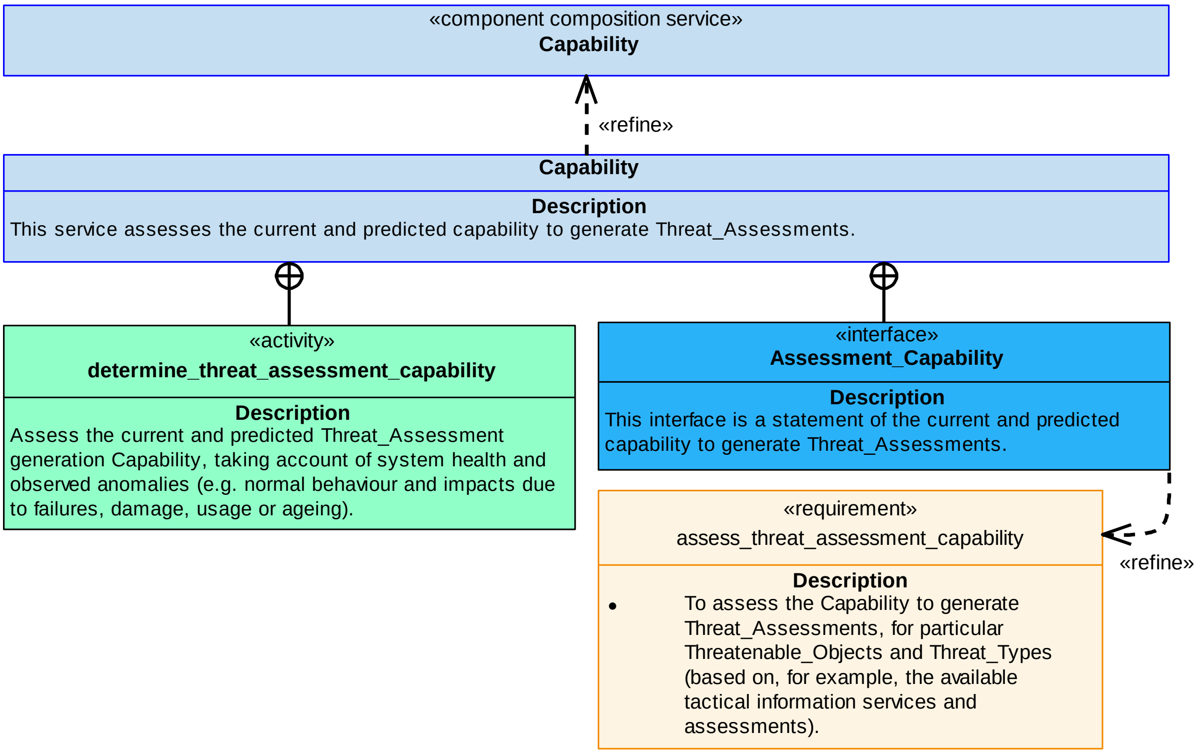


Figure 1119: Capability Service Definition


«component composition service»
Capability
«refine»
Capability
«activity»
determine_threat_assessment_capability
«interface»
Assessment_Capability
«requirement»
assess_threat_assessment_capability
«refine»

Capability

Description
This service assesses the current and predicted capability to generate Threat_Assessments.

«activity» determine_threat_assessment_capability

Description
Assess the current and predicted Threat_Assessment generation Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

«interface» Assessment_Capability

Description
This interface is a statement of the current and predicted capability to generate Threat_Assessments.

«requirement» assess_threat_assessment_capability

Description
- To assess the Capability to generate Threat_Assessments, for particular Threatenable_Objects and Threat_Types (based on, for example, the available tactical information services and assessments).

Figure 1120: Capability Service Policy

Capability

This service assesses the current and predicted capability to generate Threat_Assessments.

Interface

Assessment_Capability

This interface is a statement of the current and predicted capability to generate Threat_Assessments.

Activity

determine_threat_assessment_capability

Assess the current and predicted Threat_Assessment generation Capability, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.64.7.1.6 Capability_Evidence

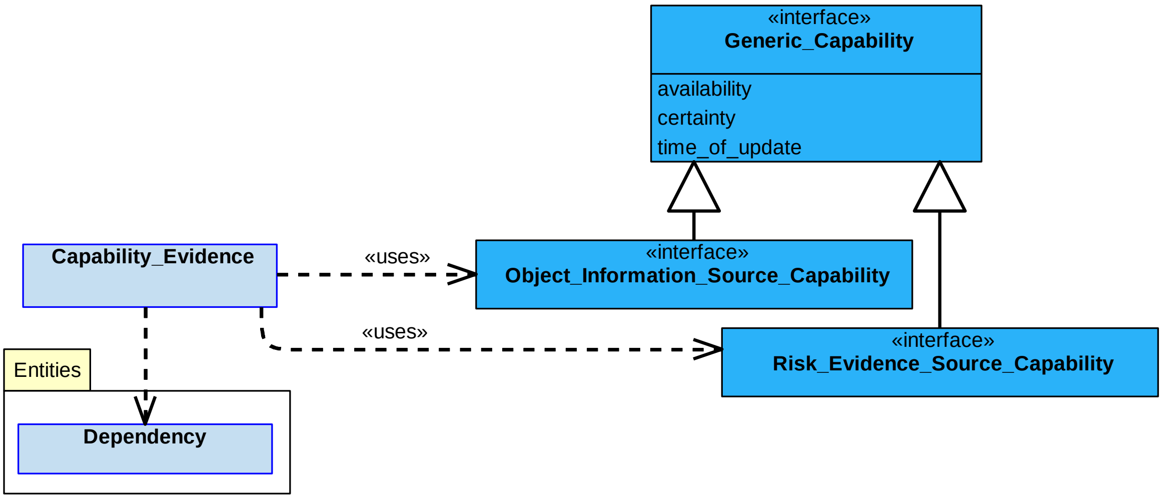


Figure 1121: Capability_Evidence Service Definition

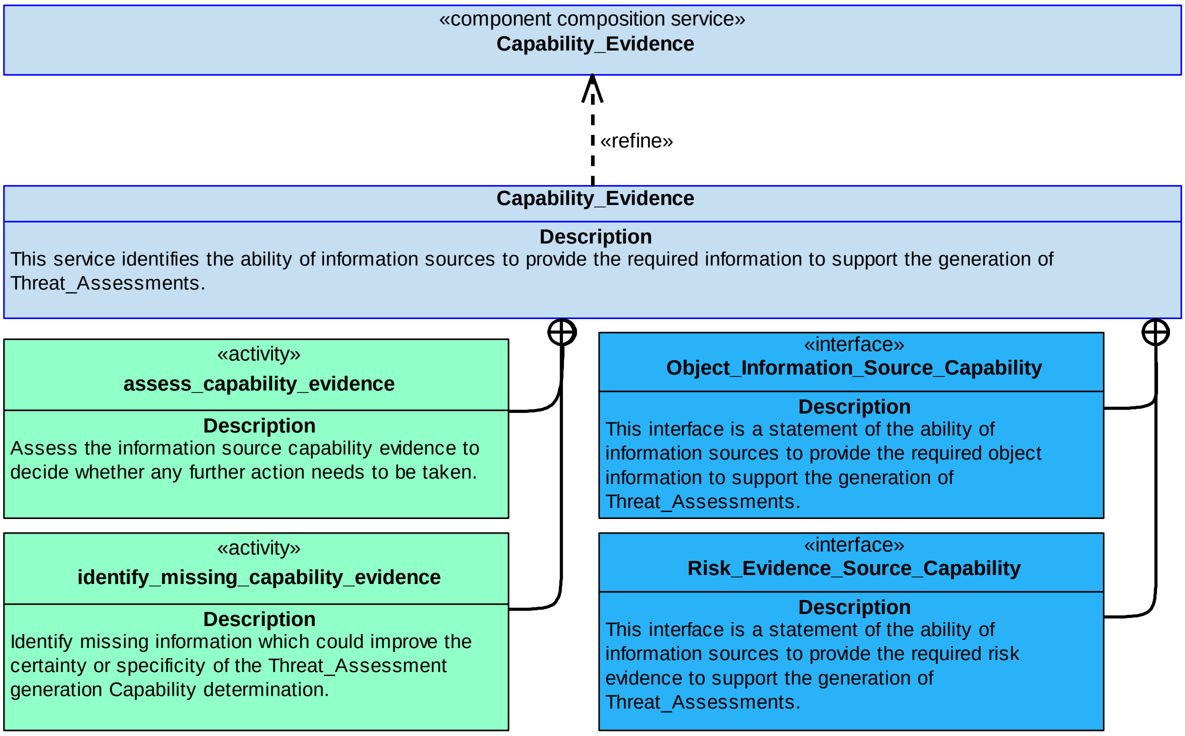

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service identifies the ability of information sources to provide the required information to support the generation of Threat_Assessments.
«activity»
assess_capability_evidence
Description
Assess the information source capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify missing information which could improve the certainty or specificity of the Threat_Assessment generation Capability determination.
«interface»
Object_Information_Source_Capability
Description
This interface is a statement of the ability of information sources to provide the required object information to support the generation of Threat_Assessments.
«interface»
Risk_Evidence_Source_Capability
Description
This interface is a statement of the ability of information sources to provide the required risk evidence to support the generation of Threat_Assessments.

Figure 1122: Capability_Evidence Service Policy

Capability_Evidence

This service identifies the ability of information sources to provide the required information to support the generation of Threat_Assessments.

Interfaces

Object_Information_Source_Capability

This interface is a statement of the ability of information sources to provide the required object information to support the generation of Threat_Assessments.

Risk_Evidence_Source_Capability

This interface is a statement of the ability of information sources to provide the required risk evidence to support the generation of Threat_Assessments.

Activities

assess_capability_evidence

Assess the information source capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify missing information which could improve the certainty or specificity of the Threat_Assessment generation Capability determination.

5.4.2.64.7.2 Service Dependencies

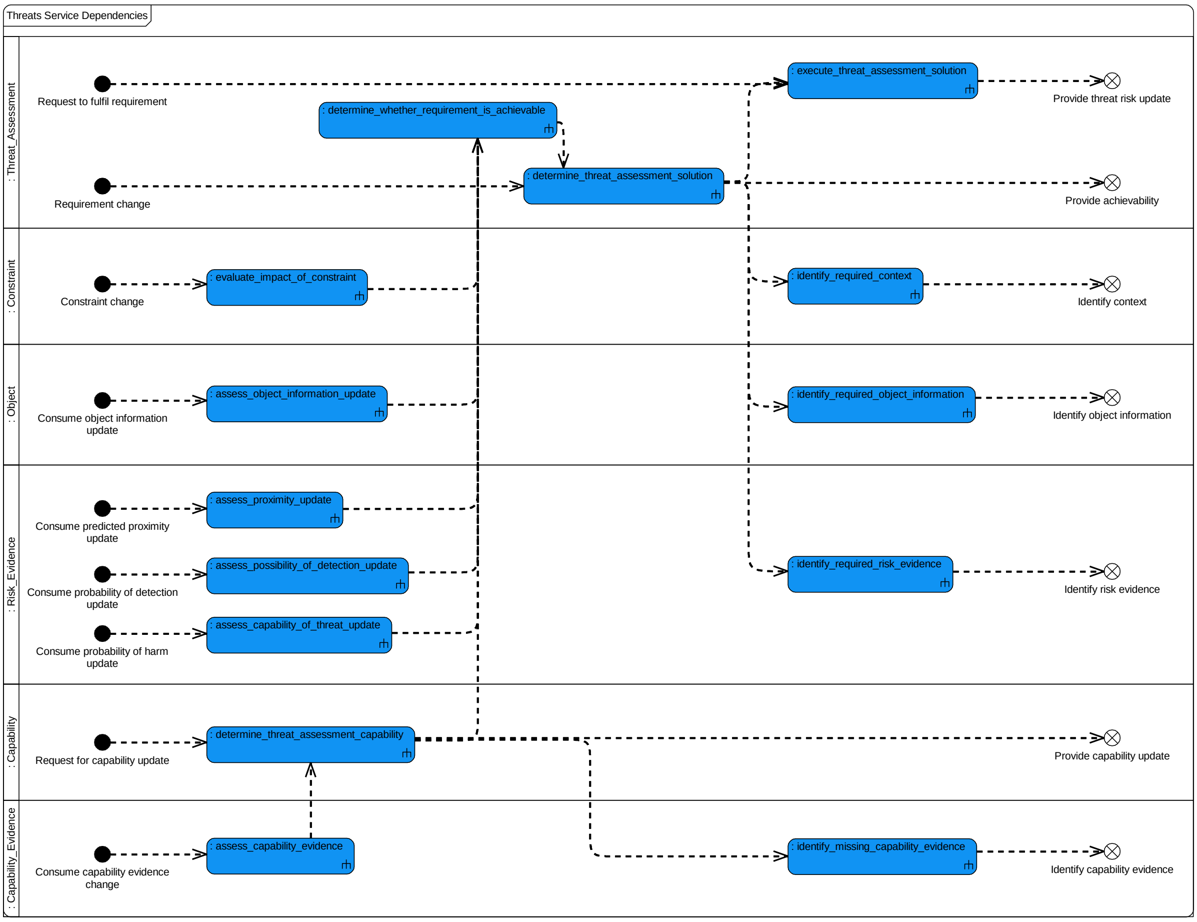

Threats Service Dependencies
: Threat_Assessment
: Constraint
: Object
: Risk_Evidence
: Capability
: Capability_Evidence
Request to fulfil requirement
Requirement change
: determine_whether_requirement_is_achievable
: determine_threat_assessment_solution
: execute_threat_assessment_solution
Provide threat risk update
Provide achievability
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume object information update
: assess_object_information_update
: identify_required_object_information
Identify object information
Consume predicted proximity update
: assess_proximity_update
Consume probability of detection update
: assess_possibility_of_detection_update
: identify_required_risk_evidence
Identify risk evidence
Consume probability of harm update
: assess_capability_of_threat_update
Request for capability update
: determine_threat_assessment_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1123: Threats Service Dependencies

5.4.2.65 Trajectory Prediction

5.4.2.65.1 Role

The role of Trajectory Prediction is to predict a spatial trajectory or trajectories for real world objects.

5.4.2.65.2 Overview

Control Architecture

Trajectory Prediction is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

Following receipt of a Requirement to provide a Trajectory_Prediction_Solution for an object, a single, or set of, predicted trajectories will be provided using defined Manoeuvring_Characteristics for that object. Accuracy and confidence levels will be ascribed to the Trajectory_Prediction_Solution in order to meet each Measurement_Criterion.

Examples of Use

Trajectory Prediction could be used to determine a Trajectory_Prediction_Solution for:

- Unguided objects following a ballistic trajectory, such as bombs or gun rounds.
- Guided objects which could follow one of a large or infinite set of possible trajectories, such as guided missiles, guided bombs, aircraft, ground vehicles, surface ships, subsurface vessels, etc. For example, aircraft trajectories are needed to assist in evading air-to-air collisions, and missile or bomb trajectories are needed to assist in evading interception with ownship and in weapon guidance.
- Signal paths, such as radio frequency trajectories.

5.4.2.65.3 Service Summary

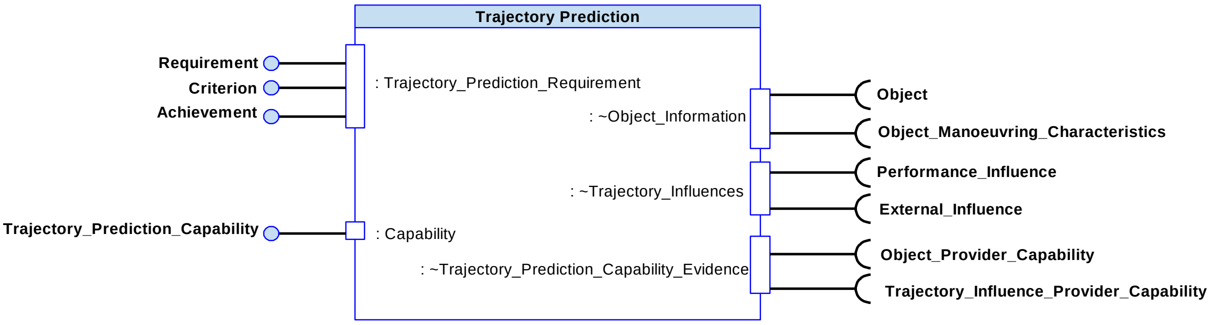

Trajectory Prediction
Requirement
Criterion
Achievement
: Trajectory_Prediction_Requirement
: ~Object_Information
Object
Object_Manoeuvring_Characteristics
Performance_Influence
: ~Trajectory_Influences
External_Influence
Trajectory_Prediction_Capability
: Capability
Object_Provider_Capability
: ~Trajectory_Prediction_Capability_Evidence
Trajectory_Influence_Provider_Capability

Figure 1124: Trajectory Prediction Service Summary

5.4.2.65.4 Responsibilities

capture_requirements_for_trajectory_prediction

- To capture the Requirements to be satisfied by the Trajectory_Prediction_Solution.

capture_measurement_criteria_for_trajectory_prediction

- To capture Measurement_Criterion/ criteria for Trajectory_Prediction_Solutions.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Capability and conditions.

determine_trajectory_prediction_solution_for_object

- To determine the Trajectory_Prediction_Solution for an Object.

capture_object_information

- To capture information about an Object's spatial state.

capture_manoeuvring_characteristics

- To capture the Manoeuvring_Characteristics of an Object or type of object.

identify_progress

- To identify the progress against a Requirement.

assess_trajectory_prediction_capability

- To assess the Capability to predict trajectories, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information that could improve the accuracy or confidence level of the Trajectory prediction Capability assessment.

predict_capability_progression

- To predict the progression of the component's Capability over time and with use.

5.4.2.65.5 Subject Matter Semantics

The subject matter of Trajectory Prediction is the prediction of a trajectory or trajectories, including the associated accuracy and confidence levels.

Exclusions

The subject matter of Trajectory Prediction does not include:

- Planning and enactment of trajectories (this includes desired or expected trajectories that result from planning or enactment, such as those used to support guidance and control algorithms).
- Comparison of intended trajectories to actual trajectories.
- Instantaneous trajectories e.g. a current velocity vector.
- Any other examples of trajectories that are not predictions.

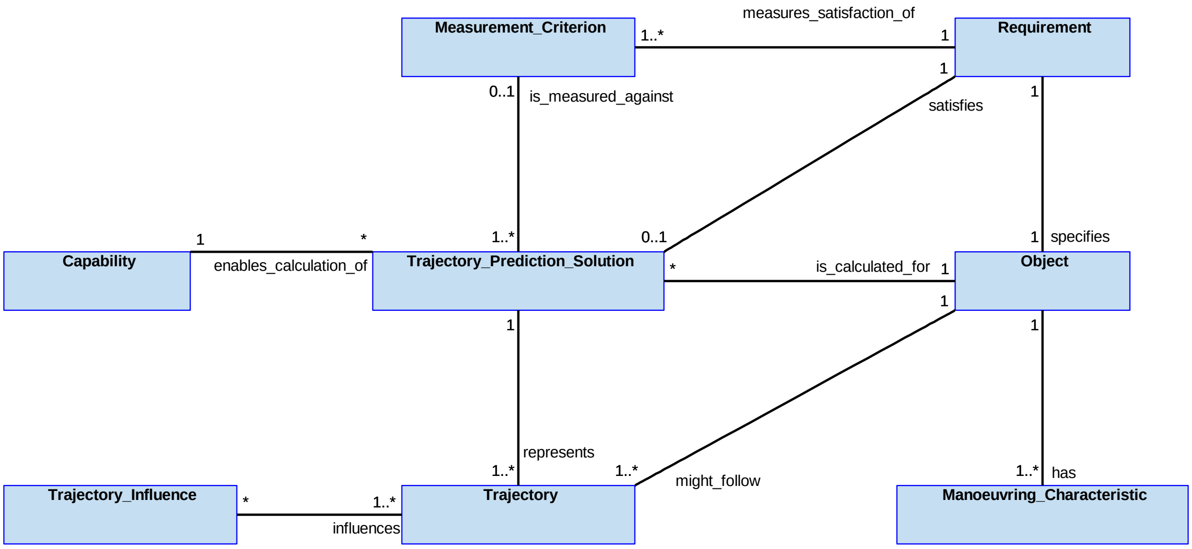


Figure 1125: Trajectory Prediction Semantics

5.4.2.65.5.1 Entities

Measurement_Criterion

A criterion against which the Trajectory_Prediction_Solution is measured (e.g. accuracy or limits of the predicted trajectories and confidence of the prediction).

Object

An object that is movable in space, including its current spatial state.

Requirement

A requirement to calculate a Trajectory_Prediction_Solution for an Object.

Trajectory

A path an Object might follow through space over time.

Trajectory_Influence

Something that affects the path of an Object through space. The Trajectory Prediction component only has an abstract understanding of these things, examples of which are environmental conditions, other battlespace objects or geographical features.

Trajectory_Prediction_Solution

The calculated Trajectory or set of Trajectories for the Object of interest, including an associated accuracy and level of confidence.

Capability

The ability to generate a Trajectory_Prediction_Solution.

Manoeuvring_Characteristic

A characteristic that defines an aspect of how an Object moves through space (e.g. capabilities and tendencies).

5.4.2.65.6 Design Rationale

5.4.2.65.6.1 Assumptions

- This component will cater for a variety of different objects (e.g. air to air missiles or moving ground based air defence units) that will have different Manoeuvring_Characteristics.
- This component may rely on ballistic or generic guidance laws when given minimal Manoeuvring_Characteristic data for an Object.
- The component is expected to provide predicted trajectories for Objects that the Exploiting Platform does not control.
- The provision of predicted trajectories for Objects by this component is made irrespective of the Object's allegiance.

5.4.2.65.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Trajectory Prediction.

- Data Driving - Could be used for various Object Manoeuvring_Characteristics such as type, variant, minimum and maximum speed, minimum turning radius or maximum altitude along with any potential Trajectory_Influences.

Extensions

- Extensions could be used to support Trajectory Prediction algorithms for specific environments or types of Object (see Component Extensions PYRAMID concept).

Exploitation Considerations

- Prior known movements of a type of Object may be available to increase the accuracy of a Trajectory_Prediction_Solution, perhaps through machine learning algorithms.
- Trajectory Prediction will be used in cases where the routing or guidance and control algorithms specific to the Object are not available (or the inputs to those algorithms), instead relying on the Object's known characteristics (e.g. maximum accelerations or minimum turn radii) and outside influences upon the Object’s trajectory.
- Trajectory Prediction will not be used to host 'start and end condition models' that are developed using the trajectories of objects, but which are abstracted away from reasoning about the trajectories of objects within their calculation. For example, it would not be used where a weapon release is calculated using a LAR model of the weapon performance that only understands the relationship between the release conditions (including position and velocity) and the possible ground impact area, and does not know anything about the possible paths between these points.

5.4.2.65.6.3 Safety Considerations

The indicative IDAL of this component is DAL B.

The rationale behind this is:

- Trajectory Prediction predicts a trajectory or trajectories for guided or unguided moving objects, including the associated accuracy and confidence levels of the prediction. The nature of the calculations performed by this component, and the potential contribution of the data to a number of hazards is such that the results provided have significant safety implications.

Incorrect trajectory prediction may contribute to:

- A number of hazards that are related to weapons, stores and countermeasures. Erroneous trajectory prediction can lead to incorrect aiming of weapons / stores, erroneous sensing and tracking of potential targets or incorrect coordination of countermeasures. These conditions may lead to collision of the weapon / store or countermeasure with the airframe post safe separation leading to death of crew (catastrophic). They may also lead to the target position being incorrect and weapons impacting locations not intended by the crew. This could potentially result in the death of other vehicle crews, collateral damage, and unintended harm to third parties. This drives a DAL B indicative IDAL.
- Where trajectory prediction data is used for the avoidance of collisions with other vehicles, erroneous trajectory prediction data could contribute to a collision with another vehicle and death of crew (catastrophic) and third parties.

The output of Trajectory Prediction will only ever be a prediction of the actual trajectory of a moving object, and therefore can only be treated as advisory. Hence, while Trajectory Prediction could contribute to a catastrophic hazard, there will need to be other mitigations in place (such as independent information sources or reliance on the probability of an accident actually occurring in the presence of a fault) and so the indicative IDAL for the component is defined as DAL B.

5.4.2.65.6.4 Security Considerations

The indicative security classification is SNEO.

This component performs a narrow but high value function, to predict the Trajectory of Objects based on an understanding of the characteristics and performance of those Objects. Such Objects may include vehicles and weapons. Interference with the availability or integrity of this function has safety implications and will have a detrimental effect on mission critical capability. As such, this component is a likely cyber target. It is therefore expected that this function will be provided by a high trust component and that the component, the data used by the component and the delivery of results provided by the component are all adequately protected.

The component will need knowledge of positional data to undertake the calculations, including own asset and target locations, though an exploitation implementation could potentially use relative positional data. If absolute positional information is used, this may increase the classification of the component. The component will necessarily use Object Manoeuvring_Characteristic and performance data that determine the behaviour and motion of the Object. In many cases, this information will be of a high classification. The component may also implement prediction algorithms that are themselves of a high classification.

In respect of security related functions, the following have been identified:

- **Maintaining Audit Records:** There may be a need to record the context behind the determination of an Object's Trajectory for accountability and non-repudiation purposes.
- **Supporting Safe Operation:** As noted above, the nature of the calculations performed by this component, and their potential use for example in relation to weapon aiming and air vehicle path prediction, is such that the results provided have significant safety implications (see also Safety Considerations).

As a service component with a narrow Trajectory calculation function, this component is not expected to implement security enforcing functions.

5.4.2.65.7 Services

5.4.2.65.7.1 Service Definitions

5.4.2.65.7.1.1 Trajectory_Prediction_Requirement

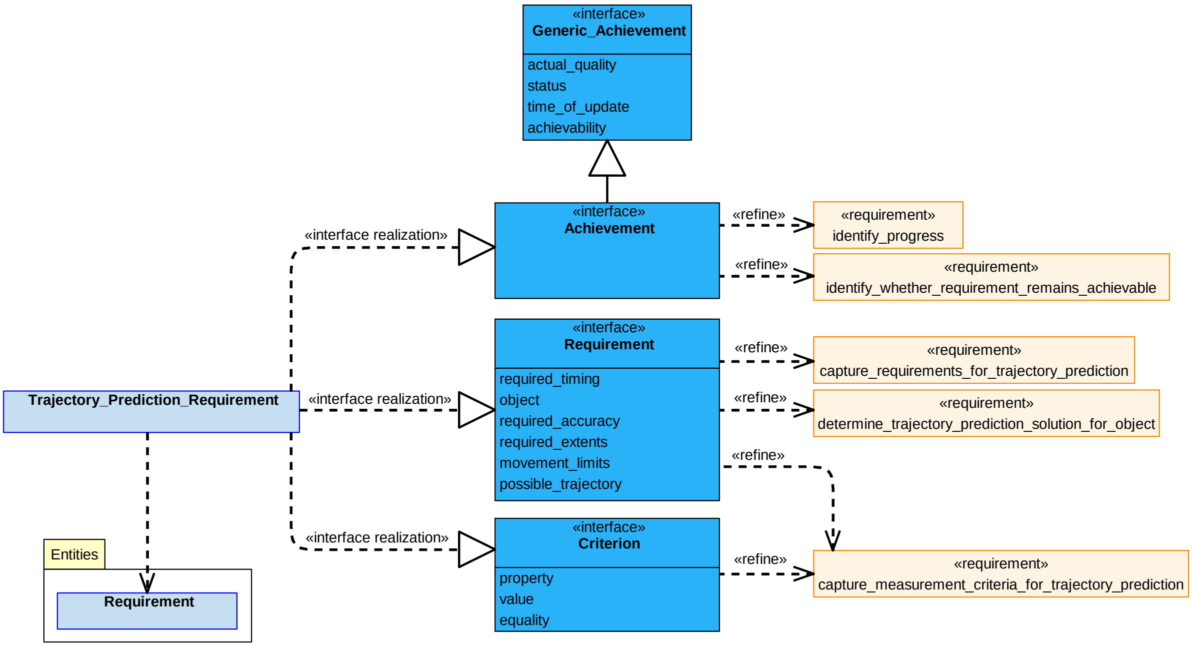


Figure 1126: Trajectory_Prediction_Requirement Service Definition

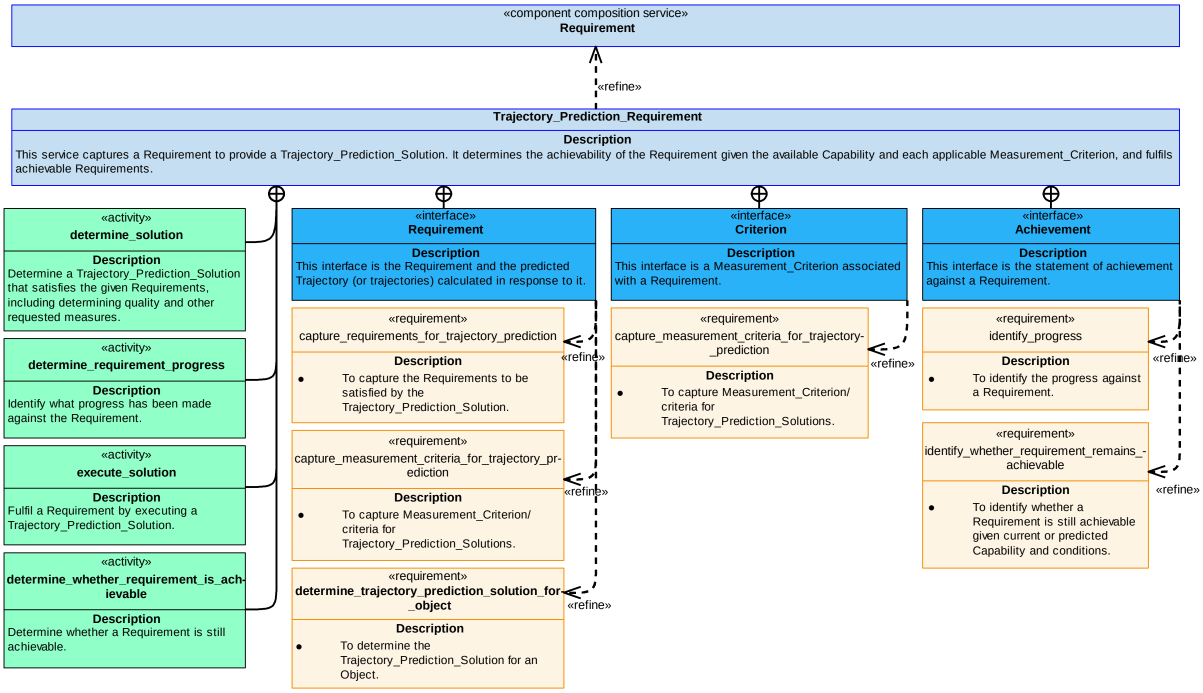

«component composition service»
Requirement
«refine»
Trajectory_Prediction_Requirement
Description
This service captures a Requirement to provide a Trajectory_Prediction_Solution. It determines the achievability of the Requirement given the available Capability and each applicable Measurement_Criterion, and fulfils achievable Requirements.
«activity»
determine_solution
Description
Determine a Trajectory_Prediction_Solution that satisfies the given Requirements, including determining quality and other requested measures.
«activity»
determine_requirement_progress
Description
Identify what progress has been made against the Requirement.
«activity»
execute_solution
Description
Fulfil a Requirement by executing a Trajectory_Prediction_Solution.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether a Requirement is still achievable.
«interface»
Requirement
Description
This interface is the Requirement and the predicted Trajectory (or trajectories) calculated in response to it.
«requirement»
capture_requirements_for_trajectory_prediction
Description
To capture the Requirements to be satisfied by the Trajectory_Prediction_Solution.
«refine»
«requirement»
capture_measurement_criteria_for_trajectory_prediction
Description
To capture Measurement_Criterion/ criteria for Trajectory_Prediction_Solutions.
«refine»
«requirement»
determine_trajectory_prediction_solution_for_object
Description
To determine the Trajectory_Prediction_Solution for an Object.
«refine»
«interface»
Criterion
Description
This interface is a Measurement_Criterion associated with a Requirement.
«requirement»
capture_measurement_criteria_for_trajectory_prediction
Description
To capture Measurement_Criterion/ criteria for Trajectory_Prediction_Solutions.
«refine»
«interface»
Achievement
Description
This interface is the statement of achievement against a Requirement.
«requirement»
identify_progress
Description
To identify the progress against a Requirement.
«refine»
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Requirement is still achievable given current or predicted Capability and conditions.
«refine»

Figure 1127: Trajectory_Prediction_Requirement Service Policy

Trajectory_Prediction_Requirement

This service captures a Requirement to provide a Trajectory_Prediction_Solution. It determines the achievability of the Requirement given the available Capability and each applicable Measurement_Criterion, and fulfils achievable Requirements.

Interfaces

Criterion

This interface is a Measurement_Criterion associated with a Requirement.

Attributes

property	The property to be measured. For example, the required accuracy level.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to). For example, requiring an accuracy above 90%.

Requirement

This interface is the Requirement and the predicted Trajectory (or trajectories) calculated in response to it.

Attributes

required_timing	The time frame within which the Trajectory_Prediction_Solution must be returned.
object	The Object for which a trajectory prediction is required.

required_accuracy	The required tolerance and probability of the Object being within the specified region. For example a Trajectory line with a tolerance of X m tangential to the line where there is a specified probability of the Object being within the tolerance; e.g. a 10m tolerance with a specified percentile (x% or x-th), CEP, Root Mean Square Error, or the x sigma.
required_extents	The extents in space and/or time to which a Trajectory is to be extrapolated.
movement_limits	The volume of space that contains all trajectories that conform to the specified time extents and to the specified required_accuracy.
possible_trajectory	A possible Trajectory that conforms to the specified time extents and to the required_accuracy.

Achievement

This interface is the statement of achievement against a Requirement.

Activities

determine_solution

Determine a Trajectory_Prediction_Solution that satisfies the given Requirements, including determining quality and other requested measures.

determine_whether_requirement_is_achievable

Determine whether a Requirement is still achievable.

determine_requirement_progress

Identify what progress has been made against the Requirement.

execute_solution

Fulfil a Requirement by executing a Trajectory_Prediction_Solution.

5.4.2.65.7.1.2 Object_Information

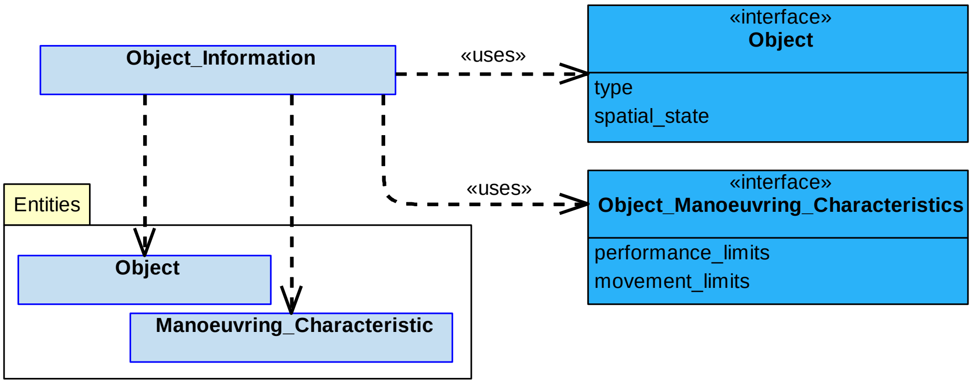


Figure 1128: Object_Information Service Definition

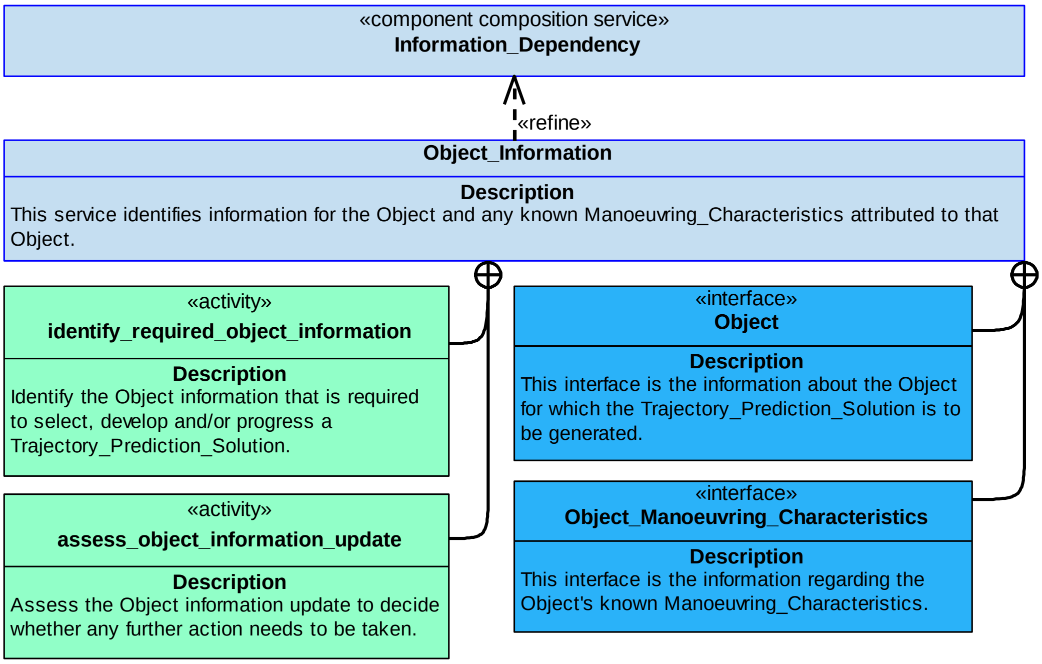

«component composition service»
Information_Dependency
«refine»
Object_Information
Description
This service identifies information for the Object and any known Manoeuvring_Characteristics attributed to that Object.
«activity»
identify_required_object_information
Description
Identify the Object information that is required to select, develop and/or progress a Trajectory_Prediction_Solution.
«activity»
assess_object_information_update
Description
Assess the Object information update to decide whether any further action needs to be taken.
«interface»
Object
Description
This interface is the information about the Object for which the Trajectory_Prediction_Solution is to be generated.
«interface»
Object_Manoeuvring_Characteristics
Description
This interface is the information regarding the Object's known Manoeuvring_Characteristics.

Figure 1129: Object_Information Service Policy

Object_Information

This service identifies information for the Object and any known Manoeuvring_Characteristics attributed to that Object.

Interfaces

Object

This interface is the information about the Object for which the Trajectory_Prediction_Solution is to be generated.

Attributes

type	The type of object.
spatial_state	The location, orientation, velocity and acceleration of the Object.

Object_Manoeuvring_Characteristics

This interface is the information regarding the Object's known Manoeuvring_Characteristics.

Attributes

performance_limits	The kinematic performance limits of the Object (e.g. maximum and minimum speed, maximum climb/descent rate or minimum turning radius).
movement_limits	The spatial limits within which an Object's trajectory is allowed (e.g. maximum altitude, traversable terrain or minimum water depth).

Activities

assess_object_information_update

Assess the Object information update to decide whether any further action needs to be taken.

identify_required_object_information

Identify the Object information that is required to select, develop and/or progress a Trajectory_Prediction_Solution.

5.4.2.65.7.1.3 Trajectory_Influences

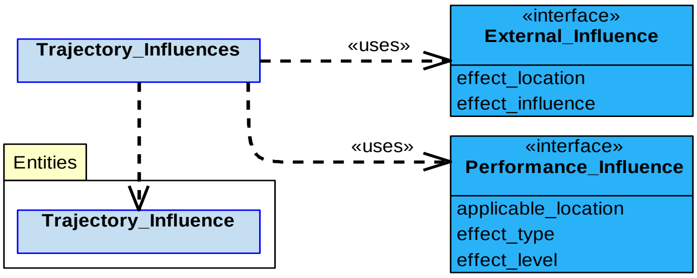


Figure 1130: Trajectory_Influences Service Definition

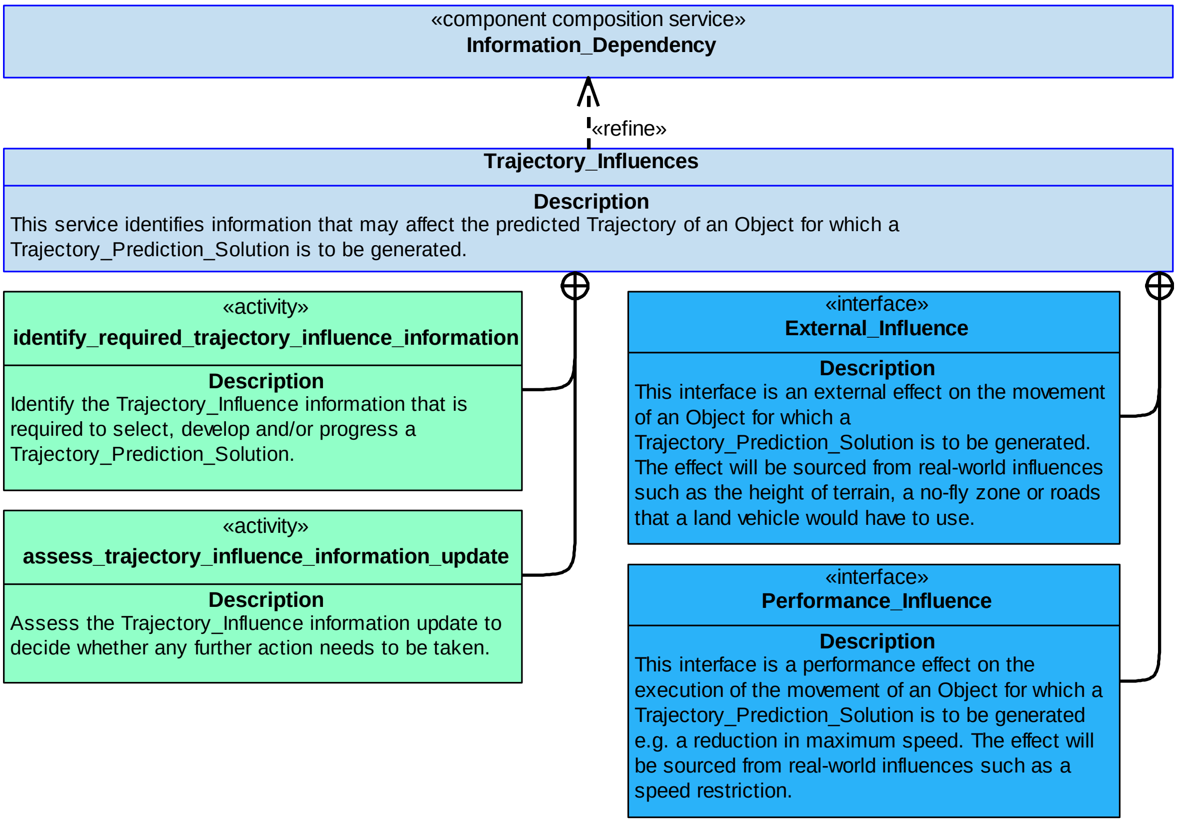

«component composition service»
Information_Dependency
«refine»
Trajectory_Influences
Description
This service identifies information that may affect the predicted Trajectory of an Object for which a Trajectory_Prediction_Solution is to be generated.
«activity»
identify_required_trajectory_influence_information
Description
Identify the Trajectory_Influence information that is required to select, develop and/or progress a Trajectory_Prediction_Solution.
«activity»
assess_trajectory_influence_information_update
Description
Assess the Trajectory_Influence information update to decide whether any further action needs to be taken.
«interface»
External_Influence
Description
This interface is an external effect on the movement of an Object for which a Trajectory_Prediction_Solution is to be generated. The effect will be sourced from real-world influences such as the height of terrain, a no-fly zone or roads that a land vehicle would have to use.
«interface»
Performance_Influence
Description
This interface is a performance effect on the execution of the movement of an Object for which a Trajectory_Prediction_Solution is to be generated e.g. a reduction in maximum speed. The effect will be sourced from real-world influences such as a speed restriction.

Figure 1131: Trajectory_Influences Service Policy

Trajectory_Influences

This service identifies information that may affect the predicted Trajectory of an Object for which a Trajectory_Prediction_Solution is to be generated.

Interfaces

External_Influence

This interface is an external effect on the movement of an Object for which a Trajectory_Prediction_Solution is to be generated. The effect will be sourced from real-world influences such as the height of terrain, a no-fly zone or roads that a land vehicle would have to use.

Attributes

effect_location	The point or volume from which the external effect on the movement of an Object originates.
effect_influence	The influence that an external effect has on the movement of an Object.

Performance_Influence

This interface is a performance effect on the execution of the movement of an Object for which a Trajectory_Prediction_Solution is to be generated e.g. a reduction in maximum speed. The effect will be sourced from real-world influences such as a speed restriction.

Attributes

applicable_location	The position and extent over which a performance influence is applicable.
effect_type	The aspect of an Object's performance in executing a Trajectory affected by a performance influence e.g. maximum speed.
effect_level	The amount to which a performance influence affects an aspect of Object's performance in executing a Trajectory.

Activities

assess_trajectory_influence_information_update

Assess the Trajectory_Influence information update to decide whether any further action needs to be taken.

identify_required_trajectory_influence_information

Identify the Trajectory_Influence information that is required to select, develop and/or progress a Trajectory_Prediction_Solution.

5.4.2.65.7.1.4 Capability

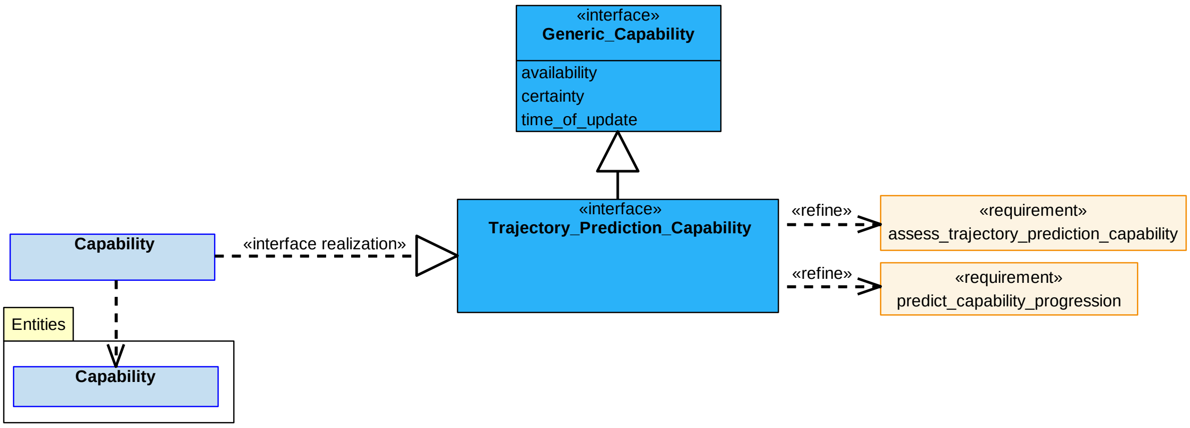


Figure 1132: Capability Service Definition

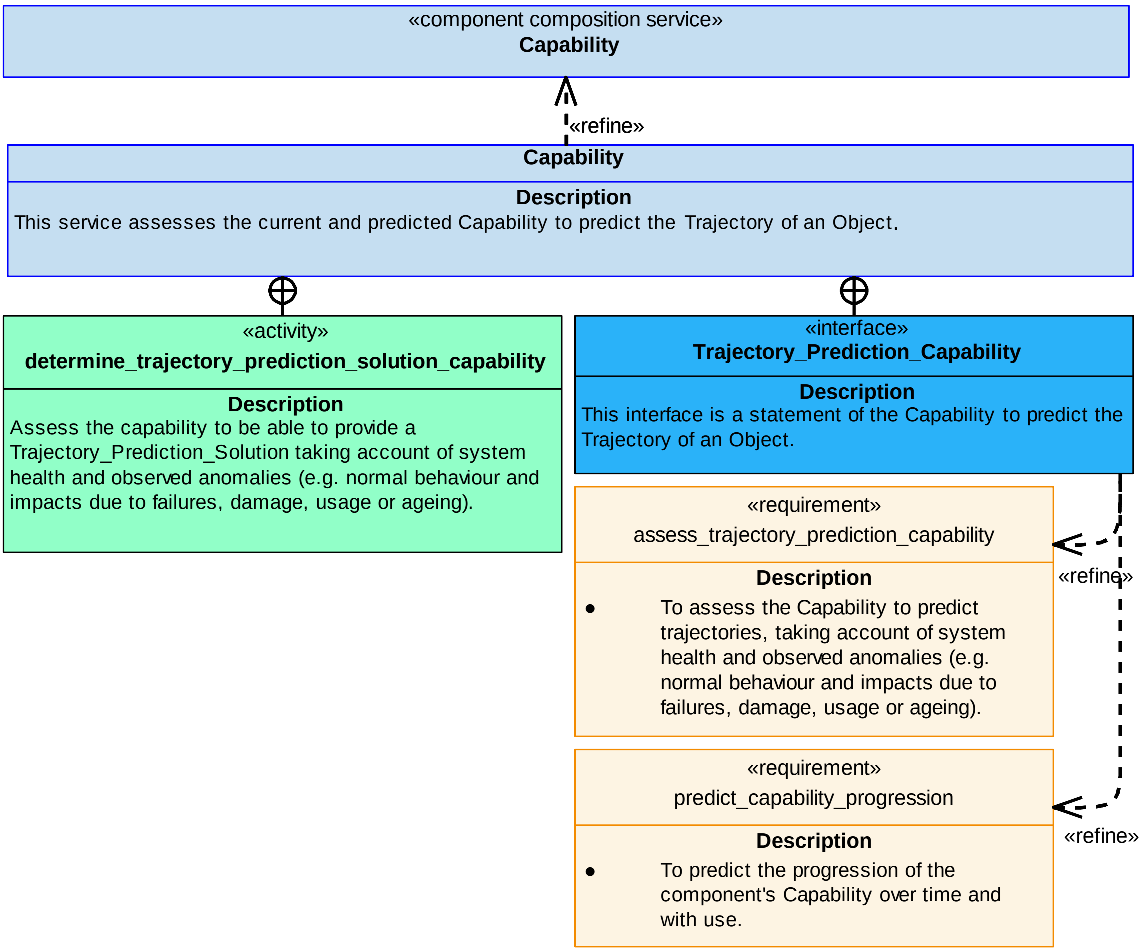

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to predict the Trajectory of an Object.
«activity»
determine_trajectory_prediction_solution_capability
Description
Assess the capability to be able to provide a Trajectory_Prediction_Solution taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Trajectory_Prediction_Capability
Description
This interface is a statement of the Capability to predict the Trajectory of an Object.
«requirement»
assess_trajectory_prediction_capability
Description
To assess the Capability to predict trajectories, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the component's Capability over time and with use.
«refine»

Figure 1133: Capability Service Policy

Capability

This service assesses the current and predicted Capability to predict the Trajectory of an Object.

Interface

Trajectory_Prediction_Capability

This interface is a statement of the Capability to predict the Trajectory of an Object.

Activity

determine_trajectory_prediction_solution_capability

Assess the capability to be able to provide a Trajectory_Prediction_Solution taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.65.7.1.5 Trajectory_Prediction_Capability_Evidence

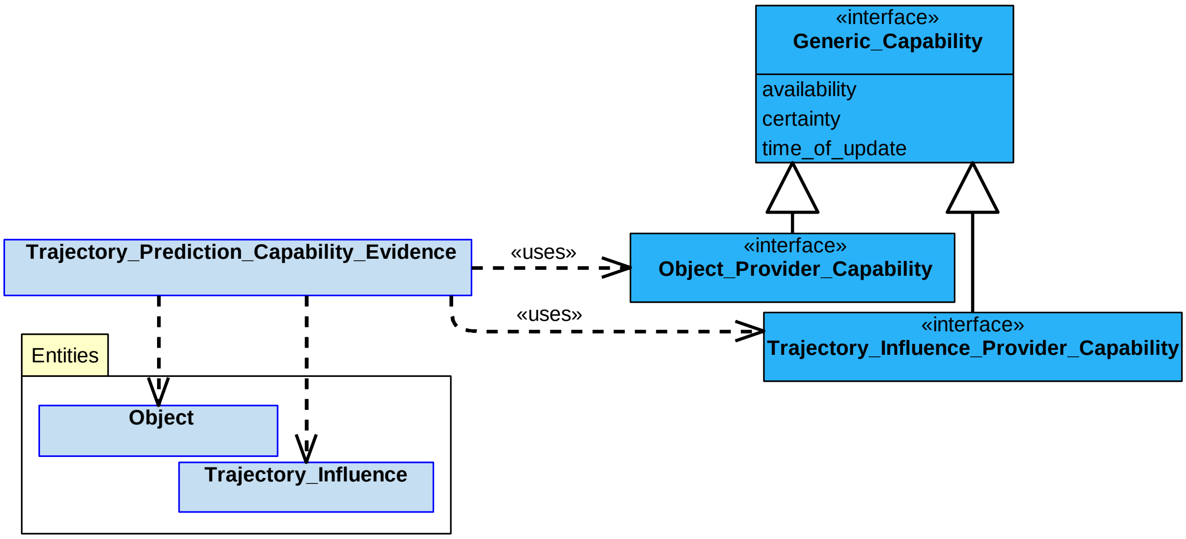


Figure 1134: Trajectory_Prediction_Capability_Evidence Service Definition

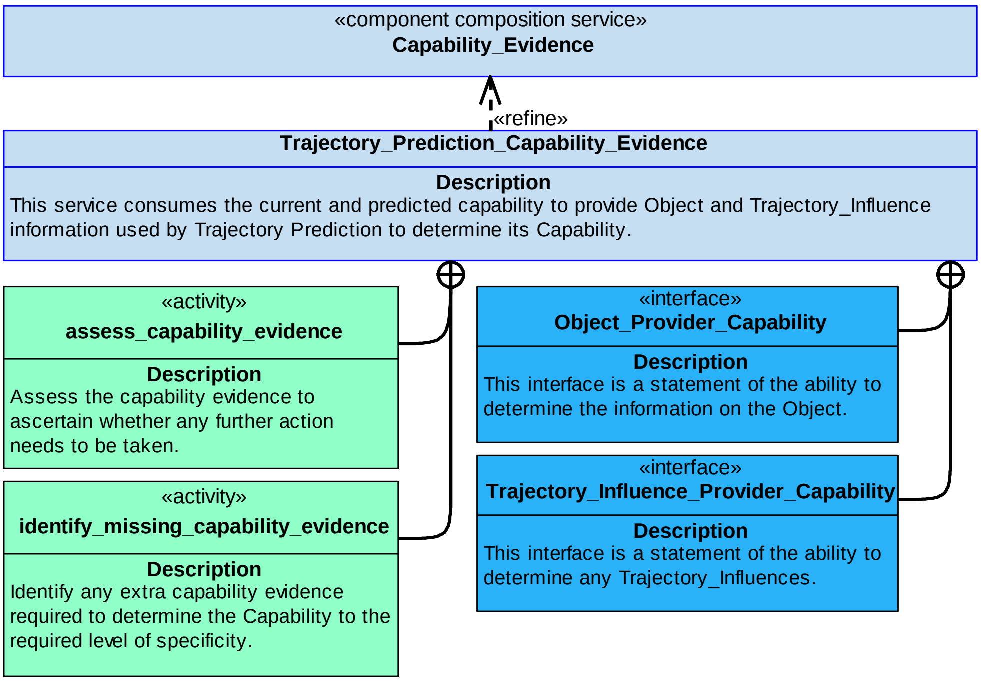

«component composition service»
Capability_Evidence
«refine»
Trajectory_Prediction_Capability_Evidence
Description
This service consumes the current and predicted capability to provide Object and Trajectory_Influence information used by Trajectory Prediction to determine its Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to ascertain whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity.
«interface»
Object_Provider_Capability
Description
This interface is a statement of the ability to determine the information on the Object.
«interface»
Trajectory_Influence_Provider_Capability
Description
This interface is a statement of the ability to determine any Trajectory_Influences.

Figure 1135: Trajectory_Prediction_Capability_Evidence Service Policy

Trajectory_Prediction_Capability_Evidence

This service consumes the current and predicted capability to provide Object and Trajectory_Influence information used by Trajectory Prediction to determine its Capability.

<u>Interfaces</u>

Object_Provider_Capability

This interface is a statement of the ability to determine the information on the Object.

Trajectory_Influence_Provider_Capability

This interface is a statement of the ability to determine any Trajectory_Influences.

<u>Activities</u>

assess_capability_evidence

Assess the capability evidence to ascertain whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity.

5.4.2.65.7.2 Service Dependencies

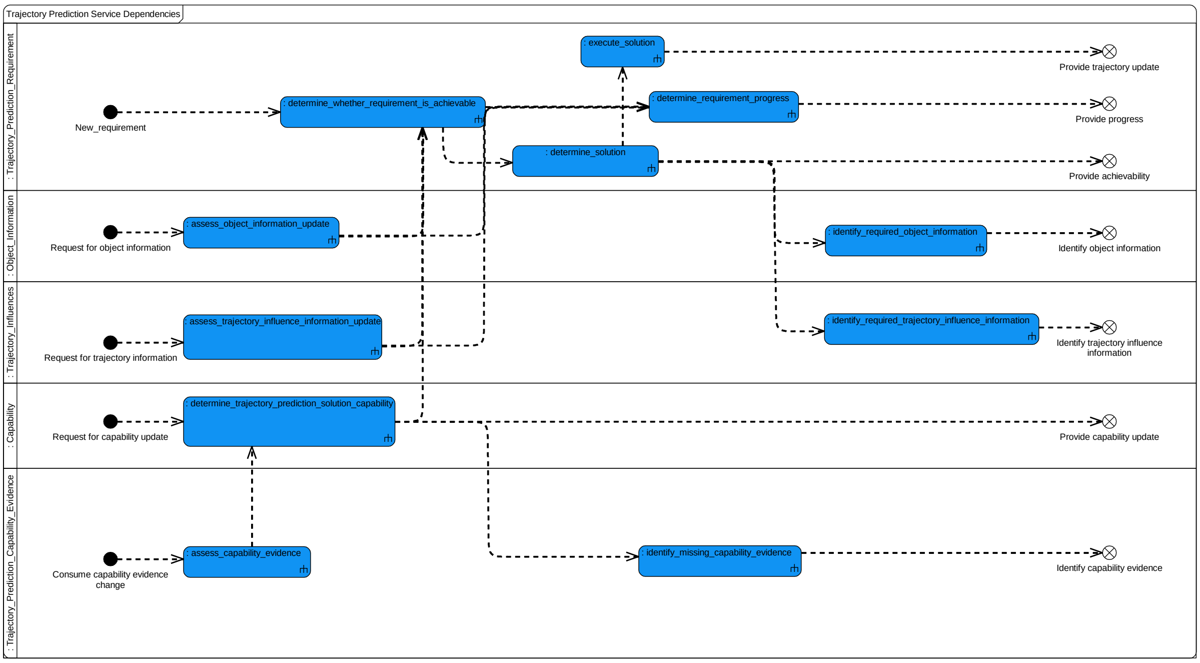

Trajectory Prediction Service Dependencies
: Trajectory_Prediction_Requirement
: Object_Information
: Trajectory_Influences
: Capability
: Trajectory_Prediction_Capability_Evidence
: execute_solution
Provide trajectory update
New_requirement
: determine_whether_requirement_is_achievable
: determine_requirement_progress
Provide progress
: determine_solution
Provide achievability
Request for object information
: assess_object_information_update
: identify_required_object_information
Identify object information
Request for trajectory information
: assess_trajectory_influence_information_update
: identify_required_trajectory_influence_information
Identify trajectory influence information
Request for capability update
: determine_trajectory_prediction_solution_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1136: Trajectory Prediction Service Dependencies

5.4.2.66 Undercarriage

5.4.2.66.1 Role

The role of Undercarriage is to control the deployment or retraction of the undercarriage.

5.4.2.66.2 Overview

Control Architecture

Undercarriage is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

On receiving a Requirement for operating the undercarriage, e.g. to retract it after take-off, the Undercarriage component will derive a solution to the Requirement that is cognisant of the available Capability and any applicable Constraints. The component then coordinates the Undercarriage_Resources, observing any Pre-conditions, to achieve the Requirement.

Examples of Use

This component can be used where:

- Software control of the undercarriage of an Exploiting Platform is required.

5.4.2.66.3 Service Summary

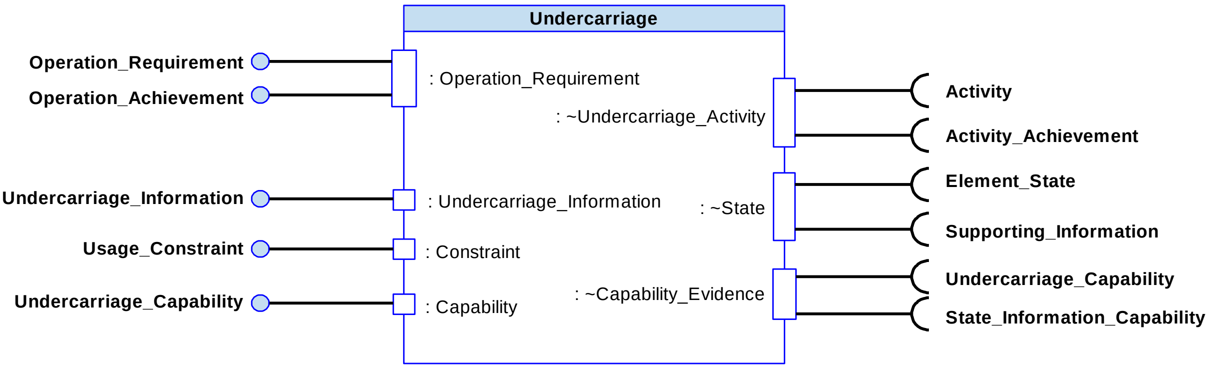


Figure 1137: Undercarriage Service Summary

5.4.2.66.4 Responsibilities

capture_undercarriage_requirement

- To capture provided Requirements for deployment or retraction of undercarriage.

capture_undercarriage_constraints

- To capture undercarriage Constraints.

determine_undercarriage_solution

- To determine an Undercarriage_Solution that meets the given Requirements and Constraints.

determine_undercarriage_state

- To determine the Undercarriage_State (e.g. up, travelling, down or weight on or off wheels).

identify_undercarriage_solution_in_progress_remains_feasible

- To identify whether an Undercarriage_Solution in progress remains feasible given current Capability.

identify_pre_condition

- To identify Pre-conditions required to support the Undercarriage_Solution or an Undercarriage_Step.

coordinate_undercarriage_movement

- To coordinate the deployment or retraction of the undercarriage to fulfil an undercarriage Requirement.

identify_progress_of_undercarriage_solution

- To identify the progress of an Undercarriage_Solution against the Requirements.

assess_undercarriage_capability

- To assess the Capability to determine the Undercarriage_State and to extend or retract the undercarriage taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the Undercarriage component Capability over time and with use.

5.4.2.66.5 Subject Matter Semantics

The subject matter of Undercarriage is the state and position of the Exploiting Platform's undercarriage.

Exclusions

The subject matter of Undercarriage does not include:

- Directional control of the Exploiting Platform (e.g. nose wheel steering).
- Application of brakes for retardation.
- Knowledge of tyres (e.g. wear), brakes (e.g. temperature), oleo struts (e.g. spring rate), etc.

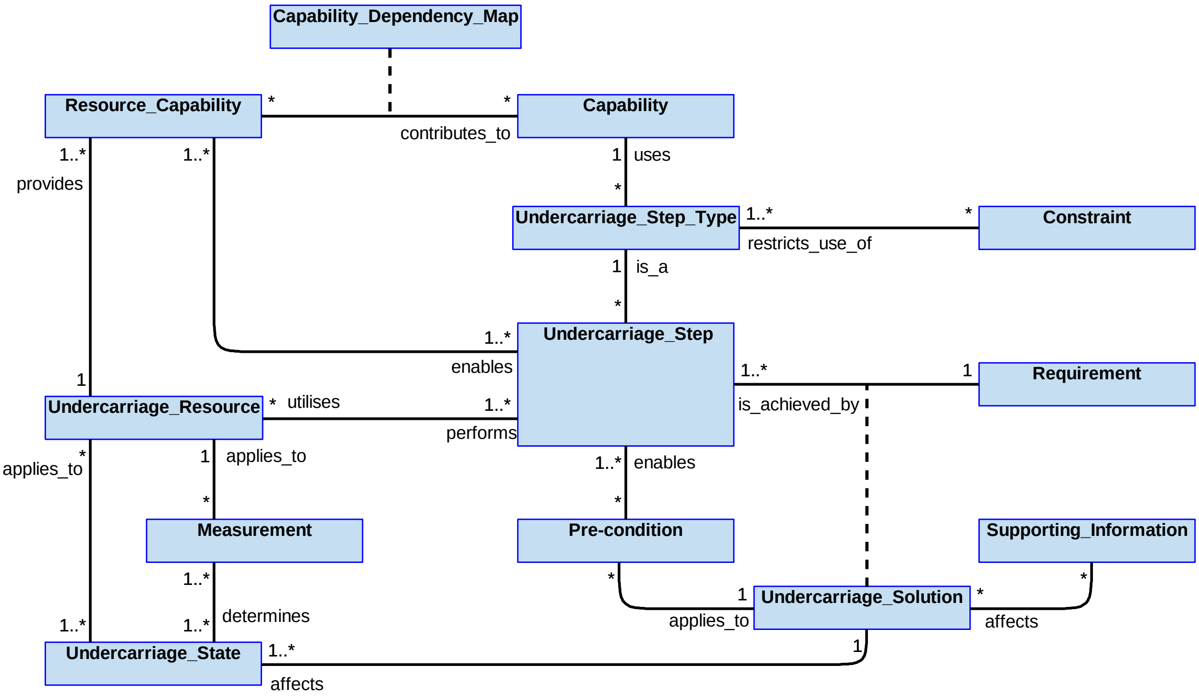


Figure 1138: Undercarriage Semantics

5.4.2.66.5.1 Entities

Capability

The capability to determine and execute Undercarriage_Solutions and determine and report the Undercarriage_State.

Capability_Dependency_Map

A mapping of how the component's Capability is dependent on the Resource_Capability.

Constraint

An externally imposed restriction.

Measurement

A representation of a measurement, e.g. the position of a weight on wheels microswitch.

Pre-condition

A condition that must be true before an action can take place, e.g. being in an appropriate phase of flight or having weight off wheels.

Requirement

A requirement placed for the undercarriage to be in a specific state (e.g. to be extended or retracted).

Resource_Capability

The capability of the resources to perform Undercarriage_Steps, e.g. open or close doors, extend or retract undercarriage assemblies.

Supporting_Information

Information not related to the undercarriage that may affect its operation, e.g. the configuration of the aircraft.

Undercarriage_Resource

The physical gear assemblies and any associated hardware such as doors, locks and position sensors.

Undercarriage_Solution

A sequence of Undercarriage_Steps that are needed to meet the Requirement.

Undercarriage_State

The state of the undercarriage, e.g. extended and locked, transitioning, weight on or off the gear, door open or closed.

Undercarriage_Step

An action the component will coordinate that, when performed, contributes to the extension or retraction of the undercarriage.

Undercarriage_Step_Type

The kind of action this component knows how to coordinate, e.g. opening doors, extending and locking the landing gear, and despinning the wheel.

5.4.2.66.6 Design Rationale

5.4.2.66.6.1 Assumptions

- Under normal conditions, undercarriage positions may be dependent on factors such as weight on or off wheels, etc.

5.4.2.66.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Undercarriage:

- Data Driving - The logic involved in controlling the undercarriage and determining whether weight is on the landing gear will vary by Exploiting Platform, this could be defined through build time data to provide configurability.

Extensions

- Extension components may be appropriate to accommodate different rules for nose and main gear, etc.

Exploitation Considerations

- Undercarriage represents the undercarriage of the vehicle, which may be of differing forms, including wheels, skis, floats, etc.

- Undercarriage monitors for transition between weight being on and being off the undercarriage.

5.4.2.66.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component controls the position of the undercarriage. Therefore, failure of this component could cause uncontrolled flight of the air vehicle if the undercarriage was extended inadvertently in flight, leading to exceedance of the flight envelope or structural limits, unless the undercarriage is cleared for operation over the full flight envelope. This could lead to an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

5.4.2.66.6.4 Security Considerations

The indicative security classification is O.

This component provides software control of the deployment or retraction of the undercarriage, therefore the security classification is considered unlikely to be above O. This component will also monitor transitions between weight on and weight off wheels, this is an interlock for a number of other operations. However, it should be noted some safety-critical cases will use a discrete interlock rather than a software interlock. Due to its role, this component will have rigorous requirements to ensure its integrity and availability.

The component may be expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** relating to control of the undercarriage and weight on/off wheels status, etc.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** with unexpected behaviour being an indicator the system may have been compromised.

The component is not expected to directly implement security enforcing functions.

5.4.2.66.7 Services

5.4.2.66.7.1 Service Definitions

5.4.2.66.7.1.1 Operation_Requirement

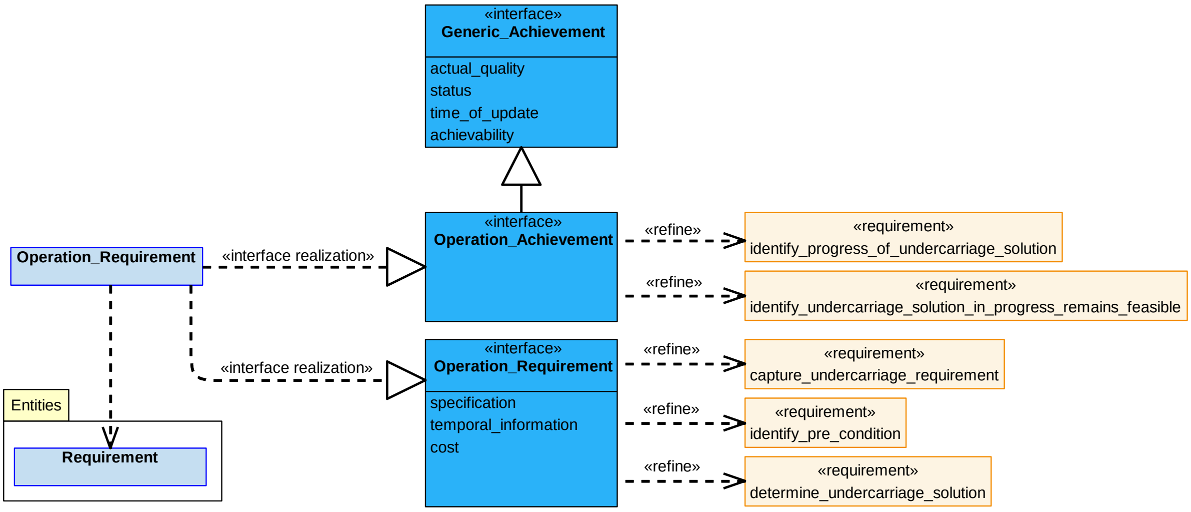
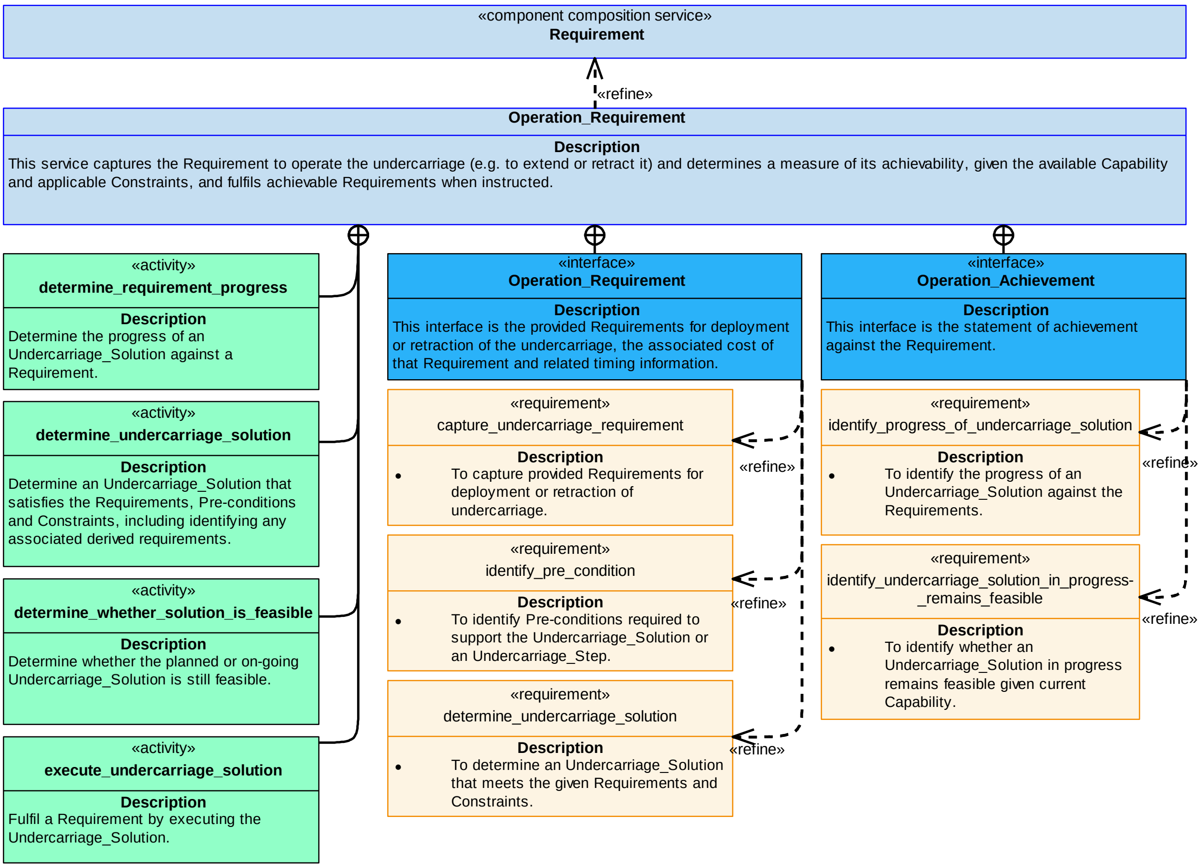


Figure 1139: Operation_Requirement Service Definition



This service captures the Requirement to operate the undercarriage (e.g. to extend or retract it) and determines a measure of its achievability, given the available Capability and applicable Constraints, and fulfils achievable Requirements when instructed.

Determine the progress of an Undercarriage_Solution against a Requirement.

Determine an Undercarriage_Solution that satisfies the Requirements, Pre-conditions and Constraints, including identifying any associated derived requirements.

Determine whether the planned or on-going Undercarriage_Solution is still feasible.

Fulfil a Requirement by executing the Undercarriage_Solution.

This interface is the provided Requirements for deployment or retraction of the undercarriage, the associated cost of that Requirement and related timing information.

This interface is the statement of achievement against the Requirement.

- To capture provided Requirements for deployment or retraction of undercarriage.

- To identify Pre-conditions required to support the Undercarriage_Solution or an Undercarriage_Step.

- To determine an Undercarriage_Solution that meets the given Requirements and Constraints.

- To identify the progress of an Undercarriage_Solution against the Requirements.

- To identify whether an Undercarriage_Solution in progress remains feasible given current Capability.

Figure 1140: Operation_Requirement Service Policy

Operation_Requirement

This service captures the Requirement to operate the undercarriage (e.g. to extend or retract it) and determines a measure of its achievability, given the available Capability and applicable Constraints, and fulfils achievable Requirements when instructed.

Interfaces

Operation_Requirement

This interface is the provided Requirements for deployment or retraction of the undercarriage, the associated cost of that Requirement and related timing information.

Attributes

specification	The specification of the Requirement. This may include the location of the undercarriage element (e.g. nose gear) and the direction of travel (e.g. extend).
temporal_information	Information covering timing, such as start or stop times.
cost	The cost of executing the solution, e.g. resources used or time taken.

Operation_Achievement

This interface is the statement of achievement against the Requirement.

Activities

determine_requirement_progress

Determine the progress of an Undercarriage_Solution against a Requirement.

determine_undercarriage_solution

Determine an Undercarriage_Solution that satisfies the Requirements, Pre-conditions and Constraints, including identifying any associated derived requirements.

execute_undercarriage_solution

Fulfil a Requirement by executing the Undercarriage_Solution.

determine_whether_solution_is_feasible

Determine whether the planned or on-going Undercarriage_Solution is still feasible.

5.4.2.66.7.1.2 Undercarriage_Activity

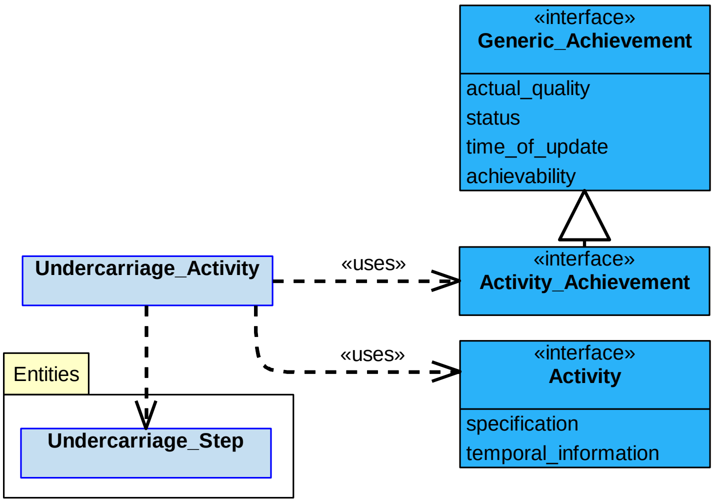


Figure 1141: Undercarriage_Activity Service Definition

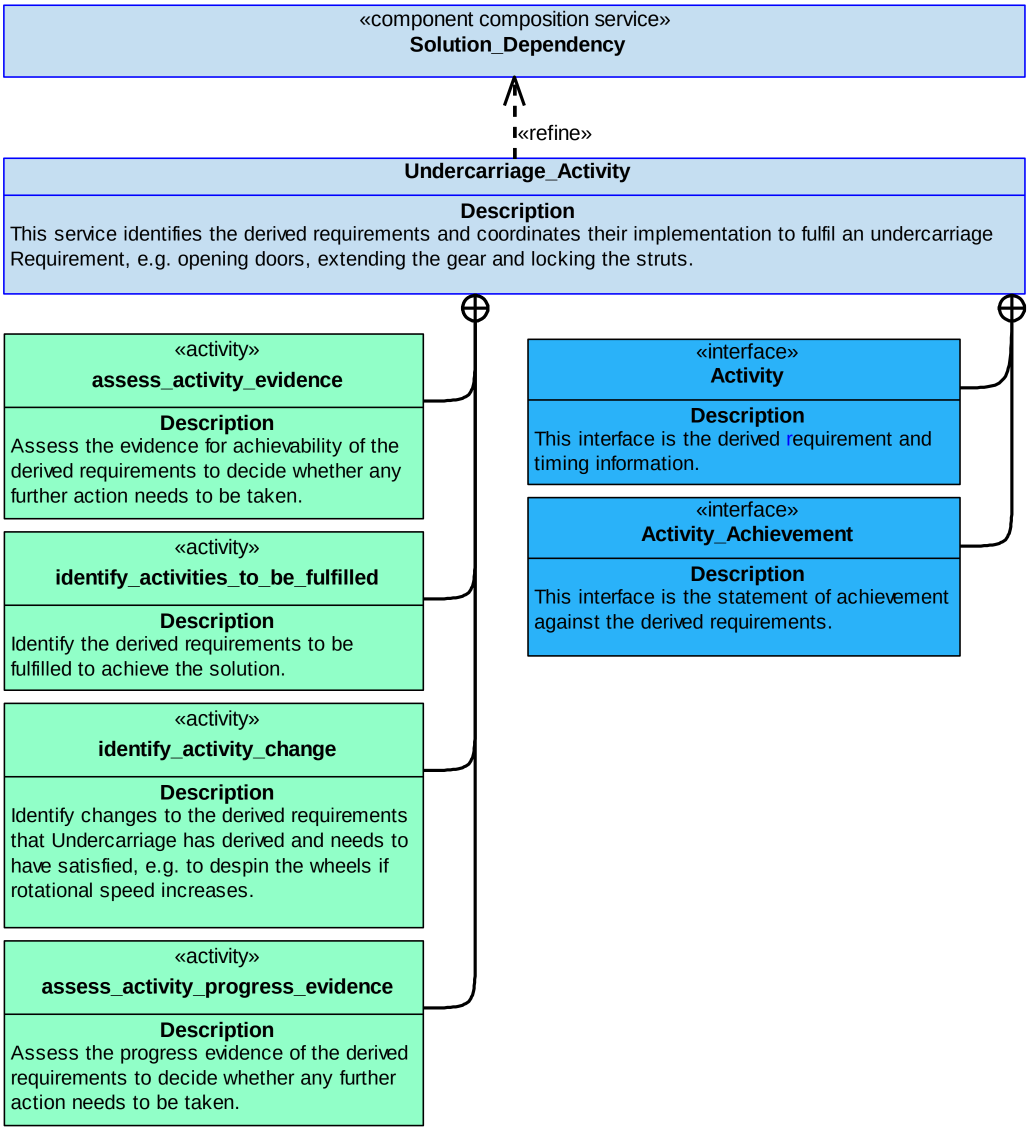

«component composition service»
Solution_Dependency
«refine»
Undercarriage_Activity
Description
This service identifies the derived requirements and coordinates their implementation to fulfil an undercarriage Requirement, e.g. opening doors, extending the gear and locking the struts.
«activity»
assess_activity_evidence
Description
Assess the evidence for achievability of the derived requirements to decide whether any further action needs to be taken.
«activity»
identify_activities_to_be_fulfilled
Description
Identify the derived requirements to be fulfilled to achieve the solution.
«activity»
identify_activity_change
Description
Identify changes to the derived requirements that Undercarriage has derived and needs to have satisfied, e.g. to despin the wheels if rotational speed increases.
«activity»
assess_activity_progress_evidence
Description
Assess the progress evidence of the derived requirements to decide whether any further action needs to be taken.
«interface»
Activity
Description
This interface is the derived requirement and timing information.
«interface»
Activity_Achievement
Description
This interface is the statement of achievement against the derived requirements.

Figure 1142: Undercarriage_Activity Service Policy

Undercarriage_Activity

This service identifies the derived requirements and coordinates their implementation to fulfil an undercarriage Requirement, e.g. opening doors, extending the gear and locking the struts.

Interfaces

Activity

This interface is the derived requirement and timing information.

Attributes

specification	The definition of the derived requirement (e.g. to align the wheels or retract the gear).
temporal_information	Information covering timing, such as start and end times.

Activity_Achievement

This interface is the statement of achievement against the derived requirements.

Activities

assess_activity_evidence

Assess the evidence for achievability of the derived requirements to decide whether any further action needs to be taken.

assess_activity_progress_evidence

Assess the progress evidence of the derived requirements to decide whether any further action needs to be taken.

identify_activities_to_be_fulfilled

Identify the derived requirements to be fulfilled to achieve the solution.

identify_activity_change

Identify changes to the derived requirements that Undercarriage has derived and needs to have satisfied, e.g. to despin the wheels if rotational speed increases.

5.4.2.66.7.1.3 Undercarriage_Information

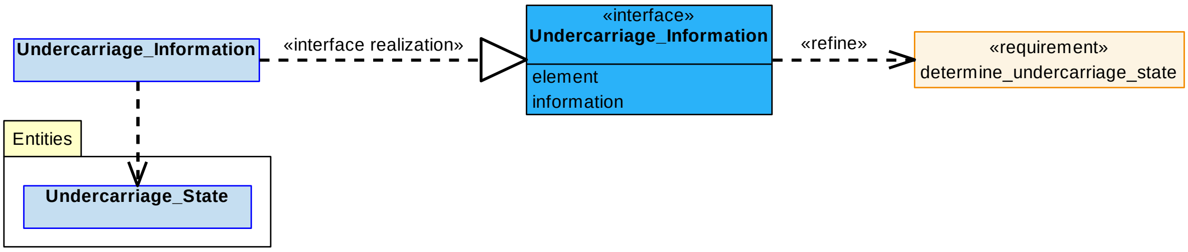


Figure 1143: Undercarriage_Information Service Definition

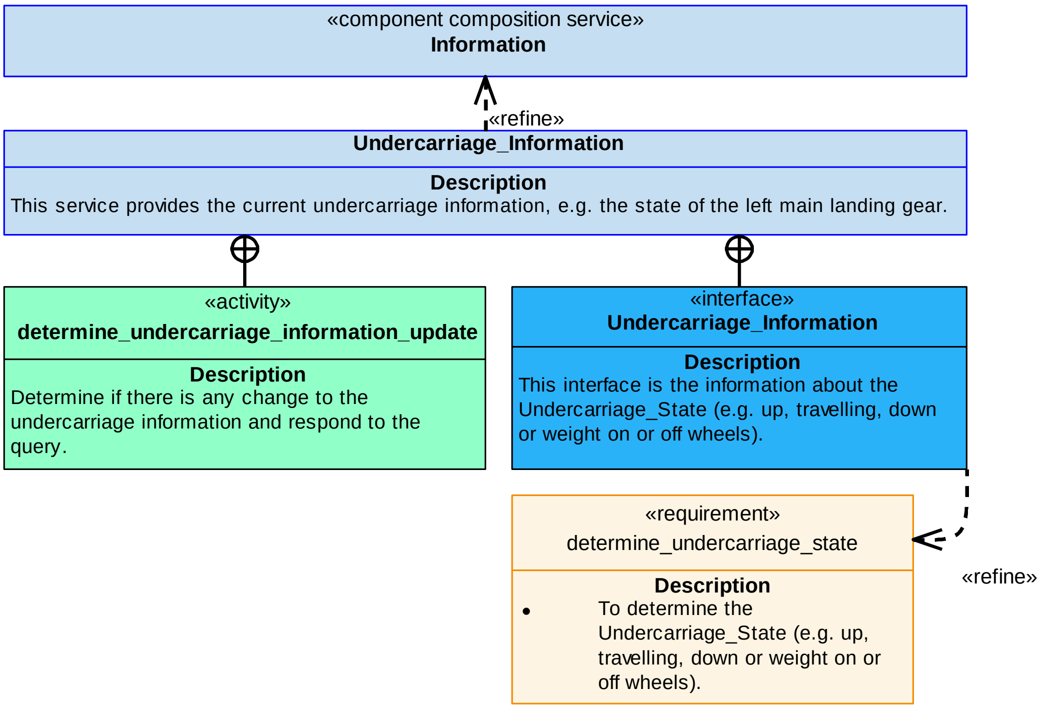

«component composition service»
Information
«refine»
Undercarriage_Information
Description
This service provides the current undercarriage information, e.g. the state of the left main landing gear.
«activity»
determine_undercarriage_information_update
Description
Determine if there is any change to the undercarriage information and respond to the query.
«interface»
Undercarriage_Information
Description
This interface is the information about the Undercarriage_State (e.g. up, travelling, down or weight on or off wheels).
«requirement»
determine_undercarriage_state
Description
To determine the Undercarriage_State (e.g. up, travelling, down or weight on or off wheels).
«refine»

Figure 1144: Undercarriage_Information Service Policy

Undercarriage_Information

This service provides the current undercarriage information, e.g. the state of the left main landing gear.

Interface

Undercarriage_Information

This interface is the information about the Undercarriage_State (e.g. up, travelling, down or weight on or off wheels).

Attributes

element	The undercarriage element the information relates to (e.g. nose wheel or left main gear).
information	The information about the element (e.g. locked or weight on wheels).

Activity

determine_undercarriage_information_update

Determine if there is any change to the undercarriage information and respond to the query.

5.4.2.66.7.1.4 State

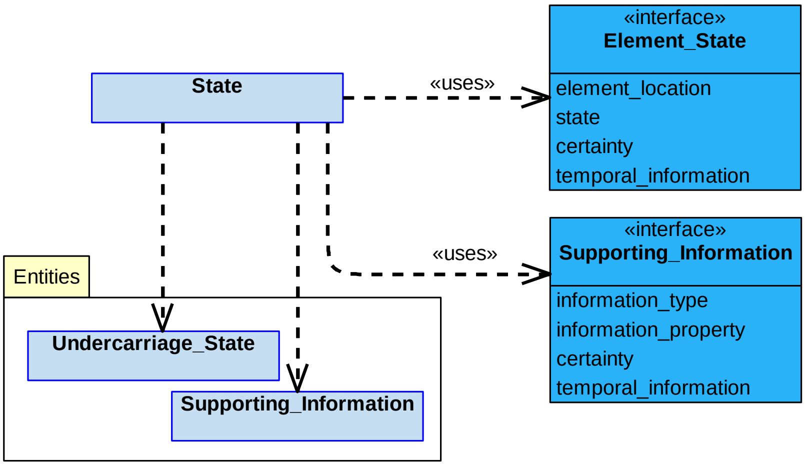


Figure 1145: State Service Definition

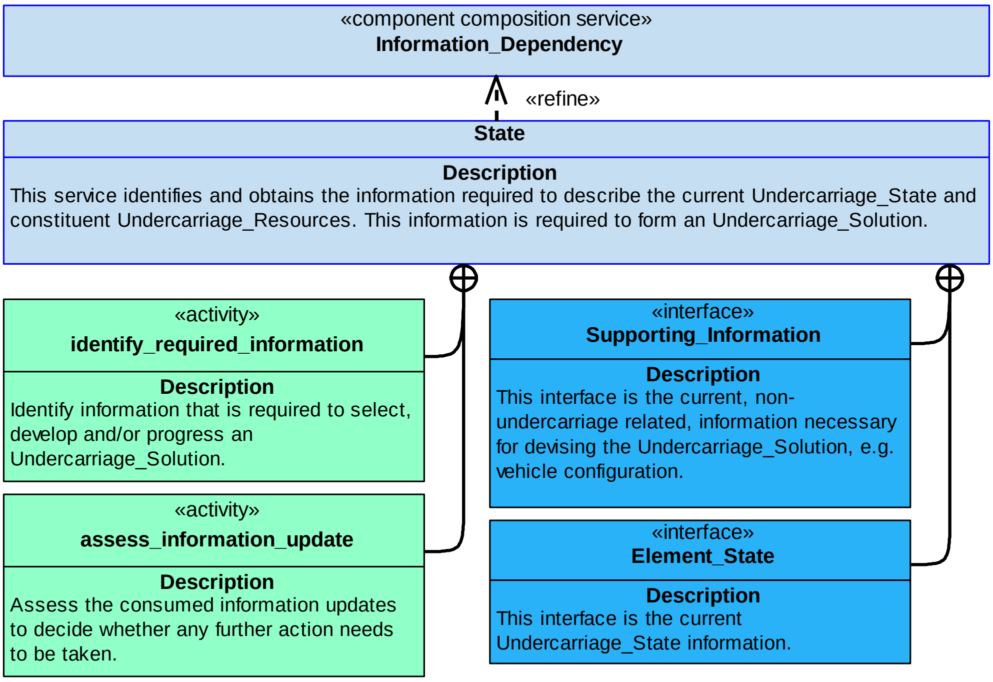

«component composition service»
Information_Dependency
«refine»
State
Description
This service identifies and obtains the information required to describe the current Undercarriage_State and constituent Undercarriage_Resources. This information is required to form an Undercarriage_Solution.
«activity»
identify_required_information
Description
Identify information that is required to select, develop and/or progress an Undercarriage_Solution.
«activity»
assess_information_update
Description
Assess the consumed information updates to decide whether any further action needs to be taken.
«interface»
Supporting_Information
Description
This interface is the current, non-undercarriage related, information necessary for devising the Undercarriage_Solution, e.g. vehicle configuration.
«interface»
Element_State
Description
This interface is the current Undercarriage_State information.

Figure 1146: State Service Policy

State

This service identifies and obtains the information required to describe the current Undercarriage_State and constituent Undercarriage_Resources. This information is required to form an Undercarriage_Solution.

Interfaces

Element_State

This interface is the current Undercarriage_State information.

Attributes

element_location	The location of the undercarriage element being reported on, e.g. nose wheel or left main gear.
state	The state of the undercarriage element, e.g. retracted or transitioning.
certainty	The level of certainty in the reported state.
temporal_information	Information covering timing of the element state being reported.

Supporting_Information

This interface is the current, non-undercarriage related, information necessary for devising the Undercarriage_Solution, e.g. vehicle configuration.

Attributes

information_type	The type of information, e.g. regarding the overall aircraft configuration.
information_property	The state or value of the Supporting_Information.
certainty	The level of certainty in the reported Supporting_Information.
temporal_information	Information covering timing of the Supporting_Information being reported.

Activities

assess_information_update

Assess the consumed information updates to decide whether any further action needs to be taken.

identify_required_information

Identify information that is required to select, develop and/or progress an Undercarriage_Solution.

5.4.2.66.7.1.5 Constraint

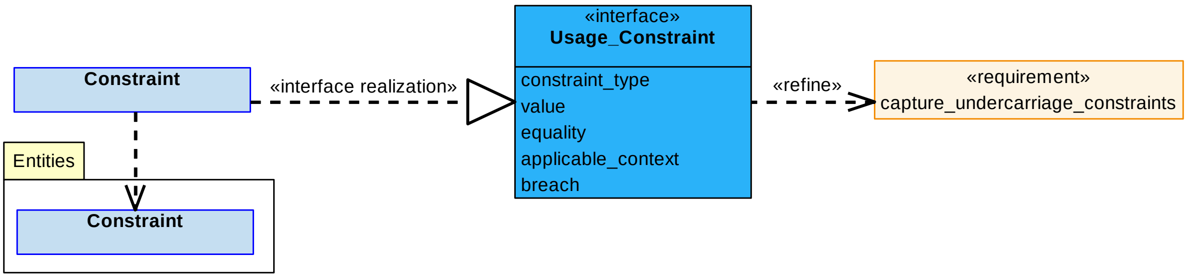


Figure 1147: Constraint Service Definition

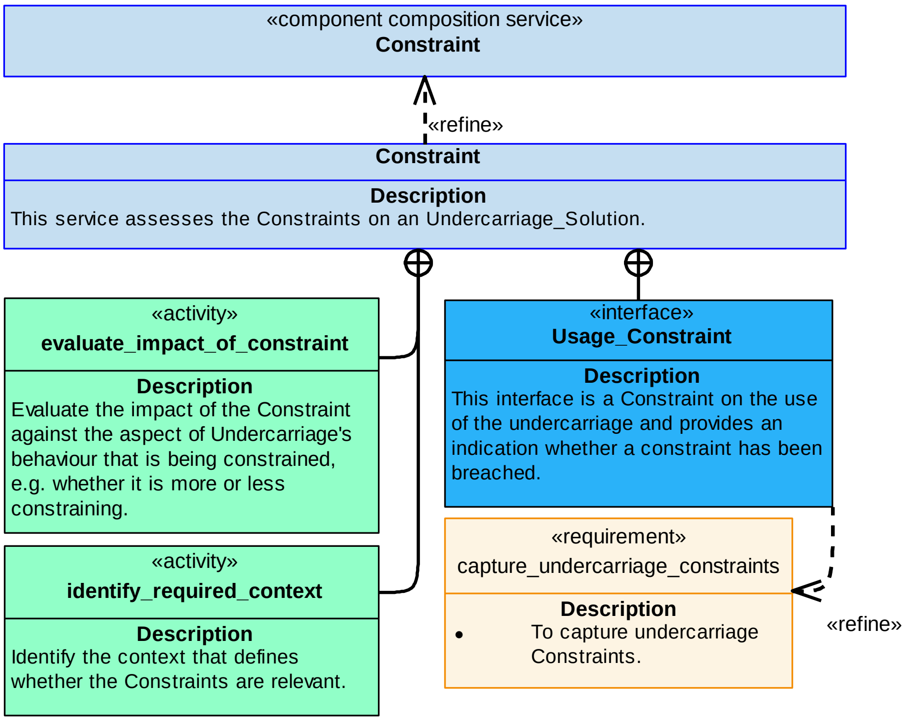

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the Constraints on an Undercarriage_Solution.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of the Constraint against the aspect of Undercarriage's behaviour that is being constrained, e.g. whether it is more or less constraining.
«interface»
Usage_Constraint
Description
This interface is a Constraint on the use of the undercarriage and provides an indication whether a constraint has been breached.
«requirement»
capture_undercarriage_constraints
Description
To capture undercarriage Constraints.
«refine»
«activity»
identify_required_context
Description
Identify the context that defines whether the Constraints are relevant.

Figure 1148: Constraint Service Policy

Constraint

This service assesses the Constraints on an Undercarriage_Solution.

Interface

Usage_Constraint

This interface is a Constraint on the use of the undercarriage and provides an indication whether a constraint has been breached.

Attributes

constraint_type	The type of limit constraining the operation of the undercarriage, e.g. preventing it being lowered.
value	The value for the constraint_type (e.g. an airspeed of 270 knots).
equality	The relationship between the value and the constraint_type, e.g. less than or equal to.
applicable_context	The context in which the Constraint is applicable, e.g. when flight control surfaces are in a flight rather than landing configuration.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of the Constraint against the aspect of Undercarriage's behaviour that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context that defines whether the Constraints are relevant.

5.4.2.66.7.1.6 Capability

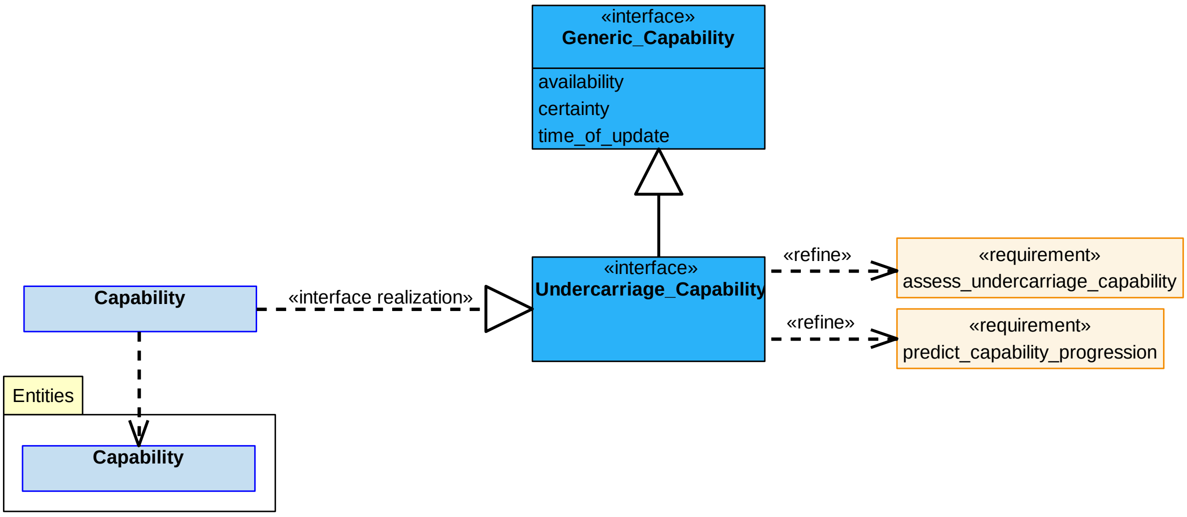


Figure 1149: Capability Service Definition

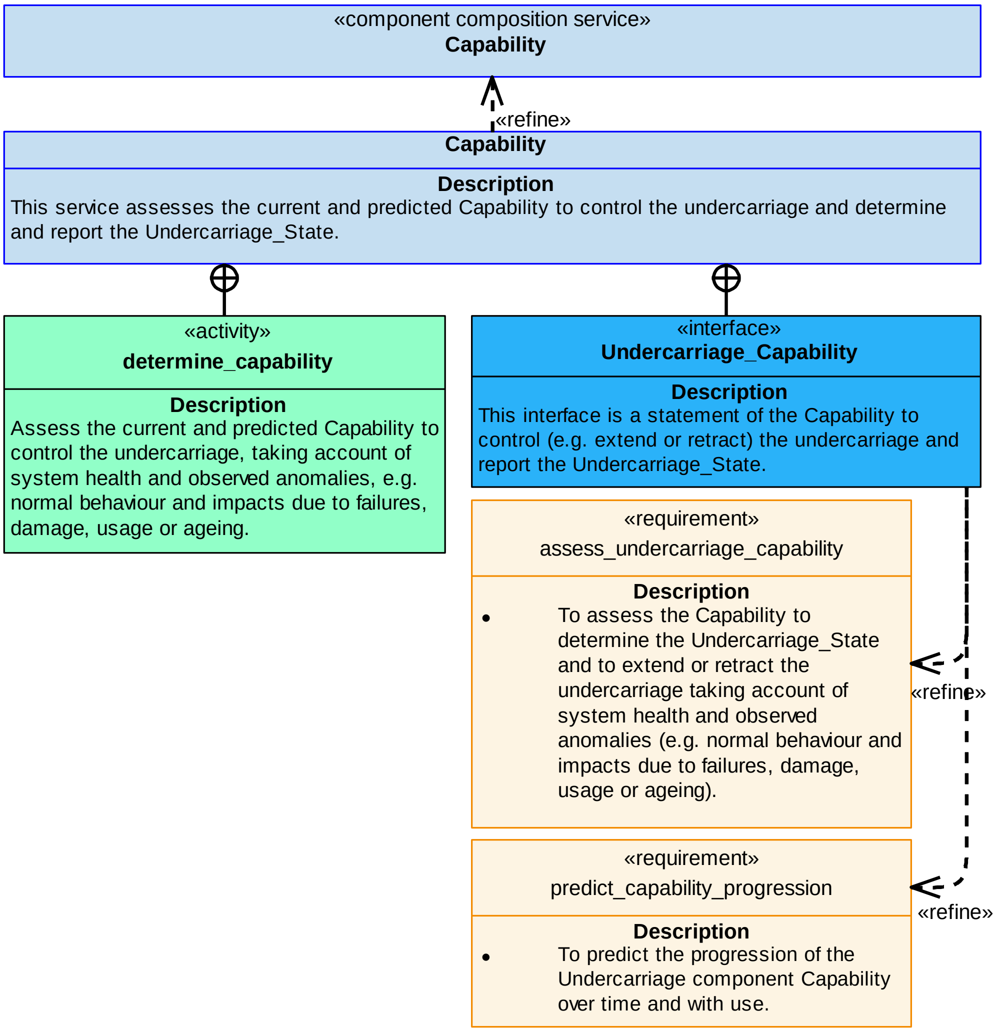

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the current and predicted Capability to control the undercarriage and determine and report the Undercarriage_State.
«activity»
determine_capability
Description
Assess the current and predicted Capability to control the undercarriage, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.
«interface»
Undercarriage_Capability
Description
This interface is a statement of the Capability to control (e.g. extend or retract) the undercarriage and report the Undercarriage_State.
«requirement»
assess_undercarriage_capability
Description
To assess the Capability to determine the Undercarriage_State and to extend or retract the undercarriage taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»
«requirement»
predict_capability_progression
Description
To predict the progression of the Undercarriage component Capability over time and with use.
«refine»

Figure 1150: Capability Service Policy

Capability

This service assesses the current and predicted Capability to control the undercarriage and determine and report the Undercarriage_State.

Interface

Undercarriage_Capability

This interface is a statement of the Capability to control (e.g. extend or retract) the undercarriage and report the Undercarriage_State.

Activity

determine_capability

Assess the current and predicted Capability to control the undercarriage, taking account of system health and observed anomalies, e.g. normal behaviour and impacts due to failures, damage, usage or ageing.

5.4.2.66.7.1.7 Capability_Evidence

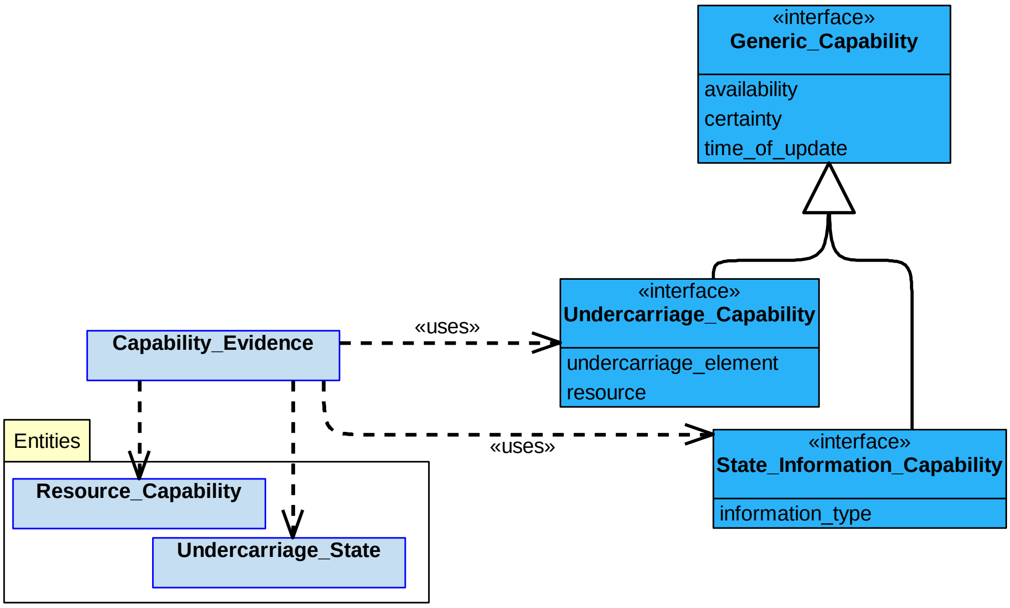


Figure 1151: Capability_Evidence Service Definition

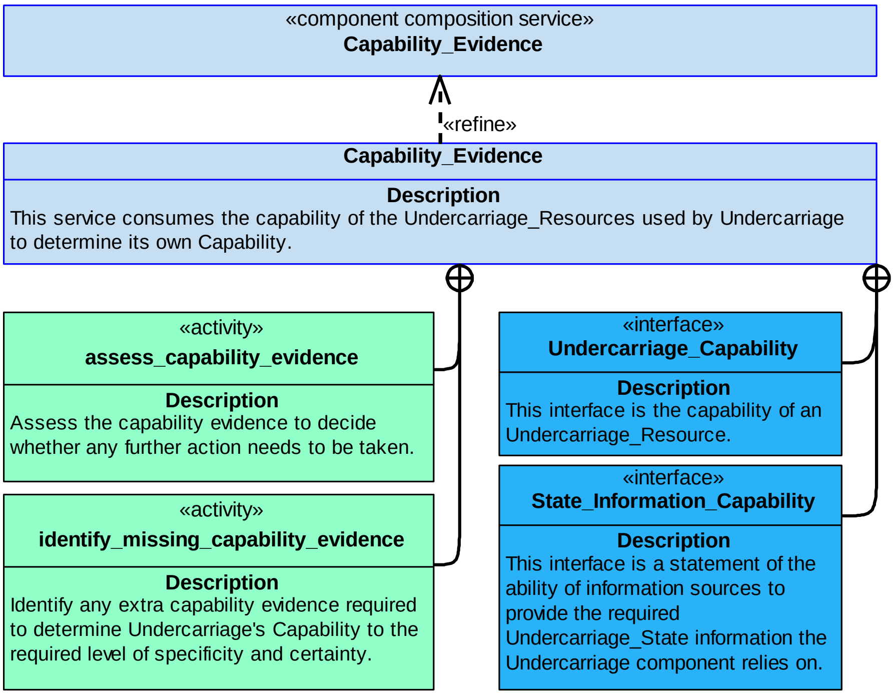

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes the capability of the Undercarriage_Resources used by Undercarriage to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine Undercarriage's Capability to the required level of specificity and certainty.
«interface»
Undercarriage_Capability
Description
This interface is the capability of an Undercarriage_Resource.
«interface»
State_Information_Capability
Description
This interface is a statement of the ability of information sources to provide the required Undercarriage_State information the Undercarriage component relies on.

Figure 1152: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the capability of the Undercarriage_Resources used by Undercarriage to determine its own Capability.

Interfaces

Undercarriage_Capability

This interface is the capability of an Undercarriage_Resource.

Attributes

undercarriage_element	The specific element of the undercarriage (e.g. nose wheel).
resource	The particular resource associated with the undercarriage_element (e.g. effectors to despin or align the wheels, retract and lock the undercarriage assembly, or to close apertures after retraction).

State_Information_Capability

This interface is a statement of the ability of information sources to provide the required Undercarriage_State information the Undercarriage component relies on.

Attribute

information_type	The specific item of information to which the evidence applies.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine Undercarriage's Capability to the required level of specificity and certainty.

5.4.2.66.7.2 Service Dependencies

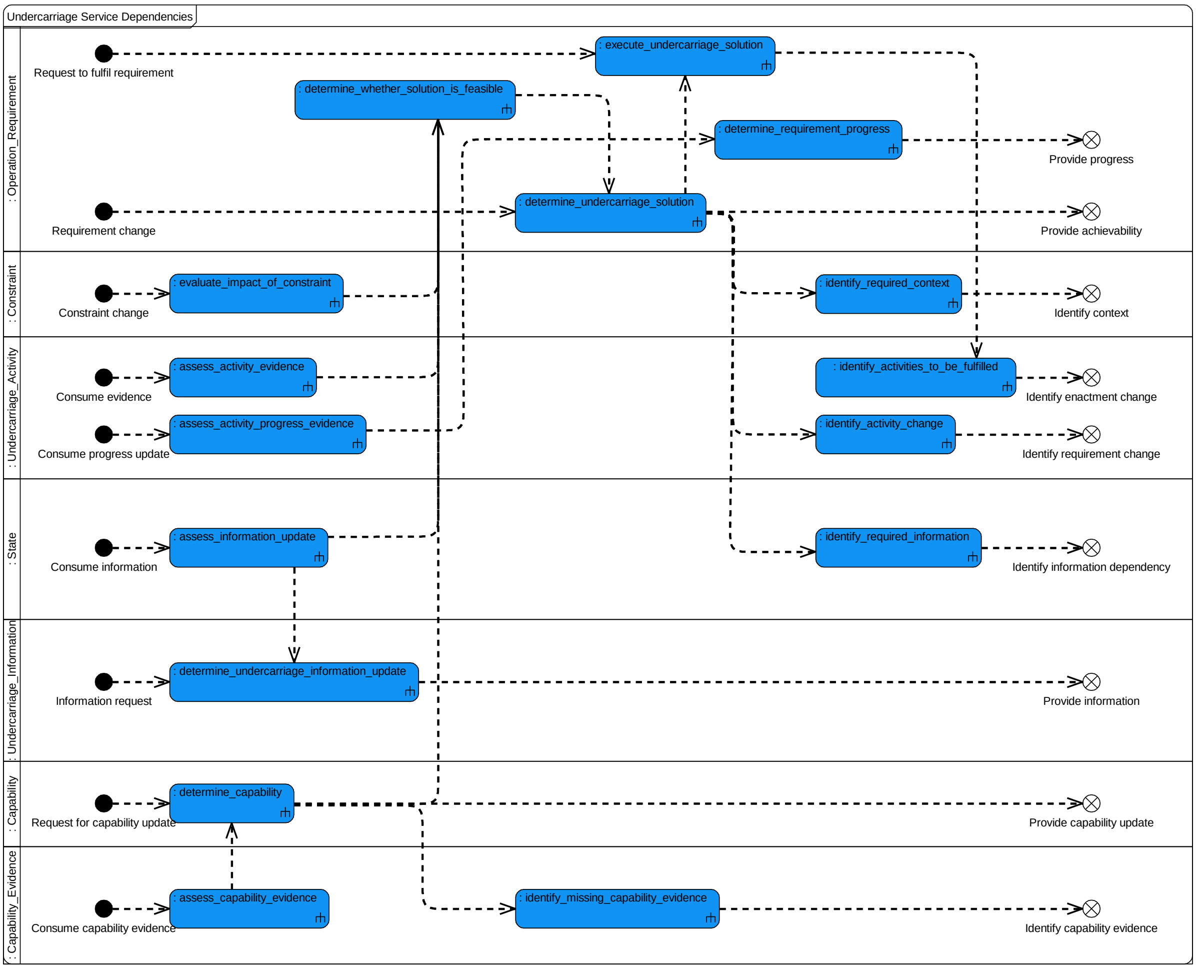

Undercarriage Service Dependencies
: Operation_Requirement
: Constraint
: Undercarriage_Activity
: State
: Undercarriage_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_undercarriage_solution
: determine_whether_solution_is_feasible
: determine_requirement_progress
Provide progress
Requirement change
: determine_undercarriage_solution
Provide achievability
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume evidence
: assess_activity_evidence
: identify_activities_to_be_fulfilled
Identify enactment change
Consume progress update
: assess_activity_progress_evidence
: identify_activity_change
Identify requirement change
Consume information
: assess_information_update
: identify_required_information
Identify information dependency
Information request
: determine_undercarriage_information_update
Provide information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1153: Undercarriage Service Dependencies

5.4.2.67 User Accounts

5.4.2.67.1 Role

The role of User Accounts is to co-ordinate access for users.

5.4.2.67.2 Overview

Control Architecture

User Accounts is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

A User interacts with an Authentication_Mechanism in order to provide valid credentials to log into a User_Account. If the credentials are invalid, the User is unable to log into the account and, depending on how the exploitation has been set up, appropriate follow on action may be taken, for example the User_Account may be locked or deleted.

Examples of Use

User Accounts can be used when there is a requirement to:

- Protect against unauthorised Users.
- Control a User's access to data.
- Identify the equipment a User needs to facilitate access.
- Limit the actions that a User can perform.

5.4.2.67.3 Service Summary

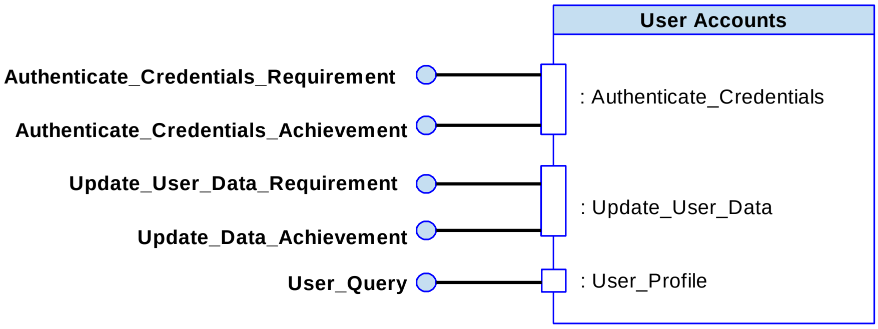

User Accounts
Authenticate_Credentials_Requirement
Authenticate_Credentials_Achievement
: Authenticate_Credentials
Update_User_Data_Requirement
Update_Data_Achievement
: Update_User_Data
User_Query
: User_Profile

Figure 1154: User Accounts Service Summary

5.4.2.67.4 Responsibilities

validate_logon_attempt

- To determine whether a log on attempt is valid (by checking the supplied User_Identity and Credentials).

validate_user_credentials

- To validate whether the supplied credentials for a User are valid.

determine_user_status

- To determine User status information (e.g. logon status, specific equipment needs or clearance levels).

determine_changes_to_user_accounts

- To add, update or delete User_Accounts according to specified rules (e.g. credential expiry or lock after failed login attempts).

capture_user_identity

- To capture the User_Identity.

capture_user_credentials

- To capture the Credentials used for the validation of Users (e.g. password or fingerprint).

capture_user_ability

- To capture the Ability of a User (e.g. whether a user is trained in performing certain actions).

capture_user_need

- To capture any User specific equipment User_Needs (e.g. wheelchair accessible workstation).

5.4.2.67.5 Subject Matter Semantics

The subject matter of User Accounts is the rules and mechanisms that determine the identity of a User, as well as the capture of user related information including qualification, approvals and any special equipment they may need.

Exclusions

The subject matter of User Accounts does not include:

- What permissions/roles a User has been given within the system.
- The capabilities or location of the User's workstation.

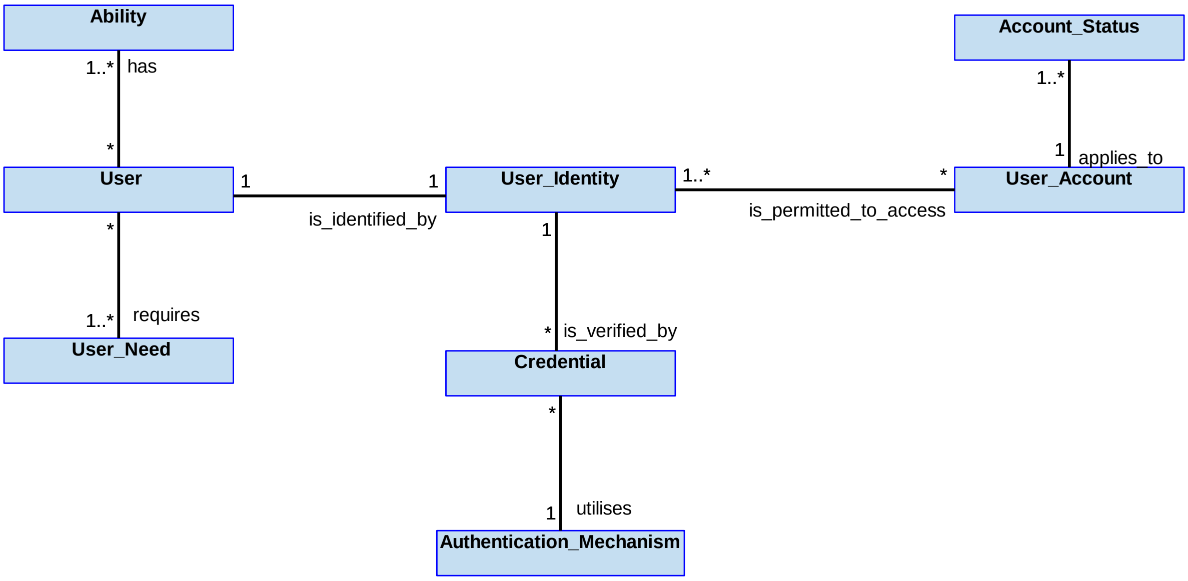


Figure 1155: User Accounts Semantics

5.4.2.67.5.1 Entities

Ability

The declared competency or clearance of the user, e.g. currently certified to pilot an X23 UAV, competent SharePoint admin or SC security cleared.

Account_Status

The current status of a User_Account (e.g. logged in, logged out, locked or expired).

Authentication_Mechanism

The mechanism that is required to allow a particular type of credential to be used (e.g. smart card reader or biometric scanner).

Credential

The evidence that a user may use to verify their authenticity, e.g. password or fingerprint.

User_Need

The equipment that is needed to support the user, independent of the role they are fulfilling (e.g. Braille interface required or left handed mouse preferred).

User

A user of the system who is uniquely identifiable, and the details of that user.

User_Account

Online representation of an individual or group for a particular application (e.g. email account, voice account or role access account).

User_Identity

The information that uniquely identifies a User.

5.4.2.67.6 Design Rationale

5.4.2.67.6.1 Assumptions

None.

5.4.2.67.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining User Accounts:

- Data Driving - Some aspects of this component may be data-driven prior to operation (e.g. User details or associations of User_Accounts to Users).
- Cyber Defence - There will be different levels of User access as described in this PYRAMID concept.
- Recording and Logging - This component will initiate logging of access attempts (successful or failed) in accordance with this PYRAMID concept and the requirements of the Exploiting Programme.

Note that the User Accounts component's capability is fixed at design time and is not dependent on consumed capability evidence and does not evolve with time, hence this component does not determine capability in the way described in the Capability Management PYRAMID concept.

Extensions

- Extension components to support different authentication mechanisms could be used.

Exploitation Considerations

- This component may be an abstract representation of (or a facade for) operating system services such as Kerberos, LDAP and Active Directory.
- A User_Account need not be associated with only one User; it may be associated with a post or shared by multiple Users.
- Most data in this component can be changed during operation through an administrative interface (e.g. adding user accounts or updating contact details).
- It will be up to the Exploiting Programme to determine the consequences of failed login attempts, for example, locking the User_Account.
- The ability to access a User_Account may vary depending on whether a User is local or remote. This is for an Exploiting Programme to determine.
- It is up to an Exploiting Programme to specify and manage the Authentication_Mechanisms.

5.4.2.67.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component may result in the inability of a User to login, have access to certain data, perform certain system admin activities or control certain functions. This may prevent the User gaining control of one or more functions of the air vehicle which may compromise safety. However, where human control is not available it is expected that the system would rely on pre-determined automatic / autonomous behaviour to perform essential functions. Therefore, it is concluded that failure of this component may result in a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.

Where instances of this component contribute to hazards that are less severe or more reliance may be placed on other barriers to an accident, then the Exploiting Platform may require a less onerous DAL.

5.4.2.67.6.4 Security Considerations

The indicative security classification is O-S.

This component is concerned with how a user identifies themselves to the system. It will have access to some personnel information that is considered O-S, with handing requirements required to comply with GDPR (article 5) Ref. [16] or similar legislation. However, it is also anticipated that instances of this component will be located within security domains that interface to a user. Where there are multiple security domains and instances, these may need to communicate and coordinate with each other. Where required, separation may be enforced by a boundary protection function located outside the component.

It will require protection against unauthorised manipulation; a high degree of integrity is expected to restrict access to permitted users only. Appropriate measures will be required to protect user details and credentials (passwords, biometrics, etc.), e.g. credentials will be hashed and salted when stored and transmitted. Credential storage will require a high degree of confidentiality and integrity protection.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** relating to login attempts etc. for later examination.
- **Maintaining Audit Records** for non-repudiation of users attached to the system at a given time.

The component will play a part in satisfying security enforcing functions relating to:

- **User Login and Authentication**, which is its fundamental purpose.

5.4.2.67.7 Services

5.4.2.67.7.1 Service Definitions

5.4.2.67.7.1.1 Authenticate_Credentials

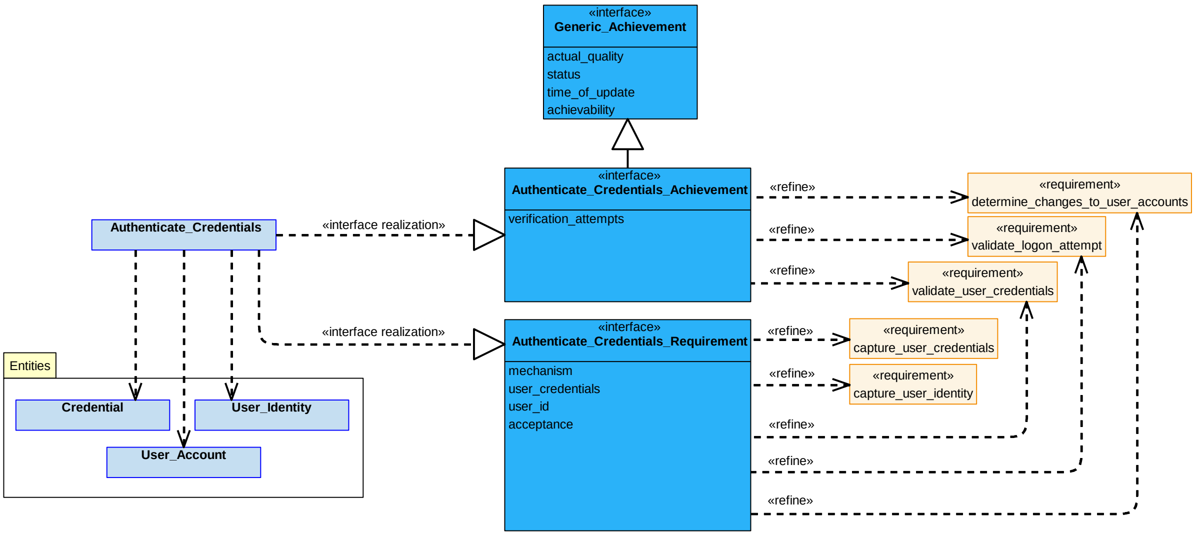


Figure 1156: Authenticate_Credentials Service Definition

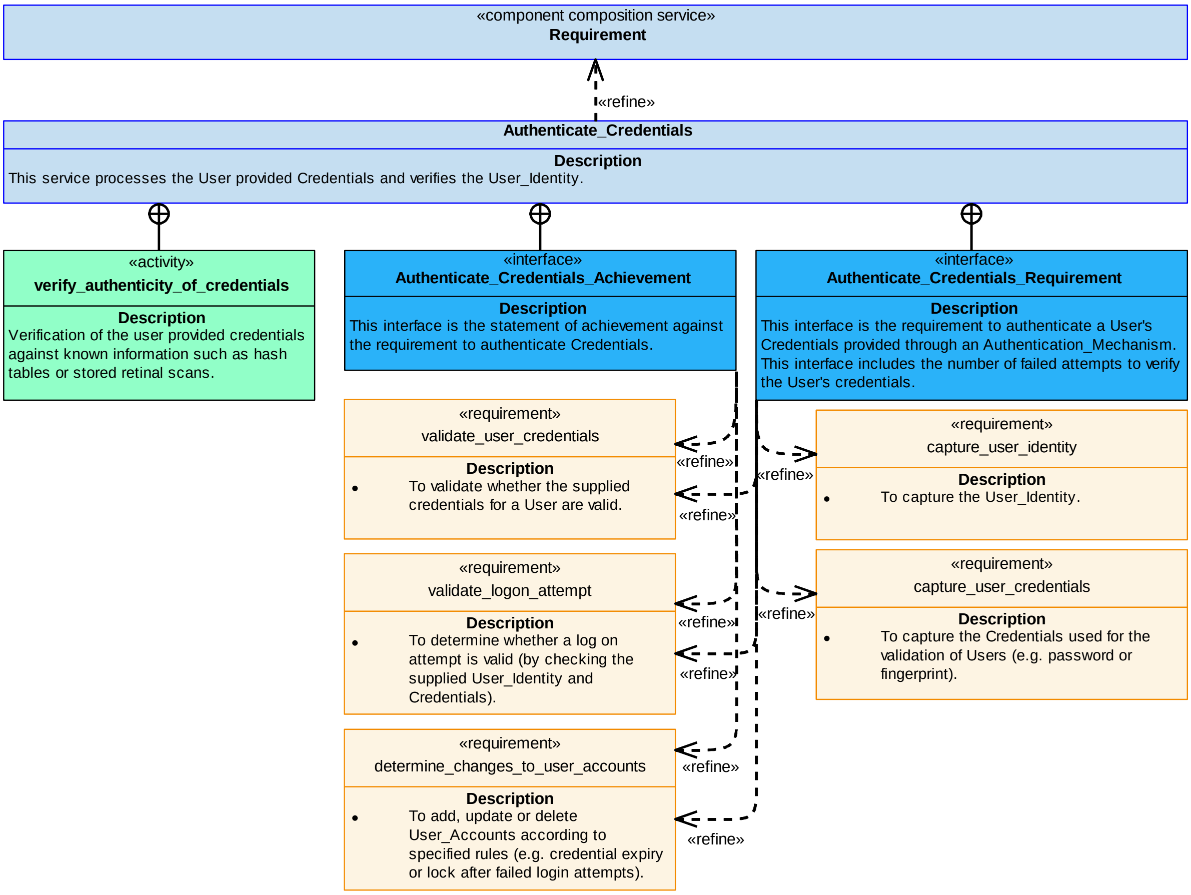

«component composition service»
Requirement
«refine»
Authenticate_Credentials
Description
This service processes the User provided Credentials and verifies the User_Identity.
«activity»
verify_authenticity_of_credentials
Description
Verification of the user provided credentials against known information such as hash tables or stored retinal scans.
«interface»
Authenticate_Credentials_Achievement
Description
This interface is the statement of achievement against the requirement to authenticate Credentials.
«interface»
Authenticate_Credentials_Requirement
Description
This interface is the requirement to authenticate a User's Credentials provided through an Authentication_Mechanism. This interface includes the number of failed attempts to verify the User's credentials.
«requirement»
validate_user_credentials
Description
To validate whether the supplied credentials for a User are valid.
«requirement»
validate_logon_attempt
Description
To determine whether a log on attempt is valid (by checking the supplied User_Identity and Credentials).
«requirement»
determine_changes_to_user_accounts
Description
To add, update or delete User_Accounts according to specified rules (e.g. credential expiry or lock after failed login attempts).
«requirement»
capture_user_identity
Description
To capture the User_Identity.
«requirement»
capture_user_credentials
Description
To capture the Credentials used for the validation of Users (e.g. password or fingerprint).
«refine»
«refine»
«refine»
«refine»
«refine»
«refine»
«refine»
«refine»

Figure 1157: Authenticate_Credentials Service Policy

Authenticate_Credentials

This service processes the User provided Credentials and verifies the User_Identity.

Interfaces

Authenticate_Credentials_Requirement

This interface is the requirement to authenticate a User's Credentials provided through an Authentication_Mechanism. This interface includes the number of failed attempts to verify the User's credentials.

Attributes

mechanism	The Authentication_Mechanism associated with the provision of a User’s Credentials (e.g. smart card reader or biometric scanner).
user_credentials	The Credentials that a User has provided to verify their authenticity.
user_id	A unique identifier for the user, e.g. john_smith26.
acceptance	Whether the Credentials have been accepted or rejected.

Authenticate_Credentials_Achievement

This interface is the statement of achievement against the requirement to authenticate Credentials.

Attribute

verification_attempts	The number of attempts a User has tried to verify their Credentials.

Activity

verify_authenticity_of_credentials

Verification of the user provided credentials against known information such as hash tables or stored retinal scans.

5.4.2.67.7.1.2 Update_User_Data

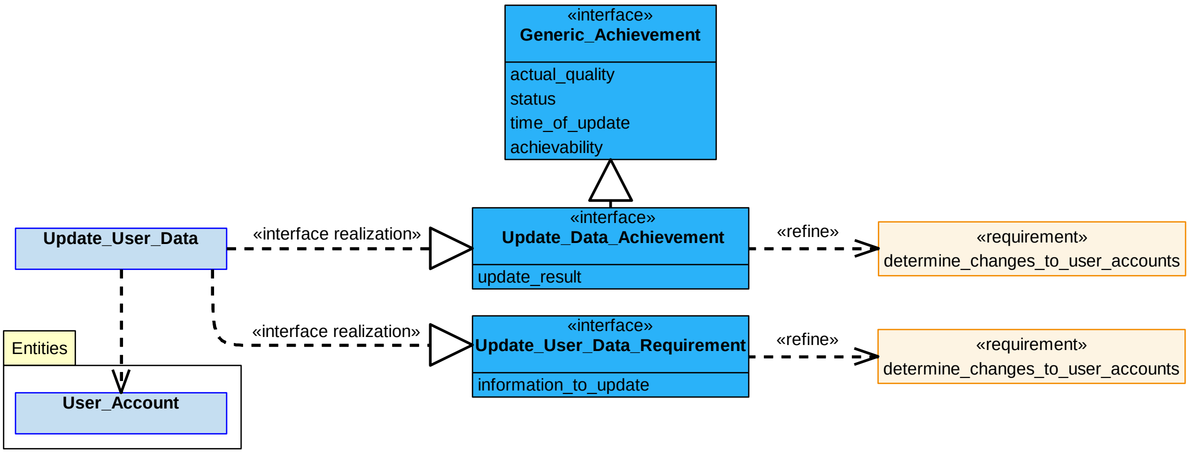
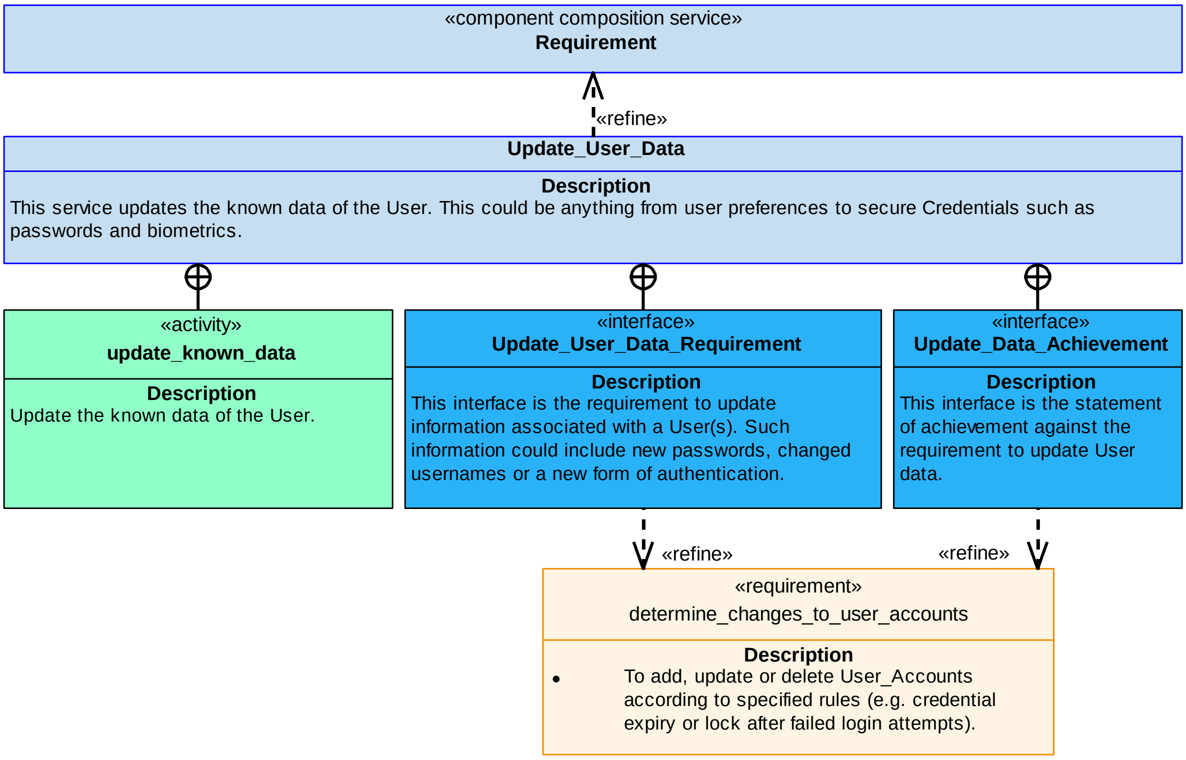


Figure 1158: Update_User_Data Service Definition



Description
This service updates the known data of the User. This could be anything from user preferences to secure Credentials such as passwords and biometrics.

Description
Update the known data of the User.

Description
This interface is the requirement to update information associated with a User(s). Such information could include new passwords, changed usernames or a new form of authentication.

Description
This interface is the statement of achievement against the requirement to update User data.

Description
- To add, update or delete User_Accounts according to specified rules (e.g. credential expiry or lock after failed login attempts).

Figure 1159: Update_User_Data Service Policy

Update_User_Data

This service updates the known data of the User. This could be anything from user preferences to secure Credentials such as passwords and biometrics.

Interfaces

Update_User_Data_Requirement

This interface is the requirement to update information associated with a User(s). Such information could include new passwords, changed usernames or a new form of authentication.

Attribute

information_to_update	This is the information that the User wishes to update, this could be a password, email address or personal data such as an address.

Update_Data_Achievement

This interface is the statement of achievement against the requirement to update User data.

Attribute

update_result	Whether the User's data has been successfully updated or not.

Activity

update_known_data

Update the known data of the User.

5.4.2.67.7.1.3 User_Profile

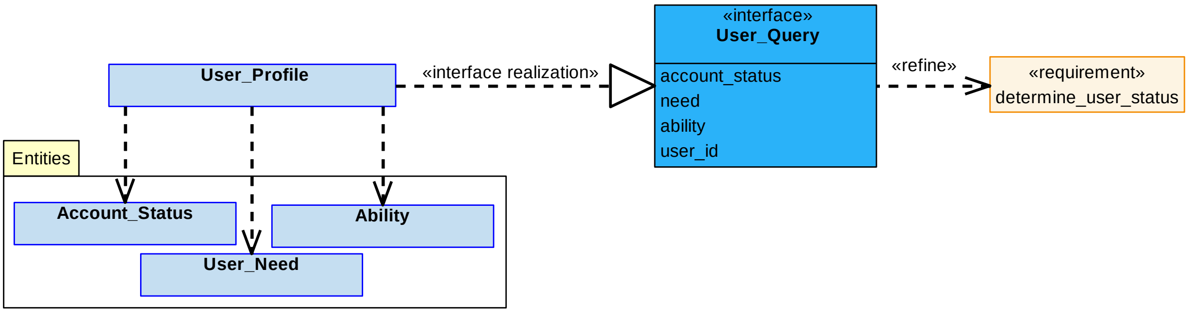


Figure 1160: User_Profile Service Definition

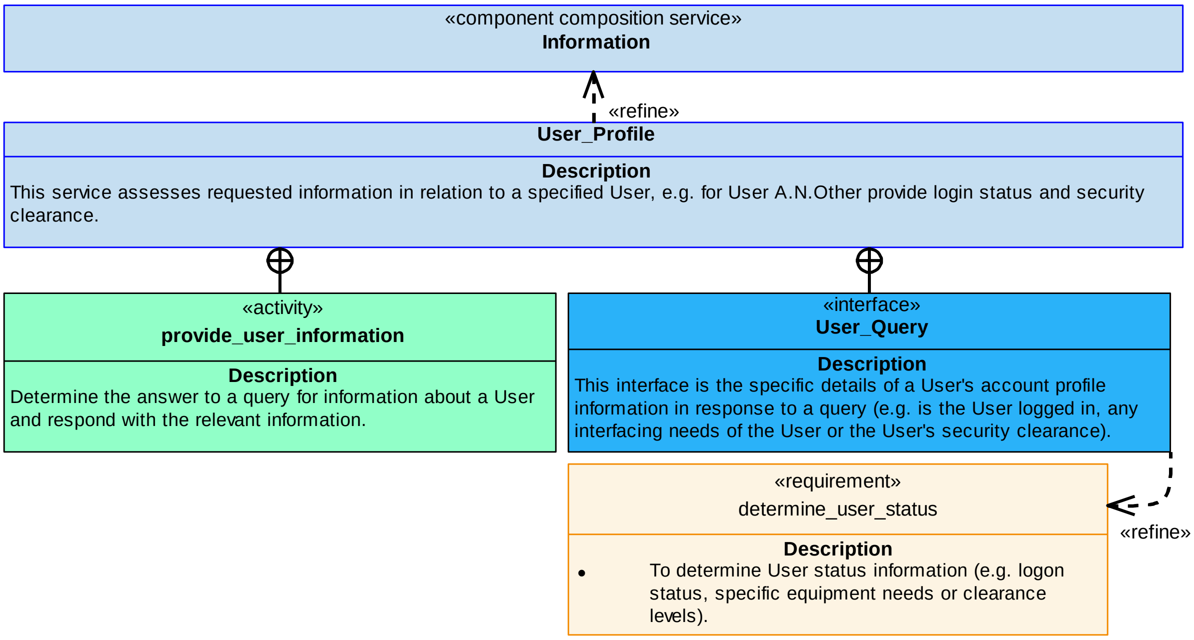

«component composition service»
Information
«refine»
User_Profile
Description
This service assesses requested information in relation to a specified User, e.g. for User A.N.Other provide login status and security clearance.
«activity»
provide_user_information
Description
Determine the answer to a query for information about a User and respond with the relevant information.
«interface»
User_Query
Description
This interface is the specific details of a User's account profile information in response to a query (e.g. is the User logged in, any interfacing needs of the User or the User's security clearance).
«requirement»
determine_user_status
Description
To determine User status information (e.g. logon status, specific equipment needs or clearance levels).
«refine»

Figure 1161: User_Profile Service Policy

User_Profile

This service assesses requested information in relation to a specified User, e.g. for User A.N.Other provide login status and security clearance.

Interface

User_Query

This interface is the specific details of a User's account profile information in response to a query (e.g. is the User logged in, any interfacing needs of the User or the User's security clearance).

Attributes

account_status	The status of a User_Account (e.g. logged in, logged out or last login date/time).
need	The needs of a User (e.g. left hand mouse settings, screen font sizes, colour settings or audio alerting).
ability	The Ability of the User (e.g. appropriate training completed, security clearance, system administrator or rank).
user_id	A unique identifier for the User, e.g. john_smith26

Activity

provide_user_information

Determine the answer to a query for information about a User and respond with the relevant information.

5.4.2.67.7.2 Service Dependencies

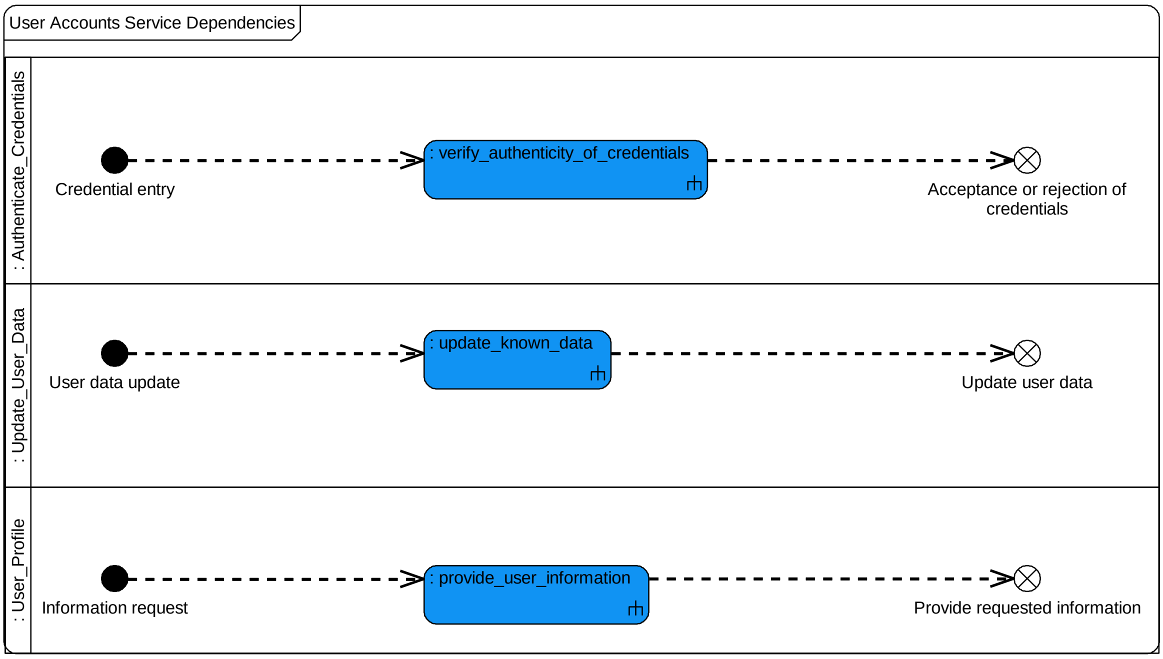

User Accounts Service Dependencies
: Authenticate_Credentials
Credential entry
: verify_authenticity_of_credentials
Acceptance or rejection of credentials
: Update_User_Data
User data update
: update_known_data
Update user data
: User_Profile
Information request
: provide_user_information
Provide requested information

Figure 1162: User Accounts Service Dependencies

5.4.2.68 User Roles

5.4.2.68.1 Role

The role of User Roles is to control the allocation of roles to authorised operators.

5.4.2.68.2 Overview

Control Architecture

User Roles is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

A User is allocated a Role if they fulfil both the required Qualification for the Role_Type and any other Constraints such as if the User is in a suitable location and is able to take on control.

Examples of Use

User Roles can be used when there is a requirement to:

- Change a Role of a User.
- Limit the handing over of Roles between Users.
- Advertise the permitted activities available to a User based on the Roles allocated.

5.4.2.68.3 Service Summary

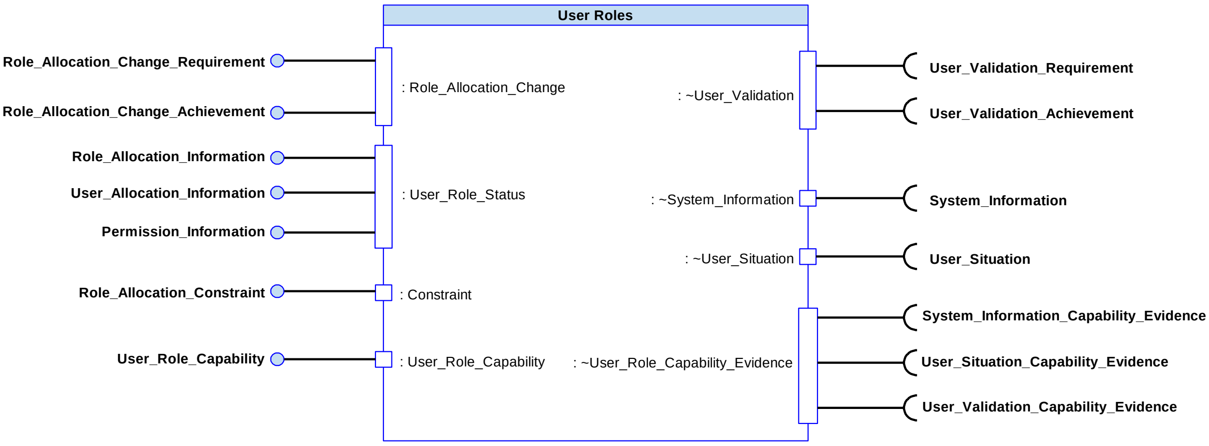

User Roles
Role_Allocation_Change_Requirement
Role_Allocation_Change_Achievement
: Role_Allocation_Change
Role_Allocation_Information
User_Allocation_Information
Permission_Information
: User_Role_Status
Role_Allocation_Constraint
: Constraint
User_Role_Capability
: User_Role_Capability
: ~User_Validation
User_Validation_Requirement
User_Validation_Achievement
: ~System_Information
System_Information
: ~User_Situation
User_Situation
: ~User_Role_Capability_Evidence
System_Information_Capability_Evidence
User_Situation_Capability_Evidence
User_Validation_Capability_Evidence

Figure 1163: User Roles Service Summary

5.4.2.68.4 Responsibilities

capture_requirements_for_role_allocation_change

- To capture provided requirements for User to Role allocation, deallocation and handover requests.

capture_constraints

- To capture the Constraints on the mapping of a User to a Role.

determine_user_permissions

- To determine the permissions allocated to a User as part of a Role_Type (e.g. whether a user has permission to change file access permissions as part of their system administrative role).

determine_operator_suitability_for_handover

- To determine the suitability of a User to receive a Role (e.g. if they are suitably qualified and the equipment is in place).

determine_equipment_required_for_a_role

- To determine the equipment required to support a User in a Role_Type.

determine_if_role_allocation_is_allowed

- To determine if allocation of a particular Role to a particular User is permissible.

determine_if_allocation_change_is_feasible

- To determine if a planned or on-going allocation, deallocation, or handover of a User or users to/from a user Role is feasible given current Allocation_Capability and Constraints.

allocate_roles

- To allocate, re-allocate and de-allocate Roles to Users.

perform_role_handover

- To perform the handover of Roles between Users.

assess_capability

- To assess the Allocation_Capability taking account of system health, observed anomalies, and availability (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Allocation_Capability assessment.

predict_capability_progression

- To predict the progression of Allocation_Capability over time and with use.

5.4.2.68.5 Subject Matter Semantics

The subject matter of User Roles is the mapping between Users and Roles.

Exclusions

The subject matter of User Roles does not include:

- Identity verification of authorised operators.
- Capability tracking of actual workstations or what equipment they are fitted with.

- Communication path tracking from the User to a remote unit.

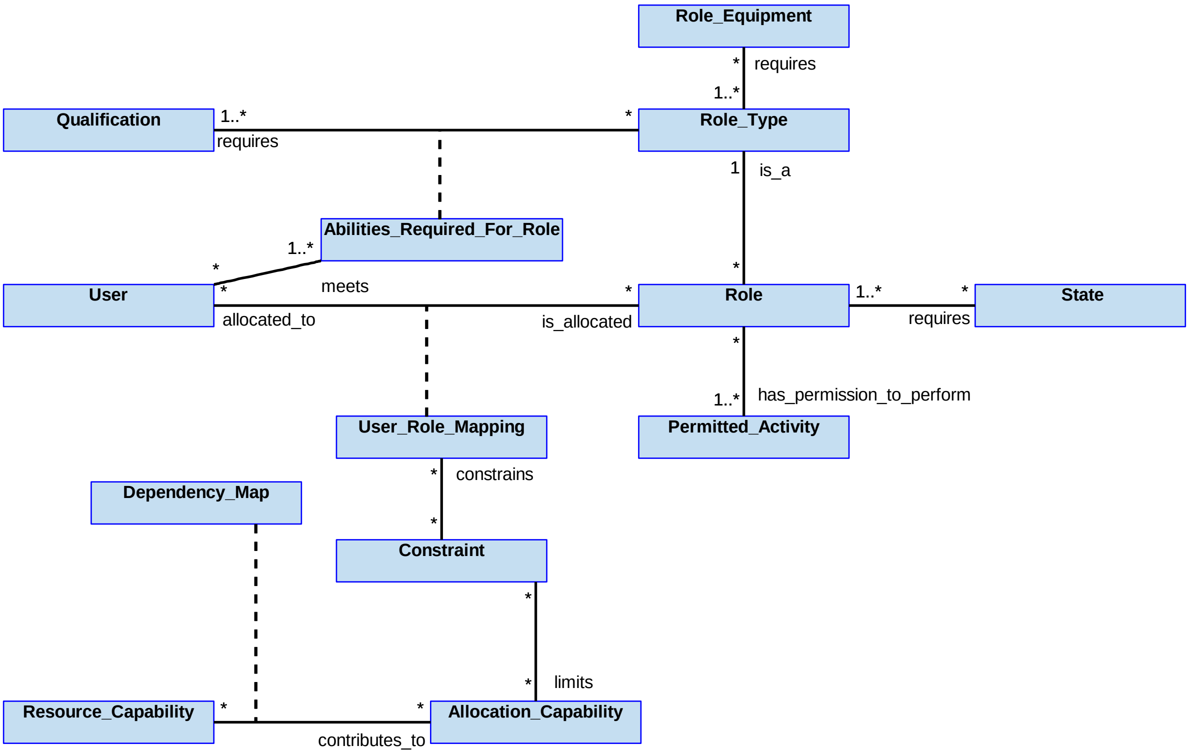


Figure 1164: User Roles Semantics

5.4.2.68.5.1 Entities

Abilities_Required_For_Role

What is considered suitably qualified or experienced for a particular role (e.g. to be a remote pilot of a particular type vehicle, you need to be trained to pilot that vehicle type, have the correct nationality and have security clearances).

Constraint

A restriction on what Roles can be allocated to, deallocated from and transferred between users, and the circumstances (such as when) in which the restriction applies. Typical constraint types are reachability, or mission phase, e.g. handover of flight control during a landing sequence needs to be to someone who can physically see the vehicle.

Permitted_Activity

The activity that a Role has responsibility and privilege level to undertake (e.g. User is allowed to control guidance on UAV with tail number ZJ929, User can read/write files for this mission or User has the rights to allocate a User to the UAV pilot role).

Role

The specific role that can be allocated (e.g. remote pilot of UAV with tail number ZJ929).

Role_Equipment

The device, equipment, or system required to support a User Role type (e.g. to be able to perform a role associated with remote operation of a platform a user may need a user terminal, some communication devices and the platform may need to be in a state where it could be controlled).

Qualification

Qualification, experience, ability or quality (e.g. trained to pilot a particular vehicle type, security clearance to see SNEO or can view UK export controlled information).

Role_Type

The type of role (e.g. a particular type vehicle remote pilot, payload operator, exploitation expert or sysadmin).

User

An entity, authorised operator or group that can take on a control Role within the system.

User_Role_Mapping

Management of the state of the association of a User with a Role (e.g. can be allocated or is allocated) and how they change (e.g. handover of roles between users).

Allocation_Capability

The ability to change the allocation of Users to Roles.

Resource_Capability

The capability to provide the source information needed to support the allocation of users to Roles (e.g. a source of information about the location of a User or devices needed to fulfil a Role, or a source stating that a User is valid and therefore allowed to be considered for allocation to a Role).

Dependency_Map

The available Allocation_Capability that can be performed with the available Resource_Capability.

State

The state of the system (e.g. weapons armed) or the state of the mission (e.g. in combat) required to support a User Role.

5.4.2.68.6 Design Rationale

5.4.2.68.6.1 Assumptions

- Roles will have different responsibilities and access to different actions, information and resources.
- The component will know who can initiate a handover. This will need to be protected to avoid an unauthorised user instigating a handover.
- This component will be able to determine the Role_Equipment (e.g. understanding the HMI requirements to perform a type of role).

- The Role_Equipment information will be used in conjunction with the actual equipment fit and user location recorded in other components to determine if a role can be transferred to a specific user.
- This component will co-ordinate handovers including automated pre-planned ones.
- This component will know which actions can be performed by each type of role (e.g. control a function, access information, provide authorisation and transfer roles).
- It is expected that a User will be capable of holding multiple Roles, e.g. 'all users' may have the role of 'read access to the system' and in addition some may have Roles to perform specific actions such as 'vehicle control'.

5.4.2.68.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining User Roles:

- Data Driving - Some aspects of the component can be data-driven, such as the rules for allowable User_Role_Mapping, the circumstances under which User to Role allocations are allowed to change, the Abilities_Required_For_Role and the permitted activities that a role provides. This therefore includes static Constraints, but not dynamic constraints that may change during a particular operating period.

Exploitation Considerations

- This component may be present within each node that owns one or more Roles (e.g. a UAV) or contains users (e.g. a UCS).
- The permission and access control relating to each Role_Type will need to be configured by the Exploiting Programme, along with what skills are expected from someone who takes on that Role_Type.
- The Role_Types and Roles used for different deployments will vary in their number and level of specificity, for example ranging from full (non administrative) access to the system for all users to an explicit role for piloting vehicle tail number ZJ924.

5.4.2.68.6.3 Safety Considerations

The indicative IDAL is DAL C*.*

The rationale behind this is:

- Failure of this component may result in the inability of an authorised operator to have access to certain data or control certain functions. It may also result in a loss of control during a handover. This may prevent the authorised operator controlling one or more functions of the air vehicle which may compromise safety. However, where human control is not available it is expected that the system would rely on pre-determined automatic or autonomous behaviour to perform essential functions. Therefore, it is concluded that failure of this component may result in a "significant reduction in safety margins", which has a major severity. Therefore, the indicative DAL is C.

- It is assumed that a failure of this component will not result in an authorised operator losing their currently assigned roles.

Where instances of this component contribute to hazards that are less severe or more reliance may be placed on other barriers to an accident, then the Exploiting Platform may require a less onerous DAL.

5.4.2.68.6.4 Security Considerations

The indicative security classification is O-S.

This component maps Roles to authorised operators and governs what they are allowed to do, as such is considered to be O-S. It is expected that this component will be required within security domains that interface to a User. Where there are multiple security domains and instances of the component, these may need to communicate with each other. Separation will be enforced by a boundary protection function located outside the component.

It is expected this component will require a high degree of protection against unauthorised manipulation, and that the integrity of input and output will be assured.

The component is expected to at least partially satisfy security related functions by:

- **Logging of Security Data** of role changes, permission elevation, etc. for later examination.
- **Maintaining Audit Records** for non-repudiation of roles held at a given time and with what permissions, role handovers requested and enacted, etc.
- **System Status and Monitoring** of role assignations and handovers; unexpected action might indicate that the system has been infiltrated by an unauthorised or malicious user.

The component will play a part in supporting security enforcing functions by:

- **Detecting Security Breaches** relating to privilege escalation, etc.
- **Restricting Access to Data** to only that which is appropriate to the role and privileges.
- Using **User Login and Authentication** information in the mapping of roles to qualified/suitable users.

5.4.2.68.7 Services

5.4.2.68.7.1 Service Definitions

5.4.2.68.7.1.1 Role_Allocation_Change

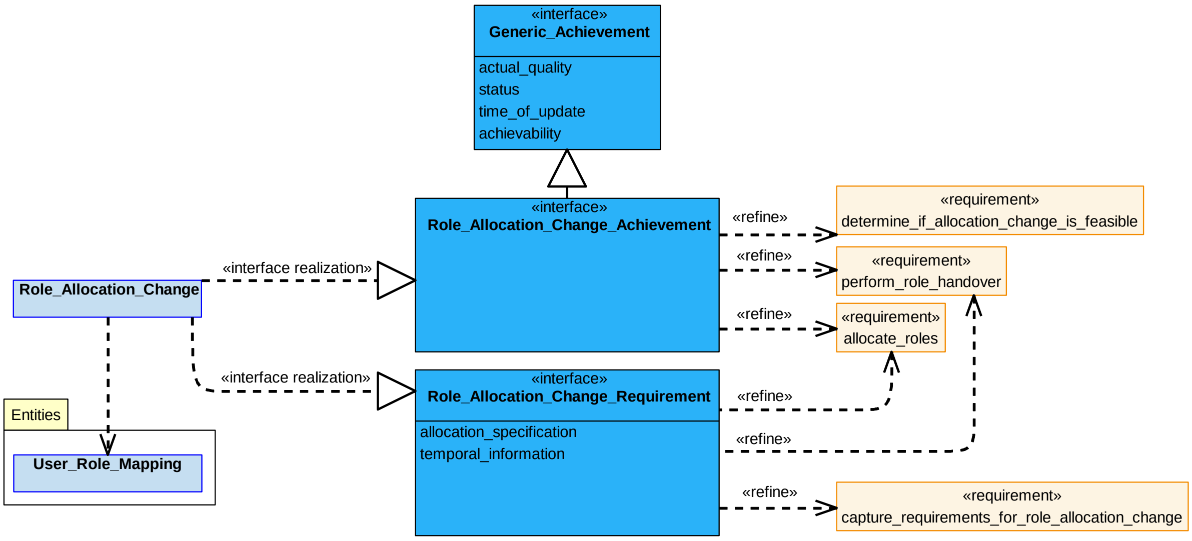


Figure 1165: Role_Allocation_Change Service Definition

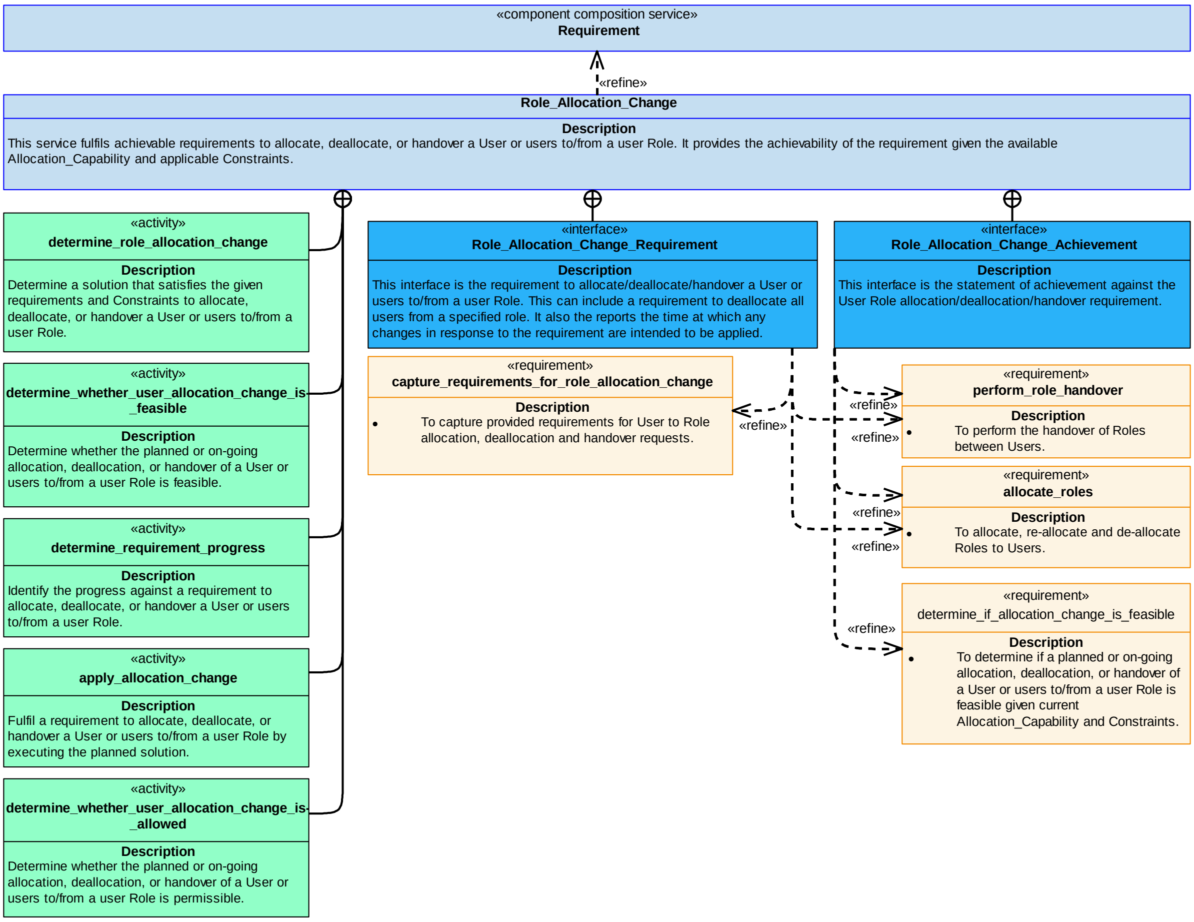

«component composition service»
Requirement
«refine»
Role_Allocation_Change
Description
This service fulfils achievable requirements to allocate, deallocate, or handover a User or users to/from a user Role. It provides the achievability of the requirement given the available Allocation_Capability and applicable Constraints.
«activity»
determine_role_allocation_change
Description
Determine a solution that satisfies the given requirements and Constraints to allocate, deallocate, or handover a User or users to/from a user Role.
«activity»
determine_whether_user_allocation_change_is_feasible
Description
Determine whether the planned or on-going allocation, deallocation, or handover of a User or users to/from a user Role is feasible.
«activity»
determine_requirement_progress
Description
Identify the progress against a requirement to allocate, deallocate, or handover a User or users to/from a user Role.
«activity»
apply_allocation_change
Description
Fulfil a requirement to allocate, deallocate, or handover a User or users to/from a user Role by executing the planned solution.
«activity»
determine_whether_user_allocation_change_is_allowed
Description
Determine whether the planned or on-going allocation, deallocation, or handover of a User or users to/from a user Role is permissible.
«interface»
Role_Allocation_Change_Requirement
Description
This interface is the requirement to allocate/deallocate/handover a User or users to/from a user Role. This can include a requirement to deallocate all users from a specified role. It also the reports the time at which any changes in response to the requirement are intended to be applied.
«interface»
Role_Allocation_Change_Achievement
Description
This interface is the statement of achievement against the User Role allocation/deallocation/handover requirement.
«requirement»
capture_requirements_for_role_allocation_change
Description
To capture provided requirements for User to Role allocation, deallocation and handover requests.
«refine»
«requirement»
perform_role_handover
Description
To perform the handover of Roles between Users.
«refine»
«refine»
«requirement»
allocate_roles
Description
To allocate, re-allocate and de-allocate Roles to Users.
«refine»
«refine»
«requirement»
determine_if_allocation_change_is_feasible
Description
To determine if a planned or on-going allocation, deallocation, or handover of a User or users to/from a user Role is feasible given current Allocation_Capability and Constraints.
«refine»

Figure 1166: Role_Allocation_Change Service Policy

Role_Allocation_Change

This service fulfils achievable requirements to allocate, deallocate, or handover a User or users to/from a user Role. It provides the achievability of the requirement given the available Allocation_Capability and applicable Constraints.

Interfaces

Role_Allocation_Change_Requirement

This interface is the requirement to allocate/deallocate/handover a User or users to/from a user Role. This can include a requirement to deallocate all users from a specified role. It also the reports the time at which any changes in response to the requirement are intended to be applied.

Attributes

allocation_specification	The definition of the requirement; for example including the User and the Role to be allocated/deallocated/handed-over, and the level of priority that the requirement has over existing or planned User Role allocations.
temporal_information	Information covering timing, such as the time at which a User will be allocated to a Role and the time at which a User will be deallocated.

Role_Allocation_Change_Achievement

This interface is the statement of achievement against the User Role allocation/deallocation/handover requirement.

Activities

determine_role_allocation_change

Determine a solution that satisfies the given requirements and Constraints to allocate, deallocate, or handover a User or users to/from a user Role.

determine_requirement_progress

Identify the progress against a requirement to allocate, deallocate, or handover a User or users to/from a user Role.

determine_whether_user_allocation_change_is_feasible

Determine whether the planned or on-going allocation, deallocation, or handover of a User or users to/from a user Role is feasible.

apply_allocation_change

Fulfil a requirement to allocate, deallocate, or handover a User or users to/from a user Role by executing the planned solution.

determine_whether_user_allocation_change_is_allowed

Determine whether the planned or on-going allocation, deallocation, or handover of a User or users to/from a user Role is permissible.

5.4.2.68.7.1.2 User_Validation

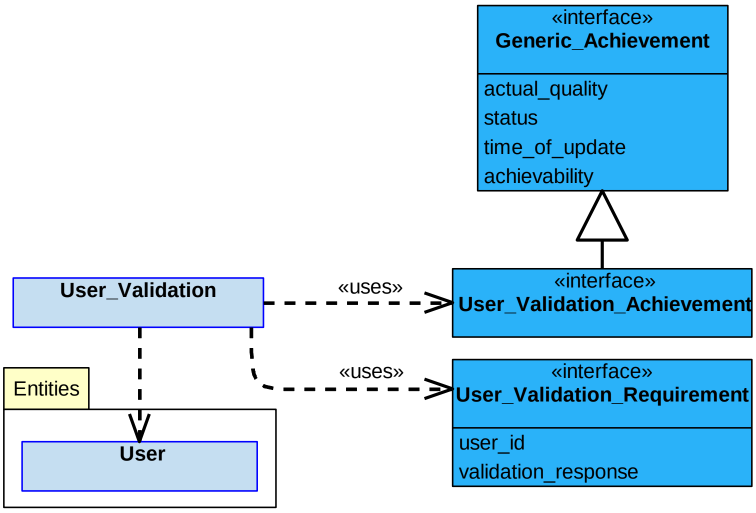


Figure 1167: User_Validation Service Definition

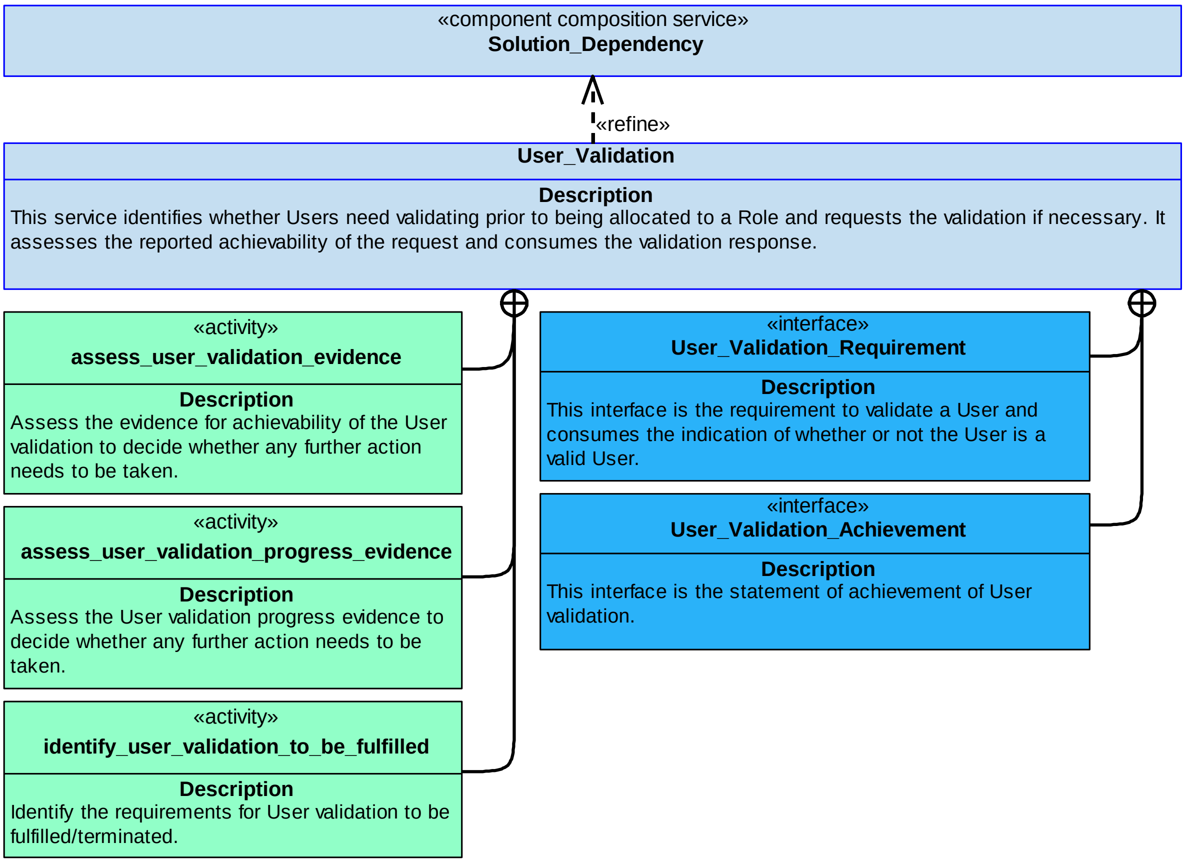

«component composition service»
Solution_Dependency
«refine»
User_Validation
Description
This service identifies whether Users need validating prior to being allocated to a Role and requests the validation if necessary. It assesses the reported achievability of the request and consumes the validation response.
«activity»
assess_user_validation_evidence
Description
Assess the evidence for achievability of the User validation to decide whether any further action needs to be taken.
«activity»
assess_user_validation_progress_evidence
Description
Assess the User validation progress evidence to decide whether any further action needs to be taken.
«activity»
identify_user_validation_to_be_fulfilled
Description
Identify the requirements for User validation to be fulfilled/terminated.
«interface»
User_Validation_Requirement
Description
This interface is the requirement to validate a User and consumes the indication of whether or not the User is a valid User.
«interface»
User_Validation_Achievement
Description
This interface is the statement of achievement of User validation.

Figure 1168: User_Validation Service Policy

User_Validation

This service identifies whether Users need validating prior to being allocated to a Role and requests the validation if necessary. It assesses the reported achievability of the request and consumes the validation response.

Interfaces

User_Validation_Achievement

This interface is the statement of achievement of User validation.

User_Validation_Requirement

This interface is the requirement to validate a User and consumes the indication of whether or not the User is a valid User.

Attributes

user_id	The user ID that is required to be validated.
validation_response	The indication of whether the User is a valid user or not.

Activities

assess_user_validation_evidence

Assess the evidence for achievability of the User validation to decide whether any further action needs to be taken.

assess_user_validation_progress_evidence

Assess the User validation progress evidence to decide whether any further action needs to be taken.

identify_user_validation_to_be_fulfilled

Identify the requirements for User validation to be fulfilled/terminated.

5.4.2.68.7.1.3 User_Role_Status

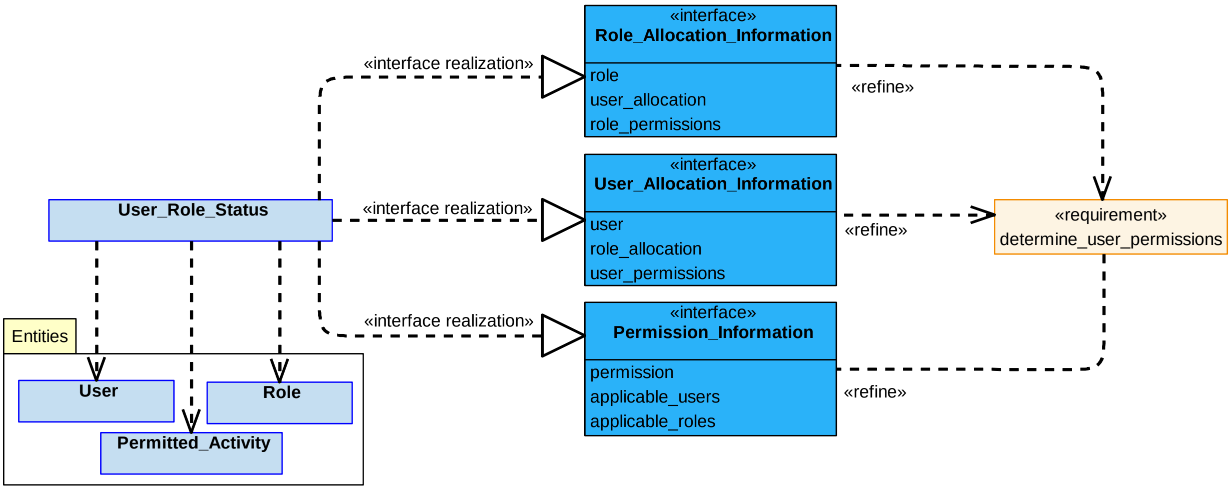
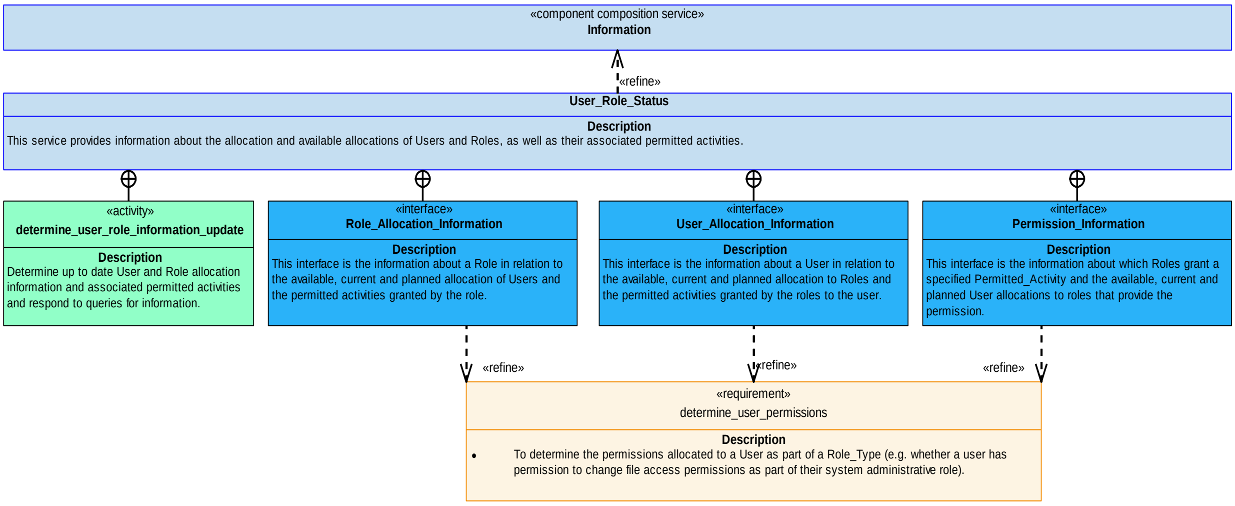


Figure 1169: User_Role_Status Service Definition


«component composition service»
Information
«refine»
User_Role_Status
«activity»
determine_user_role_information_update
«interface»
Role_Allocation_Information
«interface»
User_Allocation_Information
«interface»
Permission_Information
«refine»
«refine»
«refine»
«requirement»
determine_user_permissions

Description
This service provides information about the allocation and available allocations of Users and Roles, as well as their associated permitted activities.

Description
Determine up to date User and Role allocation information and associated permitted activities and respond to queries for information.

Description
This interface is the information about a Role in relation to the available, current and planned allocation of Users and the permitted activities granted by the role.

Description
This interface is the information about a User in relation to the available, current and planned allocation to Roles and the permitted activities granted by the roles to the user.

Description
This interface is the information about which Roles grant a specified Permitted_Activity and the available, current and planned User allocations to roles that provide the permission.

Description
- To determine the permissions allocated to a User as part of a Role_Type (e.g. whether a user has permission to change file access permissions as part of their system administrative role).

Figure 1170: User_Role_Status Service Policy

User_Role_Status

This service provides information about the allocation and available allocations of Users and Roles, as well as their associated permitted activities.

Interfaces

Role_Allocation_Information

This interface is the information about a Role in relation to the available, current and planned allocation of Users and the permitted activities granted by the role.

Attributes

role	The Role that information is being provided for.
user_allocation	The Users available to be allocated to the Role, currently allocated to the role and/or with a planned allocation to the role.
role_permissions	The permitted activities granted by the Role.

User_Allocation_Information

This interface is the information about a User in relation to the available, current and planned allocation to Roles and the permitted activities granted by the roles to the user.

Attributes

user	The User that information is being provided for.
role_allocation	The Roles available to be allocated to the User, currently allocated to the User and/or with a planned allocation to the User.
user_permissions	The available, current and planned User permitted activities granted by their available, current and planned Role allocations.

Permission_Information

This interface is the information about which Roles grant a specified Permitted_Activity and the available, current and planned User allocations to roles that provide the permission.

Attributes

permission	The Permitted_Activity that information is being provided for.
applicable_users	The available, current and planned Users allocations to Roles that provide the specified Permitted_Activity.
applicable_roles	The Roles that grant the specified Permitted_Activity.

Activity

determine_user_role_information_update

Determine up to date User and Role allocation information and associated permitted activities and respond to queries for information.

5.4.2.68.7.1.4 User_Situation

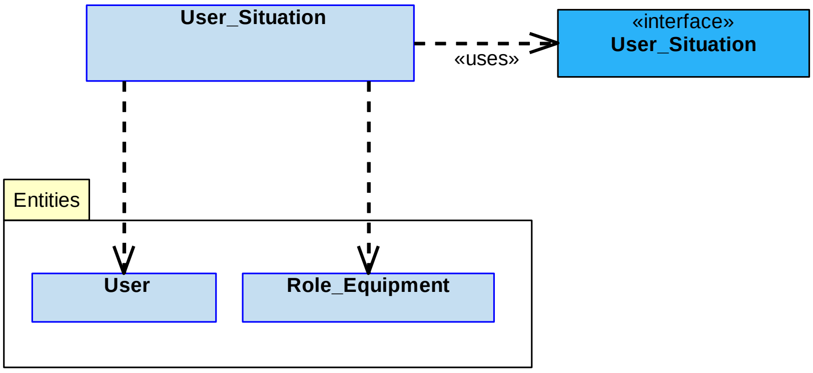


Figure 1171: User_Situation Service Definition

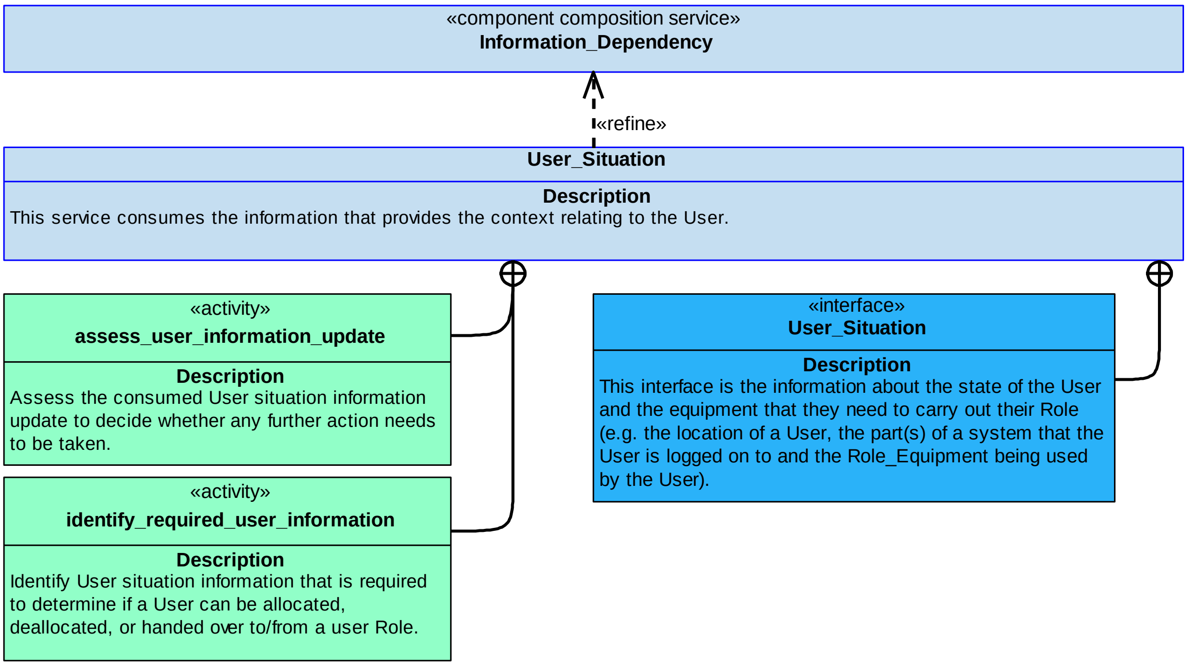

«component composition service»
Information_Dependency
«refine»
User_Situation
Description
This service consumes the information that provides the context relating to the User.
«activity»
assess_user_information_update
Description
Assess the consumed User situation information update to decide whether any further action needs to be taken.
«activity»
identify_required_user_information
Description
Identify User situation information that is required to determine if a User can be allocated, deallocated, or handed over to/from a user Role.
«interface»
User_Situation
Description
This interface is the information about the state of the User and the equipment that they need to carry out their Role (e.g. the location of a User, the part(s) of a system that the User is logged on to and the Role_Equipment being used by the User).

Figure 1172: User_Situation Service Policy

User_Situation

This service consumes the information that provides the context relating to the User.

Interface

User_Situation

This interface is the information about the state of the User and the equipment that they need to carry out their Role (e.g. the location of a User, the part(s) of a system that the User is logged on to and the Role_Equipment being used by the User).

Activities

assess_user_information_update

Assess the consumed User situation information update to decide whether any further action needs to be taken.

identify_required_user_information

Identify User situation information that is required to determine if a User can be allocated, deallocated, or handed over to/from a user Role.

5.4.2.68.7.1.5 System_Information

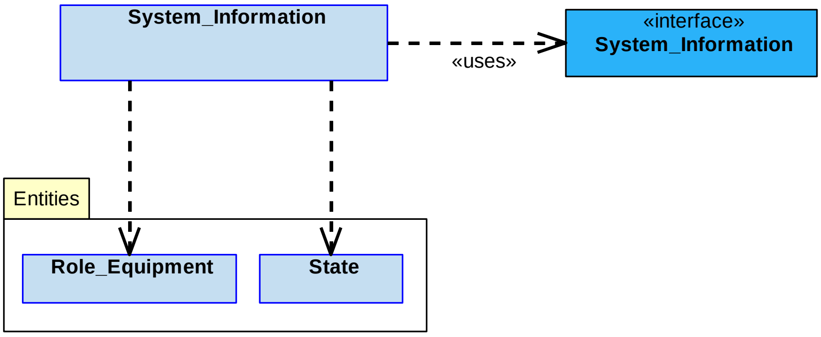
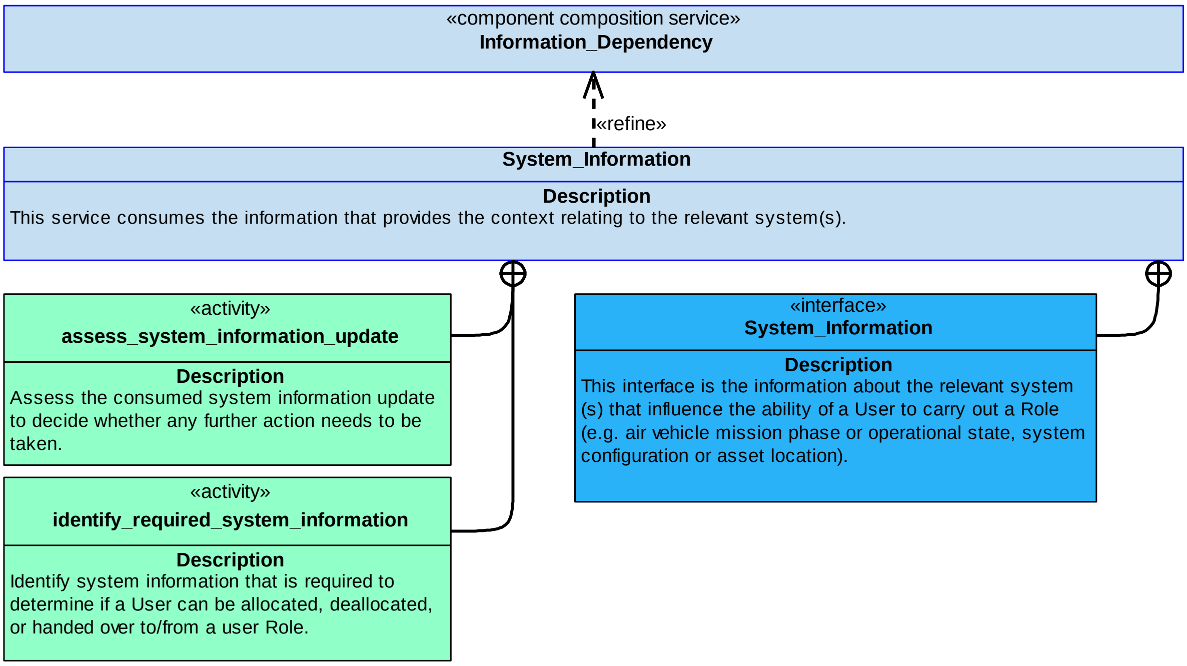


Figure 1173: System_Information Service Definition



This service consumes the information that provides the context relating to the relevant system(s).

Assess the consumed system information update to decide whether any further action needs to be taken.

Identify system information that is required to determine if a User can be allocated, deallocated, or handed over to/from a user Role.

This interface is the information about the relevant system (s) that influence the ability of a User to carry out a Role (e.g. air vehicle mission phase or operational state, system configuration or asset location).

Figure 1174: System_Information Service Policy

System_Information

This service consumes the information that provides the context relating to the relevant system(s).

Interface

System_Information

This interface is the information about the relevant system(s) that influence the ability of a User to carry out a Role (e.g. air vehicle mission phase or operational state, system configuration or asset location).

Activities

assess_system_information_update

Assess the consumed system information update to decide whether any further action needs to be taken.

identify_required_system_information

Identify system information that is required to determine if a User can be allocated, deallocated, or handed over to/from a user Role.

5.4.2.68.7.1.6 Constraint

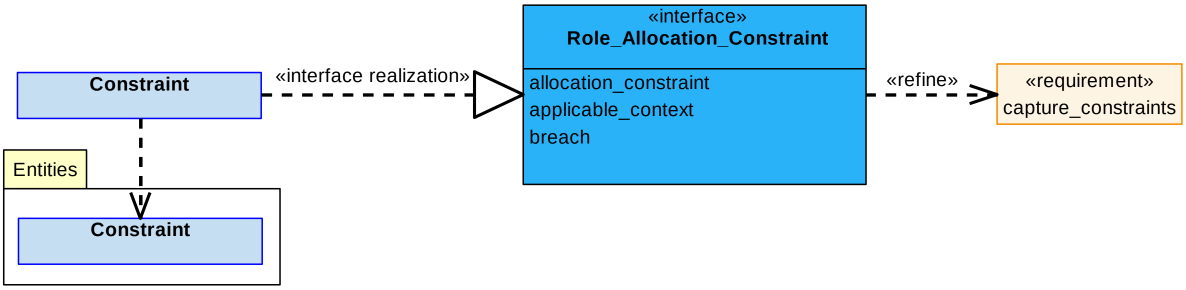


Figure 1175: Constraint Service Definition

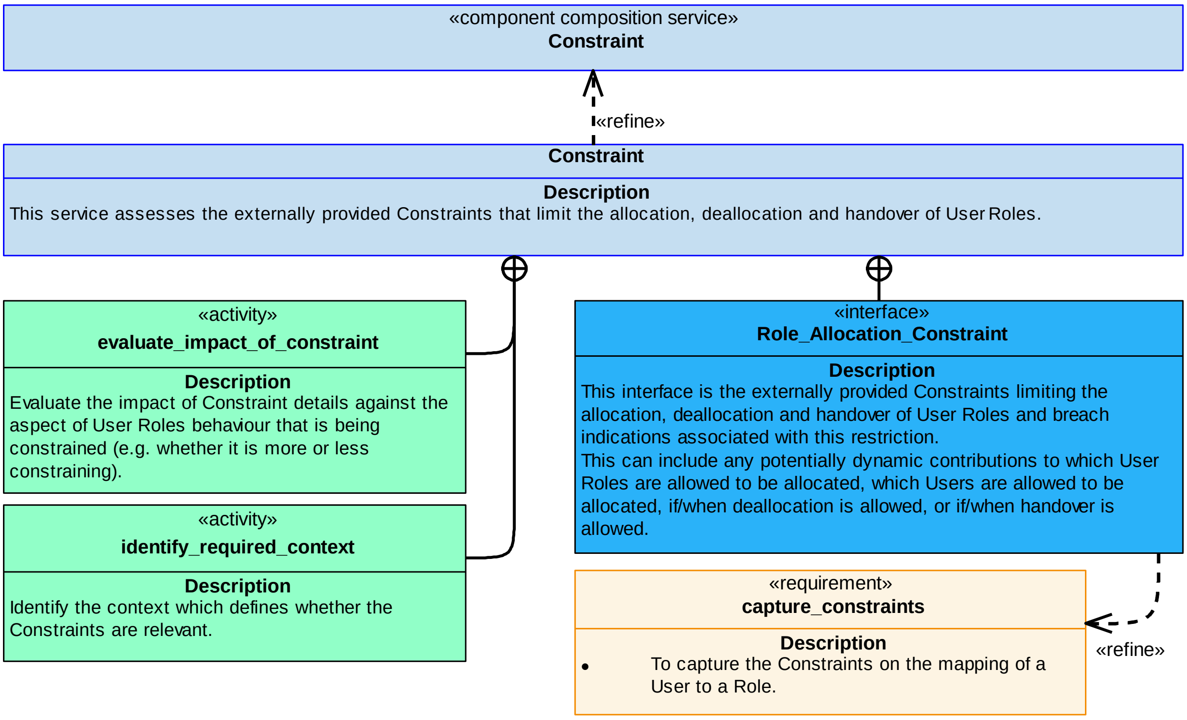

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the externally provided Constraints that limit the allocation, deallocation and handover of User Roles.
«activity»
evaluate_impact_of_constraint
Description
Evaluate the impact of Constraint details against the aspect of User Roles behaviour that is being constrained (e.g. whether it is more or less constraining).
«activity»
identify_required_context
Description
Identify the context which defines whether the Constraints are relevant.
«interface»
Role_Allocation_Constraint
Description
This interface is the externally provided Constraints limiting the allocation, deallocation and handover of User Roles and breach indications associated with this restriction.
This can include any potentially dynamic contributions to which User Roles are allowed to be allocated, which Users are allowed to be allocated, if/when deallocation is allowed, or if/when handover is allowed.
«requirement»
capture_constraints
Description
To capture the Constraints on the mapping of a User to a Role.
«refine»

Figure 1176: Constraint Service Policy

Constraint

This service assesses the externally provided Constraints that limit the allocation, deallocation and handover of User Roles.

Interface

Role_Allocation_Constraint

This interface is the externally provided Constraints limiting the allocation, deallocation and handover of User Roles and breach indications associated with this restriction.

This can include any potentially dynamic contributions to which User Roles are allowed to be allocated, which Users are allowed to be allocated, if/when deallocation is allowed, or if/when handover is allowed.

Attributes

allocation_constraint	The limit constraining Role allocation, e.g. is handover allowed.
applicable_context	The context in which the Constraint is applicable.
breach	A statement that there is an inability to meet User allocation.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of User Roles behaviour that is being constrained (e.g. whether it is more or less constraining).

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.68.7.1.7 User_Role_Capability

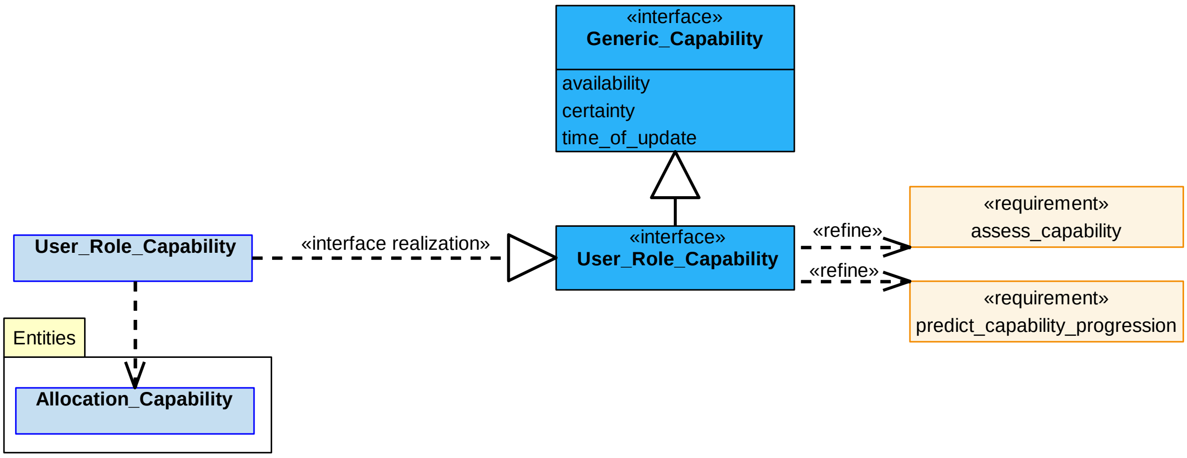
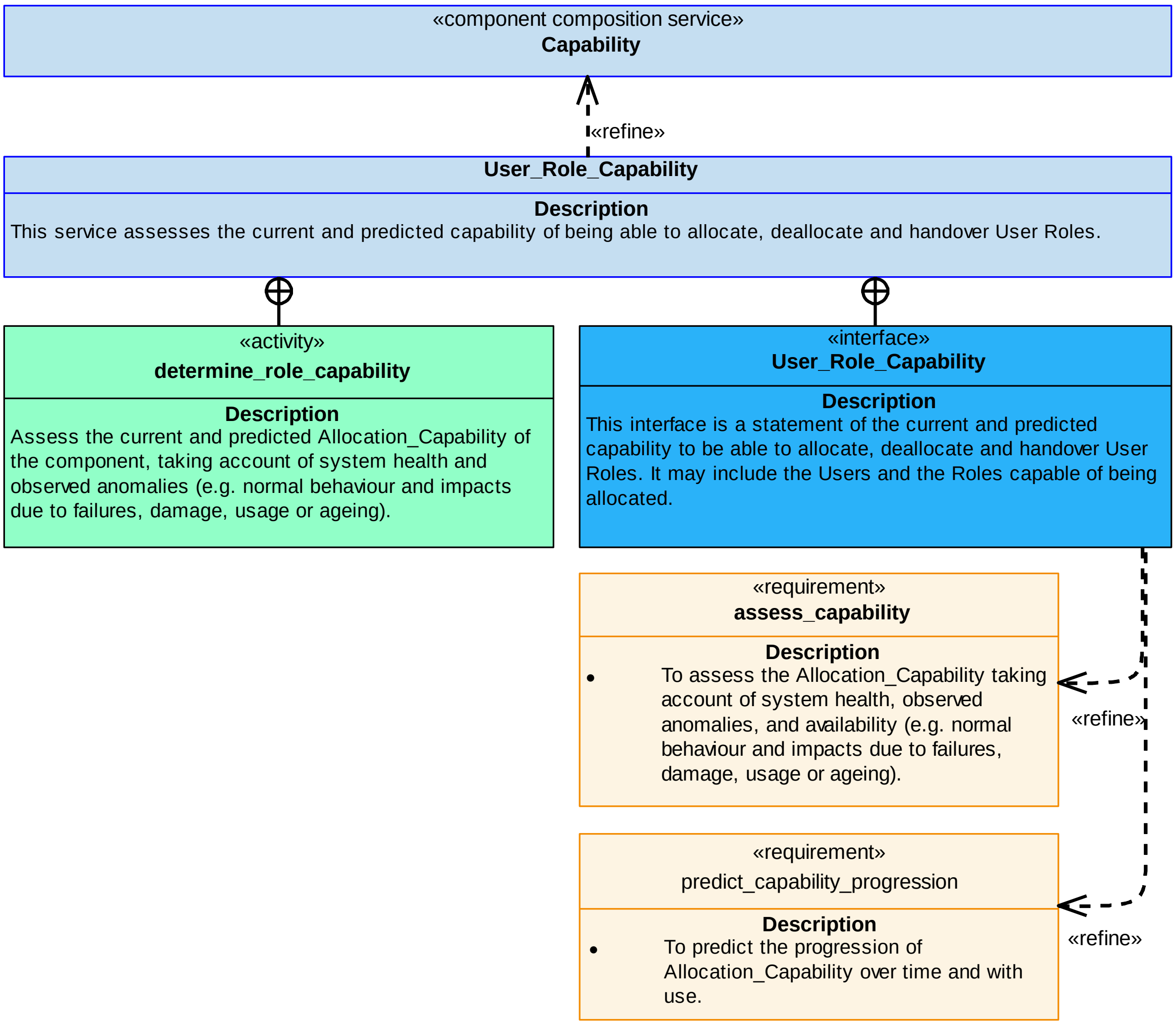


Figure 1177: User_Role_Capability Service Definition



«component composition service»
Capability

«refine»

User_Role_Capability

Description
This service assesses the current and predicted capability of being able to allocate, deallocate and handover User Roles.

«activity»
determine_role_capability

Description
Assess the current and predicted Allocation_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

«interface»
User_Role_Capability

Description
This interface is a statement of the current and predicted capability to be able to allocate, deallocate and handover User Roles. It may include the Users and the Roles capable of being allocated.

«requirement»
assess_capability

Description

- To assess the Allocation_Capability taking account of system health, observed anomalies, and availability (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

«refine»

«requirement»
predict_capability_progression

Description

- To predict the progression of Allocation_Capability over time and with use.

«refine»

Figure 1178: User_Role_Capability Service Policy

User_Role_Capability

This service assesses the current and predicted capability of being able to allocate, deallocate and handover User Roles.

Interface

User_Role_Capability

This interface is a statement of the current and predicted capability to be able to allocate, deallocate and handover User Roles. It may include the Users and the Roles capable of being allocated.

Activity

determine_role_capability

Assess the current and predicted Allocation_Capability of the component, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.68.7.1.8 User_Role_Capability_Evidence

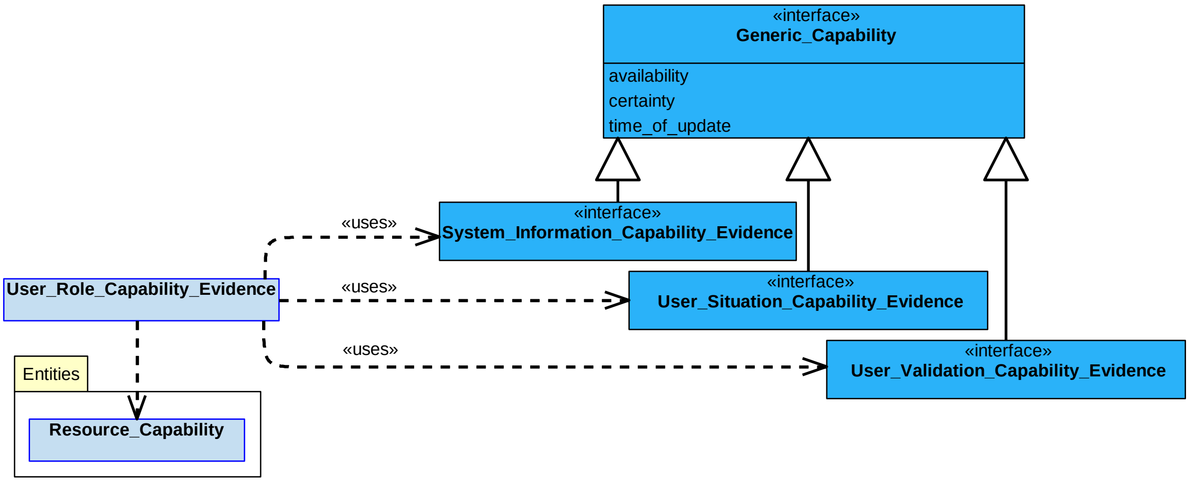
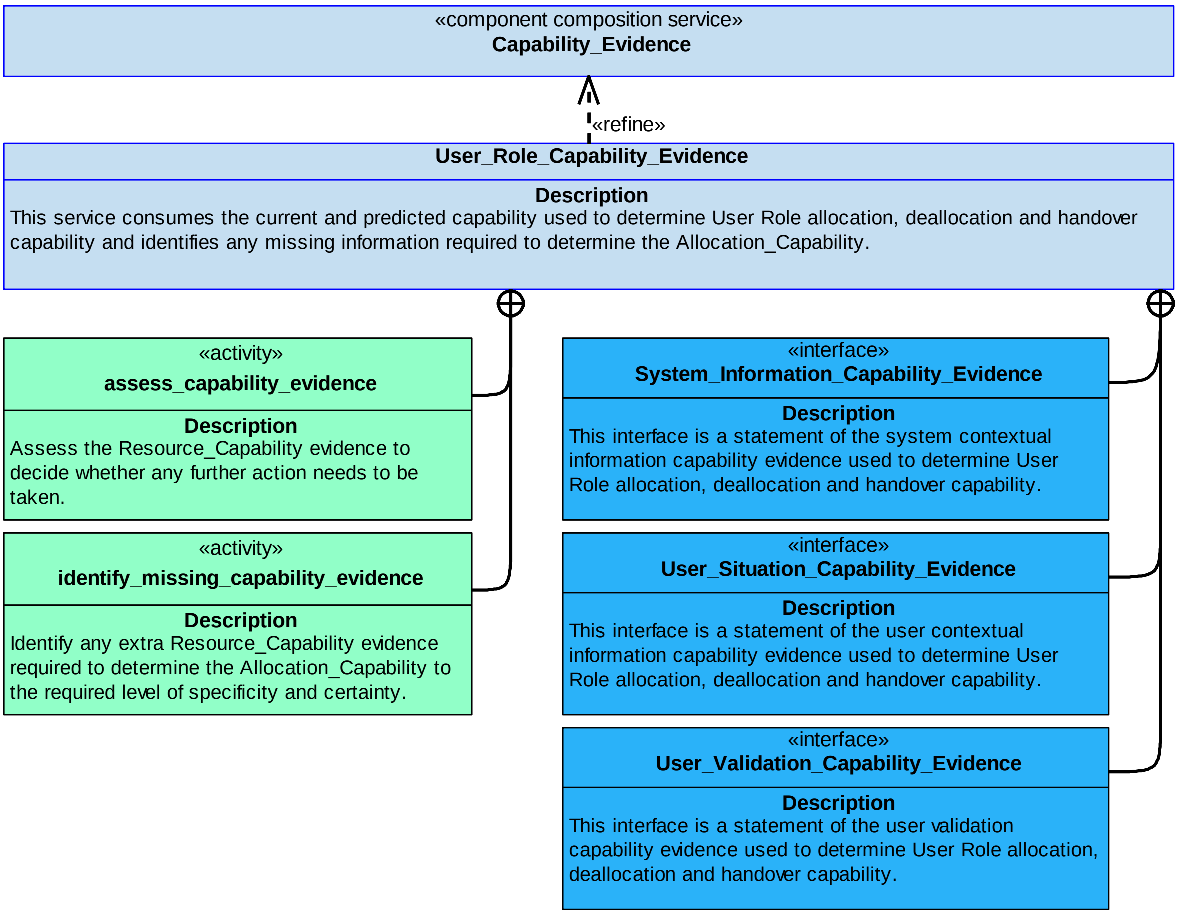


Figure 1179: User_Role_Capability_Evidence Service Definition


«component composition service»
Capability_Evidence
«refine»
User_Role_Capability_Evidence
Description
«activity»
assess_capability_evidence
Description
«activity»
identify_missing_capability_evidence
Description
«interface»
System_Information_Capability_Evidence
Description
«interface»
User_Situation_Capability_Evidence
Description
«interface»
User_Validation_Capability_Evidence
Description

This service consumes the current and predicted capability used to determine User Role allocation, deallocation and handover capability and identifies any missing information required to determine the Allocation_Capability.

Assess the Resource_Capability evidence to decide whether any further action needs to be taken.

Identify any extra Resource_Capability evidence required to determine the Allocation_Capability to the required level of specificity and certainty.

This interface is a statement of the system contextual information capability evidence used to determine User Role allocation, deallocation and handover capability.

This interface is a statement of the user contextual information capability evidence used to determine User Role allocation, deallocation and handover capability.

This interface is a statement of the user validation capability evidence used to determine User Role allocation, deallocation and handover capability.

Figure 1180: User_Role_Capability_Evidence Service Policy

User_Role_Capability_Evidence

This service consumes the current and predicted capability used to determine User Role allocation, deallocation and handover capability and identifies any missing information required to determine the Allocation_Capability.

Interfaces

System_Information_Capability_Evidence

This interface is a statement of the system contextual information capability evidence used to determine User Role allocation, deallocation and handover capability.

User_Situation_Capability_Evidence

This interface is a statement of the user contextual information capability evidence used to determine User Role allocation, deallocation and handover capability.

User_Validation_Capability_Evidence

This interface is a statement of the user validation capability evidence used to determine User Role allocation, deallocation and handover capability.

Activities

assess_capability_evidence

Assess the Resource_Capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra Resource_Capability evidence required to determine the Allocation_Capability to the required level of specificity and certainty.

5.4.2.68.7.2 Service Dependencies

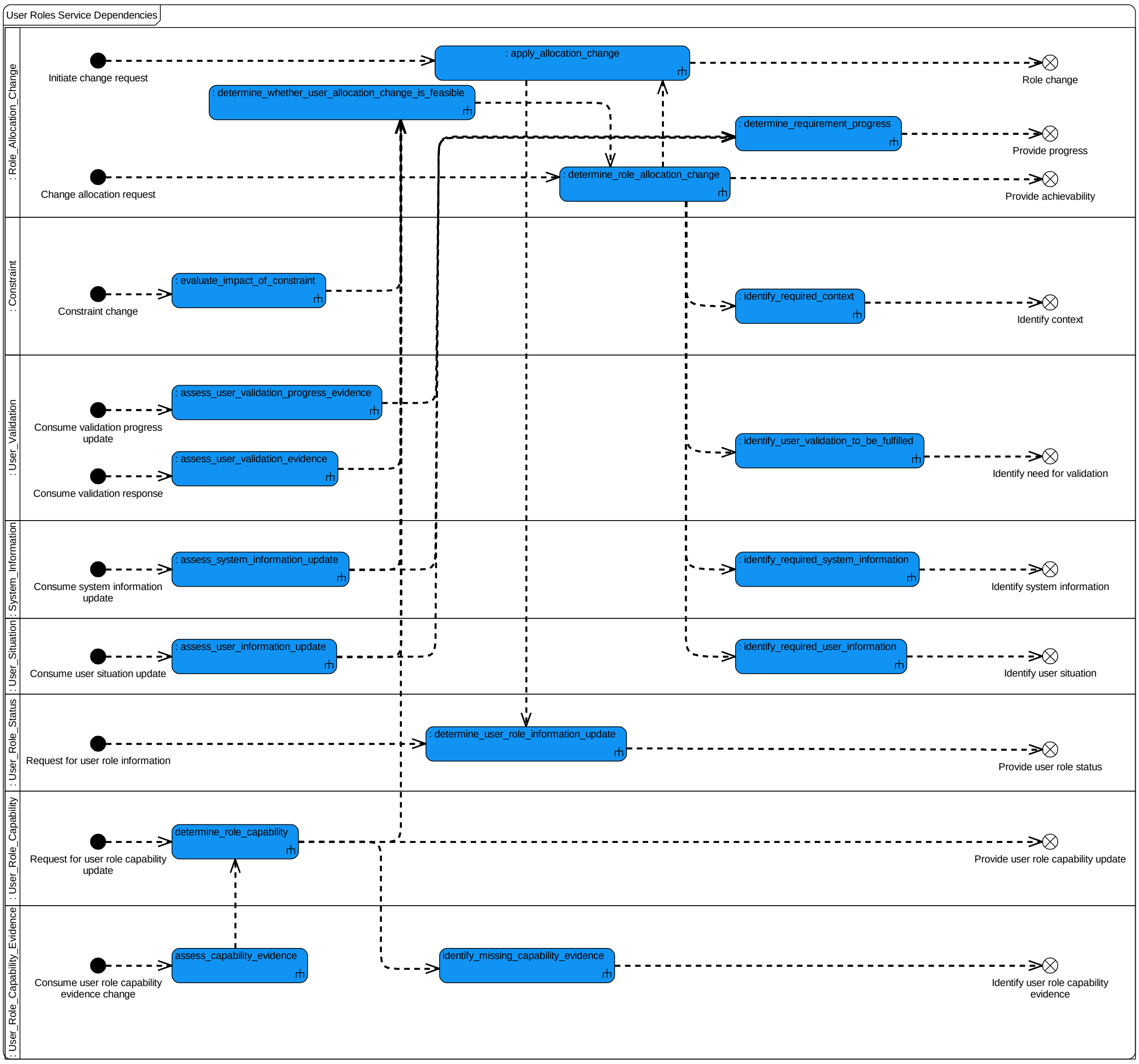

User Roles Service Dependencies
: Role_Allocation_Change
Initiate change request
: apply_allocation_change
Role change
: determine_whether_user_allocation_change_is_feasible
: determine_requirement_progress
Provide progress
Change allocation request
: determine_role_allocation_change
Provide achievability
: Constraint
: evaluate_impact_of_constraint
Constraint change
: identify_required_context
Identify context
: User_Validation
: assess_user_validation_progress_evidence
Consume validation progress update
: assess_user_validation_evidence
Consume validation response
: identify_user_validation_to_be_fulfilled
Identify need for validation
: System_Information
: assess_system_information_update
Consume system information update
: identify_required_system_information
Identify system information
: User_Situation
: assess_user_information_update
Consume user situation update
: identify_required_user_information
Identify user situation
: User_Role_Status
Request for user role information
: determine_user_role_information_update
Provide user role status
: User_Role_Capability
determine_role_capability
Request for user role capability update
Provide user role capability update
: User_Role_Capability_Evidence
assess_capability_evidence
Consume user role capability evidence change
identify_missing_capability_evidence
Identify user role capability evidence

Figure 1181: User Roles Service Dependencies

5.4.2.69 Vehicle External Environment

5.4.2.69.1 Role

The role of Vehicle External Environment is to determine information about the immediate environment surrounding a vehicle.

5.4.2.69.2 Overview

Control Architecture

Vehicle External Environment is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

On receiving a requirement to determine an Environmental_Property, e.g. Mach number, the Vehicle External Environment component will coordinate Measurement(s) of the External_Environment using Sources on the Vehicle, and use these Measurement(s) to calculate the value of the Environmental_Property, applying Correction_Factors (e.g. due to the current Vehicle Configuration or State) where required.

Examples of Use

- When properties of the immediate environment surrounding a Vehicle are required.

5.4.2.69.3 Service Summary

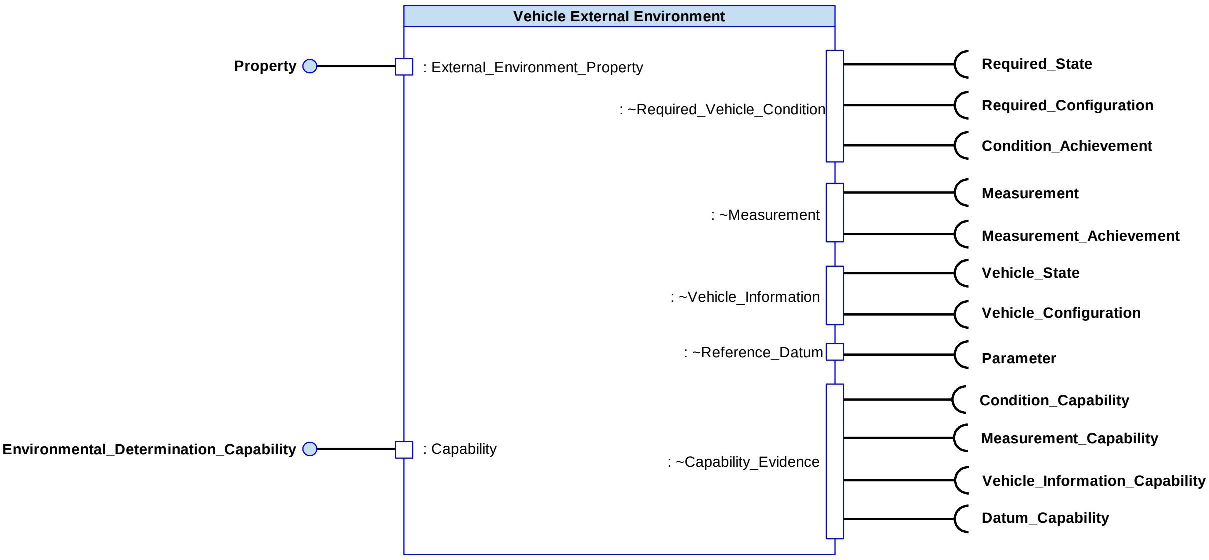


Figure 1182: Vehicle External Environment Service Summary

5.4.2.69.4 Responsibilities

determine_properties

- To determine an Environmental_Property of the immediate External_Environment surrounding a Vehicle (e.g. temperature, static pressure, indicated airspeed or pressure altitude).

determine_required_correction_factors

- To apply Correction_Factors to received sensor data.

identify_pre-conditions

- To identify pre-conditions necessary to determine an Environmental_Property of the immediate External_Environment surrounding a Vehicle, e.g. to identify a required vehicle configuration.

capture_reference_datum

- To capture appropriate Reference_Datum/data from which environmental relationships should be calculated.

assess_capability

- To identify the Capability to carry out Environmental_Property determination, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the component's Capability over time and with use.

5.4.2.69.5 Subject Matter Semantics

The subject matter of Vehicle External Environment is the external environment of the Vehicle.

Exclusions

The subject matter of Vehicle External Environment does not include:

- The control of the sensors which are used to generate Measurements.

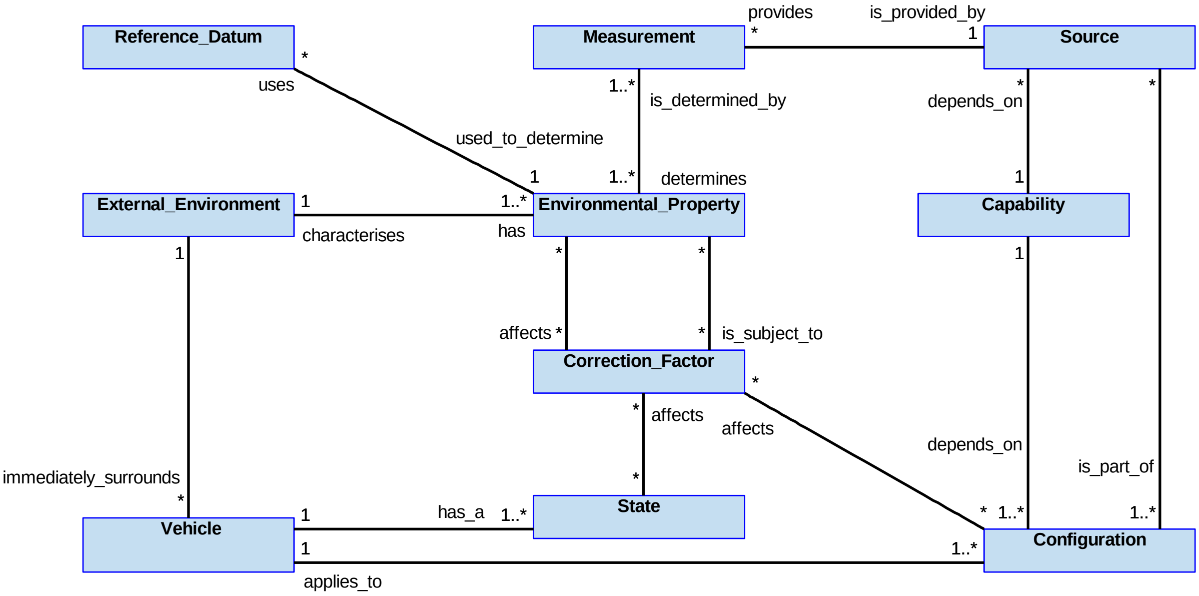


Figure 1183: Vehicle External Environment Semantics

5.4.2.69.5.1 Entities

Capability

The capability to determine environmental properties of a Vehicle's External_Environment.

Configuration

A configuration of the Vehicle that affects the disturbance of the External_Environment in the immediate proximity to the Vehicle (e.g. doors or apertures open and undercarriage deployed).

Correction_Factor

A correction applied when calculating an Environmental_Property to account for the effect of State or Configuration (e.g. for example due to a bomb bay door being open or due to aeroelastic deformation), or to account for a known disturbance.

Environmental_Property

A property of the External_Environment (e.g. pressure altitude, airspeed or Mach number).

External_Environment

The environment immediately surrounding a Vehicle.

Measurement

A sensor Measurement that supports the determination of an Environmental_Property, including the quality (accuracy, precision and validity) of the measurement.

Reference_Datum

The reference datum used to calculate an Environmental_Property (e.g. an altimeter pressure setting such as standard pressure).

Source

A sensor on a Vehicle which provide Measurements.

State

The state of a Vehicle, e.g. orientation, velocity and attitude that may affect perceived airflow.

Vehicle

An object for which the External_Environment is to be determined.

5.4.2.69.6 Design Rationale

5.4.2.69.6.1 Assumptions

- Vehicle External Environment will only process environment measurement information from sources it is configured to recognise.
- Introduction of additional Reference_Datums during operation will not be required.
- It is assumed that the data used to determine Correction_Factors is not changed during operation.
- Vehicle External Environment is only concerned with the value of the Reference_Datum (e.g. 990hPa), not whether it is QNH, QFE, etc.

5.4.2.69.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Vehicle External Environment**:**

- Data Driving - The Reference_Datums maintained by the Vehicle External Environment component for use throughout the system are designed to be data-driven using build time data. This allows the component to be configurable and flexible to cater for any set of required information. Similarly, Correction_Factors to source environmental measurements should be data-driven using build time data or dynamically calculated using data-driven rules.
- Recording and Logging - The Measurements captured and the Environmental_Property calculated, to fulfil immediate needs for filtering and combining, will be recorded in accordance with the Recording and Logging PYRAMID concept.

Extensions

- It is unlikely that extensions will be appropriate as the basic methods of determining the external environment are not likely to change. Although new sensor types may be developed these are outside the scope of this component.

Other Factors that were Taken into Account

- Whilst the subject matter of Vehicle External Environment and Location and Orientation are closely related, the information determined by each component and the Sources used to create them do not directly overlap.
- Safety concerns mean that Measurements supplied to Vehicle External Environment must be treated differently as they cover concepts such as the Vehicle's attitude to the local airflow, regardless of the platform's overall orientation, and hence impact flight control safety integrity. The Location and Orientation capability of some Exploiting Platforms may not use environment information in navigational reasoning (e.g. if determining location using navigational beacons).
- The separation of concerns between these two components aids in configurability and resilience against obsolescence requirements.

Exploitation Considerations

- Allowing the Sources supplying environment Measurements to be defined later in the development process (or be data-driven) allows the component to be reusable between multiple Exploiting Programmes.
- Vehicle External Environment may not be able to immediately determine information about the environment surrounding a Vehicle; a number of measurements may need to be taken, or a particular Configuration or State may need to be achieved prior to obtaining a measurement.

5.4.2.69.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component is expected to determine data such as airspeed, Mach number, angle of attack (alpha), sideslip (beta) and altitude. Failure of this component may cause uncontrolled flight of an air vehicle due to exceedance of the flight envelope / structural limits / loss of stability. This could lead to loss of structural integrity of the air vehicle and / or an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

5.4.2.69.6.4 Security Considerations

The indicative security classification is O.

This component determines reference data describing the properties of the environment around the vehicle, e.g. air data, which in isolation can be considered O. However, if data is allowed to accumulate within the component (e.g. for calculation or audit purposes) that allows performance data to be ascertained, the classification may need to be increased to SNEO, with associated changes in the way confidentiality is protected. The integrity and availability of air data, etc. is important for continued safe operation where the Exploiting Platform is an aircraft. Appropriate protections are required as the component is considered a legitimate target for cyber attack.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of environmental data during a mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is considered unlikely to directly implement security enforcing function.

5.4.2.69.7 Services

5.4.2.69.7.1 Service Definitions

5.4.2.69.7.1.1 External_Environment_Property

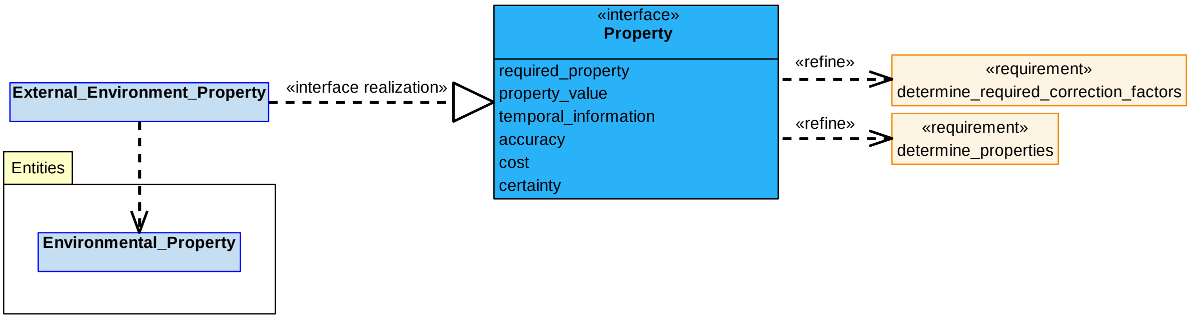


Figure 1184: External_Environment_Property Service Definition

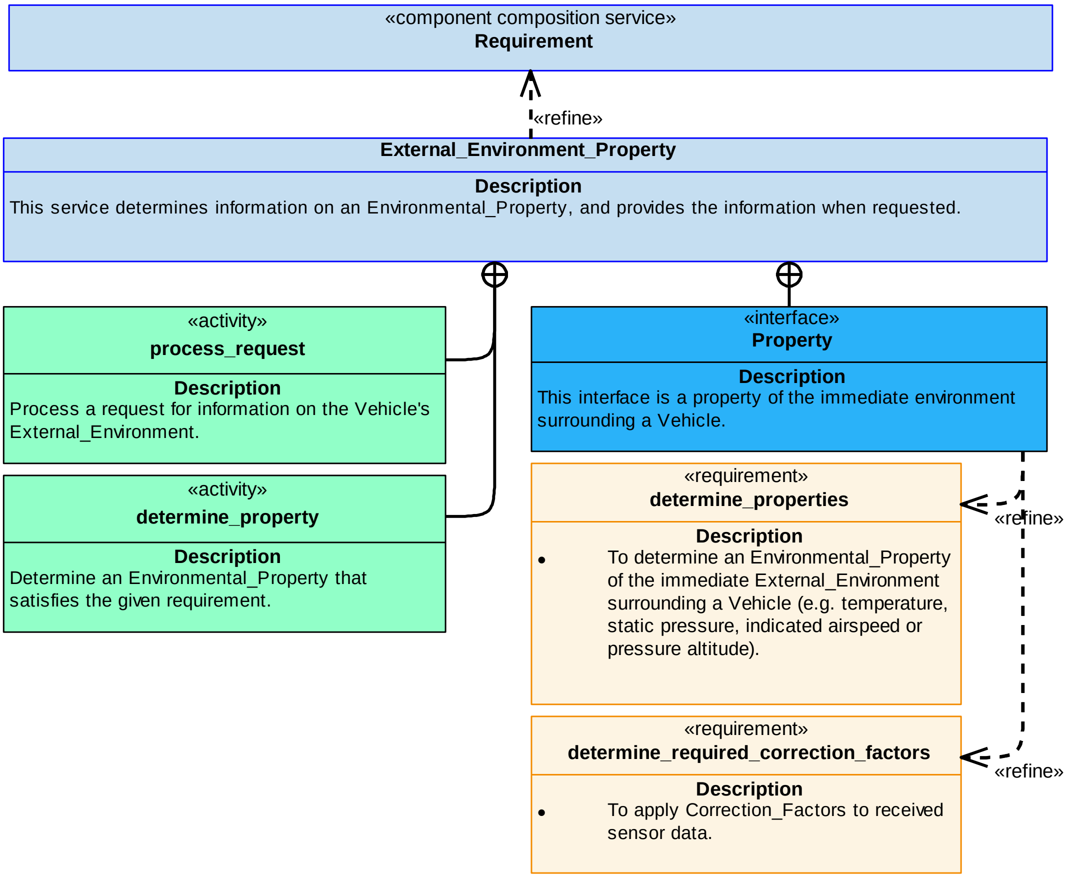

«component composition service»
Requirement
«refine»
External_Environment_Property
Description
This service determines information on an Environmental_Property, and provides the information when requested.
«activity»
process_request
Description
Process a request for information on the Vehicle's External_Environment.
«activity»
determine_property
Description
Determine an Environmental_Property that satisfies the given requirement.
«interface»
Property
Description
This interface is a property of the immediate environment surrounding a Vehicle.
«requirement»
determine_properties
Description
To determine an Environmental_Property of the immediate External_Environment surrounding a Vehicle (e.g. temperature, static pressure, indicated airspeed or pressure altitude).
«refine»
«requirement»
determine_required_correction_factors
Description
To apply Correction_Factors to received sensor data.
«refine»

Figure 1185: External_Environment_Property Service Policy

External_Environment_Property

This service determines information on an Environmental_Property, and provides the information when requested.

Interface

Property

This interface is a property of the immediate environment surrounding a Vehicle.

Attributes

required_property	The Environmental_Property for which information is required, e.g. Mach number.
property_value	The value of the Environmental_Property.
temporal_information	Information covering timing, e.g. when the request for information was made.
accuracy	The accuracy of the property value.
cost	The cost of determining the property value (e.g. resources used, time taken).
certainty	The confidence of the value being correct.

Activities

process_request

Process a request for information on the Vehicle's External_Environment.

determine_property

Determine an Environmental_Property that satisfies the given requirement.

5.4.2.69.7.1.2 Required_Vehicle_Condition

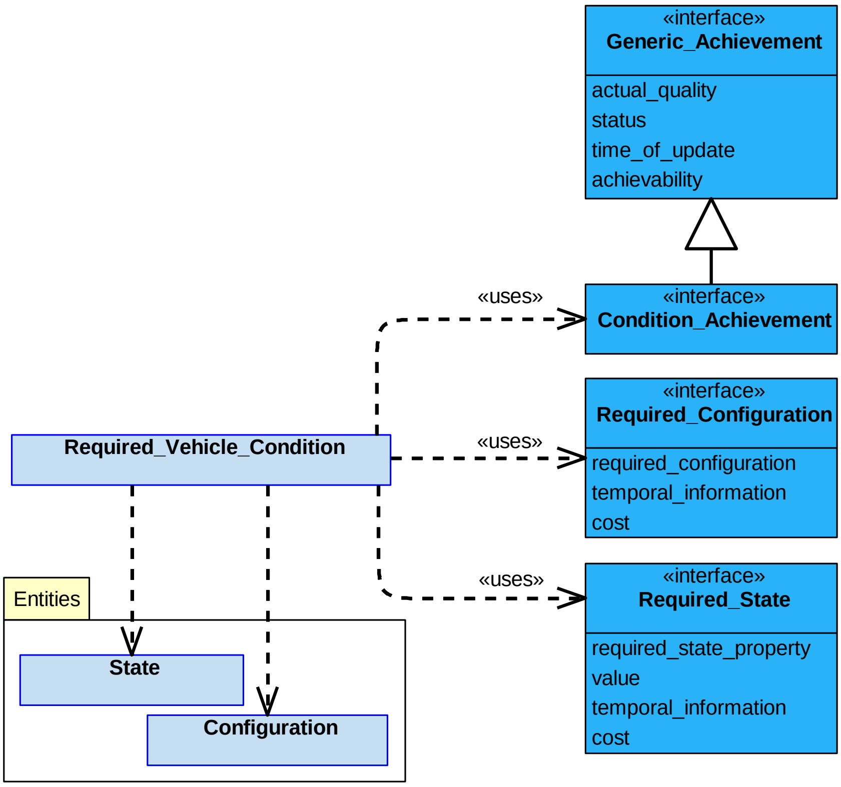


Figure 1186: Required_Vehicle_Condition Service Definition

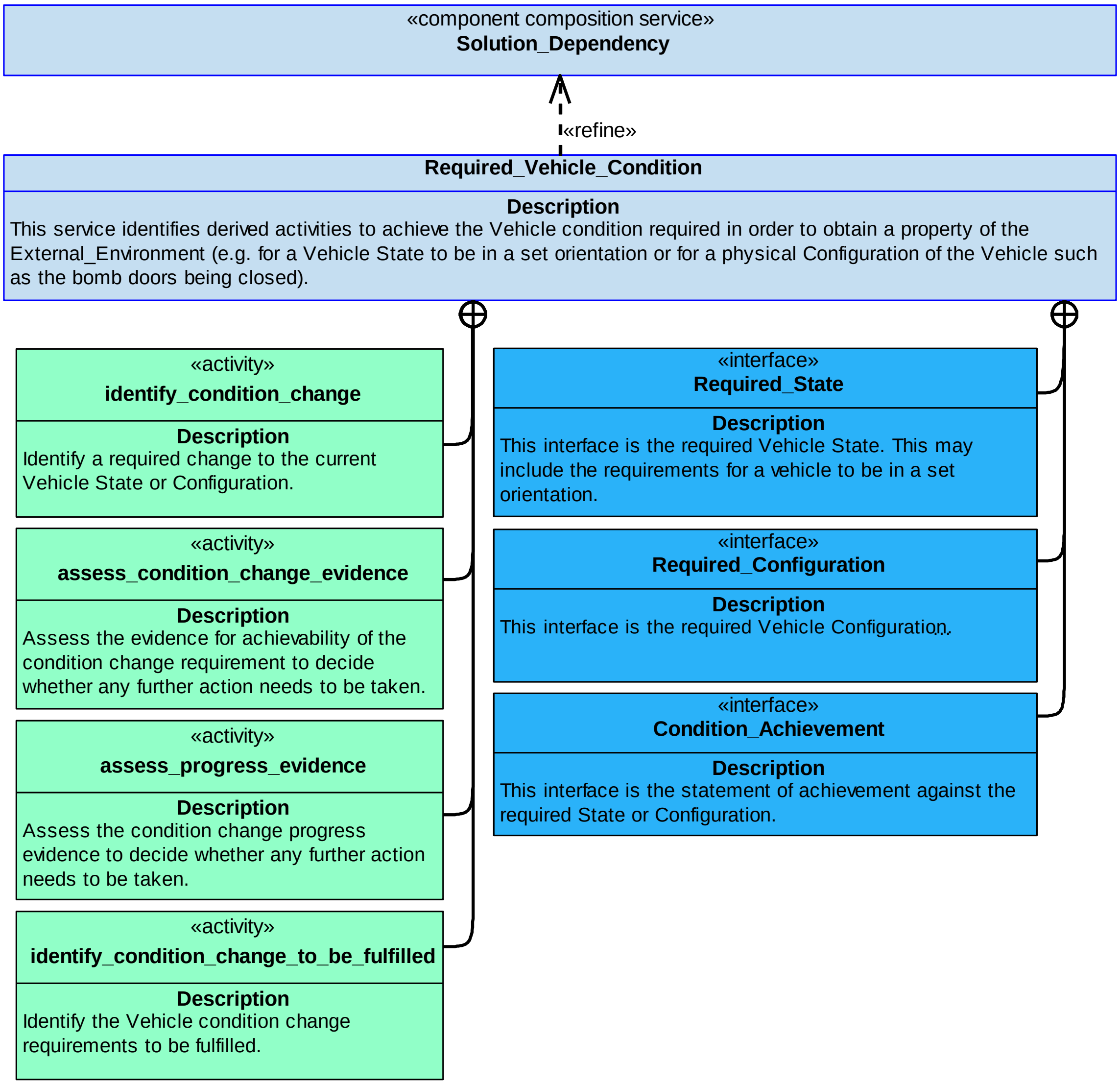

«component composition service»
Solution_Dependency
«refine»
Required_Vehicle_Condition
Description
This service identifies derived activities to achieve the Vehicle condition required in order to obtain a property of the External_Environment (e.g. for a Vehicle State to be in a set orientation or for a physical Configuration of the Vehicle such as the bomb doors being closed).
«activity»
identify_condition_change
Description
Identify a required change to the current Vehicle State or Configuration.
«activity»
assess_condition_change_evidence
Description
Assess the evidence for achievability of the condition change requirement to decide whether any further action needs to be taken.
«activity»
assess_progress_evidence
Description
Assess the condition change progress evidence to decide whether any further action needs to be taken.
«activity»
identify_condition_change_to_be_fulfilled
Description
Identify the Vehicle condition change requirements to be fulfilled.
«interface»
Required_State
Description
This interface is the required Vehicle State. This may include the requirements for a vehicle to be in a set orientation.
«interface»
Required_Configuration
Description
This interface is the required Vehicle Configuration.
«interface»
Condition_Achievement
Description
This interface is the statement of achievement against the required State or Configuration.

Figure 1187: Required_Vehicle_Condition Service Policy

Required_Vehicle_Condition

This service identifies derived activities to achieve the Vehicle condition required in order to obtain a property of the External_Environment (e.g. for a Vehicle State to be in a set orientation or for a physical Configuration of the Vehicle such as the bomb doors being closed).

Interfaces

Required_State

This interface is the required Vehicle State. This may include the requirements for a vehicle to be in a set orientation.

Attributes

required_state_property	The required State property.
value	The required value of the State property.

temporal_information	Information covering timing, such as when the Vehicle State should be enacted, and for how long.
cost	The cost of fulfilling the derived requirement for a particular Vehicle State, for example: resources used or time taken.

Required_Configuration

This interface is the required Vehicle Configuration.

Attributes

required_configuration	The required Configuration.
temporal_information	Information covering timing, such as when the required Configuration should be enacted and for how long.
cost	The cost of fulfilling the derived requirement for a particular Vehicle Configuration (e.g. resources used or time taken).

Condition_Achievement

This interface is the statement of achievement against the required State or Configuration.

Activities

identify_condition_change

Identify a required change to the current Vehicle State or Configuration.

assess_condition_change_evidence

Assess the evidence for achievability of the condition change requirement to decide whether any further action needs to be taken.

assess_progress_evidence

Assess the condition change progress evidence to decide whether any further action needs to be taken.

identify_condition_change_to_be_fulfilled

Identify the Vehicle condition change requirements to be fulfilled.

5.4.2.69.7.1.3 Measurement

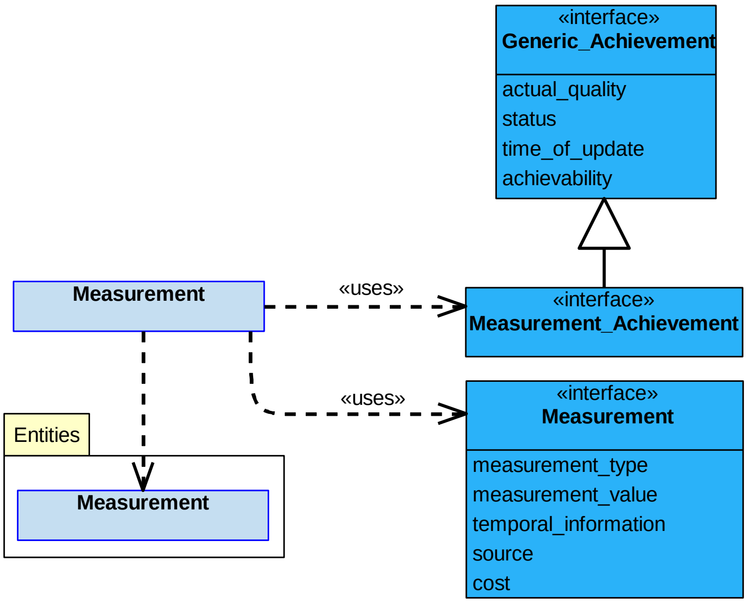


Figure 1188: Measurement Service Definition

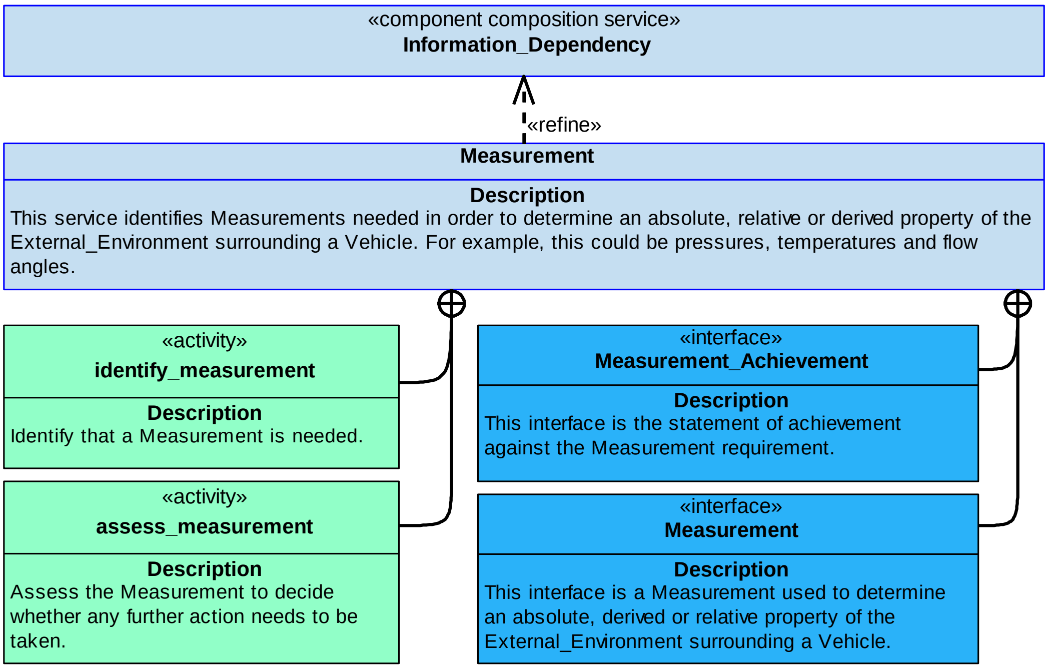

«component composition service»
Information_Dependency
«refine»
Measurement
Description
This service identifies Measurements needed in order to determine an absolute, relative or derived property of the External_Environment surrounding a Vehicle. For example, this could be pressures, temperatures and flow angles.
«activity»
identify_measurement
Description
Identify that a Measurement is needed.
«activity»
assess_measurement
Description
Assess the Measurement to decide whether any further action needs to be taken.
«interface»
Measurement_Achievement
Description
This interface is the statement of achievement against the Measurement requirement.
«interface»
Measurement
Description
This interface is a Measurement used to determine an absolute, derived or relative property of the External_Environment surrounding a Vehicle.

Figure 1189: Measurement Service Policy

Measurement

This service identifies Measurements needed in order to determine an absolute, relative or derived property of the External_Environment surrounding a Vehicle. For example, this could be pressures, temperatures and flow angles.

Interfaces

Measurement

This interface is a Measurement used to determine an absolute, derived or relative property of the External_Environment surrounding a Vehicle.

Attributes

measurement_type	The type of Measurement (e.g. temperature or flow angle).
measurement_value	The value of the Measurement.
temporal_information	Information covering timing, such as when the Measurement was made.
source	The source of the Measurement.
cost	The cost of obtaining the Measurement (e.g. resources used or time taken).

Measurement_Achievement

This interface is the statement of achievement against the Measurement requirement.

Activities

identify_measurement

Identify that a Measurement is needed.

assess_measurement

Assess the Measurement to decide whether any further action needs to be taken.

5.4.2.69.7.1.4 Vehicle_Information

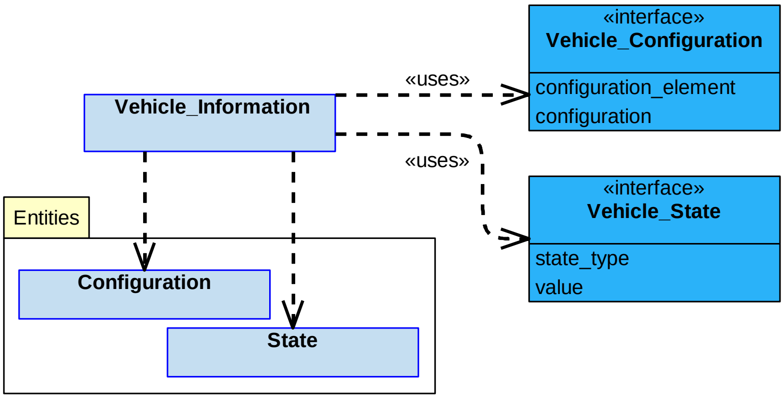


Figure 1190: Vehicle_Information Service Definition

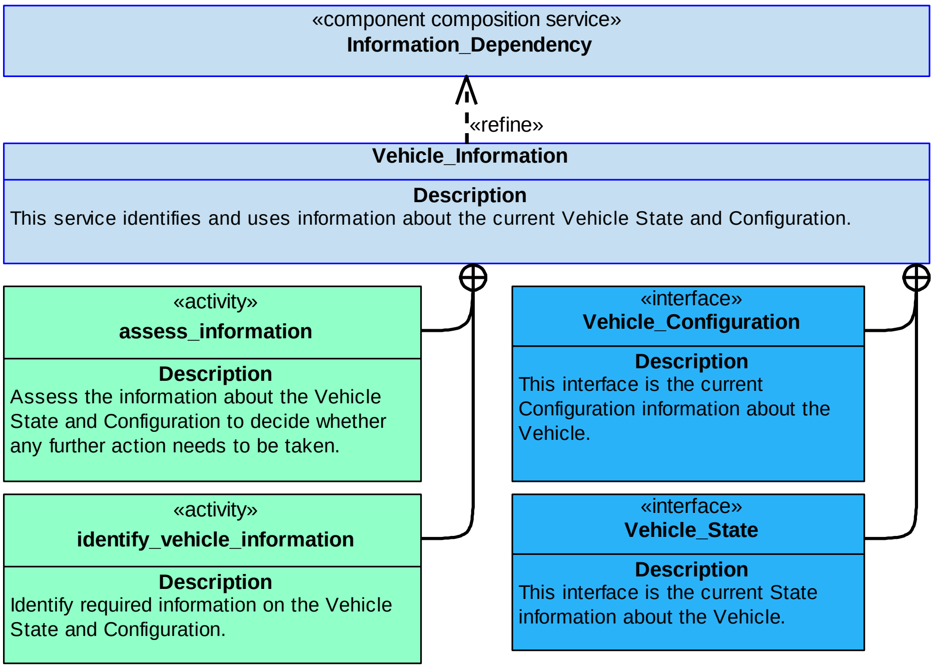

«component composition service»
Information_Dependency
«refine»
Vehicle_Information
Description
This service identifies and uses information about the current Vehicle State and Configuration.
«activity»
assess_information
Description
Assess the information about the Vehicle State and Configuration to decide whether any further action needs to be taken.
«activity»
identify_vehicle_information
Description
Identify required information on the Vehicle State and Configuration.
«interface»
Vehicle_Configuration
Description
This interface is the current Configuration information about the Vehicle.
«interface»
Vehicle_State
Description
This interface is the current State information about the Vehicle.

Figure 1191: Vehicle_Information Service Policy

Vehicle_Information

This service identifies and uses information about the current Vehicle State and Configuration.

Interfaces

Vehicle_State

This interface is the current State information about the Vehicle.

Attributes

state_type	The type of information relating to the Vehicle State, e.g. attitude.
value	The value of the state type.

Vehicle_Configuration

This interface is the current Configuration information about the Vehicle.

Attributes

configuration_element	The element of the Vehicle Configuration to which the information applies (e.g. tail flap position).
configuration	The status of the element of the Vehicle Configuration (e.g. the current angle of a moveable control surface).

Activities

assess_information

Assess the information about the Vehicle State and Configuration to decide whether any further action needs to be taken.

identify_vehicle_information

Identify required information on the Vehicle State and Configuration.

5.4.2.69.7.1.5 Reference_Datum

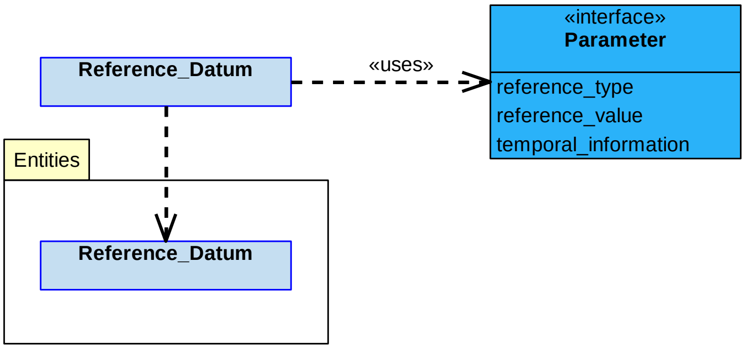


Figure 1192: Reference_Datum Service Definition

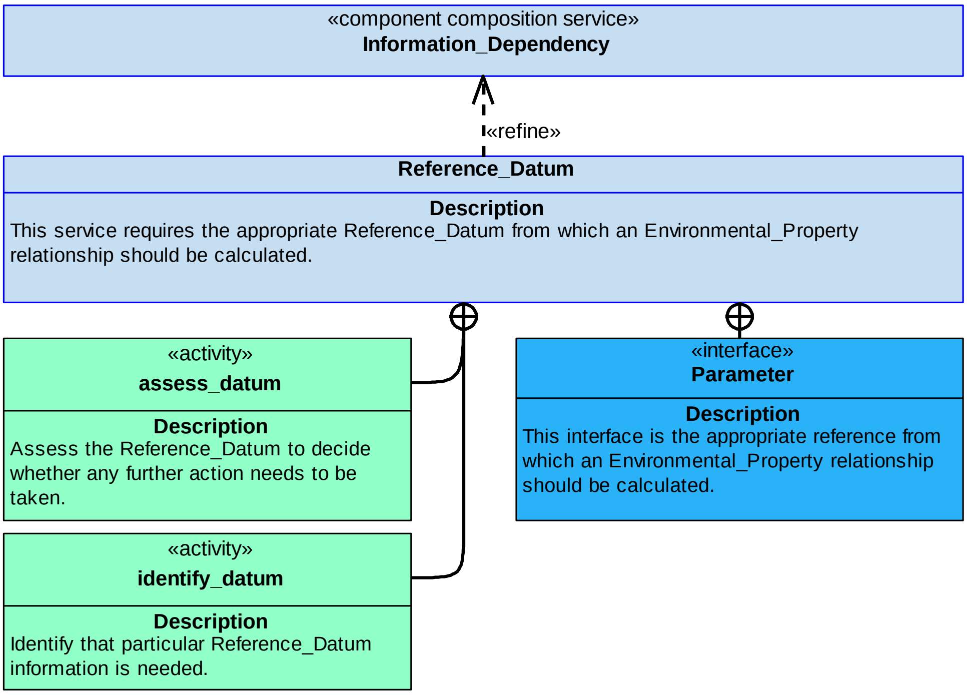

«component composition service»
Information_Dependency
«refine»
Reference_Datum
Description
This service requires the appropriate Reference_Datum from which an Environmental_Property relationship should be calculated.
«activity»
assess_datum
Description
Assess the Reference_Datum to decide whether any further action needs to be taken.
«activity»
identify_datum
Description
Identify that particular Reference_Datum information is needed.
«interface»
Parameter
Description
This interface is the appropriate reference from which an Environmental_Property relationship should be calculated.

Figure 1193: Reference_Datum Service Policy

Reference_Datum

This service requires the appropriate Reference_Datum from which an Environmental_Property relationship should be calculated.

Interface

Parameter

This interface is the appropriate reference from which an Environmental_Property relationship should be calculated.

Attributes

reference_type	The type of Reference_Datum, e.g. pressure.
reference_value	The value of the Reference_Datum parameter.
temporal_information	Information covering timing, such as when the information was obtained.

Activities

assess_datum

Assess the Reference_Datum to decide whether any further action needs to be taken.

identify_datum

Identify that particular Reference_Datum information is needed.

5.4.2.69.7.1.6 Capability

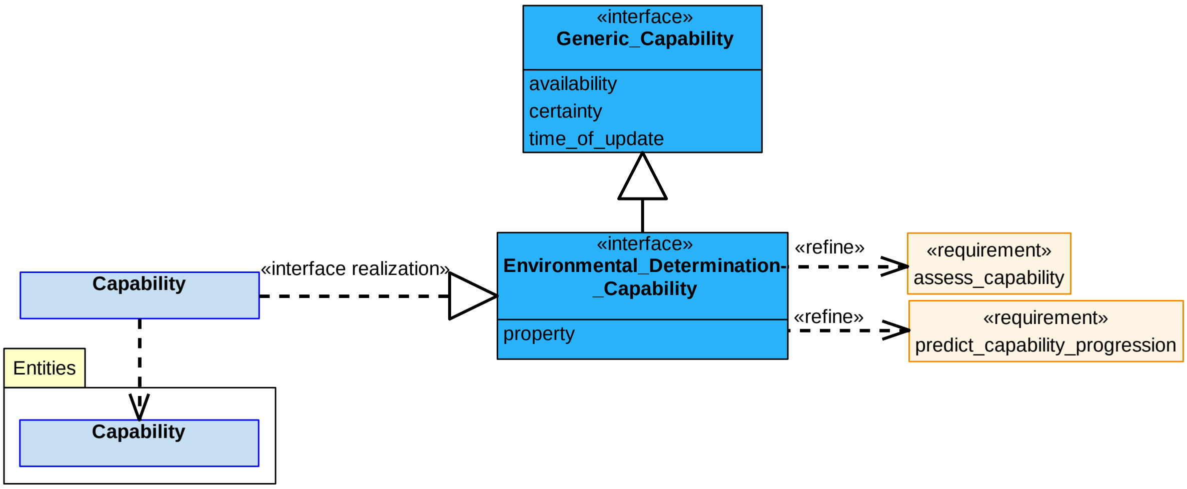


Figure 1194: Capability Service Definition

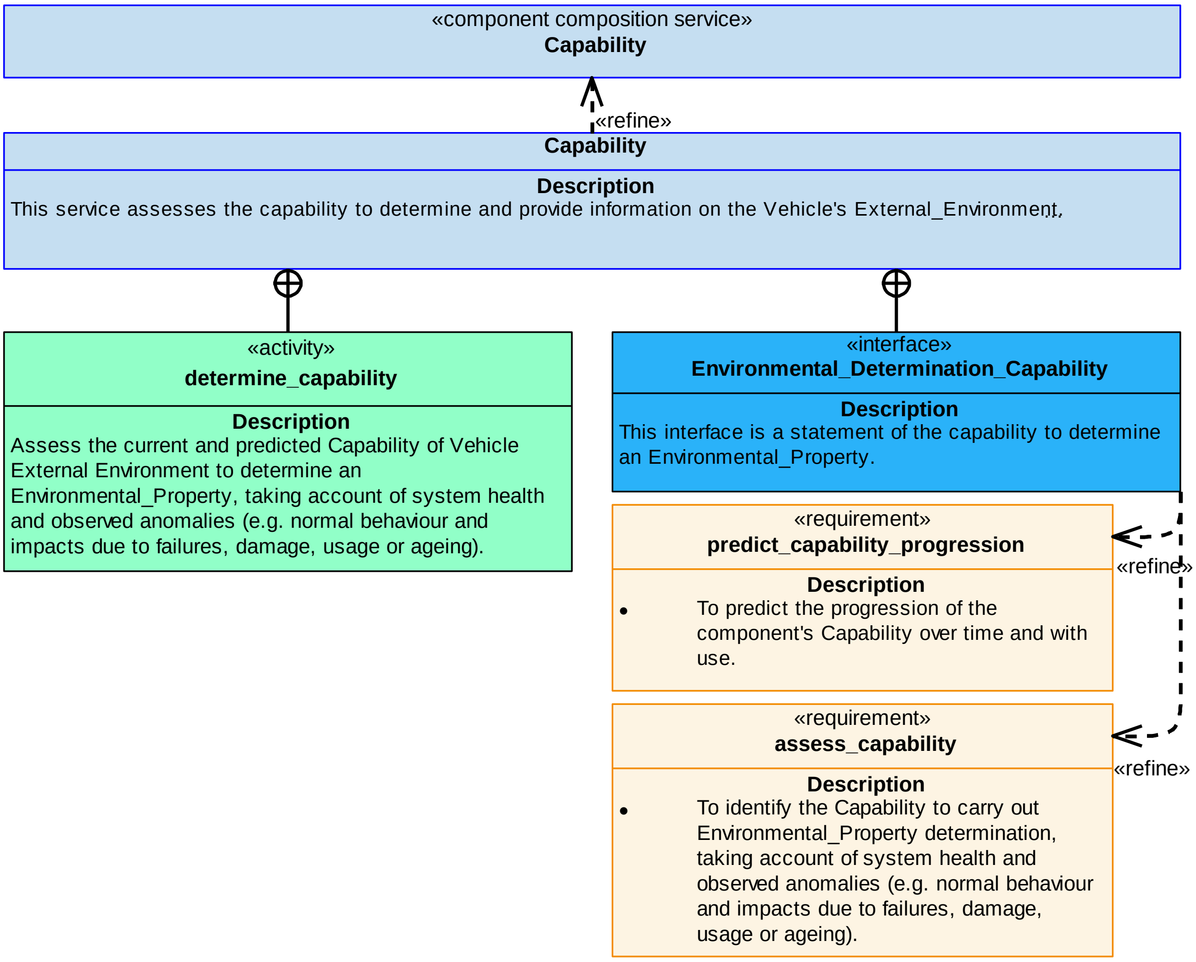

«component composition service»
Capability
«refine»
Capability
Description
This service assesses the capability to determine and provide information on the Vehicle's External_Environment.
«activity»
determine_capability
Description
Assess the current and predicted Capability of Vehicle External Environment to determine an Environmental_Property, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Environmental_Determination_Capability
Description
This interface is a statement of the capability to determine an Environmental_Property.
«requirement»
predict_capability_progression
Description
To predict the progression of the component's Capability over time and with use.
«refine»
«requirement»
assess_capability
Description
To identify the Capability to carry out Environmental_Property determination, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 1195: Capability Service Policy

Capability

This service assesses the capability to determine and provide information on the Vehicle's External_Environment.

Interface

Environmental_Determination_Capability

This interface is a statement of the capability to determine an Environmental_Property.

Attribute

property	The Environmental_Property that can be determined and provided, e.g. Mach number.

Activity

determine_capability

Assess the current and predicted Capability of Vehicle External Environment to determine an Environmental_Property, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.69.7.1.7 Capability_Evidence

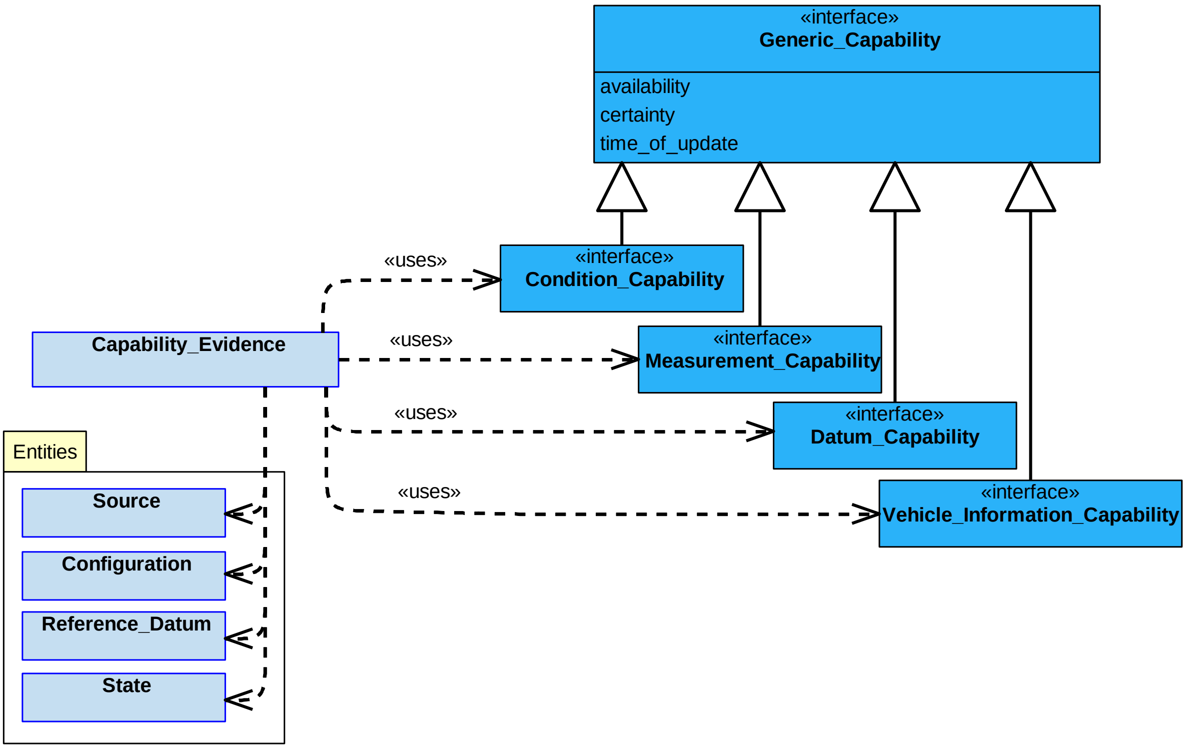


Figure 1196: Capability_Evidence Service Definition

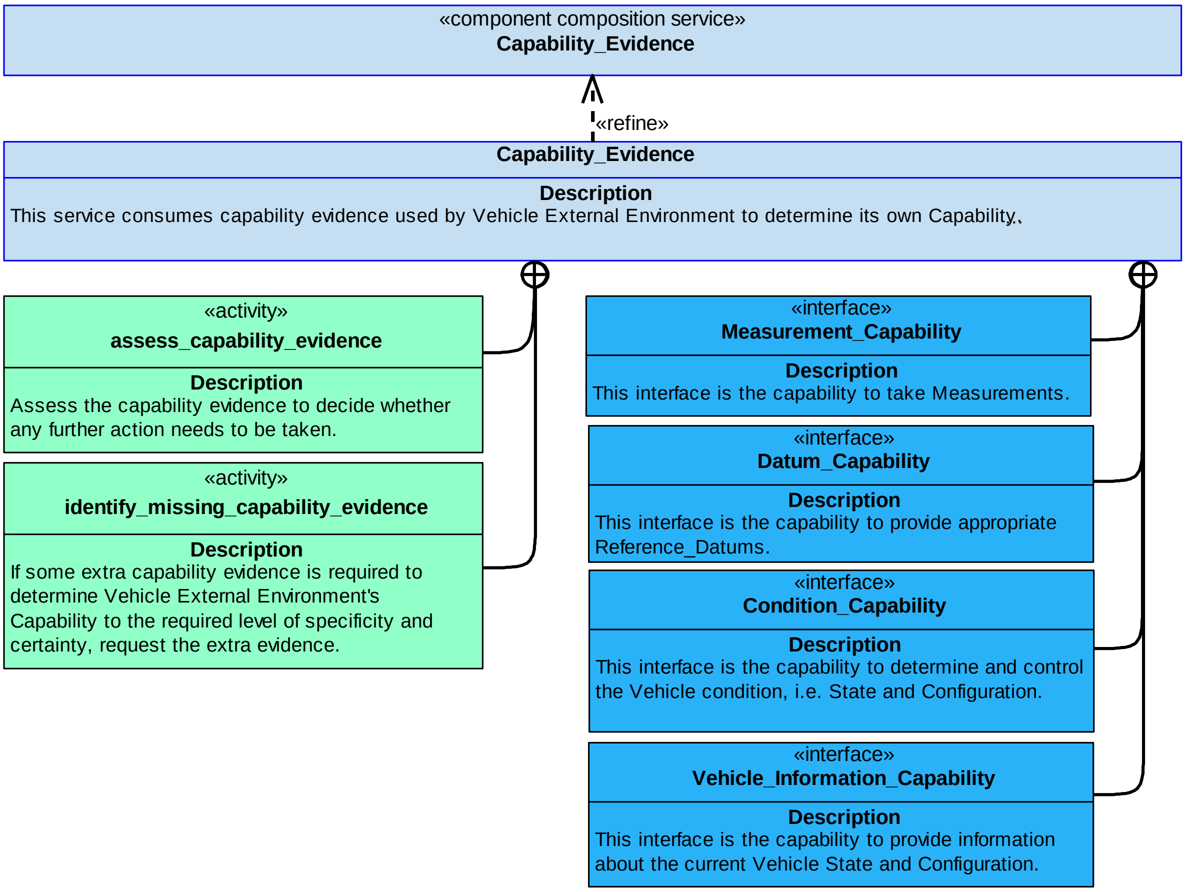

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capability evidence used by Vehicle External Environment to determine its own Capability..
«activity»
assess_capability_evidence
Description
Assess the capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
If some extra capability evidence is required to determine Vehicle External Environment's Capability to the required level of specificity and certainty, request the extra evidence.
«interface»
Measurement_Capability
Description
This interface is the capability to take Measurements.
«interface»
Datum_Capability
Description
This interface is the capability to provide appropriate Reference_Datums.
«interface»
Condition_Capability
Description
This interface is the capability to determine and control the Vehicle condition, i.e. State and Configuration.
«interface»
Vehicle_Information_Capability
Description
This interface is the capability to provide information about the current Vehicle State and Configuration.

Figure 1197: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability evidence used by Vehicle External Environment to determine its own Capability.

Interfaces

Measurement_Capability

This interface is the capability to take Measurements.

Condition_Capability

This interface is the capability to determine and control the Vehicle condition, i.e. State and Configuration.

Datum_Capability

This interface is the capability to provide appropriate Reference_Datums.

Vehicle_Information_Capability

This interface is the capability to provide information about the current Vehicle State and Configuration.

Activities

assess_capability_evidence

Assess the capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

If some extra capability evidence is required to determine Vehicle External Environment's Capability to the required level of specificity and certainty, request the extra evidence.

5.4.2.69.7.2 Service Dependencies

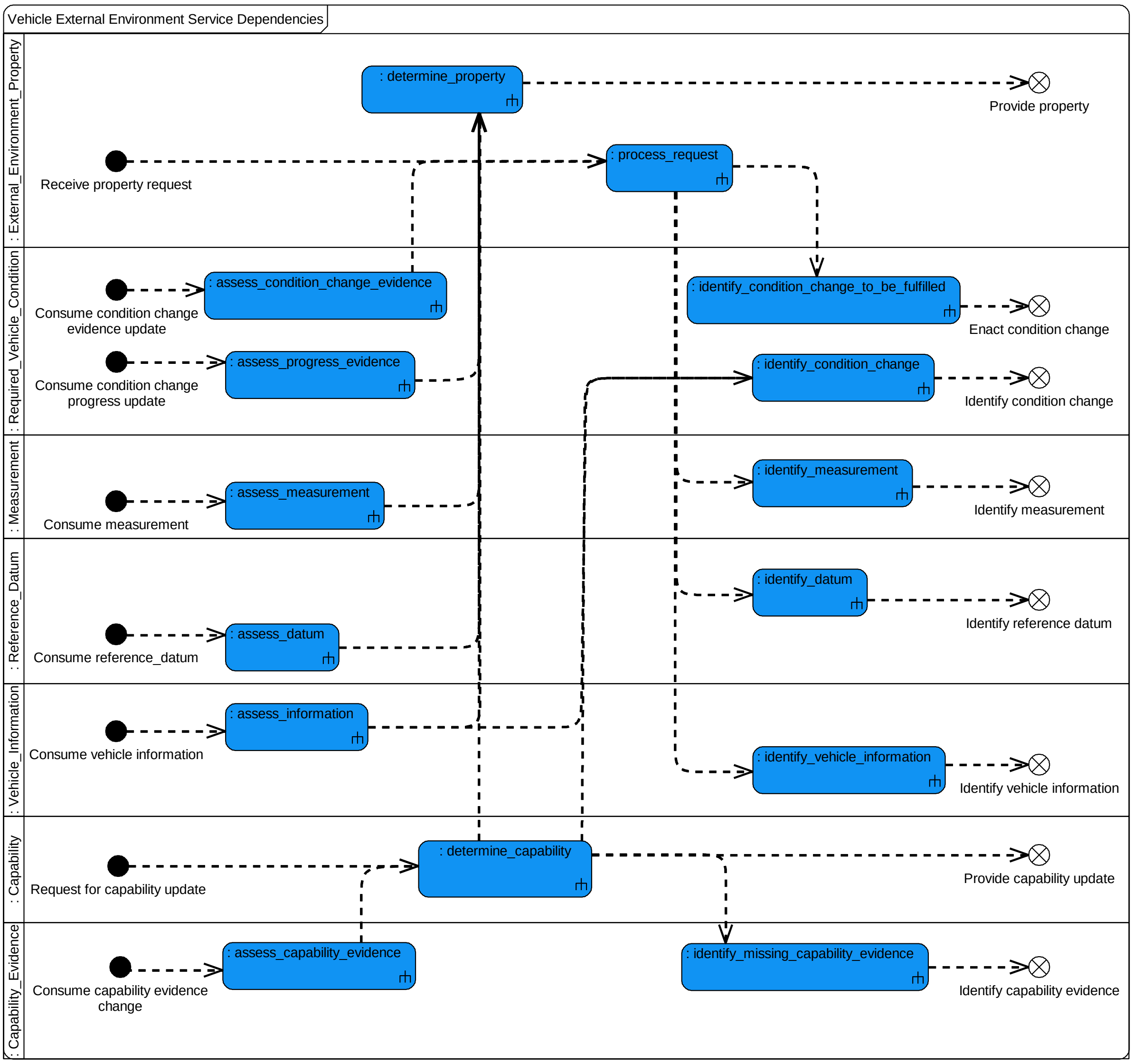

Vehicle External Environment Service Dependencies
: External_Environment_Property
: Required_Vehicle_Condition
: Measurement
: Reference_Datum
: Vehicle_Information
: Capability
: Capability_Evidence
: determine_property
Provide property
Receive property request
: process_request
: assess_condition_change_evidence
Consume condition change evidence update
: identify_condition_change_to_be_fulfilled
Enact condition change
: assess_progress_evidence
Consume condition change progress update
: identify_condition_change
Identify condition change
: assess_measurement
Consume measurement
: identify_measurement
Identify measurement
: assess_datum
Consume reference_datum
: identify_datum
Identify reference datum
: assess_information
Consume vehicle information
: identify_vehicle_information
Identify vehicle information
: determine_capability
Request for capability update
Provide capability update
: assess_capability_evidence
Consume capability evidence change
: identify_missing_capability_evidence
Identify capability evidence

Figure 1198: Vehicle External Environment Service Dependencies

5.4.2.70 Vehicle Guidance

5.4.2.70.1 Role

The role of Vehicle Guidance is to determine and execute a vehicle trajectory to fulfil provided trajectory requirements.

5.4.2.70.2 Overview

Control Architecture

Vehicle Guidance is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a request for movement (direction and/or rate of movement) of the vehicle is received (to an absolute point or relative to the own vehicle or another object) from a recognised control input, Vehicle Guidance determines the required trajectory to satisfy that movement request. The trajectory of a vehicle includes factors such as the orientation, rates of orientation and forces in any of the 6 degrees of freedom, which in some cases are of greater significance than achieving movement through specific locations in space. As the "outer loop" of the control system, Vehicle Guidance determines the Motion_Commands required to implement the trajectory. Vehicle Guidance also monitors for divergence from the required demands in order to correct the movement of the vehicle as necessary.

Examples of Use

Vehicle Guidance will be used as part of a system using electronic vehicle guidance interfaces, such as fly-by-wire flight control systems. Its use is not limited to following a path that is provided to it, since its input requirements can include requirements for an aircraft attitude or a rate of attitude change, such as where:

- A tactical sensing activity requires the vehicle to be in within specified orientation limits to enable an object of interest to be viewed as it is approached.
- A ‘loft’ attack needs to be performed, where the weapon must be released with the aircraft in the correct orientation (e.g. transition to 45 degrees in pitch).

This includes planning, on or off the vehicle, of how to meet the vehicle movement requirements, which involves pre-execution checks to ensure that a candidate Planned_Trajectory is suitable to be executed and to ensure that there is continuity between adjacent Planned_Trajectory.

5.4.2.70.3 Service Summary

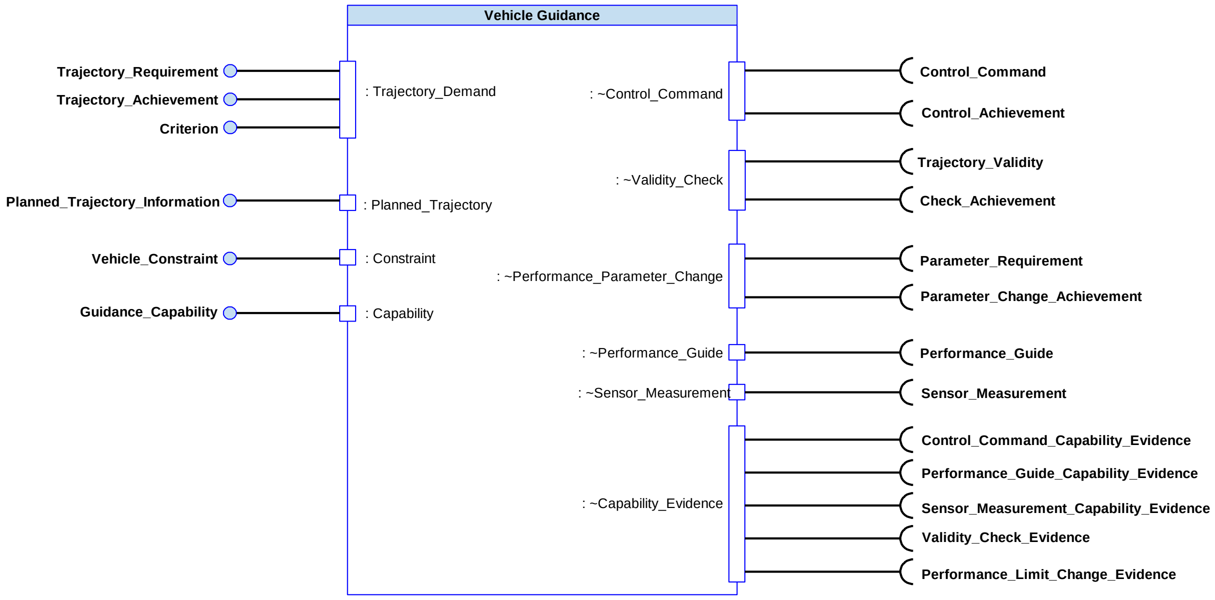


Figure 1199: Vehicle Guidance Service Summary

5.4.2.70.4 Responsibilities

capture_trajectory_requirements

- To capture given Trajectory_Requirements.

capture_measurement_criteria

- To capture Measurement_Criterion of the trajectory.

capture_vehicle_motion_constraints

- To capture given Vehicle Movement_Constraints (e.g. bank angle constraint).

ensure_solution_validity

- To ensure that all necessary Validity_Rules have been adhered to prior to enactment of a Planned_Trajectory (e.g. spatial de-confliction).

ensure_trajectory_continuity

- To ensure all adjoining Planned_Trajectory segments are continuous in terms of both location and attitude before being executed (i.e. there are no instantaneous changes).

ensure_solution_flow

- To ensure that whilst in motion there is always a Planned_Trajectory for execution. This includes planned trajectories for which there is no Trajectory_Requirement to either segue between planned trajectories, which satisfy Trajectory_Requirements, or to provide fall back guidance solutions in the absence of Trajectory_Requirements for the future.

determine_planned_vehicle_trajectory

- To determine a Planned_Trajectory to achieve Trajectory_Requirements within defined Movement_Constraints and Validity_Rules.

issue_vehicle_control_commands

- To execute the selected Planned_Trajectory by issuing Motion_Commands (e.g. attitude and speed or thrust level).

identify_whether_requirement_remains_achievable

- To identify whether a Trajectory_Requirement is still achievable given current or predicted Guidance_Capability, Performance_Guides and Movement_Constraints.

determine_trajectory_requirement_progress

- To determine the progress of a Vehicle against the Trajectory_Requirement.

determine_predicted_quality_of_deliverables

- To determine the predicted quality of the Planned_Trajectory against given Measurement_Criterion/criteria.

identify_vehicle_trajectory_divergence

- To identify Trajectory_Divergence of the Vehicle.

determine_capability

- To determine the Guidance_Capability of the Vehicle, taking into account observed anomalies.

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Guidance_Capability assessment.

predict_vehicle_guidance_capability

- To predict the progression of Guidance_Capability over time and with use.

5.4.2.70.5 Subject Matter Semantics

The subject matter of Vehicle Guidance is the control of Vehicle movement, through 6 degrees of freedom, including linear positions and attitudes, and the velocities and accelerations of these. This includes the planning and assurance of vehicle movement demands for safe Vehicle movement.

Exclusions

The subject matter of Vehicle Guidance does not include:

- Why vehicle movement needs to be performed in response to mission needs and the resulting establishment of mission movement requirements, such as the production of routes or the requirements for a particular type of manoeuvre (e.g. a collision avoidance or threat avoidance manoeuvre).

- How the vehicle movement is achieved, such as through the movement of control surfaces or propulsion.
- The prediction of vehicle trajectories.

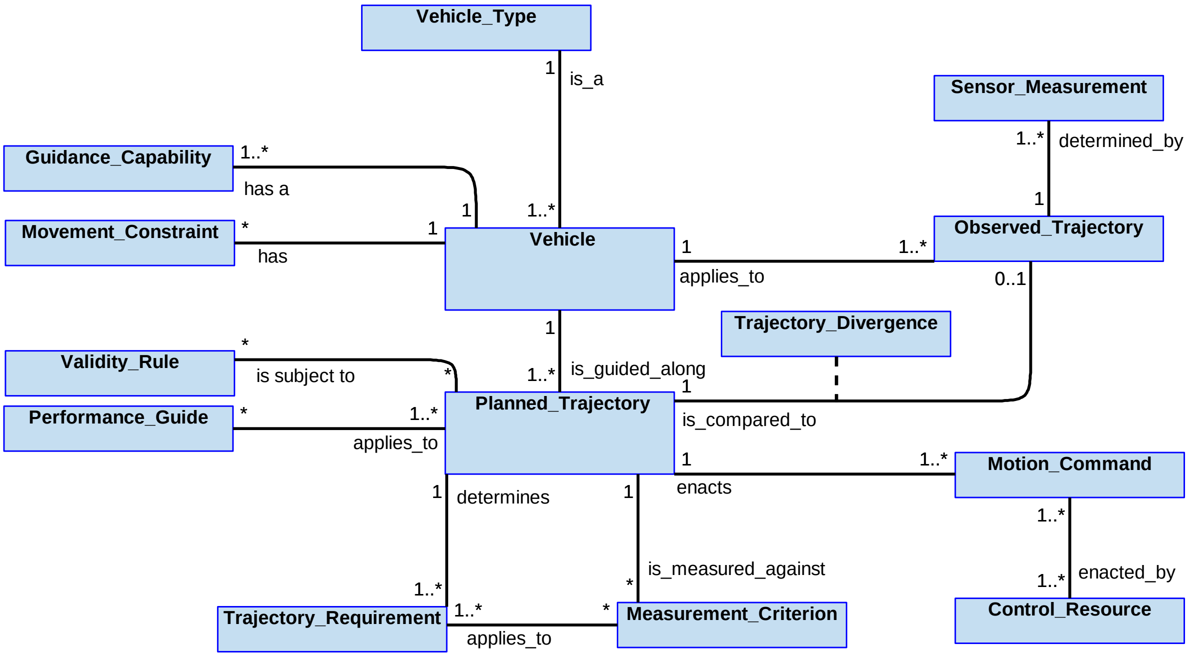


Figure 1200: Vehicle Guidance Semantics

5.4.2.70.5.1 Entities

Control_Resource

A resource or other part of the system which can be instructed to carry out Vehicle movement control.

Guidance_Capability

The capability of the Vehicle to be manoeuvred to complete the required movement. This will take into account any reported failures, etc.

Performance_Guide

A parameter that guides how the Vehicle may be manoeuvred with respect to a required performance, e.g. an ideal speed that would maximise the range of the vehicle whilst flying at the current altitude.

Motion_Command

The vehicle control commands on the motion of the vehicle as a whole (e.g. attitude, acceleration or speed) to be followed in order to enact the required manoeuvre.

Planned_Trajectory

A definition of the required 6-axis motion (location and attitude) of a vehicle over a specified time.

Trajectory_Divergence

Any divergence from the commanded Planned_Trajectory.

Trajectory_Requirement

An input demand for the Vehicle to be moved. This includes demands placed on all of the vehicles 6 degrees of freedom (e.g. orientation, acceleration, roll rate, etc.) as well as tolerances.

Vehicle

An instance of a moveable object whose movement can be controlled.

Observed_Trajectory

The actual trajectory of the Vehicle, including all 6 degrees of freedom (e.g. orientation, acceleration, roll rate, etc.).

Measurement_Criterion

Something by which the quality or cost of the trajectory will be measured.

Movement_Constraint

Any physical or operational limitations on how the Vehicle may be manoeuvred (e.g. manoeuvres limited due to the max g for the airframe or limiting speed through specified areas).

Sensor_Measurement

The determined values about a Vehicle Observed_Trajectory such as speed or altitude.

Vehicle_Type

A type of Vehicle.

Validity_Rule

The rules governing if checks need to be performed, outside of the component, against a Planned_Trajectory, and the rules for the specific types of check that need to be performed. For example, spatial de-confliction checks may be required if a Planned_Trajectory is not based on a complete path requirement provided by a high integrity source that has already checked for spatial de-confliction.

5.4.2.70.6 Design Rationale

5.4.2.70.6.1 Assumptions

- This component complies with tactical constraints relating to noise (e.g. by limiting speed through specified areas).
- This component will be used in the control of a vehicle (or a simulation of control of a vehicle) rather than in a predictive role for other objects in the environment.

5.4.2.70.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Vehicle Guidance:

- Data Driving - It is expected that variable parameters used to tune a Vehicle's performance could be data-driven. Data driving may also allow common algorithms for generic Vehicle types to be specialised for specific Vehicle or Vehicle variants.

Extensions

- Whilst not mandated, use of extensions may be appropriate in some instances where the component behaviour is not suitable for being data-driven. This could include, for example, for significantly different classes of vehicle (e.g. for fixed wing or rotary wing aircraft). However, it may be more likely that different variants of the component are required for each class of vehicle.

Exploitation Considerations

- Vehicle Guidance will likely be utilised with other flight control components, such as Vehicle Stability and Control, as part of an Exploiting Platform.
- Vehicle Guidance may receive Trajectory_Requirements from other components or through direct control from an authorised operator. The type of input commands could include:
    - a waypoint or series of waypoints
    - a specific manoeuvre
    - a position relative to another object
    - following an Instrument Landing System (ILS) localiser and glideslope
    - an altitude, flight level or relative height
    - a magnetic heading, true heading or track
    - a speed or Mach number
    - a climb / descent rate
    - direct control inputs from an authorised operator (e.g. climb/descent rate and turn rate).
- Where there are multiple sources of Trajectory_Requirements on Vehicle Guidance the Conflict Resolution component may be used to arbitrate between them. This includes cases where an authorised operator providing direct vehicle control inputs needs to be prioritised over automated Trajectory_Requirements from other components.
- When switching between satisfying mutually exclusive Trajectory_Requirements, from different sources, Vehicle Guidance may be expected to ensure that step changes to Motion_Commands do not occur.

5.4.2.70.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Failure of this component would cause uncontrollable manoeuvring of the Vehicle (e.g. uncontrolled flight in an air vehicle). In this case, whilst the air vehicle would be within the aerodynamic and stability limits of the air vehicle, the path of the air vehicle and it's velocity would not be controlled - i.e. not correctly fulfil the routing, direct authorised operator, or avoidance manoeuvre inputs. This could lead to an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

5.4.2.70.6.4 Security Considerations

The indicative security classification is SNEO.

This component represents the outer loop of vehicle control in order to meet the Trajectory_Requirements and as such has an understanding of the manoeuvre capabilities of the Exploiting Platform. This capability is considered to be SNEO and this component’s data would need protecting to ensure it is kept secret (confidentiality). The integrity and availability of this component are fundamental and it can be considered a probable target for cyber attack; tamper with it and you can remove the ability to guide the vehicle safely.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of vehicle guidance demands during a mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and will need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** of deviations from the expected Planned_Trajectory; an unexplainable change in the ability to control the vehicle or correct deviations in trajectory may indicate the component has been compromised by a cyber attack.

The component is considered unlikely to directly implement security enforcing functions, although it is dependent on the integrity of its inputs.

5.4.2.70.7 Services

5.4.2.70.7.1 Service Definitions

5.4.2.70.7.1.1 Trajectory_Demand

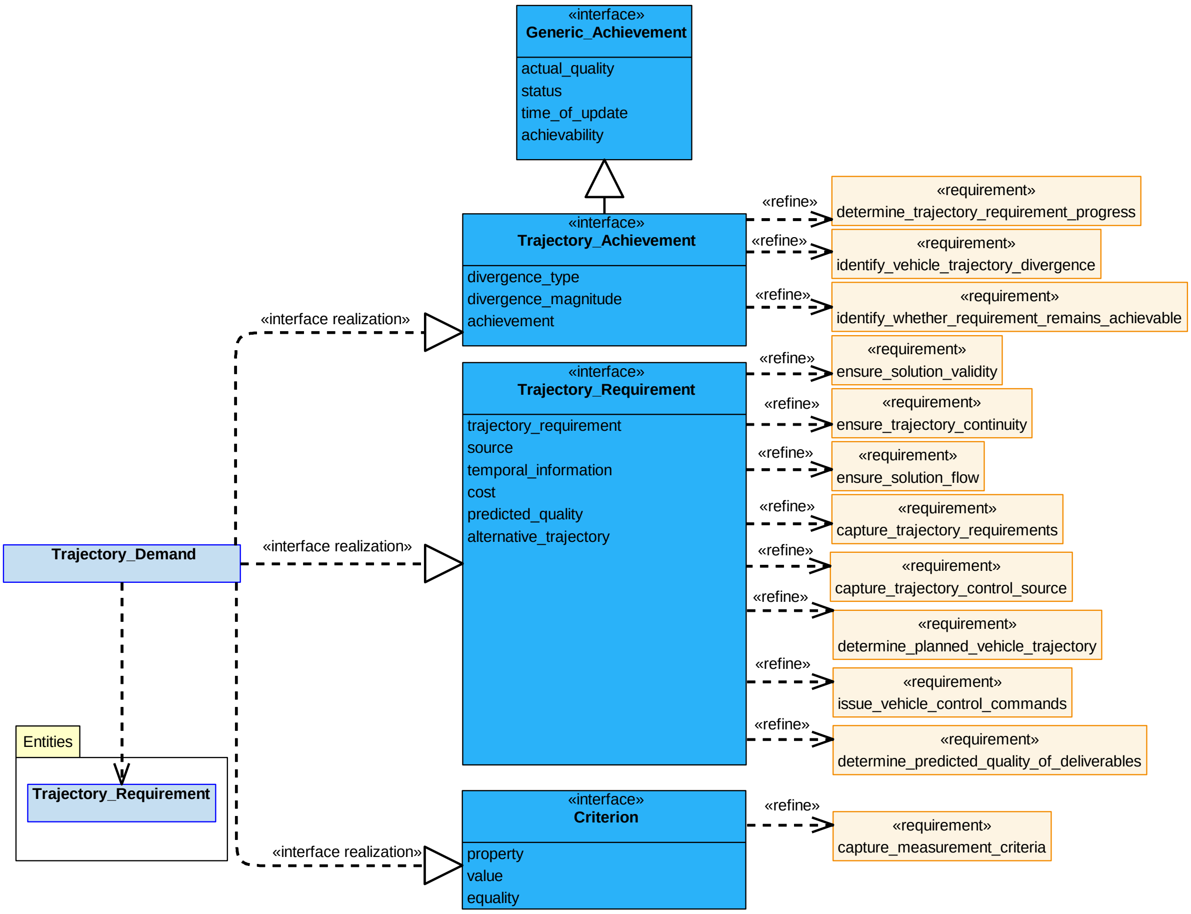


Figure 1201: Trajectory_Demand Service Definition

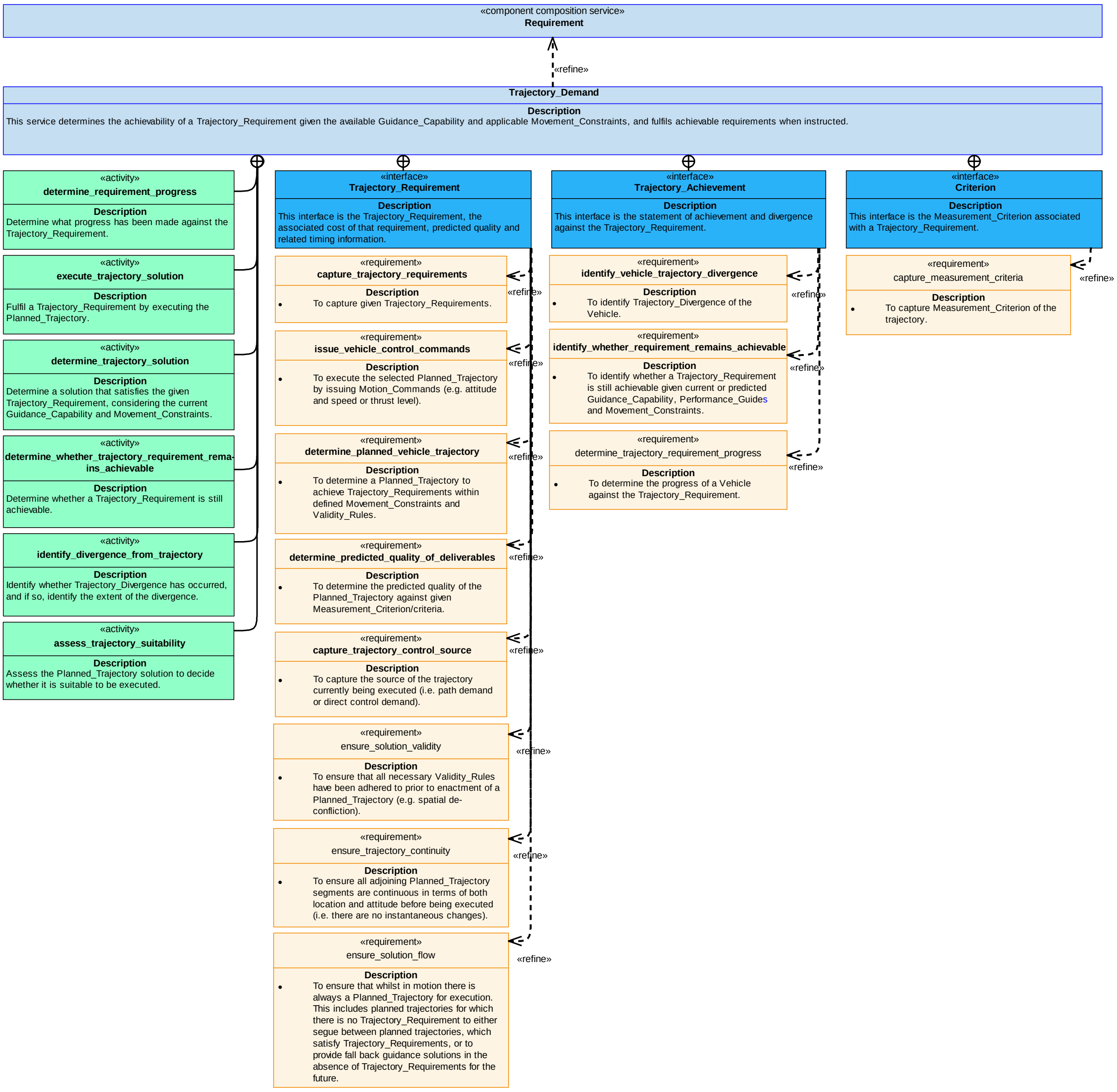

«component composition service»
Requirement
«refine»
Trajectory_Demand
Description
This service determines the achievability of a Trajectory_Requirement given the available Guidance_Capability and applicable Movement_Constraints, and fulfils achievable requirements when instructed.
«activity»
determine_requirement_progress
Description
Determine what progress has been made against the Trajectory_Requirement.
«activity»
execute_trajectory_solution
Description
Fulfil a Trajectory_Requirement by executing the Planned_Trajectory.
«activity»
determine_trajectory_solution
Description
Determine a solution that satisfies the given Trajectory_Requirement, considering the current Guidance_Capability and Movement_Constraints.
«activity»
determine_whether_trajectory_requirement_remains_achievable
Description
Determine whether a Trajectory_Requirement is still achievable.
«activity»
identify_divergence_from_trajectory
Description
Identify whether Trajectory_Divergence has occurred, and if so, identify the extent of the divergence.
«activity»
assess_trajectory_suitability
Description
Assess the Planned_Trajectory solution to decide whether it is suitable to be executed.
«interface»
Trajectory_Requirement
Description
This interface is the Trajectory_Requirement, the associated cost of that requirement, predicted quality and related timing information.
«requirement»
capture_trajectory_requirements
Description
To capture given Trajectory_Requirements.
«refine»
«requirement»
issue_vehicle_control_commands
Description
To execute the selected Planned_Trajectory by issuing Motion_Commands (e.g. attitude and speed or thrust level).
«refine»
«requirement»
determine_planned_vehicle_trajectory
Description
To determine a Planned_Trajectory to achieve Trajectory_Requirements within defined Movement_Constraints and Validity_Rules.
«refine»
«requirement»
determine_predicted_quality_of_deliverables
Description
To determine the predicted quality of the Planned_Trajectory against given Measurement_Criterion/criteria.
«refine»
«requirement»
capture_trajectory_control_source
Description
To capture the source of the trajectory currently being executed (i.e. path demand or direct control demand).
«refine»
«requirement»
ensure_solution_validity
Description
To ensure that all necessary Validity_Rules have been adhered to prior to enactment of a Planned_Trajectory (e.g. spatial de-confliction).
«refine»
«requirement»
ensure_trajectory_continuity
Description
To ensure all adjoining Planned_Trajectory segments are continuous in terms of both location and attitude before being executed (i.e. there are no instantaneous changes).
«refine»
«requirement»
ensure_solution_flow
Description
To ensure that whilst in motion there is always a Planned_Trajectory for execution. This includes planned trajectories for which there is no Trajectory_Requirement to either segue between planned trajectories, which satisfy Trajectory_Requirements, or to provide fall back guidance solutions in the absence of Trajectory_Requirements for the future.
«refine»
«interface»
Trajectory_Achievement
Description
This interface is the statement of achievement and divergence against the Trajectory_Requirement.
«requirement»
identify_vehicle_trajectory_divergence
Description
To identify Trajectory_Divergence of the Vehicle.
«refine»
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Trajectory_Requirement is still achievable given current or predicted Guidance_Capability, Performance_Guides and Movement_Constraints.
«refine»
«requirement»
determine_trajectory_requirement_progress
Description
To determine the progress of a Vehicle against the Trajectory_Requirement.
«refine»
«interface»
Criterion
Description
This interface is the Measurement_Criterion associated with a Trajectory_Requirement.
«requirement»
capture_measurement_criteria
Description
To capture Measurement_Criterion of the trajectory.
«refine»

Figure 1202: Trajectory_Demand Service Policy

Trajectory_Demand

This service determines the achievability of a Trajectory_Requirement given the available Guidance_Capability and applicable Movement_Constraints, and fulfils achievable requirements when instructed.

Interfaces

Trajectory_Requirement

This interface is the Trajectory_Requirement, the associated cost of that requirement, predicted quality and related timing information.

Attributes

trajectory_requirement	The definition of the Trajectory_Requirement.

source	The source of the Trajectory_Requirement (e.g. from the system or an operator).
temporal_information	Information covering timing, such as start and end times.
cost	The cost of executing the Planned_Trajectory (e.g. resources used or time taken).
predicted_quality	How well the Planned_Trajectory is predicted to satisfy the Trajectory_Requirement.
alternative_trajectory	A proposed alternative to the Trajectory_Requirement.

Trajectory_Achievement

This interface is the statement of achievement and divergence against the Trajectory_Requirement.

Attributes

divergence_type	The type of divergence from the required trajectory (e.g. direction and/or movement).
divergence_magnitude	The magnitude of the divergence from the required trajectory.
achievement	An indication of whether or not the Trajectory_Requirement can be achieved.

Criterion

This interface is the Measurement_Criterion associated with a Trajectory_Requirement.

Attributes

property	The property to be measured.
value	The measured value of the property.
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Activities

determine_requirement_progress

Determine what progress has been made against the Trajectory_Requirement.

determine_trajectory_solution

Determine a solution that satisfies the given Trajectory_Requirement, considering the current Guidance_Capability and Movement_Constraints.

execute_trajectory_solution

Fulfil a Trajectory_Requirement by executing the Planned_Trajectory.

determine_whether_trajectory_requirement_remains_achievable

Determine whether a Trajectory_Requirement is still achievable.

identify_divergence_from_trajectory

Identify whether Trajectory_Divergence has occurred, and if so, identify the extent of the divergence.

assess_trajectory_suitability

Assess the Planned_Trajectory solution to decide whether it is suitable to be executed.

5.4.2.70.7.1.2 Control_Command

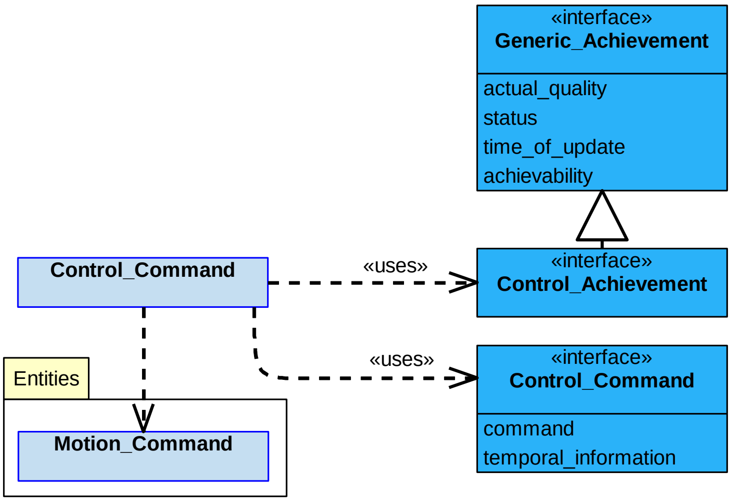


Figure 1203: Control_Command Service Definition

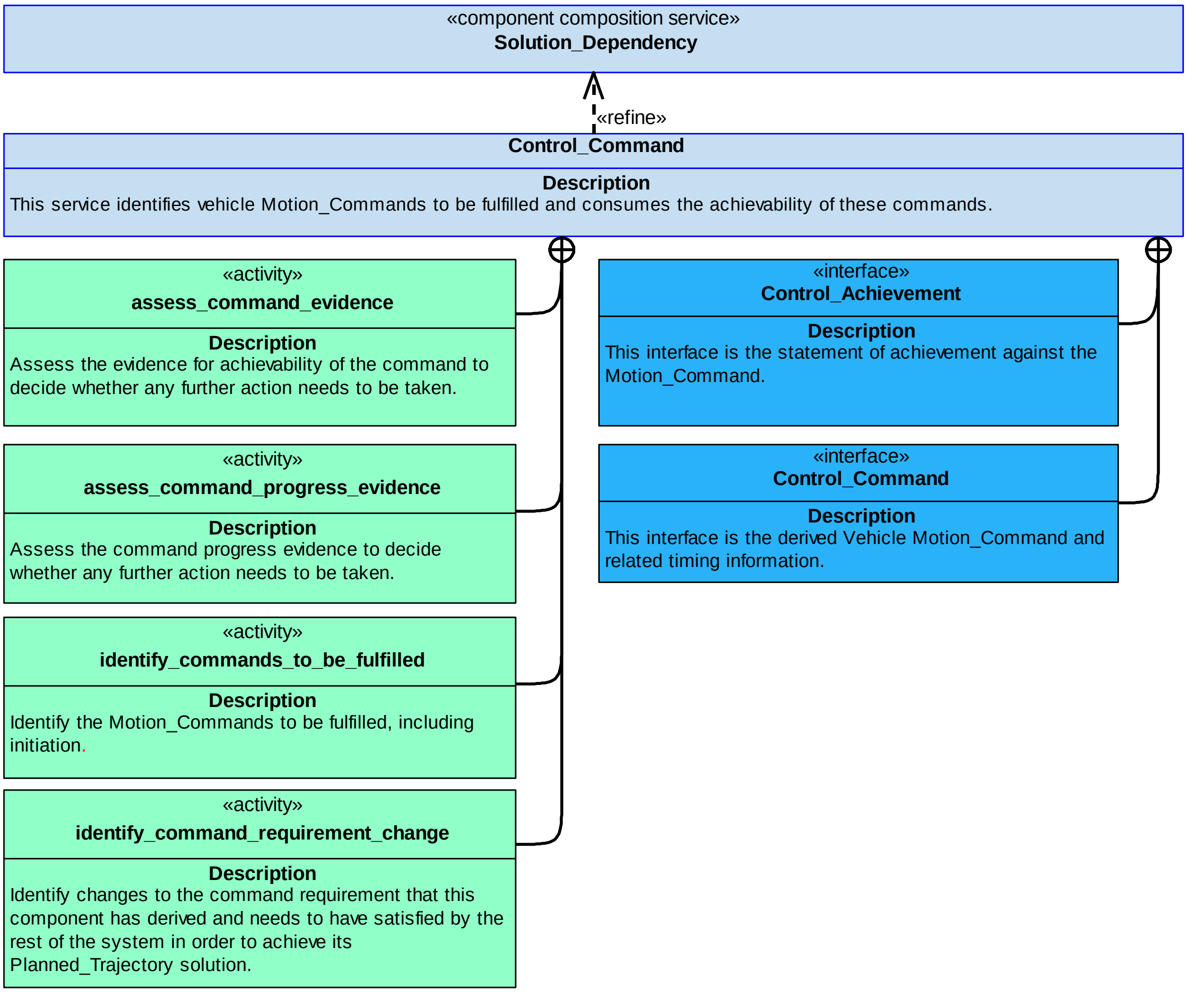

«component composition service»
Solution_Dependency
«refine»
Control_Command
Description
This service identifies vehicle Motion_Commands to be fulfilled and consumes the achievability of these commands.
«activity»
assess_command_evidence
Description
Assess the evidence for achievability of the command to decide whether any further action needs to be taken.
«activity»
assess_command_progress_evidence
Description
Assess the command progress evidence to decide whether any further action needs to be taken.
«activity»
identify_commands_to_be_fulfilled
Description
Identify the Motion_Commands to be fulfilled, including initiation.
«activity»
identify_command_requirement_change
Description
Identify changes to the command requirement that this component has derived and needs to have satisfied by the rest of the system in order to achieve its Planned_Trajectory solution.
«interface»
Control_Achievement
Description
This interface is the statement of achievement against the Motion_Command.
«interface»
Control_Command
Description
This interface is the derived Vehicle Motion_Command and related timing information.

Figure 1204: Control_Command Service Policy

Control_Command

This service identifies vehicle Motion_Commands to be fulfilled and consumes the achievability of these commands.

Interfaces

Control_Command

This interface is the derived Vehicle Motion_Command and related timing information.

Attributes

command	The derived Motion_Command.
temporal_information	Information covering timing, such as start and end times.

Control_Achievement

This interface is the statement of achievement against the Motion_Command.

Activities

assess_command_evidence

Assess the evidence for achievability of the command to decide whether any further action needs to be taken.

assess_command_progress_evidence

Assess the command progress evidence to decide whether any further action needs to be taken.

identify_command_requirement_change

Identify changes to the command requirement that this component has derived and needs to have satisfied by the rest of the system in order to achieve its Planned_Trajectory solution.

identify_commands_to_be_fulfilled

Identify the Motion_Commands to be fulfilled, including initiation.

5.4.2.70.7.1.3 Validity_Check

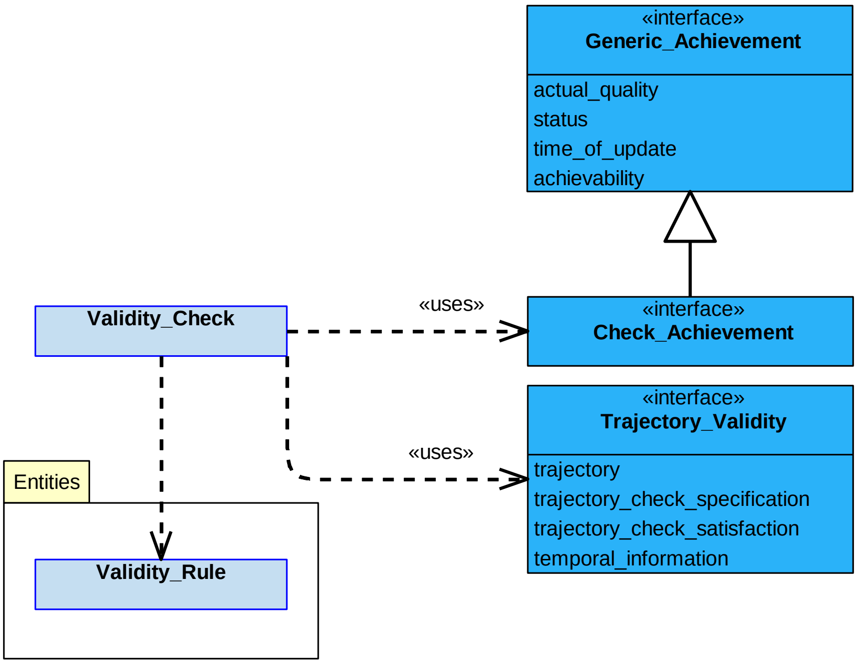


Figure 1205: Validity_Check Service Definition

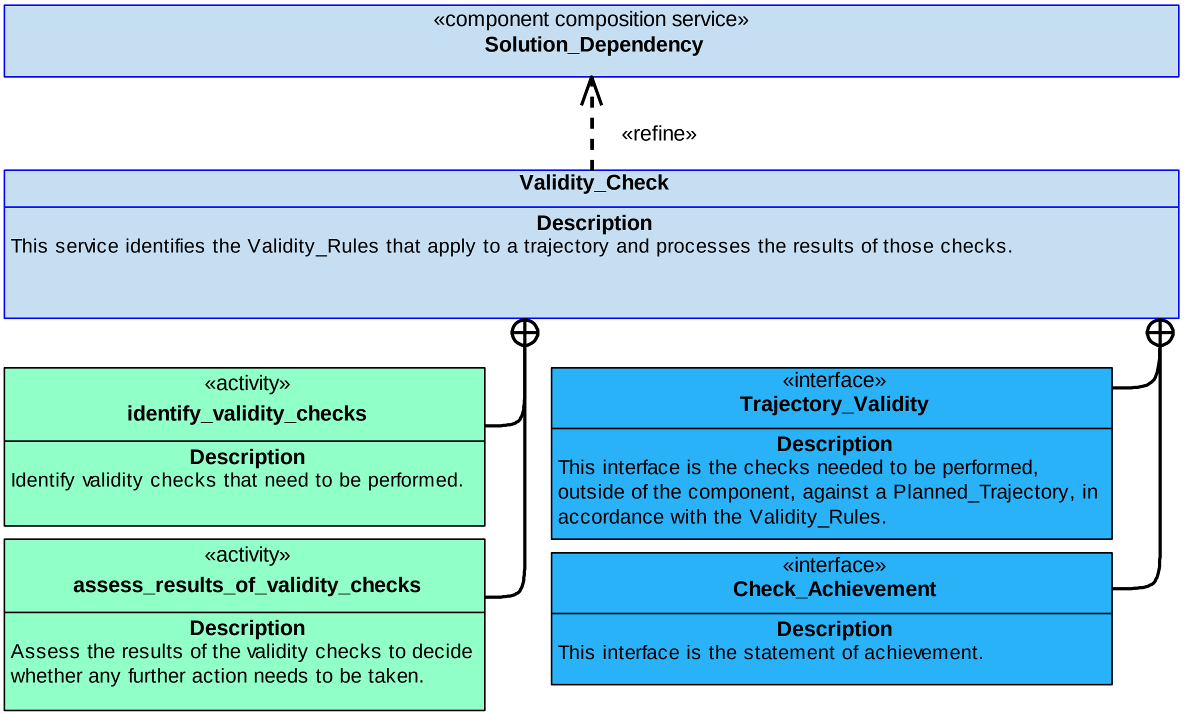

«component composition service»
Solution_Dependency
«refine»
Validity_Check
Description
This service identifies the Validity_Rules that apply to a trajectory and processes the results of those checks.
«activity»
identify_validity_checks
Description
Identify validity checks that need to be performed.
«activity»
assess_results_of_validity_checks
Description
Assess the results of the validity checks to decide whether any further action needs to be taken.
«interface»
Trajectory_Validity
Description
This interface is the checks needed to be performed, outside of the component, against a Planned_Trajectory, in accordance with the Validity_Rules.
«interface»
Check_Achievement
Description
This interface is the statement of achievement.

Figure 1206: Validity_Check Service Policy

Validity_Check

This service identifies the Validity_Rules that apply to a trajectory and processes the results of those checks.

Interfaces

Trajectory_Validity

This interface is the checks needed to be performed, outside of the component, against a Planned_Trajectory, in accordance with the Validity_Rules.

Attributes

trajectory	The trajectory to be checked.
trajectory_check_specification	Details of the check required to be carried out on the trajectory, e.g. a check that the trajectory does not infringe a no-fly zone.
trajectory_check_satisfaction	A measure of whether or not the trajectory passed all the necessary trajectory checks (such as safety checks).
temporal_information	Information covering timing, such as when the check was carried out, and for how long it remains valid.

Check_Achievement

This interface is the statement of achievement.

Activities

identify_validity_checks

Identify validity checks that need to be performed.

assess_results_of_validity_checks

Assess the results of the validity checks to decide whether any further action needs to be taken.

5.4.2.70.7.1.4 Performance_Parameter_Change

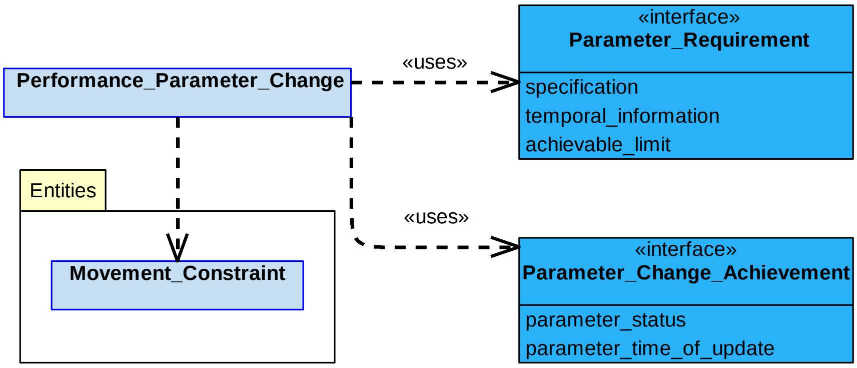


Figure 1207: Performance Parameter Change Service Definition

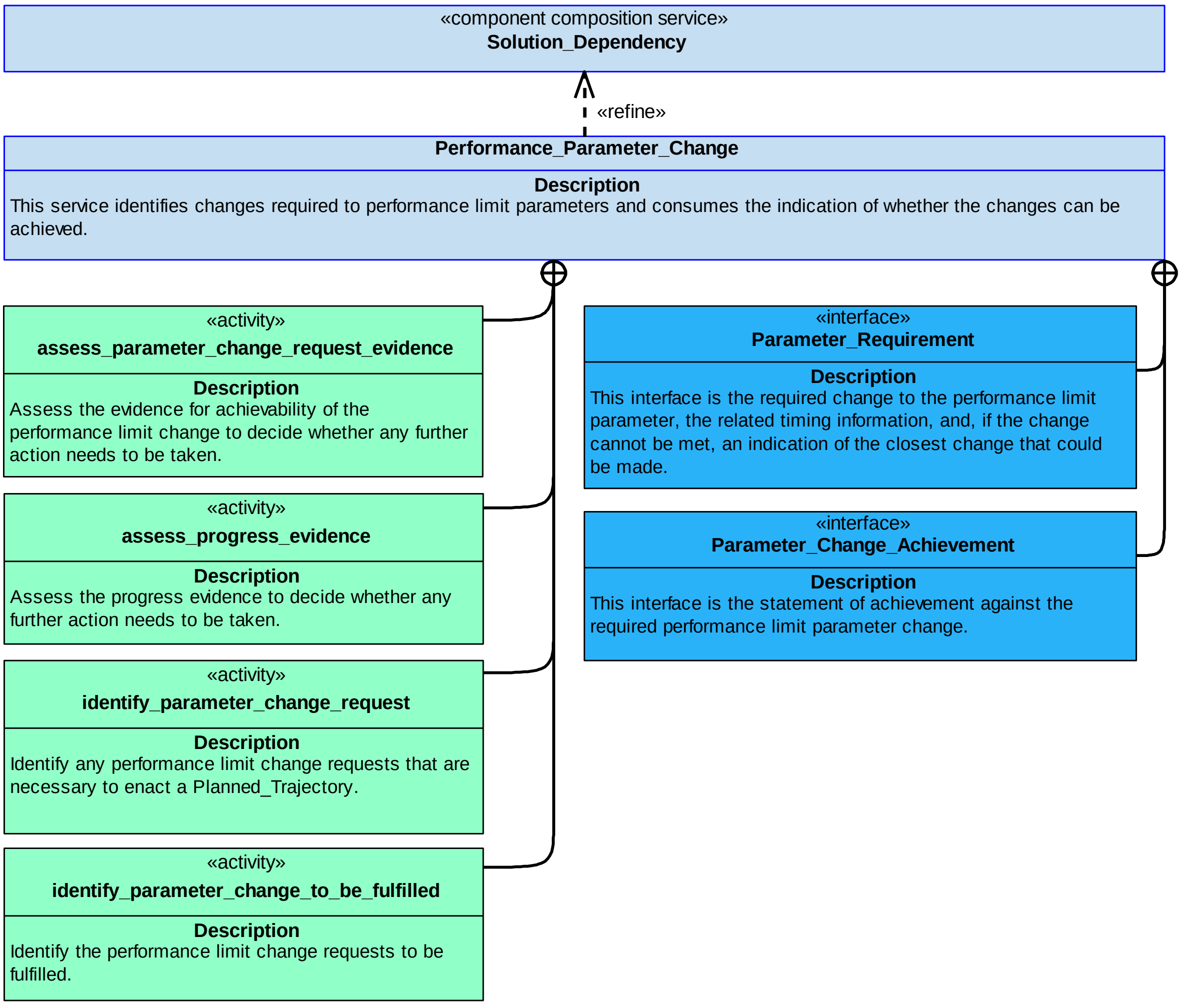

«component composition service»
Solution_Dependency
«refine»
Performance_Parameter_Change
Description
This service identifies changes required to performance limit parameters and consumes the indication of whether the changes can be achieved.
«activity»
assess_parameter_change_request_evidence
Description
Assess the evidence for achievability of the performance limit change to decide whether any further action needs to be taken.
«activity»
assess_progress_evidence
Description
Assess the progress evidence to decide whether any further action needs to be taken.
«activity»
identify_parameter_change_request
Description
Identify any performance limit change requests that are necessary to enact a Planned_Trajectory.
«activity»
identify_parameter_change_to_be_fulfilled
Description
Identify the performance limit change requests to be fulfilled.
«interface»
Parameter_Requirement
Description
This interface is the required change to the performance limit parameter, the related timing information, and, if the change cannot be met, an indication of the closest change that could be made.
«interface»
Parameter_Change_Achievement
Description
This interface is the statement of achievement against the required performance limit parameter change.

Figure 1208: Performance Parameter Change Service Policy

Performance_Parameter_Change

This service identifies changes required to performance limit parameters and consumes the indication of whether the changes can be achieved.

Interfaces

Parameter_Requirement

This interface is the required change to the performance limit parameter, the related timing information, and, if the change cannot be met, an indication of the closest change that could be made.

Attributes

specification	The definition of the required change to the performance limit parameter.
temporal_information	Information covering timing, such as start and end times of the required change to the performance limit parameter.
achievable_limit	How close to a request a parameter's value and timings can be set.

Parameter_Change_Achievement

This interface is the statement of achievement against the required performance limit parameter change.

Attributes

parameter_status	A high-level representation of achievement in relation to the requested change to the performance limit property (e.g. not started, in progress, or complete).
parameter_time_of_update	The time at which a performance limit parameter change achievement update occurred.

Activities

assess_parameter_change_request_evidence

Assess the evidence for achievability of the performance limit change to decide whether any further action needs to be taken.

identify_parameter_change_to_be_fulfilled

Identify the performance limit change requests to be fulfilled.

assess_progress_evidence

Assess the progress evidence to decide whether any further action needs to be taken.

identify_parameter_change_request

Identify any performance limit change requests that are necessary to enact a Planned_Trajectory.

5.4.2.70.7.1.5 Planned_Trajectory

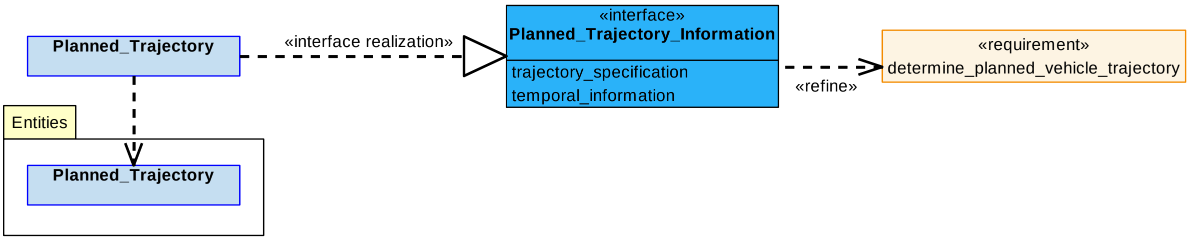


Figure 1209: Planned_Trajectory Service Definition

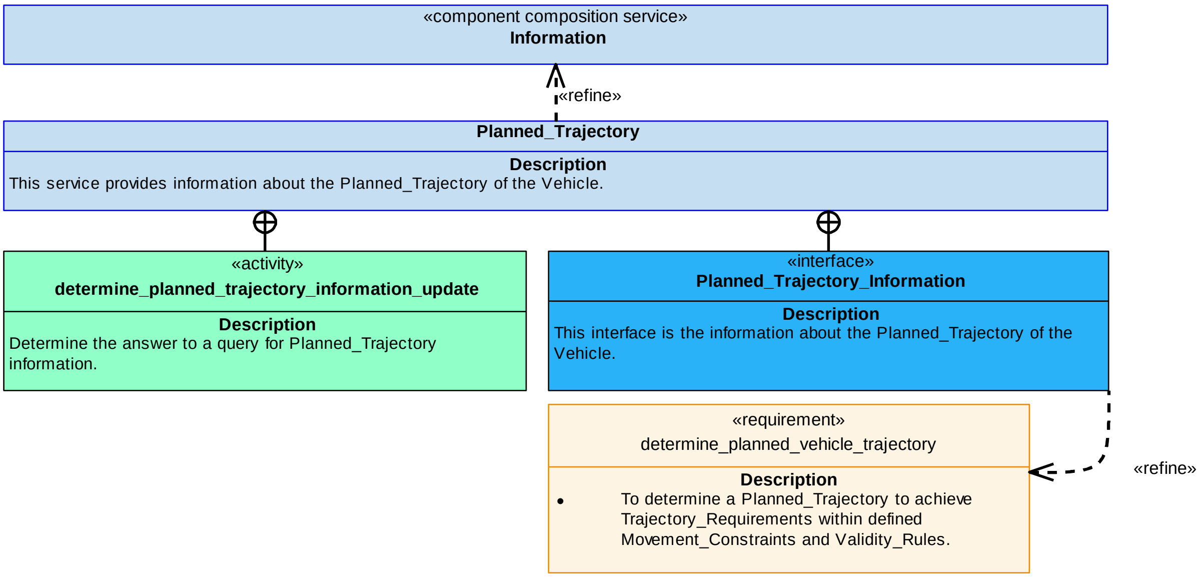

«component composition service»
Information
«refine»
Planned_Trajectory
Description
This service provides information about the Planned_Trajectory of the Vehicle.
«activity»
determine_planned_trajectory_information_update
Description
Determine the answer to a query for Planned_Trajectory information.
«interface»
Planned_Trajectory_Information
Description
This interface is the information about the Planned_Trajectory of the Vehicle.
«requirement»
determine_planned_vehicle_trajectory
Description
To determine a Planned_Trajectory to achieve Trajectory_Requirements within defined Movement_Constraints and Validity_Rules.
«refine»

Figure 1210: Planned_Trajectory Service Policy

Planned_Trajectory

This service provides information about the Planned_Trajectory of the Vehicle.

Interface

Planned_Trajectory_Information

This interface is the information about the Planned_Trajectory of the Vehicle.

Attributes

trajectory_specification	The details of the planned trajectory of the platform.
temporal_information	Information covering timing, such as start and end times.

Activity

determine_planned_trajectory_information_update

Determine the answer to a query for Planned_Trajectory information.

5.4.2.70.7.1.6 Performance_Guide

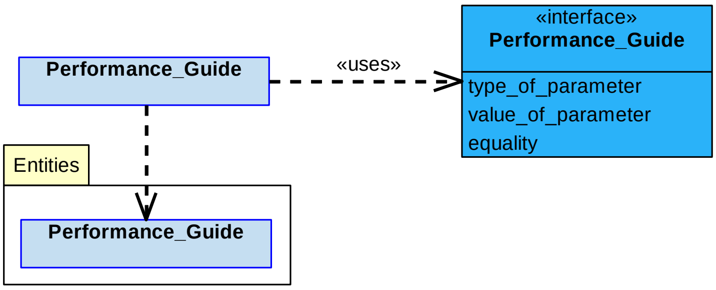


Figure 1211: Performance_Guide Service Definition

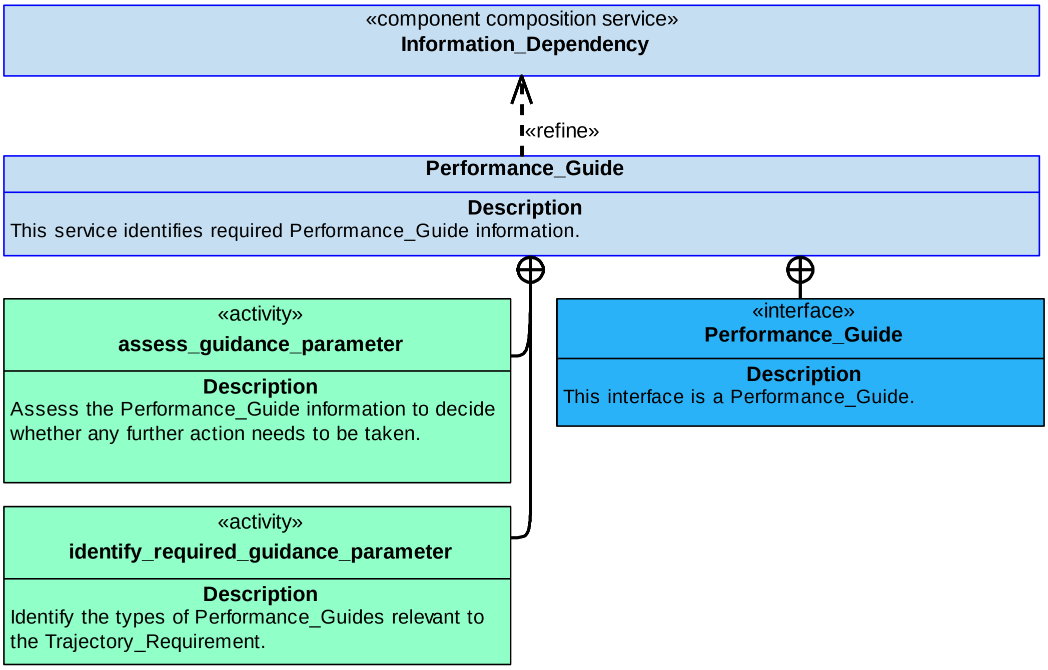

«component composition service»
Information_Dependency
«refine»
Performance_Guide
Description
This service identifies required Performance_Guide information.
«activity»
assess_guidance_parameter
Description
Assess the Performance_Guide information to decide whether any further action needs to be taken.
«interface»
Performance_Guide
Description
This interface is a Performance_Guide.
«activity»
identify_required_guidance_parameter
Description
Identify the types of Performance_Guides relevant to the Trajectory_Requirement.

Figure 1212: Performance_Guide Service Policy

Performance_Guide

This service identifies required Performance_Guide information.

Interface

Performance_Guide

This interface is a Performance_Guide.

Attributes

type_of_parameter	A type of Performance_Guide that the component uses to support a Trajectory_Requirement (e.g. airspeed to maximise range).
value_of_parameter	The value of the Performance_Guide (e.g. Mach 2).
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Activities

assess_guidance_parameter

Assess the Performance_Guide information to decide whether any further action needs to be taken.

identify_required_guidance_parameter

Identify the types of Performance_Guides relevant to the Trajectory_Requirement.

5.4.2.70.7.1.7 Sensor_Measurement

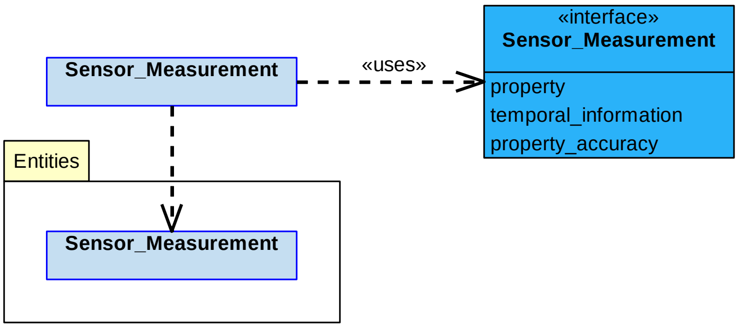


Figure 1213: Sensor_Measurement Service Definition

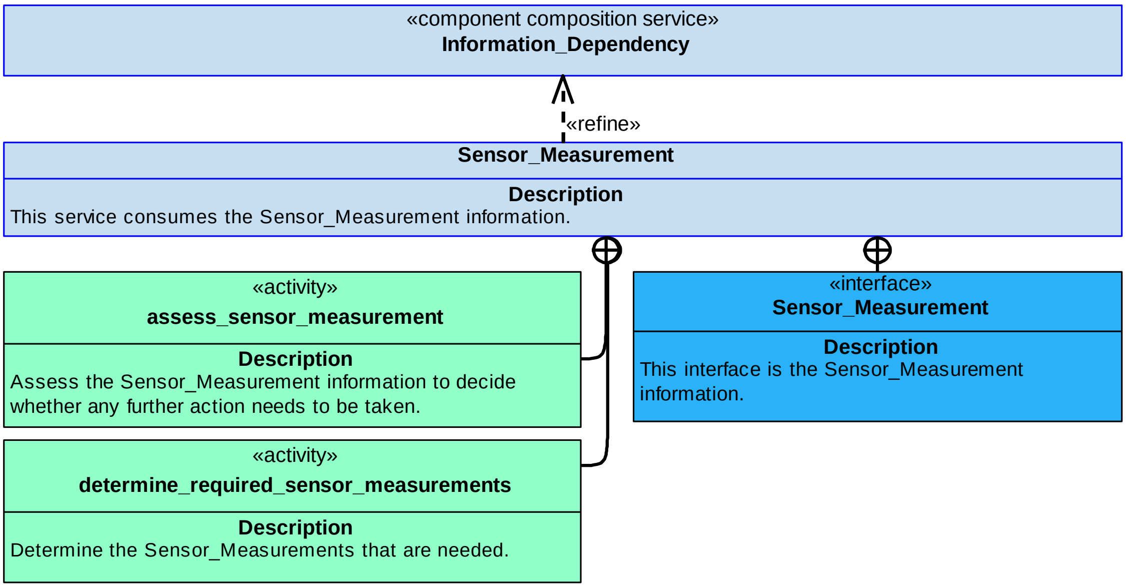

«component composition service»
Information_Dependency
«refine»
Sensor_Measurement
Description
This service consumes the Sensor_Measurement information.
«activity»
assess_sensor_measurement
Description
Assess the Sensor_Measurement information to decide whether any further action needs to be taken.
«activity»
determine_required_sensor_measurements
Description
Determine the Sensor_Measurements that are needed.
«interface»
Sensor_Measurement
Description
This interface is the Sensor_Measurement information.

Figure 1214: Sensor_Measurement Service Policy

Sensor_Measurement

This service consumes the Sensor_Measurement information.

Interface

Sensor_Measurement

This interface is the Sensor_Measurement information.

Attributes

property	The property and value of a Sensor_Measurement.
temporal_information	Information covering timing of the Sensor_Measurement information provided.
property_accuracy	The accuracy of the value of a particular Sensor_Measurement property.

Activities

assess_sensor_measurement

Assess the Sensor_Measurement information to decide whether any further action needs to be taken.

determine_required_sensor_measurements

Determine the Sensor_Measurements that are needed.

5.4.2.70.7.1.8 Constraint

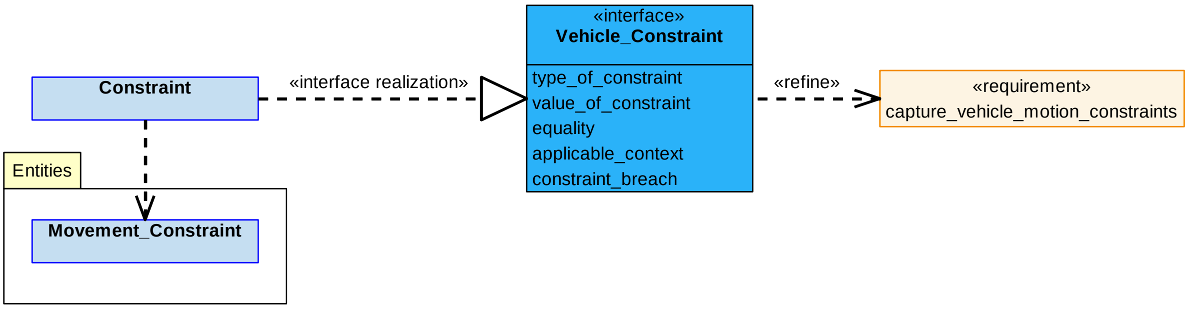


Figure 1215: Constraint Service Definition

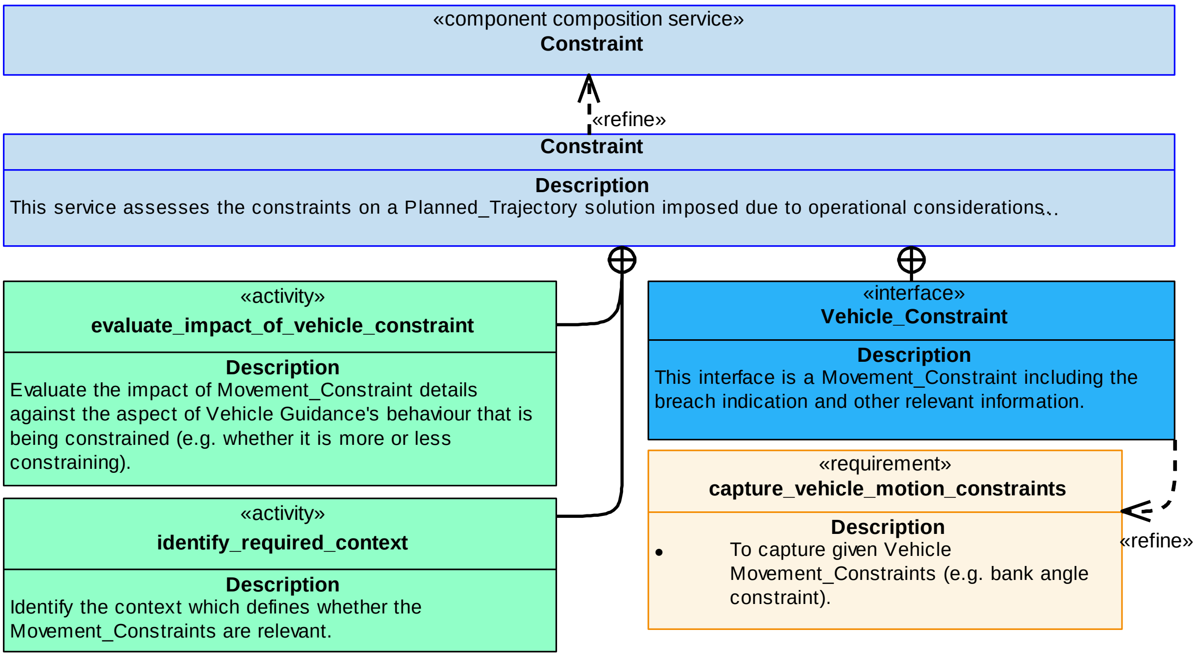

«component composition service»
Constraint
«refine»
Constraint
Description
This service assesses the constraints on a Planned_Trajectory solution imposed due to operational considerations..
«activity»
evaluate_impact_of_vehicle_constraint
Description
Evaluate the impact of Movement_Constraint details against the aspect of Vehicle Guidance's behaviour that is being constrained (e.g. whether it is more or less constraining).
«interface»
Vehicle_Constraint
Description
This interface is a Movement_Constraint including the breach indication and other relevant information.
«requirement»
capture_vehicle_motion_constraints
Description
To capture given Vehicle Movement_Constraints (e.g. bank angle constraint).
«refine»
«activity»
identify_required_context
Description
Identify the context which defines whether the Movement_Constraints are relevant.

Figure 1216: Constraint Service Policy

Constraint

This service assesses the constraints on a Planned_Trajectory solution imposed due to operational considerations.

Interface

Vehicle_Constraint

This interface is a Movement_Constraint including the breach indication and other relevant information.

Attributes

type_of_constraint	A type of limit that the component needs to adhere to (e.g. speed limit).
value_of_constraint	The value of the type of limit (e.g. Mach 2).
equality	The relationship between the value of the limit and the type of limit (e.g. less than or greater than).
applicable_context	The context in which the Movement_Constraint is applicable.
constraint_breach	A statement that the Movement_Constraint has been breached.

Activities

evaluate_impact_of_vehicle_constraint

Evaluate the impact of Movement_Constraint details against the aspect of Vehicle Guidance's behaviour that is being constrained (e.g. whether it is more or less constraining).

identify_required_context

Identify the context which defines whether the Movement_Constraints are relevant.

5.4.2.70.7.1.9 Capability

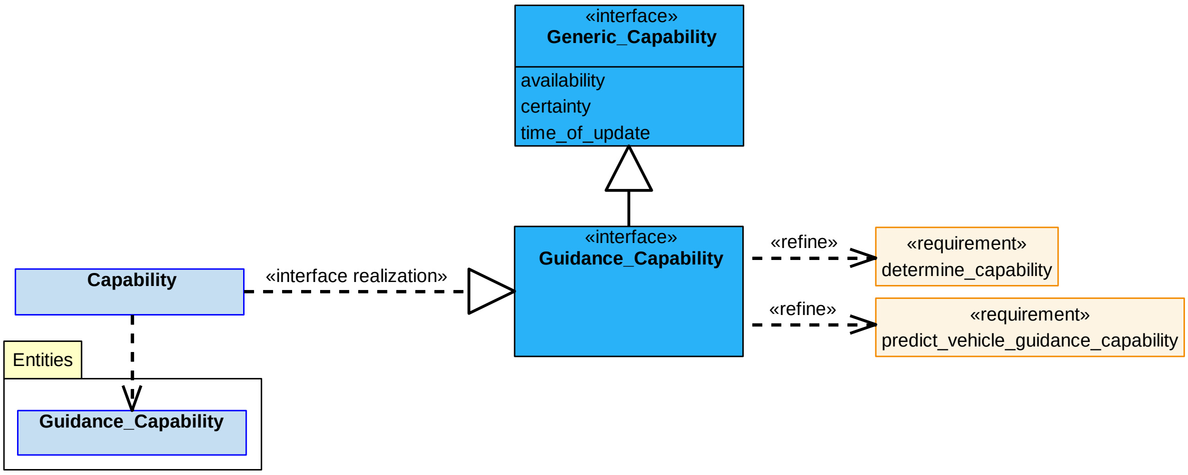


Figure 1217: Capability Service Definition

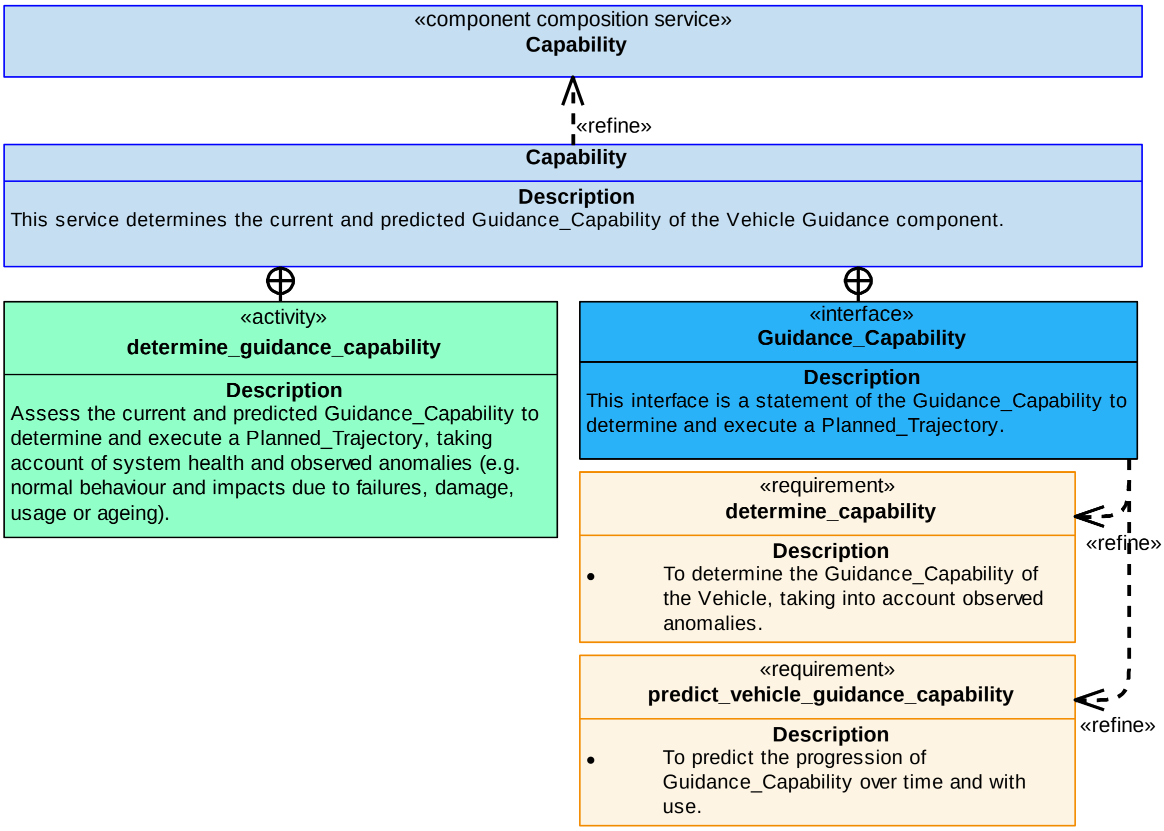

«component composition service»
Capability
«refine»
Capability
Description
This service determines the current and predicted Guidance_Capability of the Vehicle Guidance component.
«activity»
determine_guidance_capability
Description
Assess the current and predicted Guidance_Capability to determine and execute a Planned_Trajectory, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«interface»
Guidance_Capability
Description
This interface is a statement of the Guidance_Capability to determine and execute a Planned_Trajectory.
«requirement»
determine_capability
Description
To determine the Guidance_Capability of the Vehicle, taking into account observed anomalies.
«refine»
«requirement»
predict_vehicle_guidance_capability
Description
To predict the progression of Guidance_Capability over time and with use.
«refine»

Figure 1218: Capability Service Policy

Capability

This service determines the current and predicted Guidance_Capability of the Vehicle Guidance component.

Interface

Guidance_Capability

This interface is a statement of the Guidance_Capability to determine and execute a Planned_Trajectory.

Activity

determine_guidance_capability

Assess the current and predicted Guidance_Capability to determine and execute a Planned_Trajectory, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

5.4.2.70.7.1.10 Capability_Evidence

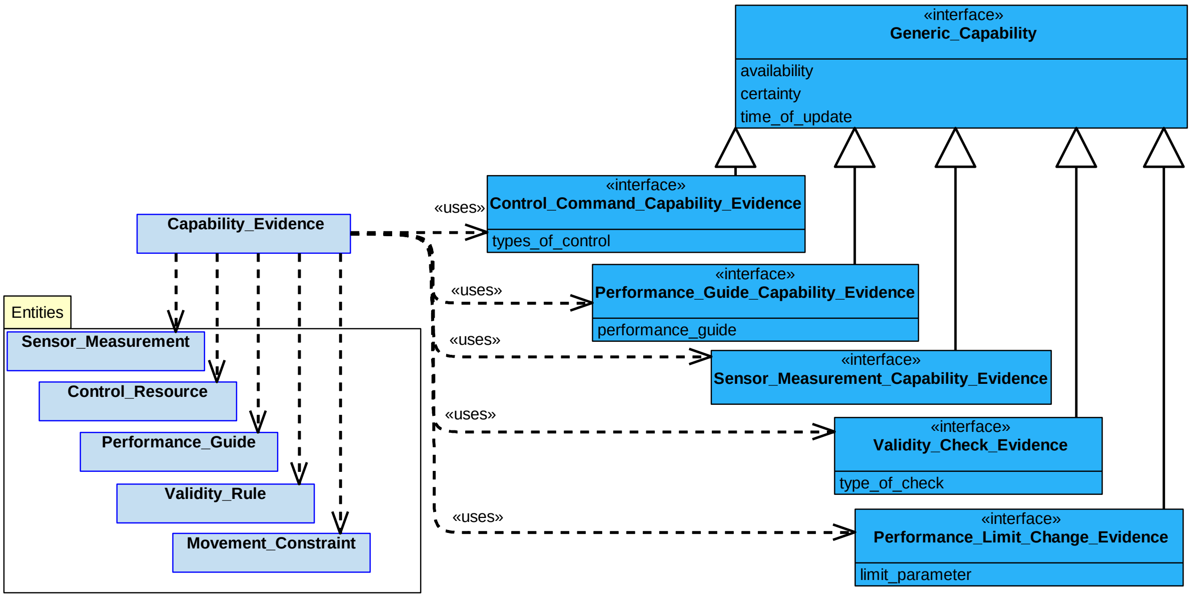


Figure 1219: Capability_Evidence Service Definition

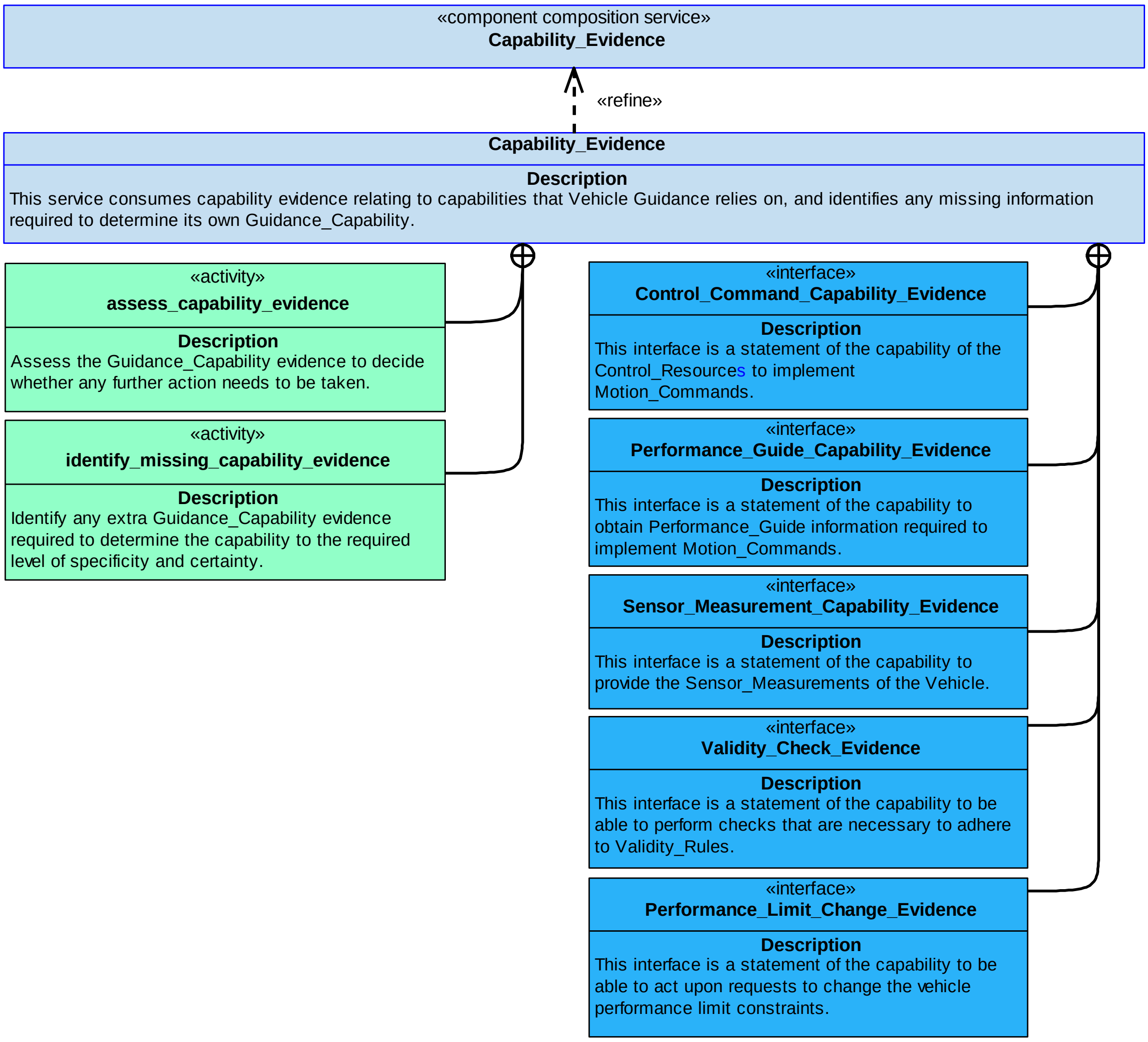

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capability evidence relating to capabilities that Vehicle Guidance relies on, and identifies any missing information required to determine its own Guidance_Capability.
«activity»
assess_capability_evidence
Description
Assess the Guidance_Capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra Guidance_Capability evidence required to determine the capability to the required level of specificity and certainty.
«interface»
Control_Command_Capability_Evidence
Description
This interface is a statement of the capability of the Control_Resources to implement Motion_Commands.
«interface»
Performance_Guide_Capability_Evidence
Description
This interface is a statement of the capability to obtain Performance_Guide information required to implement Motion_Commands.
«interface»
Sensor_Measurement_Capability_Evidence
Description
This interface is a statement of the capability to provide the Sensor_Measurements of the Vehicle.
«interface»
Validity_Check_Evidence
Description
This interface is a statement of the capability to be able to perform checks that are necessary to adhere to Validity_Rules.
«interface»
Performance_Limit_Change_Evidence
Description
This interface is a statement of the capability to be able to act upon requests to change the vehicle performance limit constraints.

Figure 1220: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability evidence relating to capabilities that Vehicle Guidance relies on, and identifies any missing information required to determine its own Guidance_Capability.

Interfaces

Control_Command_Capability_Evidence

This interface is a statement of the capability of the Control_Resources to implement Motion_Commands.

Attribute

types_of_control	The types of Motion_Commands that can be actioned.

Performance_Guide_Capability_Evidence

This interface is a statement of the capability to obtain Performance_Guide information required to implement Motion_Commands.

Attribute

performance_guide	The type of Performance_Guide information provided.

Sensor_Measurement_Capability_Evidence

This interface is a statement of the capability to provide the Sensor_Measurements of the Vehicle.

Validity_Check_Evidence

This interface is a statement of the capability to be able to perform checks that are necessary to adhere to Validity_Rules.

Attribute

type_of_check	The type of validity check against which the capability evidence is being stated.

Performance_Limit_Change_Evidence

This interface is a statement of the capability to be able to act upon requests to change the vehicle performance limit constraints.

Attribute

limit_parameter	The type of limit parameter against which the capability evidence is being stated.

Activities

assess_capability_evidence

Assess the Guidance_Capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra Guidance_Capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.70.7.2 Service Dependencies

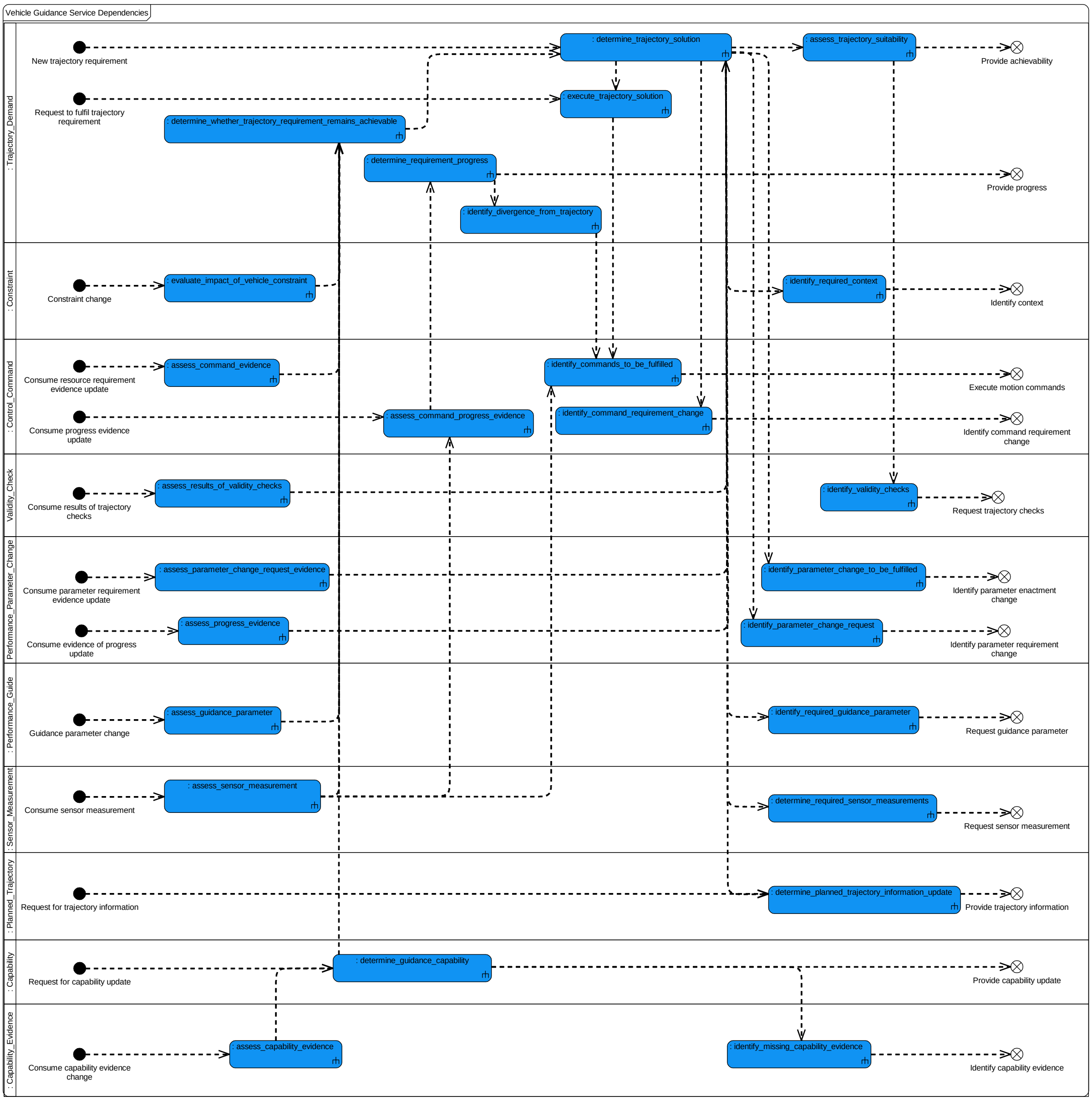

Vehicle Guidance Service Dependencies
: Trajectory_Demand
: Constraint
: Control_Command
: Validity_Check
: Performance_Parameter_Change
: Performance_Guide
: Sensor_Measurement
: Planned_Trajectory
: Capability
: Capability_Evidence
New trajectory requirement
: determine_trajectory_solution
: assess_trajectory_suitability
Provide achievability
Request to fulfil trajectory requirement
: execute_trajectory_solution
: determine_whether_trajectory_requirement_remains_achievable
: determine_requirement_progress
Provide progress
: identify_divergence_from_trajectory
Constraint change
: evaluate_impact_of_vehicle_constraint
: identify_required_context
Identify context
Consume resource requirement evidence update
: assess_command_evidence
: identify_commands_to_be_fulfilled
Execute motion commands
Consume progress evidence update
: assess_command_progress_evidence
: identify_command_requirement_change
Identify command requirement change
Consume results of trajectory checks
: assess_results_of_validity_checks
: identify_validity_checks
Request trajectory checks
Consume parameter requirement evidence update
: assess_parameter_change_request_evidence
: identify_parameter_change_to_be_fulfilled
Identify parameter enactment change
Consume evidence of progress update
: assess_progress_evidence
: identify_parameter_change_request
Identify parameter requirement change
Guidance parameter change
: assess_guidance_parameter
: identify_required_guidance_parameter
Request guidance parameter
Consume sensor measurement
: assess_sensor_measurement
: determine_required_sensor_measurements
Request sensor measurement
Request for trajectory information
: determine_planned_trajectory_information_update
Provide trajectory information
Request for capability update
: determine_guidance_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1221: Vehicle Guidance Service Dependencies

5.4.2.71 Vehicle Performance

5.4.2.71.1 Role

The role of Vehicle Performance is to determine the vehicle movement performance parameters for different vehicle activities and configurations.

5.4.2.71.2 Overview

Control Architecture

Vehicle Performance is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In response to a query about the Vehicle_Configuration or Performance_Parameters, Vehicle Performance determines the appropriate Performance_Parameters against the applicable Performance_Envelope. For example, for a Parameter_Query, the Vehicle_Activity, External_Conditions and Vehicle_Configuration will be used to determine the maximum speed during banking.

Additionally, Vehicle Performance identifies Performance_Parameter constraints based on the Vehicle_Configuration, Vehicle_Activity and External_Conditions.

Examples of Use

- Vehicle Performance will be required where allowable vehicle Performance_Parameter values vary, depending on the Vehicle_Configuration and/or Vehicle_Activity. For example, it will be required to determine a maximum speed allowable when the undercarriage is lowered, or to determine an appropriate Vehicle_Configuration given a particular set of Performance_Parameter values.

5.4.2.71.3 Service Summary

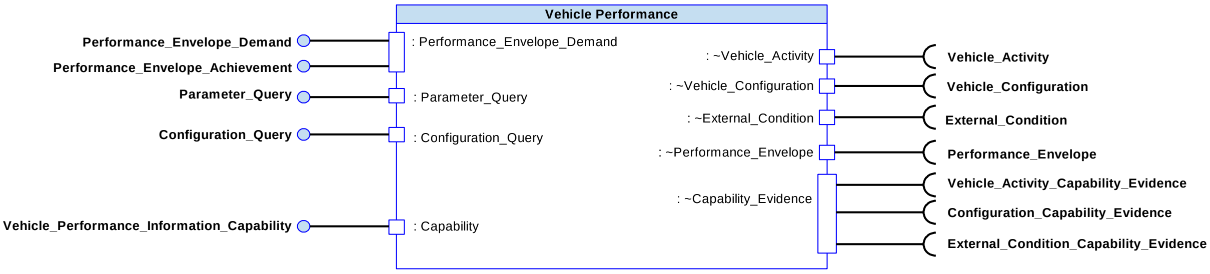


Figure 1222: Vehicle Performance Service Summary

5.4.2.71.4 Responsibilities

determine_performance_envelope

- To determine and actively manage the suitable Performance_Envelope for a defined Vehicle_Activity and/or Vehicle_Configuration and identify conflicting demands which prevent a suitable Performance_Envelope from being determined based on the applicable Performance_Regimes and constraints.

determine_applicable_values

- To determine applicable Performance_Parameter values, which apply to a Performance_Regime or Performance_Envelope, for combinations of Vehicle_Activity, External_Condition and Vehicle_Configuration.

determine_vehicle_configuration

- To determine Vehicle_Configurations for which a combination of Performance_Parameter values is allowable.

manage_performance_regime

- To determine and actively manage the Performance_Regime(s) in response to Vehicle_Activity, Vehicle_Configuration and External_Conditions and to identify conflicts which prevent an allowable Performance_Regime being determined.

capture_performance_demands

- To capture all demands that will impact the Performance_Envelope.

capture_vehicle_configuration

- To capture the Vehicle_Configuration.

capture_vehicle_activity

- To capture any Vehicle_Activity that will affect vehicle performance.

capture_conditions

- To capture External_Conditions that affect vehicle performance.

assess_capability

- To assess the Capability to determine applicable Performance_Envelopes, identify Performance_Parameter constraints and answer queries, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Vehicle Performance Capability assessment.

predict_capability_progression

- To predict the progression of the component's Capability over time and with use.

5.4.2.71.5 Subject Matter Semantics

The subject matter of Vehicle Performance is the vehicle movement Performance_Parameters and the Performance_Regime rulesets by which they are determined.

Exclusions

The subject matter of Vehicle Performance does not include:

- The determination of whether Performance_Parameter limits or restrictions for Vehicle_Configuration or vehicle activities are complied with or exceeded, Vehicle Performance merely determines those that are applicable.

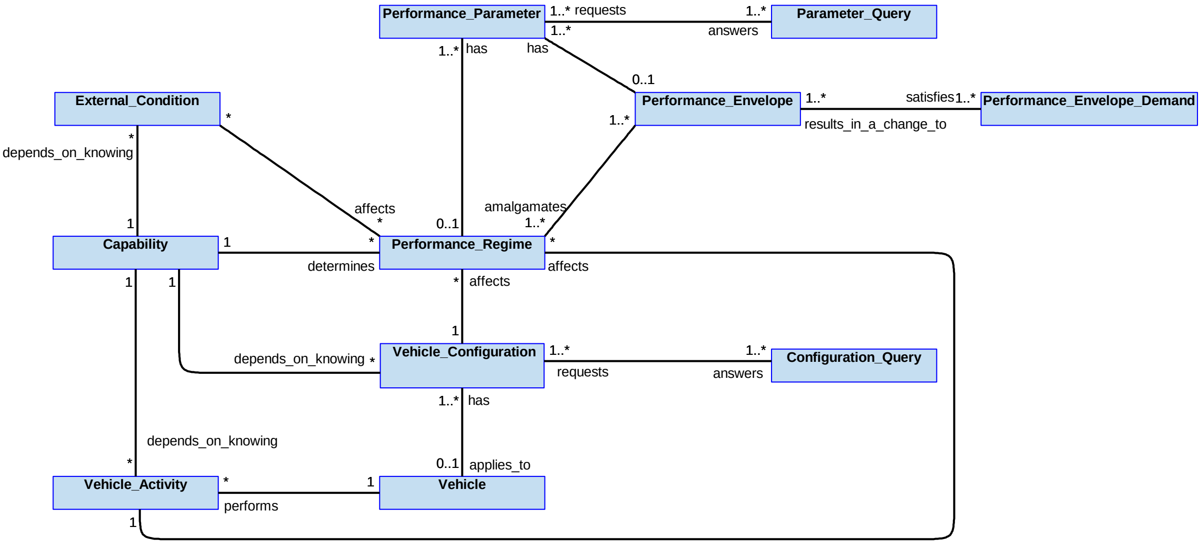


Figure 1223: Vehicle Performance Semantics

5.4.2.71.5.1 Entities

Configuration_Query

A query to determine the appropriate configuration needed to achieve or maintain Performance_Parameters within a Performance_Envelope or a Performance_Regime given certain External_Conditions and type of Vehicle_Activity.

External_Condition

An external condition (e.g. the immediate environment surrounding the vehicle, the runway length or an imposed speed limit).

Parameter_Query

A query to determine the permitted parameter values for a Vehicle_Configuration and/or Vehicle_Activity, in a Performance_Regime, given certain External_Conditions.

Performance_Regime

A range of vehicle movement performance conditions relating to a specific mode of operation of a vehicle, which may be either a, current or required, specific vehicle configuration (e.g. the state of the landing gear or the weapon bay doors) or specific activity (e.g. as landing or weapons release). The conditions define the allowable limits (e.g. the minimum and maximum speed allowed for weapon release) and ways to optimise any relevant specific aspects of vehicle performance.

Vehicle

A moveable entity to which performance characteristics apply.

Vehicle_Configuration

A combination of vehicle elements, the position or state of which affects vehicle performance.

Capability

The capability to provide accurate information about vehicle performance.

Performance_Parameter

A specific characteristic of a Performance_Envelope/Performance_Regime (e.g. optimum altitude or maximum altitude) or a specific characteristic relating to a condition that results in the need for a Performance_Envelope/Performance_Regime (e.g. maximum take-off mass).

Vehicle_Activity

An activity being executed by the vehicle which may affect vehicle performance.

Performance_Envelope

A range of vehicle movement performance conditions that results from the combination of all relevant Performance_Regimes, which are either currently active of planned to be active at the same time.

Performance_Envelope_Demand

A demand to determine the required Performance_Envelope based on the applicable vehicle activities, vehicle configurations, and External_Conditions.

5.4.2.71.6 Design Rationale

5.4.2.71.6.1 Assumptions

- The methods of determining applicable Performance_Parameters are not expected to change after build time.
- The set of possible Vehicle_Configurations are expected to change after build time.
- The set of possible Performance_Regimes are expected to change after build time.

5.4.2.71.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Vehicle Performance:

- Data Driving - Traditional vehicle control systems typically define multiple performance envelopes consisting of relationships between vehicle Performance_Parameters. The appropriate envelopes are then selected dynamically to suit the current configuration of the vehicle or the conditions/environment it is operating in. If these design authority approved envelopes are represented as fixed tables they could be accommodated by data driving.

Extensions

- If the design authority approved envelopes (discussed above) are represented as algorithms instead of fixed tables, these could be accommodated by extension components.

Exploitation Considerations

- It is expected that some performance regimes are mutually exclusive (e.g. take-off and landing). Other regimes may be valid simultaneously (e.g. normal flight envelope and a store release envelope). This component would be expected to determine the most restrictive limits.
- This component will provide the best performance information based upon the information it receives. If such information is of low certainty or unavailable (e.g. the state of an aperture is unknown due to a sensor failure), then the response to a query may be overly conservative.

5.4.2.71.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- This component determines Performance_Parameter limits. In the case of an air vehicle, exceeding these limits may result in uncontrolled flight. This could lead to loss of structural integrity of the air vehicle and / or an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

5.4.2.71.6.4 Security Considerations

The indicative security classification is SNEO.

This component has information about the minimum, maximum and optimum Performance_Parameters, possible Performance_Regimes and Vehicle_Configurations of the Exploiting Platform, therefore the indicative security classification is SNEO, with appropriate protection required to be in place to protect the confidentiality of this information. This is one of a series of components that will assist in identifying if form and fit integrity has been interfered with.

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of the performance data being used during a mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.71.7 Services

5.4.2.71.7.1 Service Definitions

5.4.2.71.7.1.1 Performance_Envelope_Demand

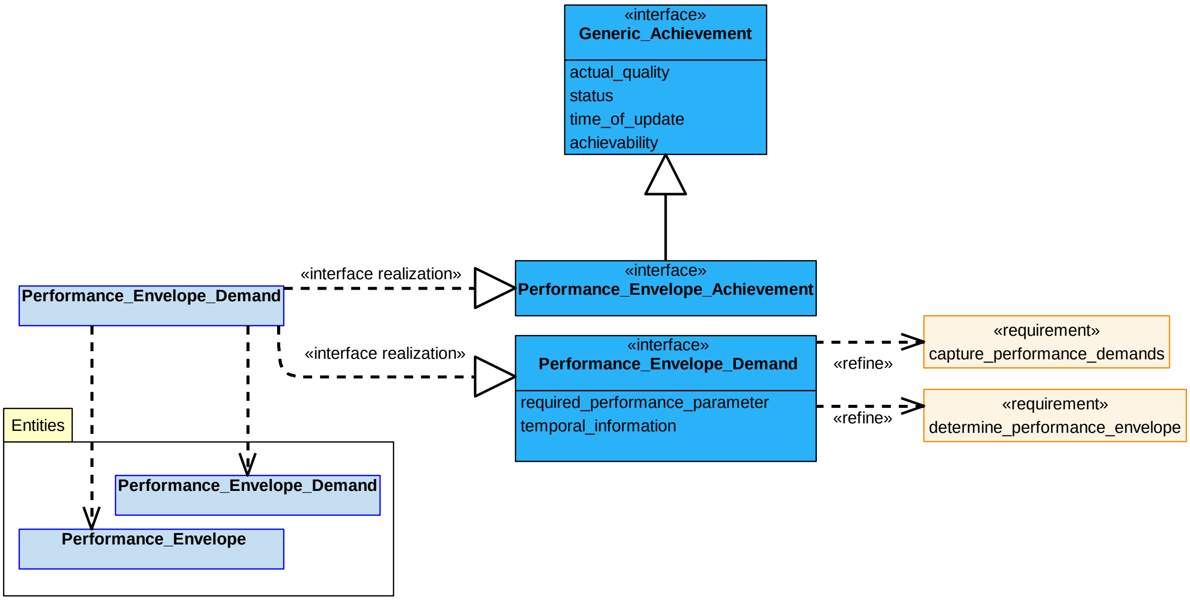


Figure 1224: Performance_Envelope_Demand Service Definition

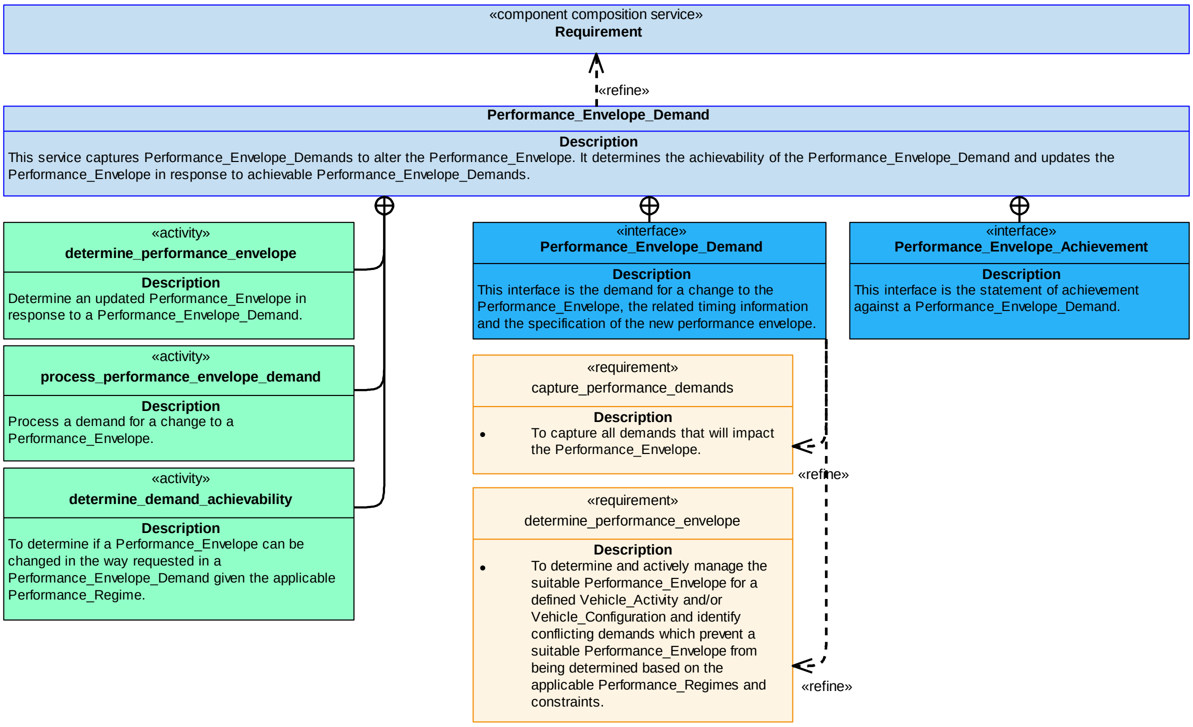

«component composition service»
Requirement
«refine»
Performance_Envelope_Demand
Description
This service captures Performance_Envelope_Demands to alter the Performance_Envelope. It determines the achievability of the Performance_Envelope_Demand and updates the Performance_Envelope in response to achievable Performance_Envelope_Demands.
«activity»
determine_performance_envelope
Description
Determine an updated Performance_Envelope in response to a Performance_Envelope_Demand.
«activity»
process_performance_envelope_demand
Description
Process a demand for a change to a Performance_Envelope.
«activity»
determine_demand_achievability
Description
To determine if a Performance_Envelope can be changed in the way requested in a Performance_Envelope_Demand given the applicable Performance_Regime.
«interface»
Performance_Envelope_Demand
Description
This interface is the demand for a change to the Performance_Envelope, the related timing information and the specification of the new performance envelope.
«interface»
Performance_Envelope_Achievement
Description
This interface is the statement of achievement against a Performance_Envelope_Demand.
«requirement»
capture_performance_demands
Description
• To capture all demands that will impact the Performance_Envelope.
«refine»
«requirement»
determine_performance_envelope
Description
• To determine and actively manage the suitable Performance_Envelope for a defined Vehicle_Activity and/or Vehicle_Configuration and identify conflicting demands which prevent a suitable Performance_Envelope from being determined based on the applicable Performance_Regimes and constraints.
«refine»

Figure 1225: Performance_Envelope_Demand Service Policy

Performance_Envelope_Demand

This service captures Performance_Envelope_Demands to alter the Performance_Envelope. It determines the achievability of the Performance_Envelope_Demand and updates the Performance_Envelope in response to achievable Performance_Envelope_Demands.

Interfaces

Performance_Envelope_Demand

This interface is the demand for a change to the Performance_Envelope, the related timing information and the specification of the new performance envelope.

Attributes

required_performance_parameter	The Performance_Parameter(s) requested to be altered by the Performance_Envelope_Demand (including allowable tolerances and whether the required value is a target, minimum or maximum).
temporal_information	Information covering timing, such as start and end times, e.g. the start and end times that the Performance_Envelope will attain to for the given Vehicle_Activity.

Performance_Envelope_Achievement

This interface is the statement of achievement against a Performance_Envelope_Demand.

Activities

determine_performance_envelope

Determine an updated Performance_Envelope in response to a Performance_Envelope_Demand.

process_performance_envelope_demand

Process a demand for a change to a Performance_Envelope.

determine_demand_achievability

To determine if a Performance_Envelope can be changed in the way requested in a Performance_Envelope_Demand given the applicable Performance_Regime.

5.4.2.71.7.1.2 Parameter_Query

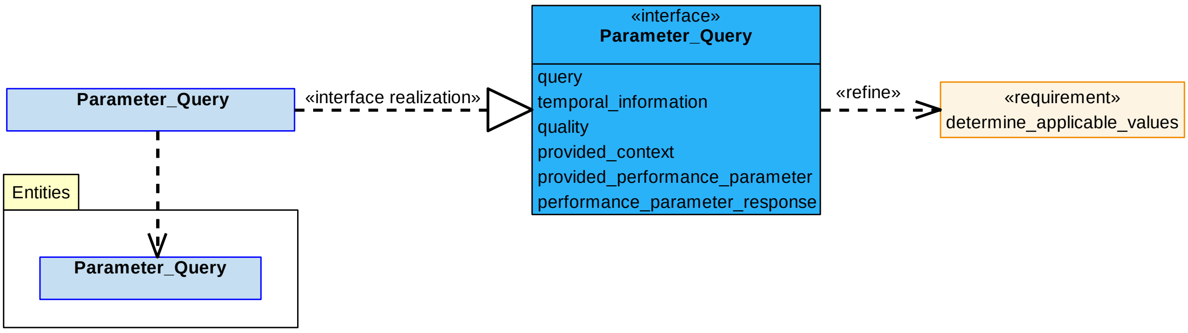


Figure 1226: Parameter_Query Service Definition

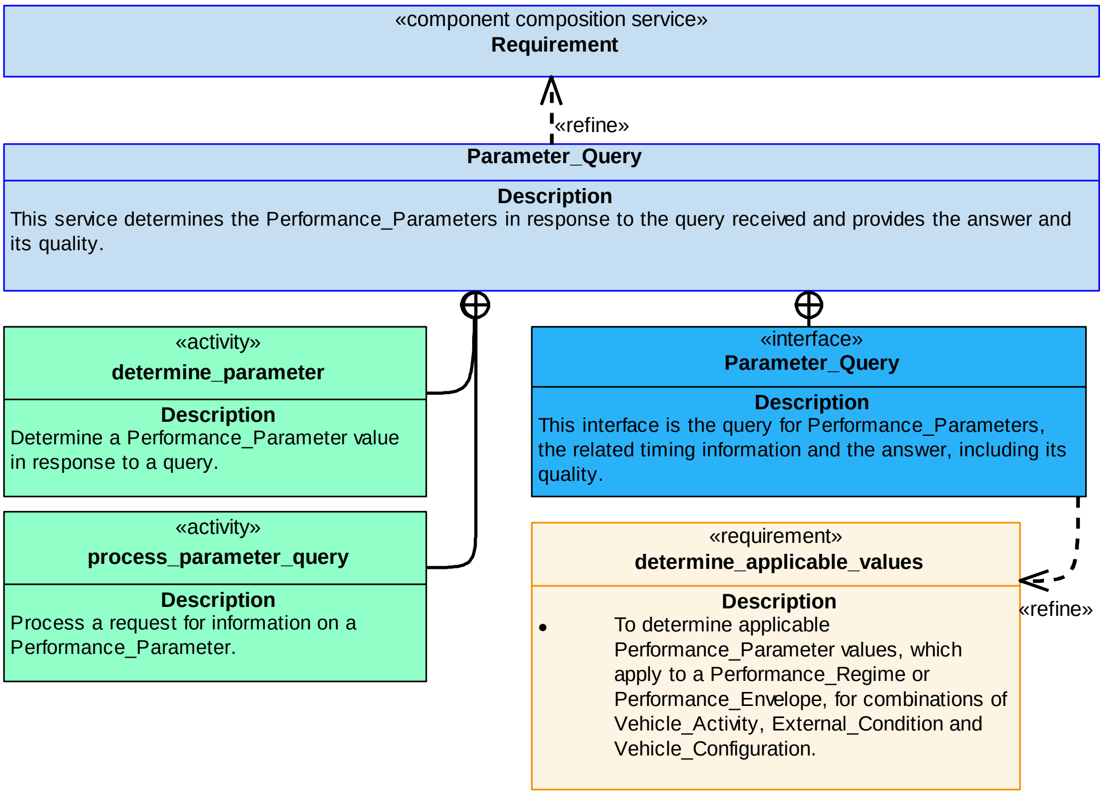

«component composition service»
Requirement
«refine»
Parameter_Query
Description
This service determines the Performance_Parameters in response to the query received and provides the answer and its quality.
«activity»
determine_parameter
Description
Determine a Performance_Parameter value in response to a query.
«activity»
process_parameter_query
Description
Process a request for information on a Performance_Parameter.
«interface»
Parameter_Query
Description
This interface is the query for Performance_Parameters, the related timing information and the answer, including its quality.
«requirement»
determine_applicable_values
Description
To determine applicable Performance_Parameter values, which apply to a Performance_Regime or Performance_Envelope, for combinations of Vehicle_Activity, External_Condition and Vehicle_Configuration.
«refine»

Figure 1227: Parameter_Query Service Policy

Parameter_Query

This service determines the Performance_Parameters in response to the query received and provides the answer and its quality.

Interface

Parameter_Query

This interface is the query for Performance_Parameters, the related timing information and the answer, including its quality.

Attributes

query	The question being asked, e.g. what is the maximum permissible speed with the current configuration and conditions?
temporal_information	Information covering timing, such as start and end times, e.g. the Performance_Parameter limitations start and end times, given that the vehicle will be undergoing certain Vehicle_Configuration evolutions at a certain time.
quality	The quality (e.g. conservativeness and compliance to requirements) in the provided response.
provided_context	The Vehicle_Configuration, Vehicle_Activity and External_Conditions provided within the query.
provided_performance_parameter	A Performance_Parameter fixed as part of a query.

performance_parameter_response	The Performance_Parameter returned as a solution to a Parameter_Query.

Activities

determine_parameter

Determine a Performance_Parameter value in response to a query.

process_parameter_query

Process a request for information on a Performance_Parameter.

5.4.2.71.7.1.3 Configuration_Query

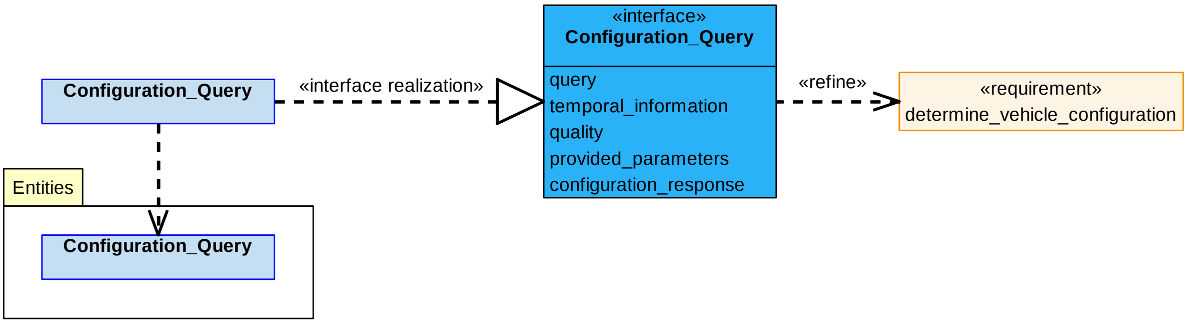


Figure 1228: Configuration_Query Service Definition

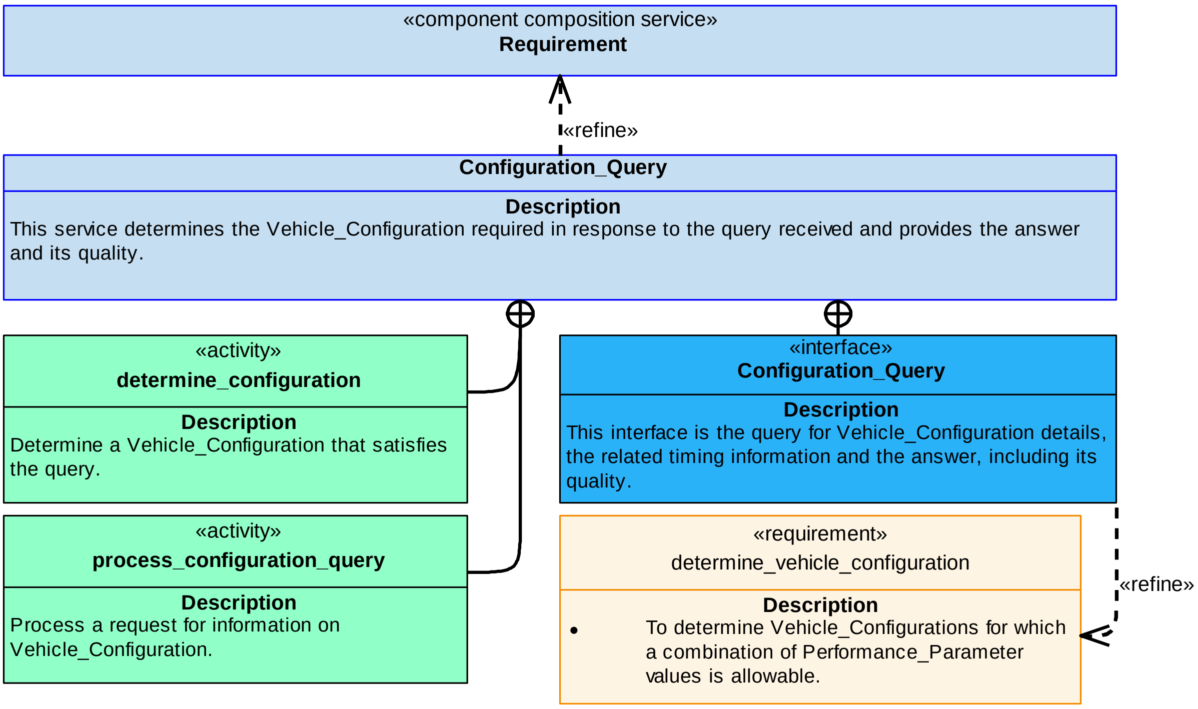

«component composition service»
Requirement
«refine»
Configuration_Query
Description
This service determines the Vehicle_Configuration required in response to the query received and provides the answer and its quality.
«activity»
determine_configuration
Description
Determine a Vehicle_Configuration that satisfies the query.
«activity»
process_configuration_query
Description
Process a request for information on Vehicle_Configuration.
«interface»
Configuration_Query
Description
This interface is the query for Vehicle_Configuration details, the related timing information and the answer, including its quality.
«requirement»
determine_vehicle_configuration
Description
• To determine Vehicle_Configurations for which a combination of Performance_Parameter values is allowable.
«refine»

Figure 1229: Configuration_Query Service Policy

Configuration_Query

This service determines the Vehicle_Configuration required in response to the query received and provides the answer and its quality.

Interface

Configuration_Query

This interface is the query for Vehicle_Configuration details, the related timing information and the answer, including its quality.

Attributes

query	The question that is being asked, e.g. positions of aerodynamic surfaces required to achieve the lowest drag on aircraft.
temporal_information	Information covering timing, such as start and end times, e.g. the Vehicle_Configuration start and end times, given that the vehicle will be in a certain location at a certain time.
quality	The quality (e.g. conservativeness and compliance to requirements) in the provided answer.
provided_parameters	The parameters provided within the query (e.g. vehicle weight or speed).
configuration_response	The Vehicle_Configuration(s) which are returned as a solution to a query.

Activities

determine_configuration

Determine a Vehicle_Configuration that satisfies the query.

process_configuration_query

Process a request for information on Vehicle_Configuration.

5.4.2.71.7.1.4 Vehicle_Activity

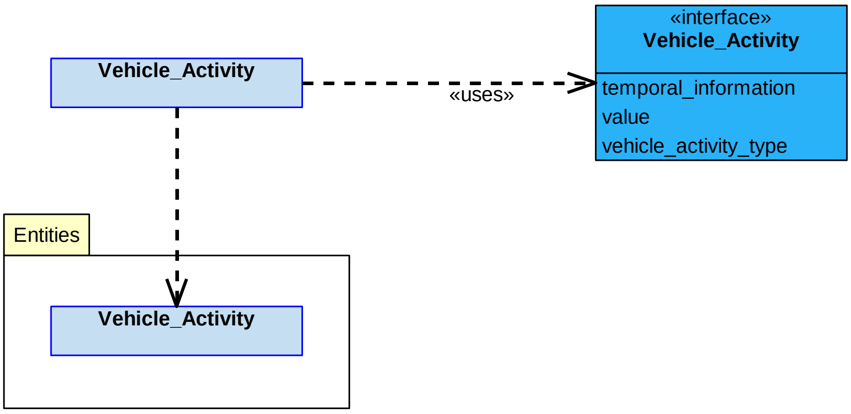


Figure 1230: Vehicle_Activity Service_Definition

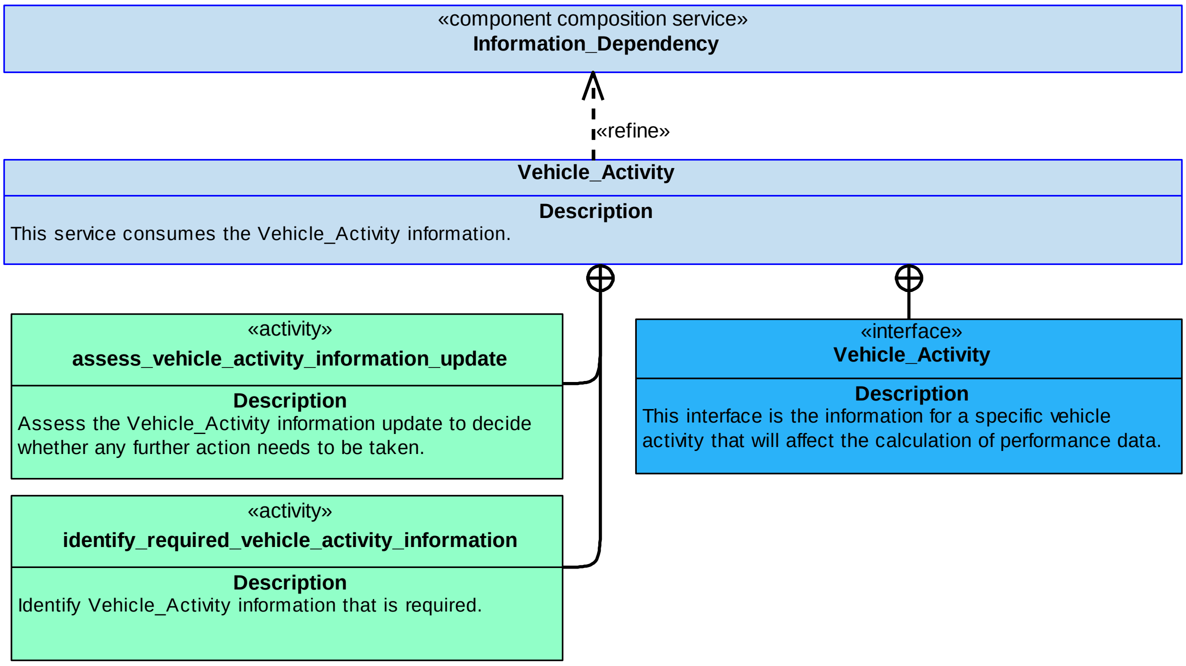

«component composition service»
Information_Dependency
«refine»
Vehicle_Activity
Description
This service consumes the Vehicle_Activity information.
«activity»
assess_vehicle_activity_information_update
Description
Assess the Vehicle_Activity information update to decide whether any further action needs to be taken.
«activity»
identify_required_vehicle_activity_information
Description
Identify Vehicle_Activity information that is required.
«interface»
Vehicle_Activity
Description
This interface is the information for a specific vehicle activity that will affect the calculation of performance data.

Figure 1231: Vehicle_Activity Service_Policy

Vehicle_Activity

This service consumes the Vehicle_Activity information.

Interface

Vehicle_Activity

This interface is the information for a specific vehicle activity that will affect the calculation of performance data.

Attributes

temporal_information	Information covering timing, such as start and end times, e.g. the period over which a vehicle activity occurs.
value	A numerical parameter associated with the vehicle activity.
vehicle_activity_type	Information on the activity type being executed by the vehicle which will affect the calculation of performance data.

Activities

assess_vehicle_activity_information_update

Assess the Vehicle_Activity information update to decide whether any further action needs to be taken.

identify_required_vehicle_activity_information

Identify Vehicle_Activity information that is required.

5.4.2.71.7.1.5 Vehicle_Configuration

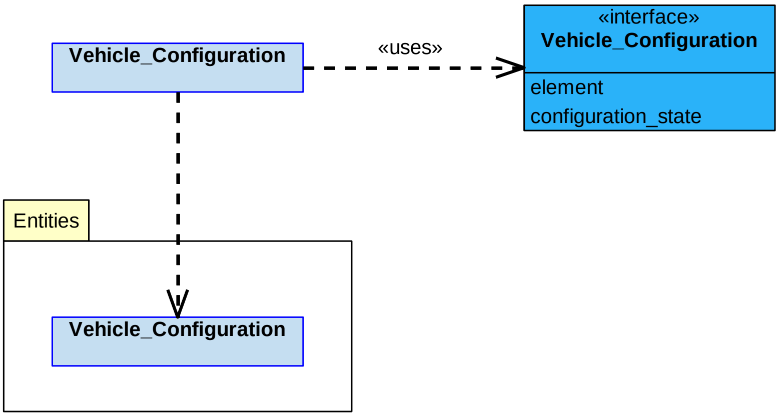


Figure 1232: Vehicle_Configuration Service Definition

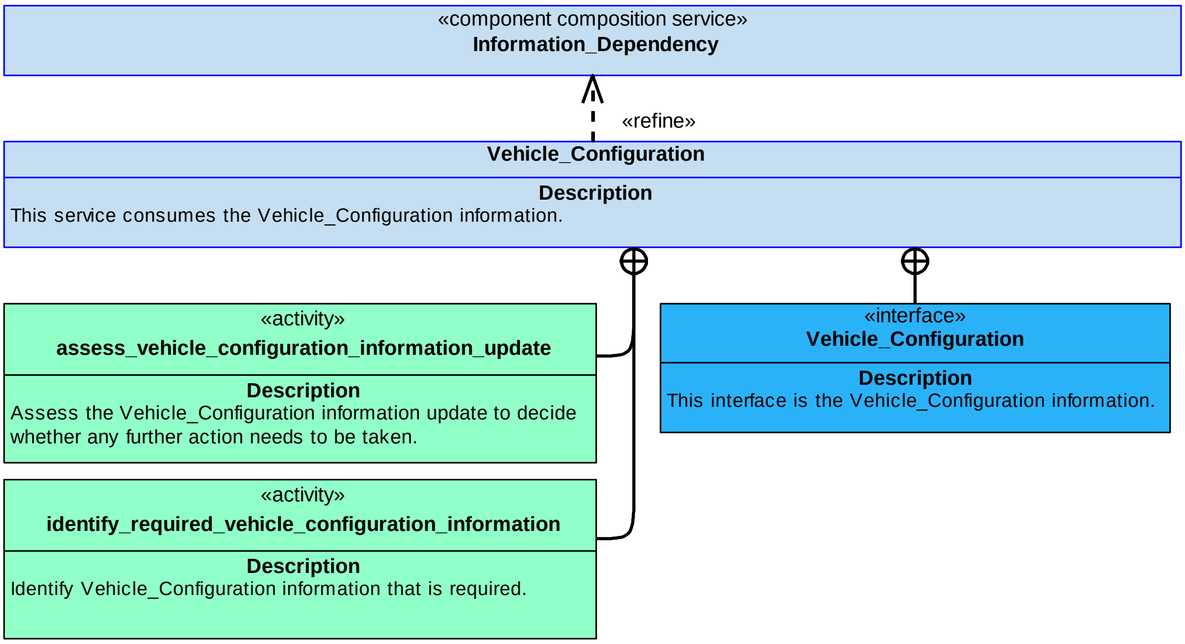

«component composition service»
Information_Dependency
«refine»
Vehicle_Configuration
Description
This service consumes the Vehicle_Configuration information.
«activity»
assess_vehicle_configuration_information_update
Description
Assess the Vehicle_Configuration information update to decide whether any further action needs to be taken.
«interface»
Vehicle_Configuration
Description
This interface is the Vehicle_Configuration information.
«activity»
identify_required_vehicle_configuration_information
Description
Identify Vehicle_Configuration information that is required.

Figure 1233: Vehicle_Configuration Service Policy

Vehicle_Configuration

This service consumes the Vehicle_Configuration information.

Interface

Vehicle_Configuration

This interface is the Vehicle_Configuration information.

Attributes

element	A part of the vehicle that is configurable (e.g. the undercarriage or flaps).
configuration_state	The state of the element (e.g. up or down, open or closed, or the degree of extension).

Activities

assess_vehicle_configuration_information_update

Assess the Vehicle_Configuration information update to decide whether any further action needs to be taken.

identify_required_vehicle_configuration_information

Identify Vehicle_Configuration information that is required.

5.4.2.71.7.1.6 External_Condition

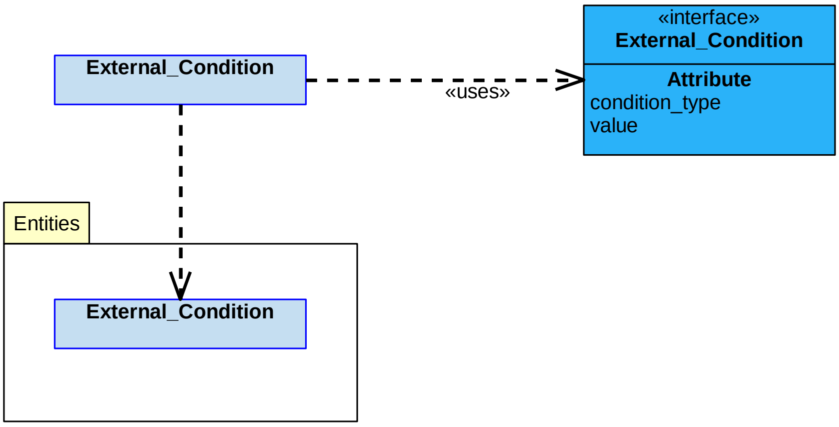


Figure 1234: External_Condition Service Definition

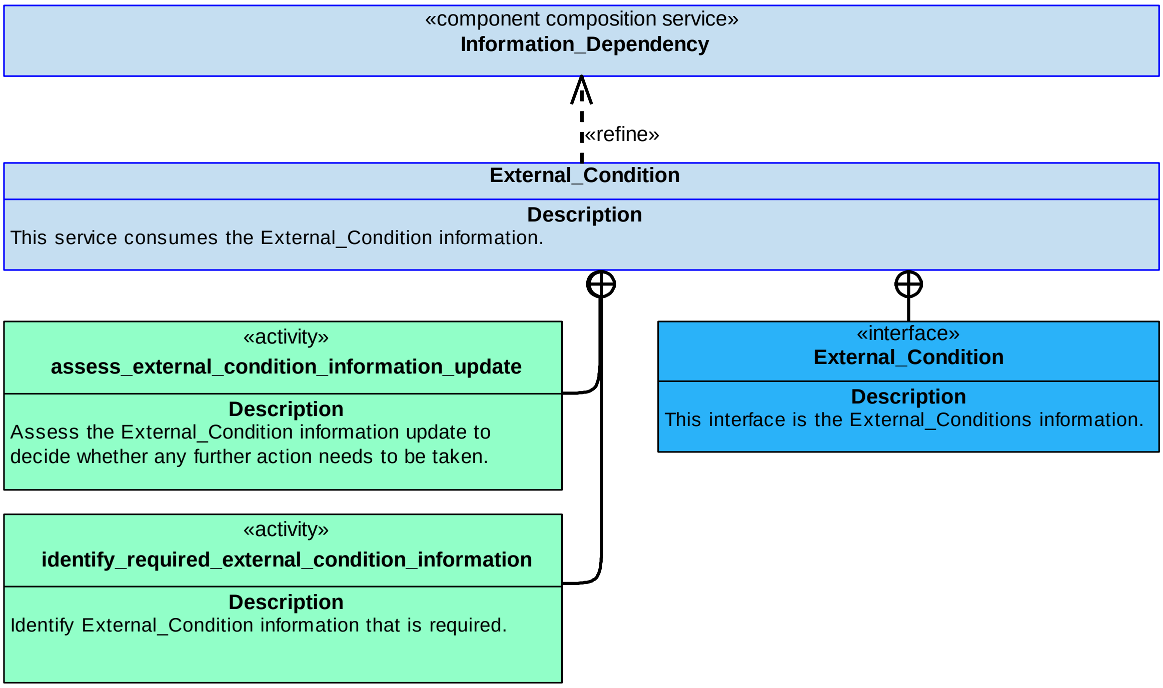

«component composition service»
Information_Dependency
«refine»
External_Condition
Description
This service consumes the External_Condition information.
«activity»
assess_external_condition_information_update
Description
Assess the External_Condition information update to decide whether any further action needs to be taken.
«activity»
identify_required_external_condition_information
Description
Identify External_Condition information that is required.
«interface»
External_Condition
Description
This interface is the External_Conditions information.

Figure 1235: External_Condition Service Policy

External_Condition

This service consumes the External_Condition information.

Interface

External_Condition

This interface is the External_Conditions information.

Attributes

condition_type	The type of condition that affects the calculation of performance data (e.g. pressure altitude, wind speed, or runway length).
value	The value associated with the condition.

Activities

assess_external_condition_information_update

Assess the External_Condition information update to decide whether any further action needs to be taken.

identify_required_external_condition_information

Identify External_Condition information that is required.

5.4.2.71.7.1.7 Performance_Envelope

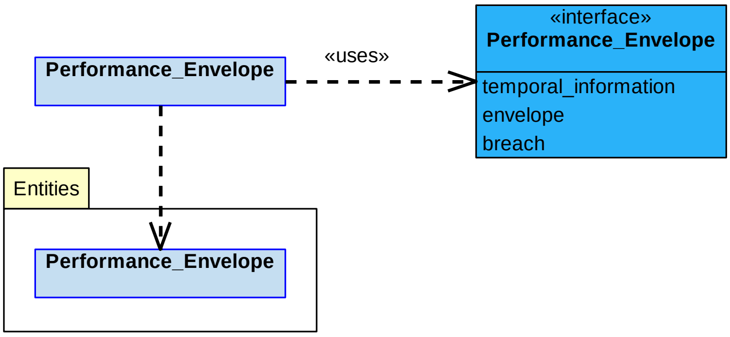


Figure 1236: Performance_Envelope Service Definition

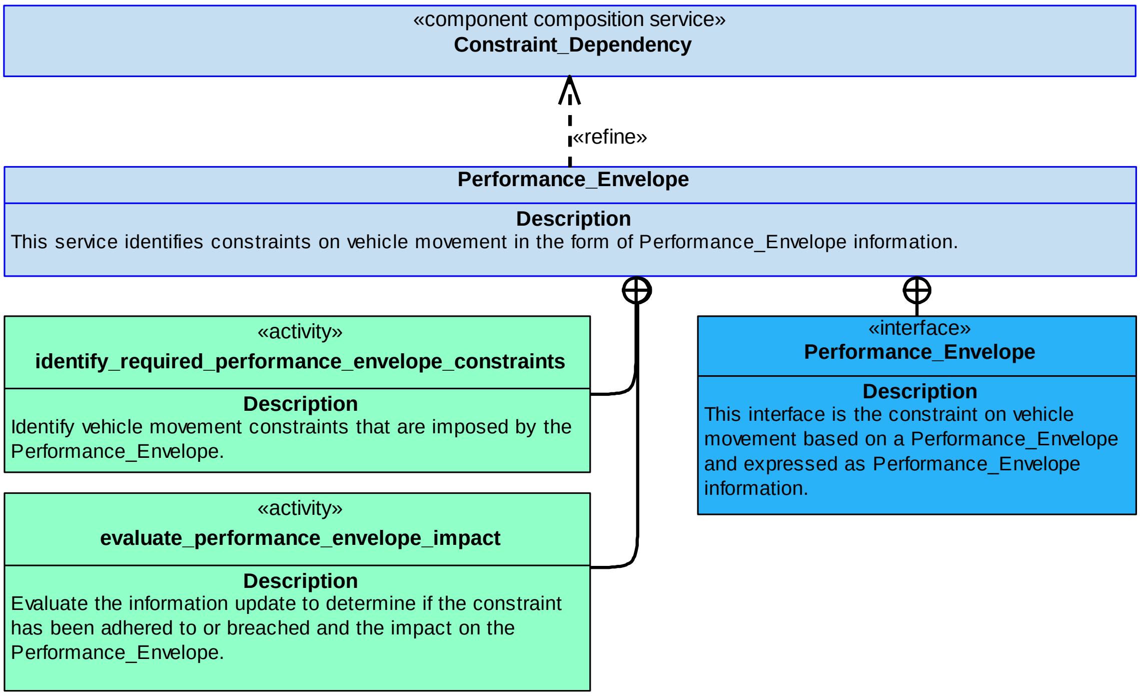

«component composition service»
Constraint_Dependency
«refine»
Performance_Envelope
Description
This service identifies constraints on vehicle movement in the form of Performance_Envelope information.
«activity»
identify_required_performance_envelope_constraints
Description
Identify vehicle movement constraints that are imposed by the Performance_Envelope.
«activity»
evaluate_performance_envelope_impact
Description
Evaluate the information update to determine if the constraint has been adhered to or breached and the impact on the Performance_Envelope.
«interface»
Performance_Envelope
Description
This interface is the constraint on vehicle movement based on a Performance_Envelope and expressed as Performance_Envelope information.

Figure 1237: Performance_Envelope Service Policy

Performance_Envelope

This service identifies constraints on vehicle movement in the form of Performance_Envelope information.

Interface

Performance_Envelope

This interface is the constraint on vehicle movement based on a Performance_Envelope and expressed as Performance_Envelope information.

Attributes

temporal_information	Information covering timing, such as start and end times, e.g. the period over which the Performance_Envelope applies.
envelope	Information defining the specific parameter and its upper and lower limits.
breach	Information on whether the existing performance parameters are outside the safe performance envelope, or are likely to be outside of the safe performance envelope if enforced.

Activities

evaluate_performance_envelope_impact

Evaluate the information update to determine if the constraint has been adhered to or breached and the impact on the Performance_Envelope.

identify_required_performance_envelope_constraints

Identify vehicle movement constraints that are imposed by the Performance_Envelope.

5.4.2.71.7.1.8 Capability

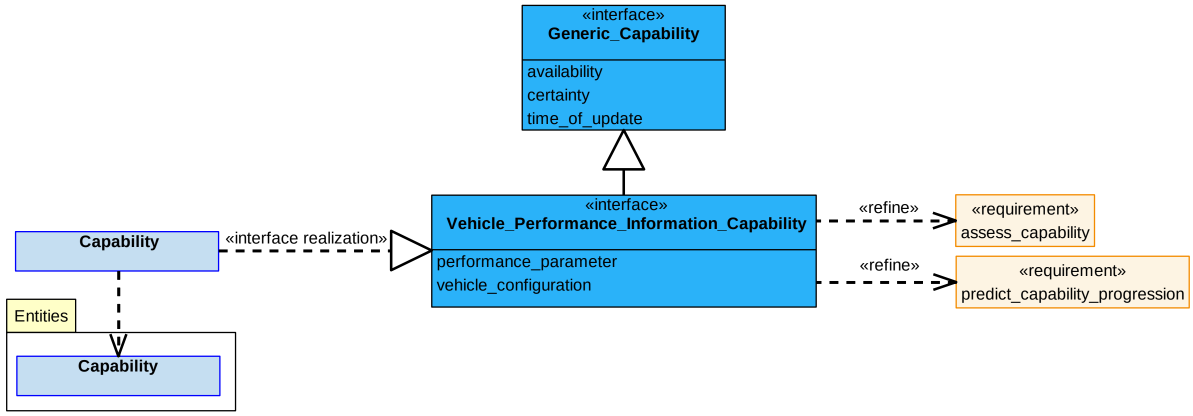
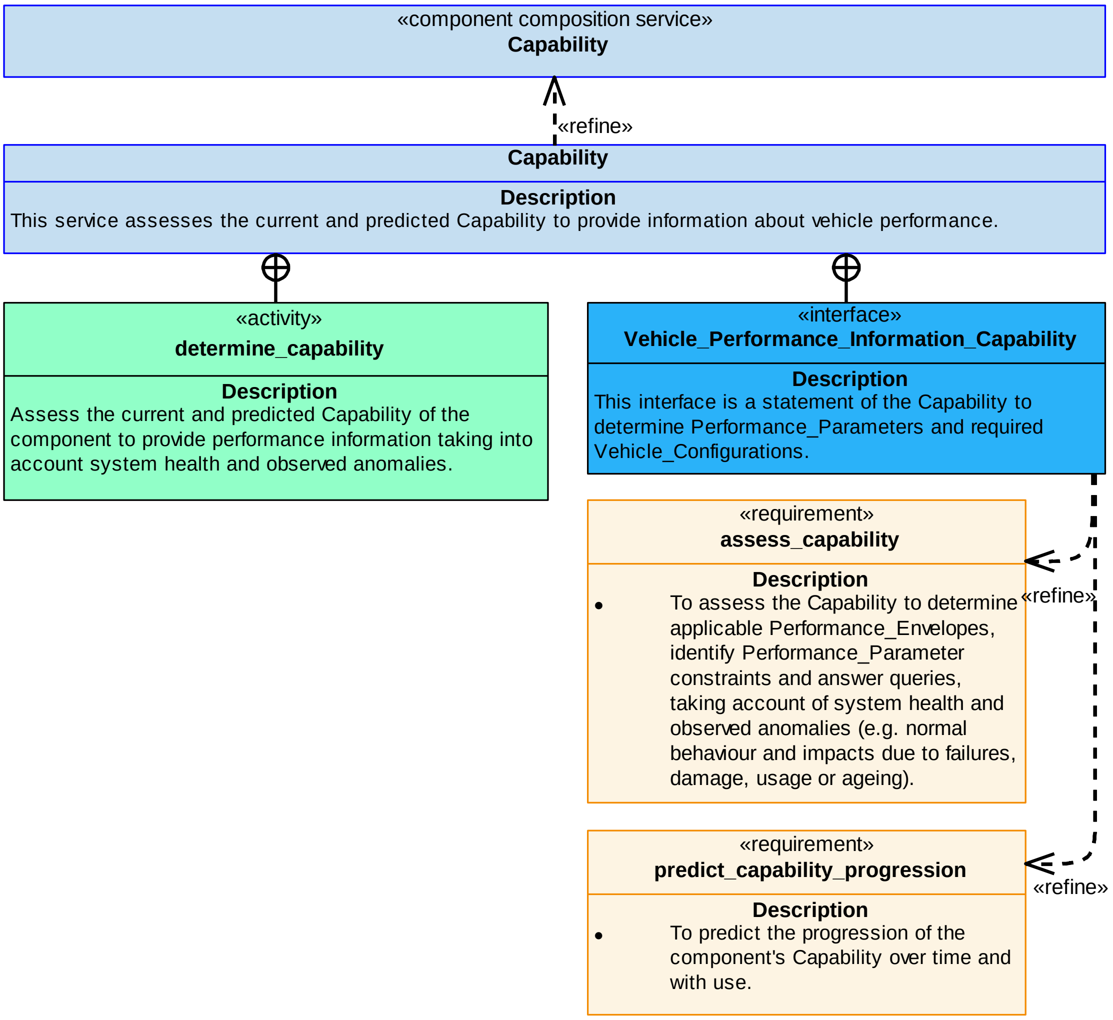


Figure 1238: Capability Service Definition



Description
This service assesses the current and predicted Capability to provide information about vehicle performance.

Description
Assess the current and predicted Capability of the component to provide performance information taking into account system health and observed anomalies.

Description
This interface is a statement of the Capability to determine Performance_Parameters and required Vehicle_Configurations.

Description
- To assess the Capability to determine applicable Performance_Envelopes, identify Performance_Parameter constraints and answer queries, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

Description
- To predict the progression of the component's Capability over time and with use.

Figure 1239: Capability Service Policy

Capability

This service assesses the current and predicted Capability to provide information about vehicle performance.

Interface

Vehicle_Performance_Information_Capability

This interface is a statement of the Capability to determine Performance_Parameters and required Vehicle_Configurations.

Attributes

performance_parameter	The property of vehicle performance that can be determined and provided (e.g. minimum speed).
vehicle_configuration	A particular Vehicle_Configuration that can be determined and provided.

Activity

determine_capability

Assess the current and predicted Capability of the component to provide performance information taking into account system health and observed anomalies.

5.4.2.71.7.1.9 Capability_Evidence

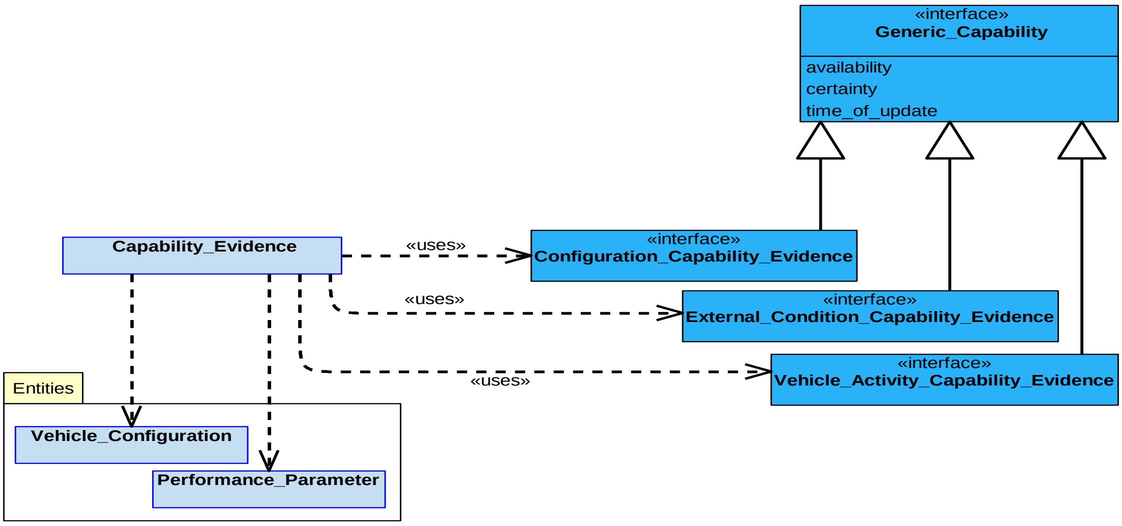


Figure 1240: Capability_Evidence Service Definition

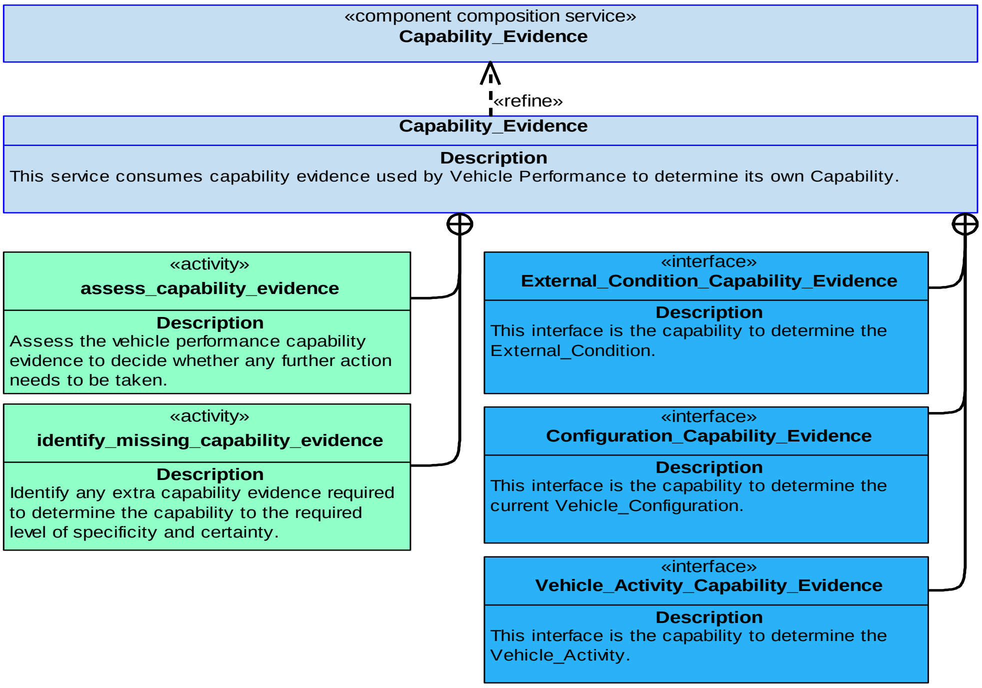

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes capability evidence used by Vehicle Performance to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the vehicle performance capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.
«interface»
External_Condition_Capability_Evidence
Description
This interface is the capability to determine the External_Condition.
«interface»
Configuration_Capability_Evidence
Description
This interface is the capability to determine the current Vehicle_Configuration.
«interface»
Vehicle_Activity_Capability_Evidence
Description
This interface is the capability to determine the Vehicle_Activity.

Figure 1241: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability evidence used by Vehicle Performance to determine its own Capability.

Interfaces

Configuration_Capability_Evidence

This interface is the capability to determine the current Vehicle_Configuration.

External_Condition_Capability_Evidence

This interface is the capability to determine the External_Condition.

Vehicle_Activity_Capability_Evidence

This interface is the capability to determine the Vehicle_Activity.

Activities

assess_capability_evidence

Assess the vehicle performance capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the capability to the required level of specificity and certainty.

5.4.2.71.7.2 Service Dependencies

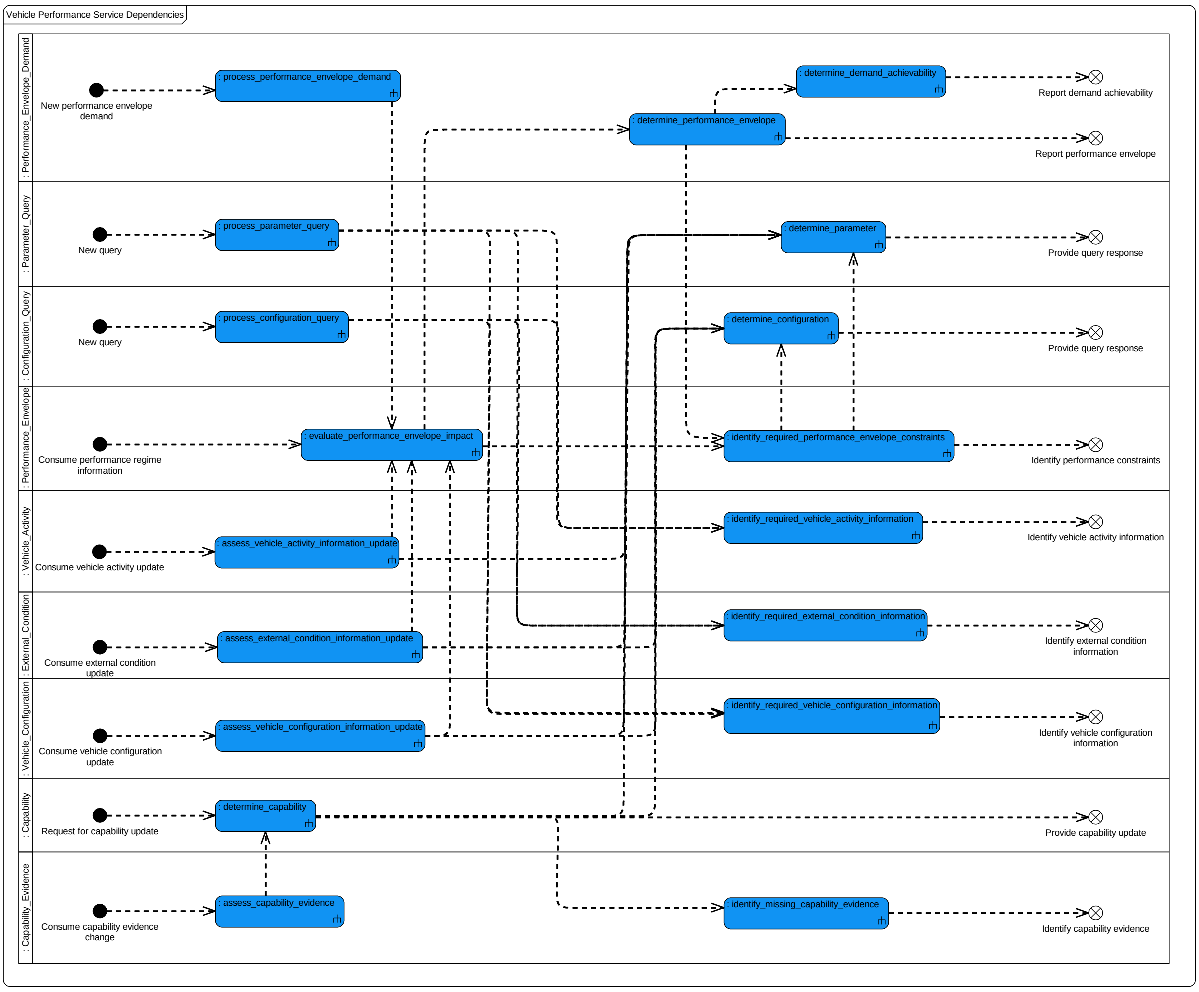

Vehicle Performance Service Dependencies
: Performance_Envelope_Demand
: Parameter_Query
: Configuration_Query
: Performance_Envelope
: Vehicle_Activity
: External_Condition
: Vehicle_Configuration
: Capability
: Capability_Evidence
New performance envelope demand
: process_performance_envelope_demand
: determine_demand_achievability
Report demand achievability
: determine_performance_envelope
Report performance envelope
New query
: process_parameter_query
: determine_parameter
Provide query response
New query
: process_configuration_query
: determine_configuration
Provide query response
Consume performance regime information
: evaluate_performance_envelope_impact
: identify_required_performance_envelope_constraints
Identify performance constraints
Consume vehicle activity update
: assess_vehicle_activity_information_update
: identify_required_vehicle_activity_information
Identify vehicle activity information
Consume external condition update
: assess_external_condition_information_update
: identify_required_external_condition_information
Identify external condition information
Consume vehicle configuration update
: assess_vehicle_configuration_information_update
: identify_required_vehicle_configuration_information
Identify vehicle configuration information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1242: Vehicle Performance Service Dependencies

5.4.2.72 Vehicle Stability and Control

5.4.2.72.1 Role

The role of Vehicle Stability and Control is to determine control effector commands to achieve vehicle control requirements whilst ensuring vehicle stability.

5.4.2.72.2 Overview

Control Architecture

Vehicle Stability and Control is an action component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

When a request to control the attitude or velocity of a Vehicle is received, in the form of a Vehicle_Control_Requirement, Vehicle Stability and Control will determine Control_Effector_Commands that fulfil the Vehicle_Control_Requirement whilst remaining within current Stability_Control_Limits. Vehicle Stability and Control will then issue the Control_Effector_Commands onto Control_Effectors that will control the Vehicle. In this way, Vehicle Stability and Control is part of the "inner loops" of a traditional flight control system.

Examples of Use

- Vehicle Stability and Control will be used as part of a system using electronic Vehicle guidance interfaces, such as a flight control system.

5.4.2.72.3 Service Summary

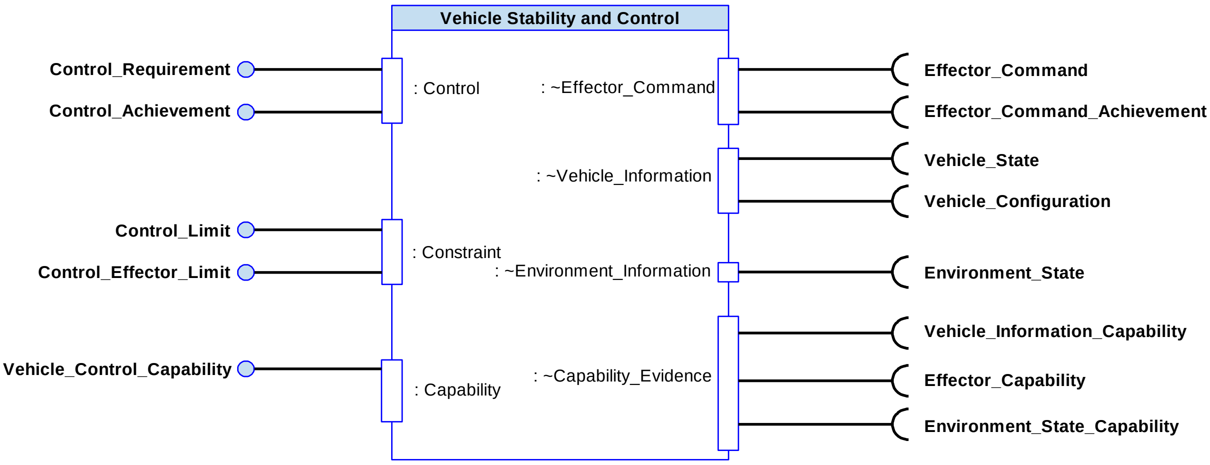


Figure 1243: Vehicle Stability and Control Service Summary

5.4.2.72.4 Responsibilities

capture_vehicle_control_requirements

- To capture Vehicle_Control_Requirements (e.g. attitude or velocity).

capture_limits

- To capture given Stability_Control_Limits.

identify_whether_requirement_remains_achievable

- To identify whether a Vehicle_Control_Requirement is still achievable given current or predicted Capability and Stability_Control_Limits.

determine_control_effector_commands

- To determine Control_Effector_Commands (e.g. control surface movement or thrust demand) to achieve Vehicle_Control_Requirements.

fulfil_requirement

- To fulfil a Vehicle_Control_Requirement by executing the planned Control_Effector_Commands.

assess_capability

- To assess the Capability to control the Vehicle, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage, or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of the Capability assessment.

predict_capability_progression

- To predict the progression of the component's Capability over time and with use.

5.4.2.72.5 Subject Matter Semantics

The subject matter of Vehicle Stability and Control is the attitude and velocity of the Vehicle, and the Control_Effectors that can be used to control them.

Exclusions

The subject matter of Vehicle Stability and Control does not include:

- The Vehicle trajectory demands, or the translation of the trajectory demands into attitude and velocity demands, also called the "outer loops".

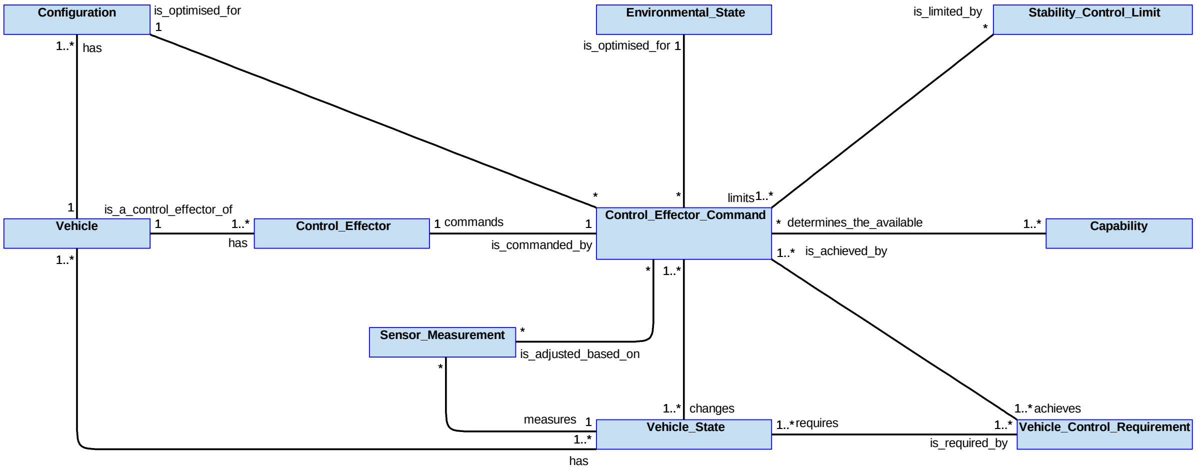


Figure 1244: Vehicle Stability and Control Semantics

5.4.2.72.5.1 Entities

Capability

The capability of controlling the attitude and velocity of the Vehicle taking account of system health and observed anomalies.

Configuration

The configuration of the Vehicle (e.g. bay doors open or the current centre of gravity).

Control_Effector

A controllable device which causes a Vehicle to move or cease to move. This may be either a control surface, a dedicated propulsion device or a dedicated breaking device.

Control_Effector_Command

A command to change the state of a Control_Effector (e.g. control surface movement or thrust demand).

Stability_Control_Limit

The stability control limits that all of Control_Effector_Commands have to conform to, either determined by this component or externally imposed.

Vehicle_State

The state (e.g. velocity or attitude) of the Vehicle.

Vehicle

The object whose attitude and velocity are to be controlled.

Vehicle_Control_Requirement

A requirement to control the attitude or velocity of the Vehicle.

Environmental_State

The state of the environment around the Vehicle (e.g. air pressure or airspeed over lifting surfaces).

Sensor_Measurement

Sensor measurements of the Vehicle_State used by the Control_Effector_Command to achieve the intended command.

5.4.2.72.6 Design Rationale

5.4.2.72.6.1 Assumptions

- This component will be used in 'real-time' control of a platform rather than in a planning role.

5.4.2.72.6.2 Design Considerations

Related PYRAMID Concepts

- No PYRAMID concepts were specifically taken into account in defining Vehicle Stability and Control.

Extensions

- The use of extensions for the Vehicle Stability and Control component is not considered appropriate, as the functionality is highly dependent on the particular control system hardware.

Exploitation Considerations

- The knowledge of how the Vehicle's Control_Effectors can be used to provide control of attitude and velocity resides within Vehicle Stability and Control (e.g. it understands the relationships between a number of Control_Effectors that together move a control surface).
- It is likely a new component would be developed for each Vehicle type.
- It is likely that this component would be receiving Vehicle_Control_Requirement updates every few milliseconds, and therefore any achievement reporting done by the component is probably more useful when it can’t fulfil a requirement, rather than when it can.
- Vehicle Stability and Control must always have a Vehicle_Control_Requirement to meet. It is up to the Exploiting Platform to determine how this is achieved, for example, by defining a default requirement to be met if no input is present.

5.4.2.72.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Failure of this component may cause uncontrolled flight of the air vehicle. This could lead to loss of structural integrity of the air vehicle and / or an uncontrolled crash. The result is likely to be loss of the air vehicle and fatalities.

5.4.2.72.6.4 Security Considerations

The indicative security classification is SNEO.

This component represents the inner loop of vehicle control and as such has knowledge of flight control/performance and limitations of the Exploiting Platform, which suggests it will be SNEO, although it may be possible to declassify using relative data.

This would need protecting to ensure that its data is kept secret (confidentiality) and to ensure that it will control the vehicle in the manner intended (integrity) when required (availability).

The component is expected to at least partially satisfy security related functions by:

- **Maintaining Audit Records** of flight control demands placed during a mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.
- Performing **System Status and Monitoring** of deviations from expected operation; any unexplainable behaviour may indicate the component has been compromised by a cyber attack.
- Generation of **Warnings and Notifications** that may provide awareness of unexpected behaviour of vehicle controls and therefore possible cyber attack.

The component is considered unlikely to directly implement security enforcing functions, although it is dependent on the integrity of its inputs.

5.4.2.72.7 Services

5.4.2.72.7.1 Service Definitions

5.4.2.72.7.1.1 Control

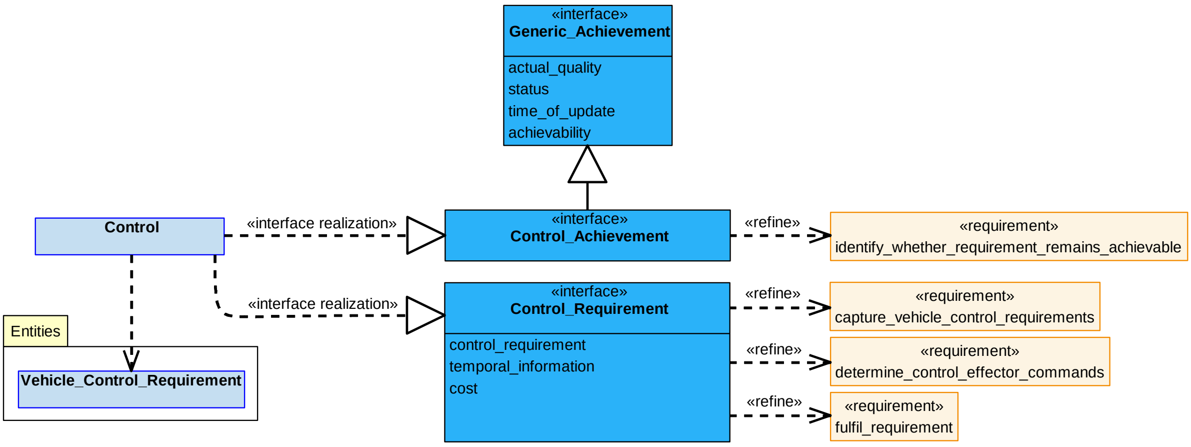


Figure 1245: Control Service Definition

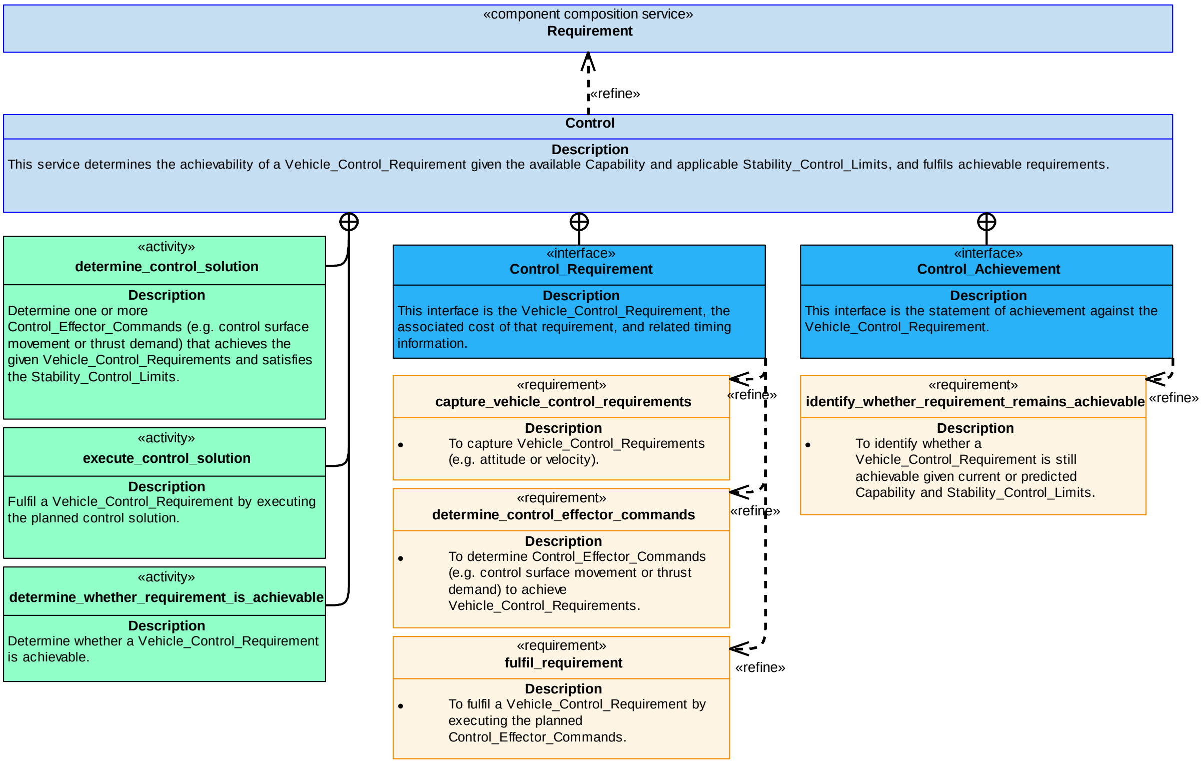

«component composition service»
Requirement
«refine»
Control
Description
This service determines the achievability of a Vehicle_Control_Requirement given the available Capability and applicable Stability_Control_Limits, and fulfils achievable requirements.
«activity»
determine_control_solution
Description
Determine one or more Control_Effector_Commands (e.g. control surface movement or thrust demand) that achieves the given Vehicle_Control_Requirements and satisfies the Stability_Control_Limits.
«activity»
execute_control_solution
Description
Fulfil a Vehicle_Control_Requirement by executing the planned control solution.
«activity»
determine_whether_requirement_is_achievable
Description
Determine whether a Vehicle_Control_Requirement is achievable.
«interface»
Control_Requirement
Description
This interface is the Vehicle_Control_Requirement, the associated cost of that requirement, and related timing information.
«requirement»
capture_vehicle_control_requirements
Description
To capture Vehicle_Control_Requirements (e.g. attitude or velocity).
«refine»
«requirement»
determine_control_effector_commands
Description
To determine Control_Effector_Commands (e.g. control surface movement or thrust demand) to achieve Vehicle_Control_Requirements.
«refine»
«requirement»
fulfil_requirement
Description
To fulfil a Vehicle_Control_Requirement by executing the planned Control_Effector_Commands.
«refine»
«interface»
Control_Achievement
Description
This interface is the statement of achievement against the Vehicle_Control_Requirement.
«requirement»
identify_whether_requirement_remains_achievable
Description
To identify whether a Vehicle_Control_Requirement is still achievable given current or predicted Capability and Stability_Control_Limits.
«refine»

Figure 1246: Control Service Policy

Control

This service determines the achievability of a Vehicle_Control_Requirement given the available Capability and applicable Stability_Control_Limits, and fulfils achievable requirements.

Interfaces

Control_Requirement

This interface is the Vehicle_Control_Requirement, the associated cost of that requirement, and related timing information.

Attributes

control_requirement	The requirement to control the attitude or velocity of the Vehicle.
temporal_information	Information covering timing, such as start and end times.
cost	The cost of achieving the Vehicle_Control_Requirement (e.g. Control_Effectors used or time taken).

Control_Achievement

This interface is the statement of achievement against the Vehicle_Control_Requirement.

Activities

determine_control_solution

Determine one or more Control_Effector_Commands (e.g. control surface movement or thrust demand) that achieves the given Vehicle_Control_Requirements and satisfies the Stability_Control_Limits.

execute_control_solution

Fulfil a Vehicle_Control_Requirement by executing the planned control solution.

determine_whether_requirement_is_achievable

Determine whether a Vehicle_Control_Requirement is achievable.

5.4.2.72.7.1.2 Effector_Command

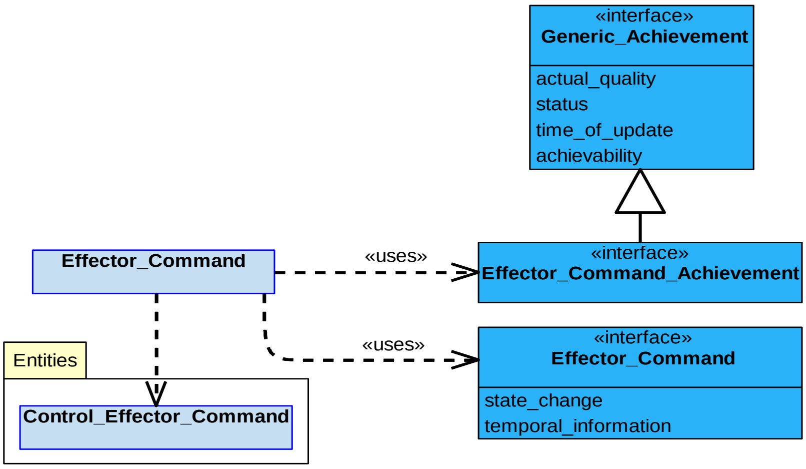


Figure 1247: Effector_Command Service Definition

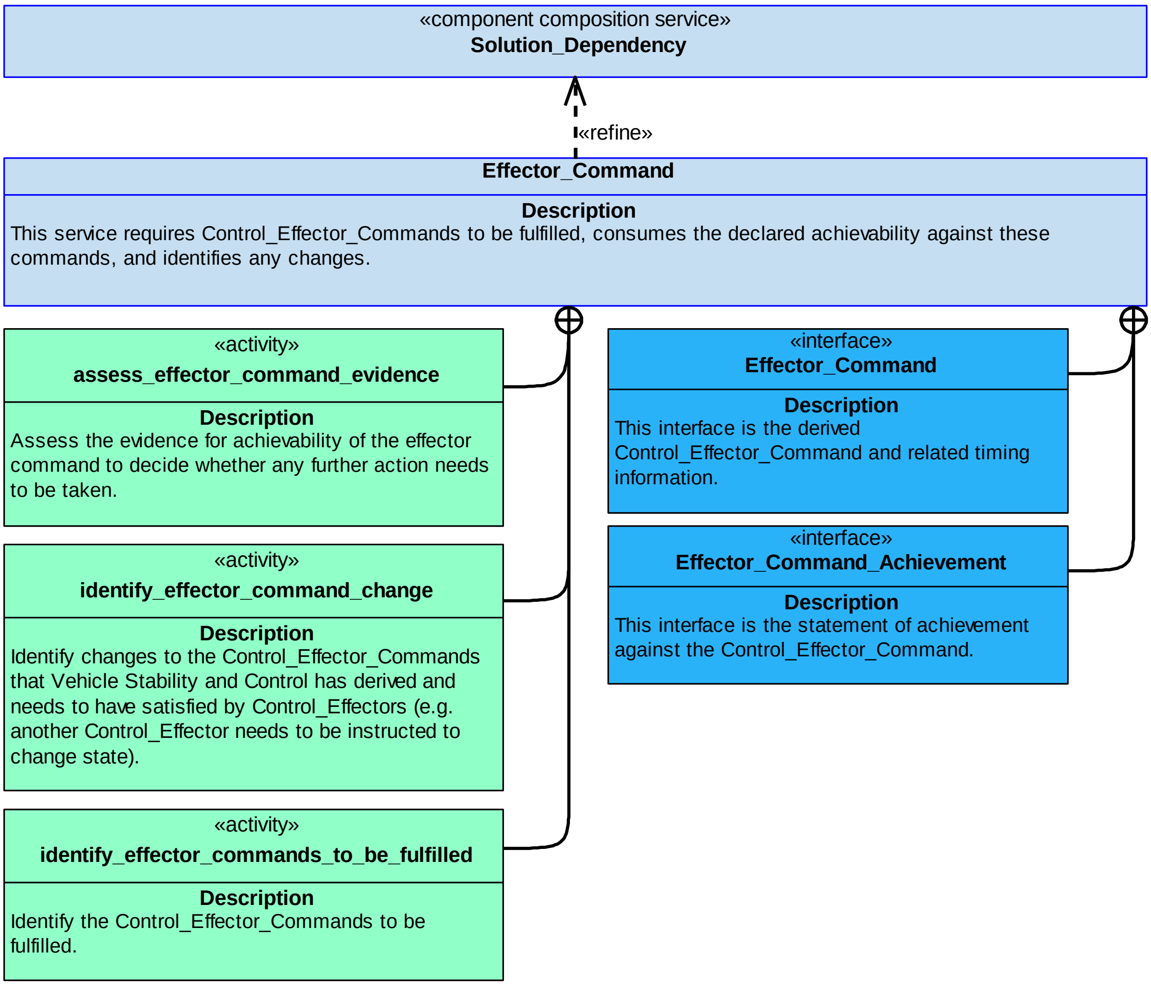

«component composition service»
Solution_Dependency
«refine»
Effector_Command
Description
This service requires Control_Effector_Commands to be fulfilled, consumes the declared achievability against these commands, and identifies any changes.
«activity»
assess_effector_command_evidence
Description
Assess the evidence for achievability of the effector command to decide whether any further action needs to be taken.
«activity»
identify_effector_command_change
Description
Identify changes to the Control_Effector_Commands that Vehicle Stability and Control has derived and needs to have satisfied by Control_Effectors (e.g. another Control_Effector needs to be instructed to change state).
«activity»
identify_effector_commands_to_be_fulfilled
Description
Identify the Control_Effector_Commands to be fulfilled.
«interface»
Effector_Command
Description
This interface is the derived Control_Effector_Command and related timing information.
«interface»
Effector_Command_Achievement
Description
This interface is the statement of achievement against the Control_Effector_Command.

Figure 1248: Effector_Command Service Policy

Effector_Command

This service requires Control_Effector_Commands to be fulfilled, consumes the declared achievability against these commands, and identifies any changes.

Interfaces

Effector_Command

This interface is the derived Control_Effector_Command and related timing information.

Attributes

state_change	The derived requirement for a state change in one or more Control_Effectors (e.g. control surface movement).
temporal_information	Information covering timing, such as start and end times.

Effector_Command_Achievement

This interface is the statement of achievement against the Control_Effector_Command.

Activities

assess_effector_command_evidence

Assess the evidence for achievability of the effector command to decide whether any further action needs to be taken.

identify_effector_command_change

Identify changes to the Control_Effector_Commands that Vehicle Stability and Control has derived and needs to have satisfied by Control_Effectors (e.g. another Control_Effector needs to be instructed to change state).

identify_effector_commands_to_be_fulfilled

Identify the Control_Effector_Commands to be fulfilled.

5.4.2.72.7.1.3 Vehicle_Information

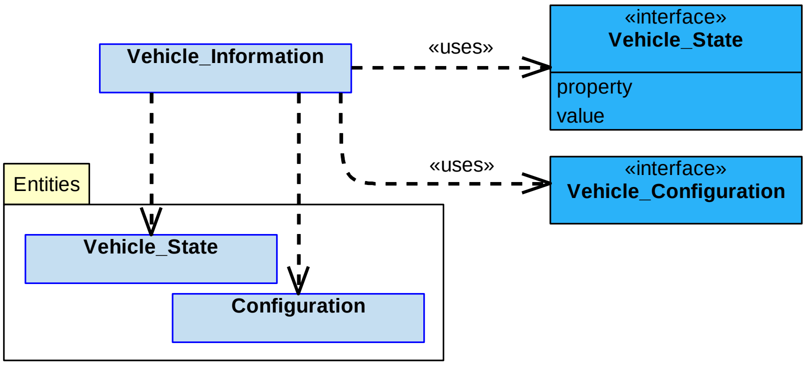


Figure 1249: Vehicle_Information Service Definition

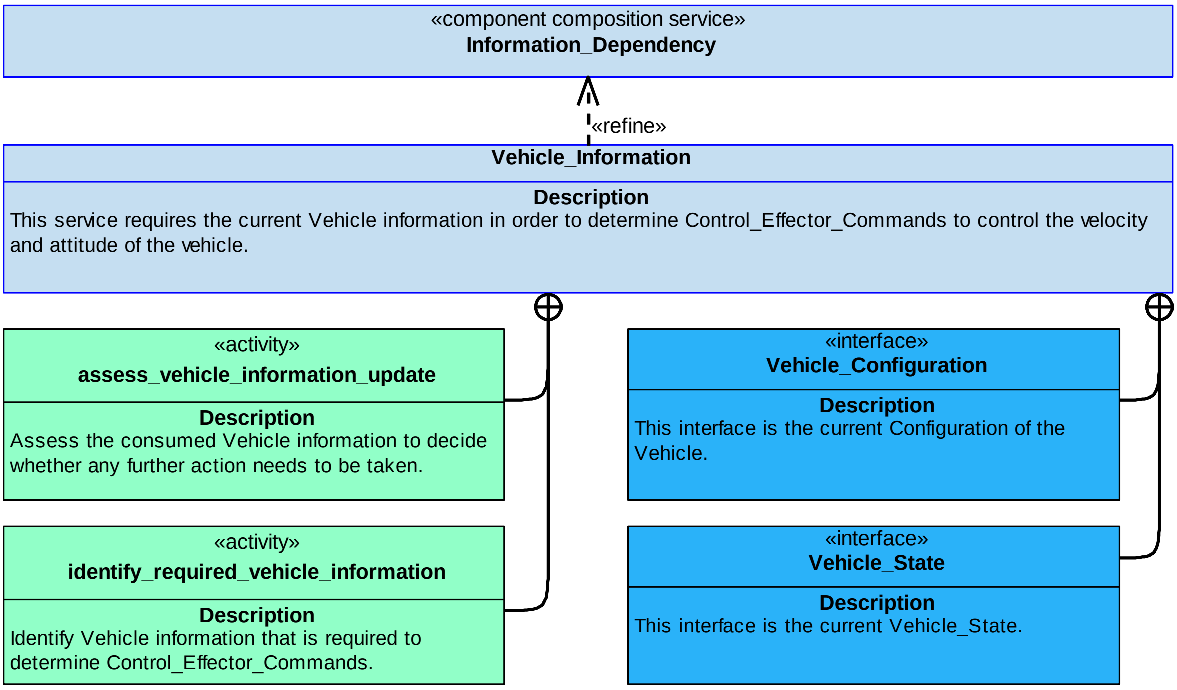

«component composition service»
Information_Dependency
«refine»
Vehicle_Information
Description
This service requires the current Vehicle information in order to determine Control_Effector_Commands to control the velocity and attitude of the vehicle.
«activity»
assess_vehicle_information_update
Description
Assess the consumed Vehicle information to decide whether any further action needs to be taken.
«activity»
identify_required_vehicle_information
Description
Identify Vehicle information that is required to determine Control_Effector_Commands.
«interface»
Vehicle_Configuration
Description
This interface is the current Configuration of the Vehicle.
«interface»
Vehicle_State
Description
This interface is the current Vehicle_State.

Figure 1250: Vehicle_Information Service Policy

Vehicle_Information

This service requires the current Vehicle information in order to determine Control_Effector_Commands to control the velocity and attitude of the vehicle.

<u>Interfaces</u>

Vehicle_State

This interface is the current Vehicle_State.

<u>Attributes</u>

property	The property relating to the Vehicle_State (e.g. velocity, pitch, or roll).
value	The value of the property.

Vehicle_Configuration

This interface is the current Configuration of the Vehicle.

<u>Activities</u>

assess_vehicle_information_update

Assess the consumed Vehicle information to decide whether any further action needs to be taken.

identify_required_vehicle_information

Identify Vehicle information that is required to determine Control_Effector_Commands.

5.4.2.72.7.1.4 Environment_Information

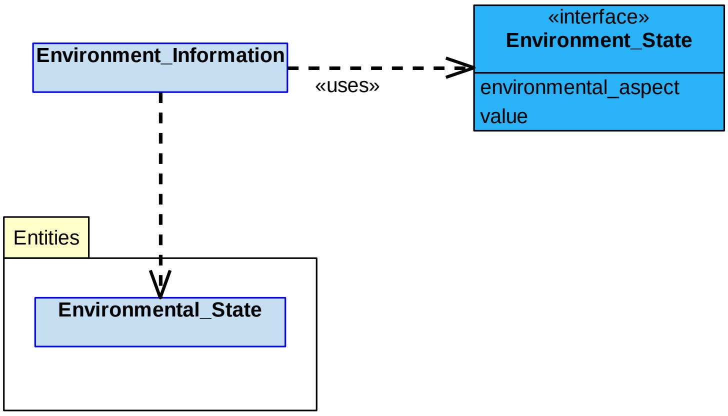


Figure 1251: Environment_Information Service Definition

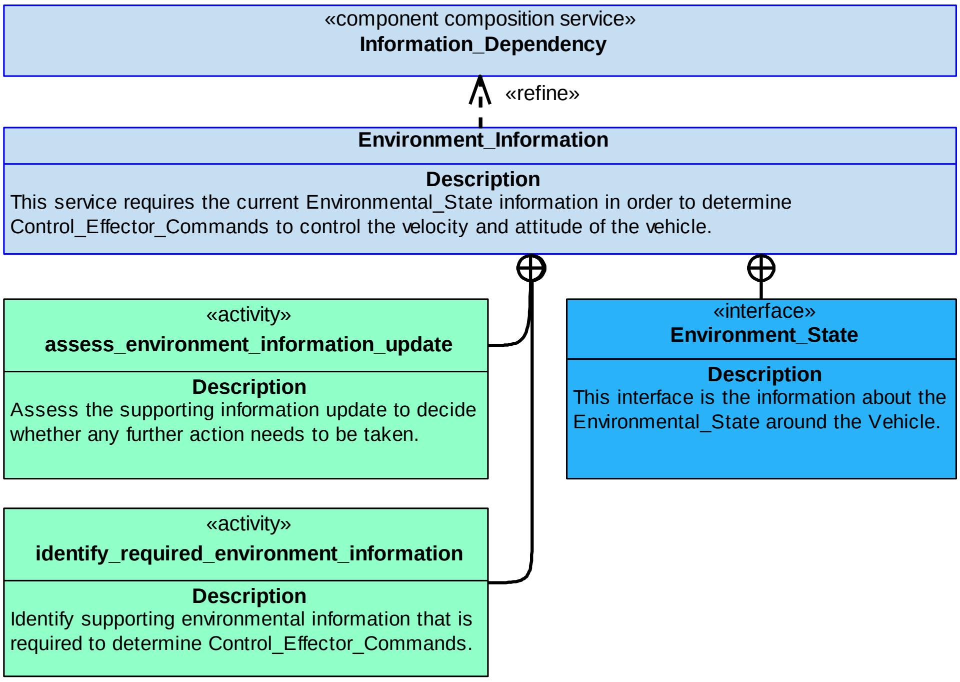

«component composition service»
Information_Dependency
«refine»
Environment_Information
Description
This service requires the current Environmental_State information in order to determine Control_Effector_Commands to control the velocity and attitude of the vehicle.
«activity»
assess_environment_information_update
Description
Assess the supporting information update to decide whether any further action needs to be taken.
«interface»
Environment_State
Description
This interface is the information about the Environmental_State around the Vehicle.
«activity»
identify_required_environment_information
Description
Identify supporting environmental information that is required to determine Control_Effector_Commands.

Figure 1252: Environment_Information Service Policy

Environment_Information

This service requires the current Environmental_State information in order to determine Control_Effector_Commands to control the velocity and attitude of the vehicle.

Interface

Environment_State

This interface is the information about the Environmental_State around the Vehicle.

Attributes

environmental_aspect	The specific type of Environmental_State information (e.g. the angle of attack of a specific lifting surface).
value	The value associated with an Environmental_State (e.g. an angle of attack of 2.5 degrees).

Activities

identify_required_environment_information

Identify supporting environmental information that is required to determine Control_Effector_Commands.

assess_environment_information_update

Assess the supporting information update to decide whether any further action needs to be taken.

5.4.2.72.7.1.5 Constraint

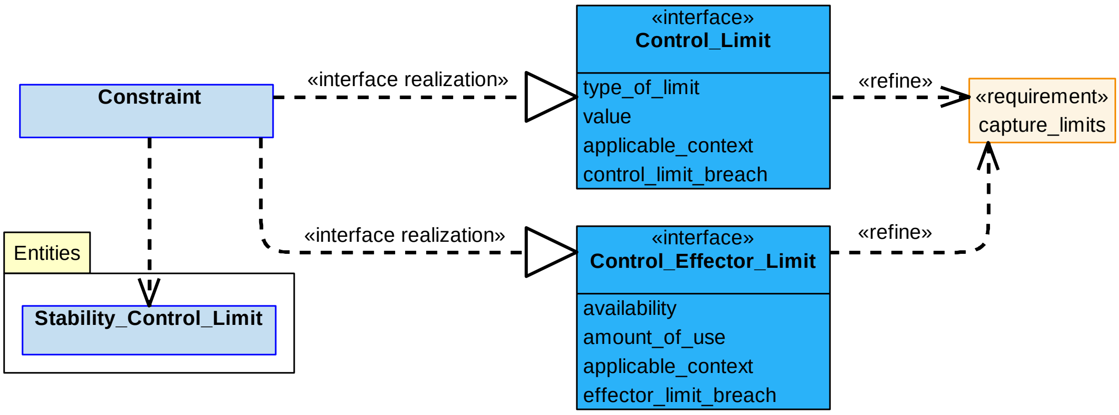
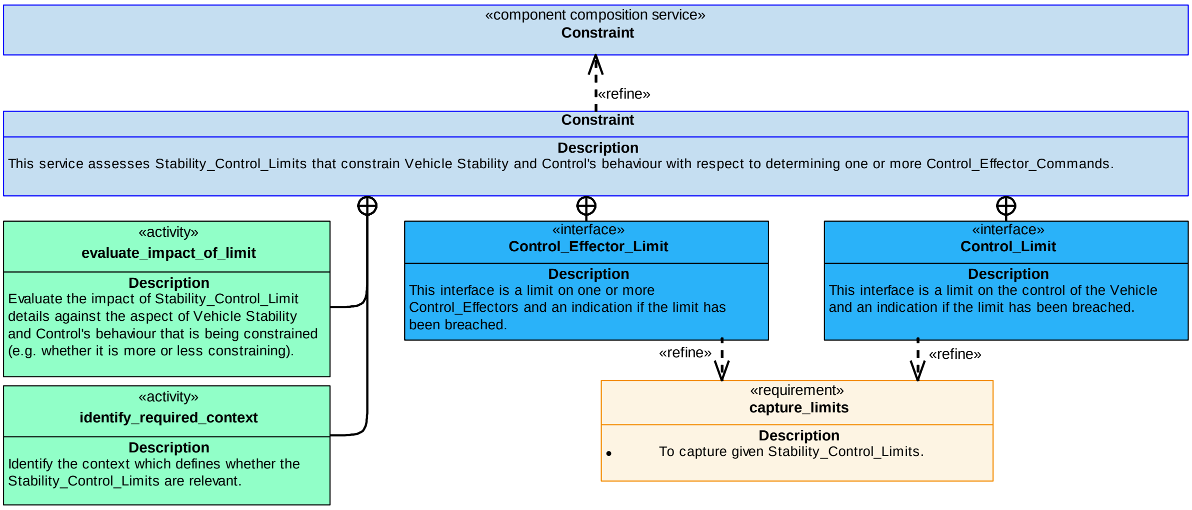


Figure 1253: Constraint Service Definition



This service assesses Stability_Control_Limits that constrain Vehicle Stability and Control's behaviour with respect to determining one or more Control_Effector_Commands.

Evaluate the impact of Stability_Control_Limit details against the aspect of Vehicle Stability and Control's behaviour that is being constrained (e.g. whether it is more or less constraining).

This interface is a limit on one or more Control_Effectors and an indication if the limit has been breached.

This interface is a limit on the control of the Vehicle and an indication if the limit has been breached.

- To capture given Stability_Control_Limits.

Identify the context which defines whether the Stability_Control_Limits are relevant.

Figure 1254: Constraint Service Policy

Constraint

This service assesses Stability_Control_Limits that constrain Vehicle Stability and Control's behaviour with respect to determining one or more Control_Effector_Commands.

Interfaces

Control_Limit

This interface is a limit on the control of the Vehicle and an indication if the limit has been breached.

Attributes

type_of_limit	The type of limit constraining Vehicle Stability and Control (e.g. roll rate and roll acceleration).
value	The value of the type of limit.
applicable_context	The context in which the limit is applicable.
control_limit_breach	A statement that the limit on the control of the vehicle has been breached.

Control_Effector_Limit

This interface is a limit on one or more Control_Effectors and an indication if the limit has been breached.

Attributes

availability	Whether a Control_Effector is allowed to be used.
amount_of_use	The utilisation of a particular Control_Effector (e.g. limitations on how much a control surface can be moved, such as what the limits of allowed positions of the control surface are).
applicable_context	The context in which the limit is applicable.
effector_limit_breach	A statement that the Control_Effector limit has been breached.

Activities

evaluate_impact_of_limit

Evaluate the impact of Stability_Control_Limit details against the aspect of Vehicle Stability and Control's behaviour that is being constrained (e.g. whether it is more or less constraining).

identify_required_context

Identify the context which defines whether the Stability_Control_Limits are relevant.

5.4.2.72.7.1.6 Capability

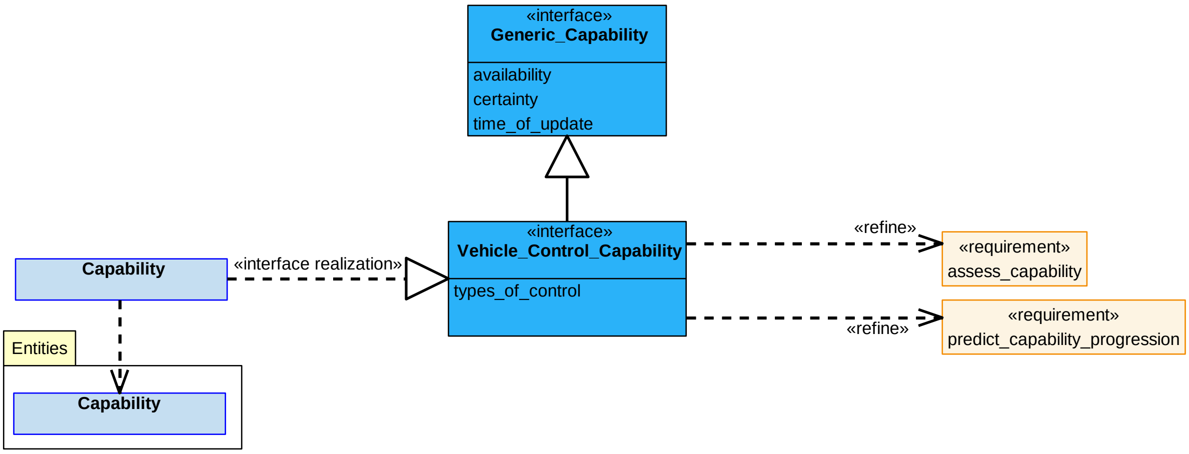
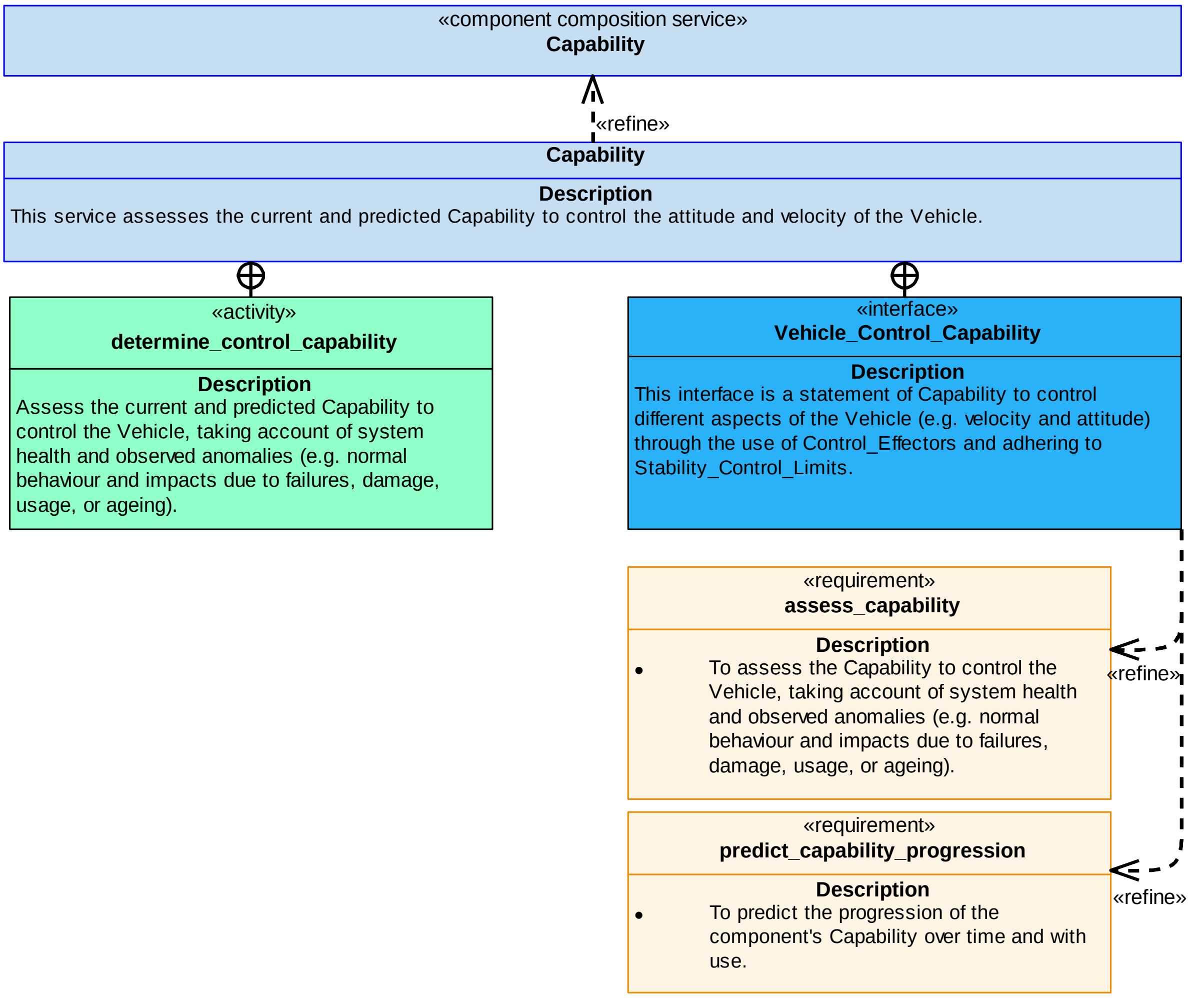


Figure 1255: Capability Service Definition


«component composition service»
Capability
«refine»
Capability
«activity»
determine_control_capability
«interface»
Vehicle_Control_Capability
«requirement»
assess_capability
«refine»
«requirement»
predict_capability_progression
«refine»

Description

This service assesses the current and predicted Capability to control the attitude and velocity of the Vehicle.

Description

Assess the current and predicted Capability to control the Vehicle, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage, or ageing).

Description

This interface is a statement of Capability to control different aspects of the Vehicle (e.g. velocity and attitude) through the use of Control_Effectors and adhering to Stability_Control_Limits.

Description

- To assess the Capability to control the Vehicle, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage, or ageing).

Description

- To predict the progression of the component's Capability over time and with use.

Figure 1256: Capability Service Policy

Capability

This service assesses the current and predicted Capability to control the attitude and velocity of the Vehicle.

Interface

Vehicle_Control_Capability

This interface is a statement of Capability to control different aspects of the Vehicle (e.g. velocity and attitude) through the use of Control_Effectors and adhering to Stability_Control_Limits.

Attribute

types_of_control	The different types of control that can be determined and issued (e.g. on ground control or in flight control).

Activity

determine_control_capability

Assess the current and predicted Capability to control the Vehicle, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage, or ageing).

5.4.2.72.7.1.7 Capability_Evidence

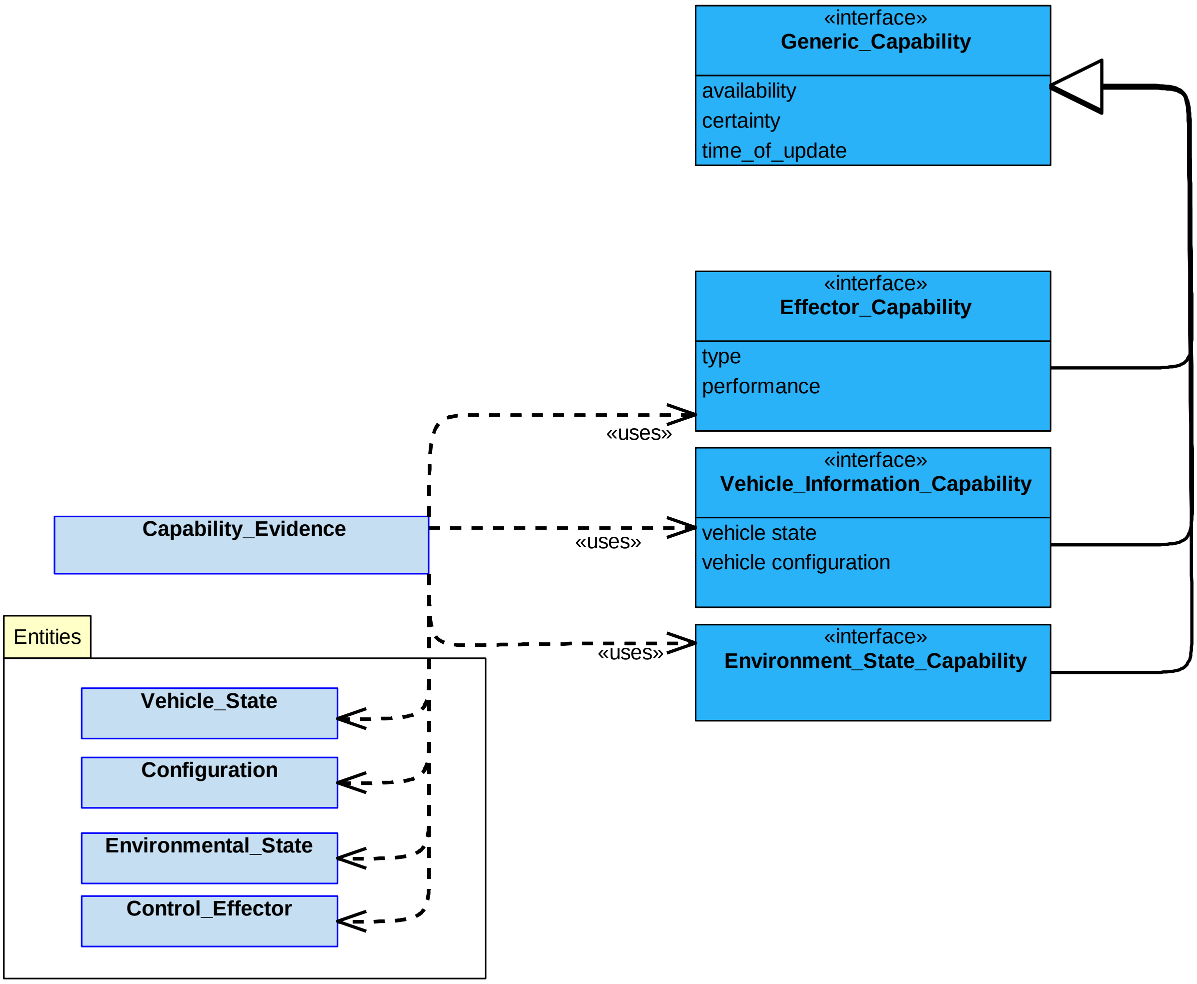


Figure 1257: Capability_Evidence Service Definition

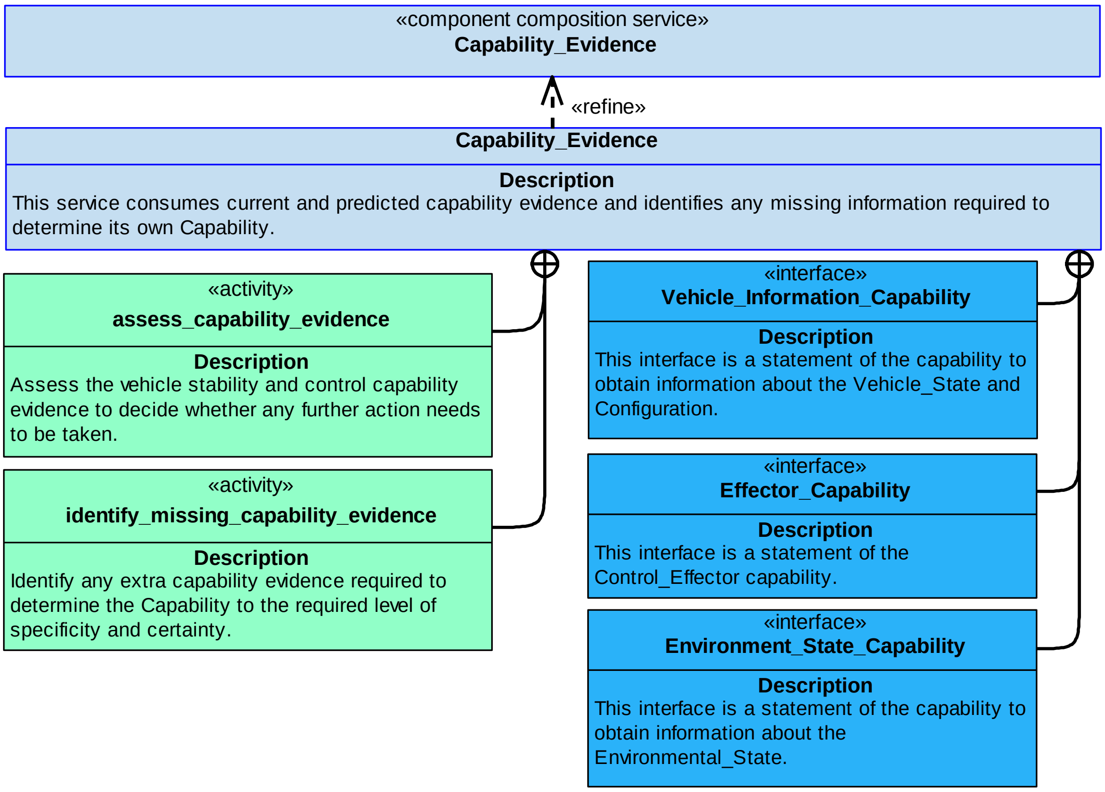

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes current and predicted capability evidence and identifies any missing information required to determine its own Capability.
«activity»
assess_capability_evidence
Description
Assess the vehicle stability and control capability evidence to decide whether any further action needs to be taken.
«activity»
identify_missing_capability_evidence
Description
Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.
«interface»
Vehicle_Information_Capability
Description
This interface is a statement of the capability to obtain information about the Vehicle_State and Configuration.
«interface»
Effector_Capability
Description
This interface is a statement of the Control_Effector capability.
«interface»
Environment_State_Capability
Description
This interface is a statement of the capability to obtain information about the Environmental_State.

Figure 1258: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability evidence and identifies any missing information required to determine its own Capability.

Interfaces

Effector_Capability

This interface is a statement of the Control_Effector capability.

Attributes

type	The type of Control_Effector the capability applies to.
performance	An indication of the available performance of a Control_Effector (e.g. range of movement).

Environment_State_Capability

This interface is a statement of the capability to obtain information about the Environmental_State.

Vehicle_Information_Capability

This interface is a statement of the capability to obtain information about the Vehicle_State and Configuration.

Attributes

vehicle state	An indication of the type of information about the Vehicle_State.

vehicle configuration	An indication of the type of information about the vehicle Configuration.

Activities

assess_capability_evidence

Assess the vehicle stability and control capability evidence to decide whether any further action needs to be taken.

identify_missing_capability_evidence

Identify any extra capability evidence required to determine the Capability to the required level of specificity and certainty.

5.4.2.72.7.2 Service Dependencies

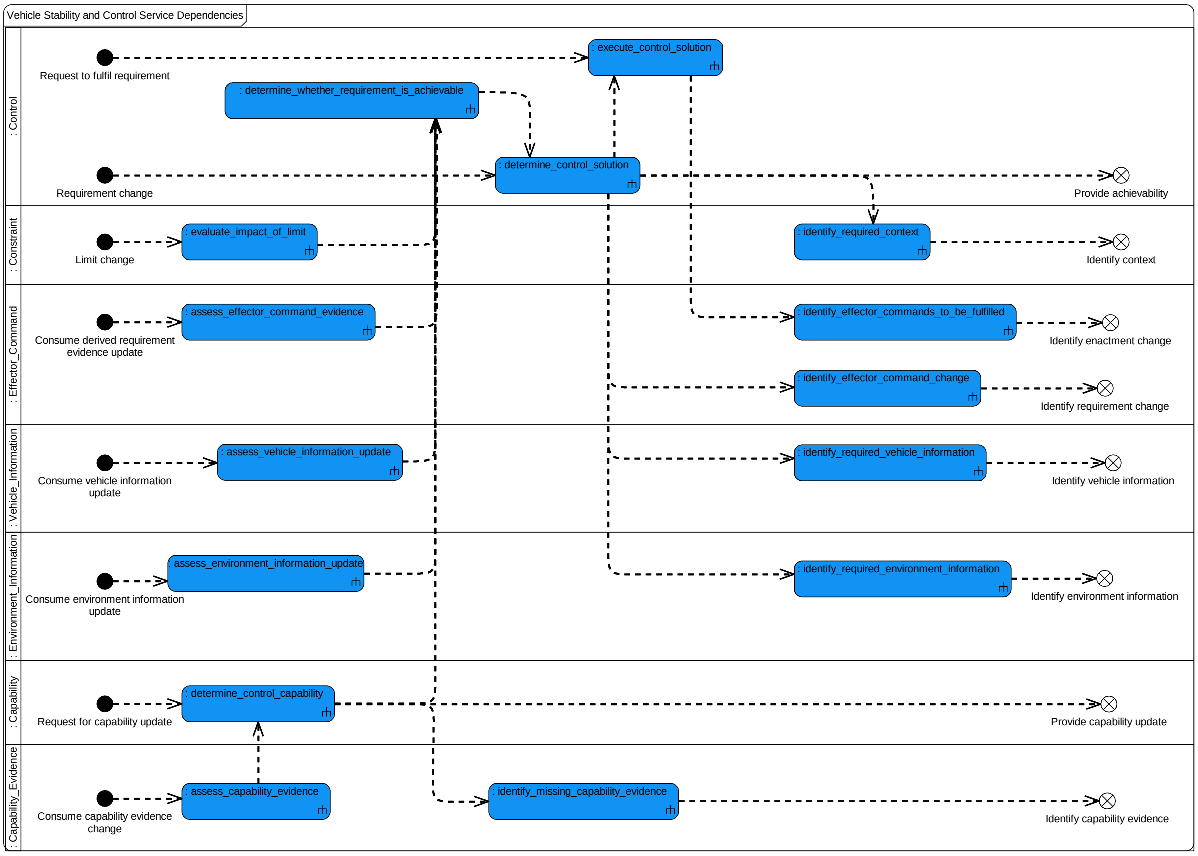

Vehicle Stability and Control Service Dependencies
: Control
: Constraint
: Effector_Command
: Vehicle_Information
: Environment_Information
: Capability
: Capability_Evidence
Request to fulfil requirement
: execute_control_solution
: determine_whether_requirement_is_achievable
Requirement change
: determine_control_solution
Provide achievability
Limit change
evaluate_impact_of_limit
: identify_required_context
Identify context
Consume derived requirement evidence update
: assess_effector_command_evidence
: identify_effector_commands_to_be_fulfilled
Identify enactment change
: identify_effector_command_change
Identify requirement change
Consume vehicle information update
: assess_vehicle_information_update
: identify_required_vehicle_information
Identify vehicle information
Consume environment information update
: assess_environment_information_update
: identify_required_environment_information
Identify environment information
Request for capability update
: determine_control_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence
: identify_missing_capability_evidence
Identify capability evidence

Figure 1259: Vehicle Stability and Control Service Dependencies

5.4.2.73 Weather

5.4.2.73.1 Role

The role of Weather is to determine the overall weather picture (including predicted weather based on received forecasts) in the operating environment.

5.4.2.73.2 Overview

Control Architecture

Weather is a service component as defined in the Control Architecture PYRAMID concept.

Standard Pattern of Use

In response to a query about the Weather_Picture, Weather obtains information on Weather_Conditions from on-board and off-board sources taking into account any Constraints on Sources. Weather may also derive Weather_Conditions using Measurements, again taking into account any Constraints. Weather compiles the Weather_Conditions into a Weather_Picture.

Examples of Use

Weather will be used where route planning needs to take into account regions of bad weather to be avoided.

5.4.2.73.3 Service Summary

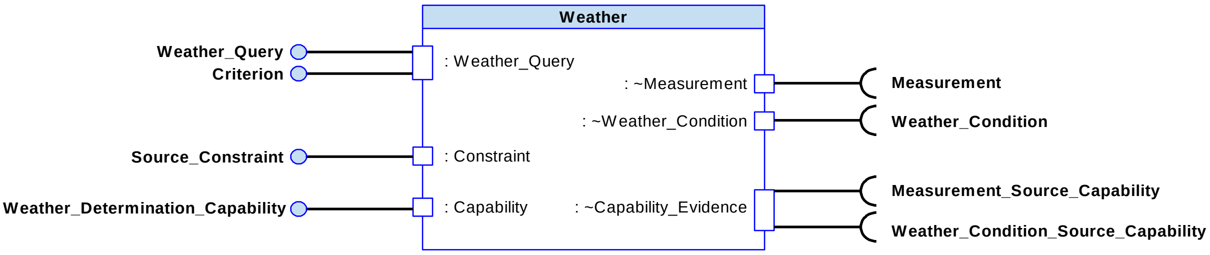


Figure 1260: Weather Service Summary

5.4.2.73.4 Responsibilities

capture_weather_picture_requirements

- To capture provided Requirements for the provision of a Weather_Picture.

capture_measurement_criteria

- To capture provided Measurement_Criterion/criteria for the Weather_Picture, e.g. confidence of prediction.

capture_constraints_for_weather_picture

- To capture Constraints relating to which Sources can be used for Weather_Types and Measurement_Types.

determine_relevant_weather_information

- To determine the Weather_Picture at particular times (or time windows) and locations based on forecasts and Measurements.

capture_measurements

- To capture Measurements from which Weather_Conditions can be determined, e.g. inputs from sensors.

capture_weather_condition_information

- To capture information on Weather_Conditions, e.g. forecasts.

determine_quality_of_weather_picture

- To determine the quality of the Weather_Picture against given Measurement_Criterion/criteria.

assess_weather_capability

- To assess the Capability to determine the overall Weather_Picture taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

identify_missing_information

- To identify missing information which could improve the certainty or specificity of a Weather_Picture, e.g. a more recent forecast.

predict_capability_progression

- To predict the progression of the Capability to determine the overall Weather_Picture over time and with use.

5.4.2.73.5 Subject Matter Semantics

The subject matter of Weather is meteorological conditions.

Exclusions

The subject matter of Weather does not include:

- The effects that meteorological conditions may have on an Exploiting Platform or a mission.

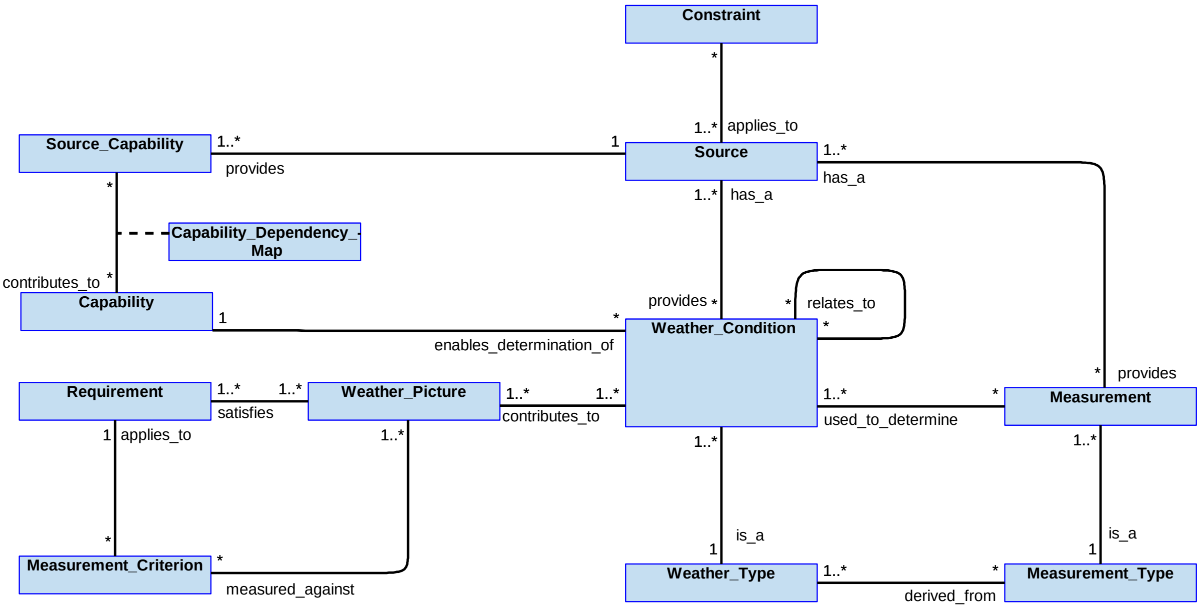


Figure 1261: Weather Semantics

5.4.2.73.5.1 Entities

Capability

The capability of Weather to determine the overall Weather_Picture (including predicted weather) in the operating environment.

Capability_Dependency_Map

A mapping of how the component's Capability is dependent on the capability of the information sources.

Constraint

An externally imposed restriction relating to which Sources can be used for Weather_Types and Measurement_Types. For example, limiting the use of information on wind speed from a particular Source which is suspected to be affected by a cyber attack.

Measurement

A quantification of a variable at a particular time and location. For example, temperature from which an icing Weather_Condition can be determined, or airspeed which can be used to calculate wind speed.

Measurement_Criterion

A criterion that the quality of the Weather_Picture will be measured against. For example, the accuracy of a weather forecast, which may be impacted by the source and age of meteorological information used to compile the forecast.

Measurement_Type

A type of Measurement.

Requirement

A request to provide the Weather_Picture.

Source

A provider of weather information, or Measurements from which weather can be determined (for example, airspeed and groundspeed from which wind speed can be calculated). It may be on-board (such as a sensor or a processor of information), or it may be off-board (such as SIGMET via ATS or ATIS).

Source_Capability

The ability of an information provider to provide information.

Weather_Type

The type of meteorological condition, such as temperature, wind speed, visibility, precipitation and pressure.

Weather_Condition

The state of a type of meteorological condition at a given time and place. This may take into account the validity, expiration and/or confidence of the information. There may be more than one instance of a specific weather condition if information about it is available from multiple sources.

Weather_Picture

A set of Weather_Conditions at a particular time and place. For example, raining with a strong northerly wind.

5.4.2.73.6 Design Rationale

5.4.2.73.6.1 Assumptions

- Weather will consider the provenance (source, validity and confidence) of meteorological information when determining the overall Weather_Picture.
- The quality of the Weather_Picture provided may deteriorate with time or the availability of forecasts or Measurements. For example, with increased time since a weather forecast was received.
- Weather includes information on atmospheric particulates including volcanic ash, chemical clouds and nuclear fallout.

5.4.2.73.6.2 Design Considerations

Related PYRAMID Concepts

These PYRAMID concepts were specifically taken into account in defining Weather:

- Data Driving - It is expected that Weather will be highly data-driven. For example, the types of weather that are reasoned about, how sources of information may be changed, how Weather_Type is determined from a Measurement_Type, etc.

Extensions

- It is possible that extension components will be developed to support specific types of equipment, e.g. weather radars and LIDARs.

Other Factors that were Taken into Account

- Weather will use measurements of the external environment to predict Weather_Conditions that the aircraft can be expected to encounter. For example, using temperature to predict CAT (Clear Air Turbulence) on the aircraft path.

Exploitation Considerations

- It is up to the Exploiting Programme to determine which Sources of information may be used.
- It is expected that Measurements and Weather_Conditions will be available to this component at all times. Where they need to be requested, as a derived requirement, Solution_Dependency services will be required.

5.4.2.73.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- As shown on the Weather IV this component consolidates weather related information from forecasts and on board sensors - i.e. is the single source of weather information. Flight in weather conditions that exceed the capability of the air vehicle could result in uncontrolled flight (e.g. if the air vehicle flies into a cumulonimbus cloud) and an uncontrolled crash. This would result in loss of the air vehicle and potentially fatalities.
- No credit has been assumed for the crew controlling the air vehicle directly observing the local weather or its effect on the air vehicle. For Exploiting Programmes where this is possible, DAL requirements may be less onerous.

5.4.2.73.6.4 Security Considerations

The indicative security classification is O.

This component determines the actual or forecast Weather_Picture using on-board and off-board sources. The weather is considered O, however the local weather picture may reveal positional information and therefore have more stringent confidentiality requirements. Loss of integrity in the weather information may cause the Exploiting Platform to unnecessarily re-route or alter mission parameters.

The component is expected to at least partially satisfy security related functions by:

- **Identifying Data Sources** used for determining weather as being allowable sources.
- **Maintaining Audit Records** of changes to the Weather_Picture during the course of the mission.
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and may therefore need to be protected to assure continued airworthiness.

- Performing **System Status and Monitoring** relating to the accuracy of weather information; whilst forecasting is not expected to be 100% accurate, large variations in forecast may indicate a source has been compromised.

The component is considered unlikely to directly implement security enforcing functions.

5.4.2.73.7 Services

5.4.2.73.7.1 Service Definitions

5.4.2.73.7.1.1 Weather_Query

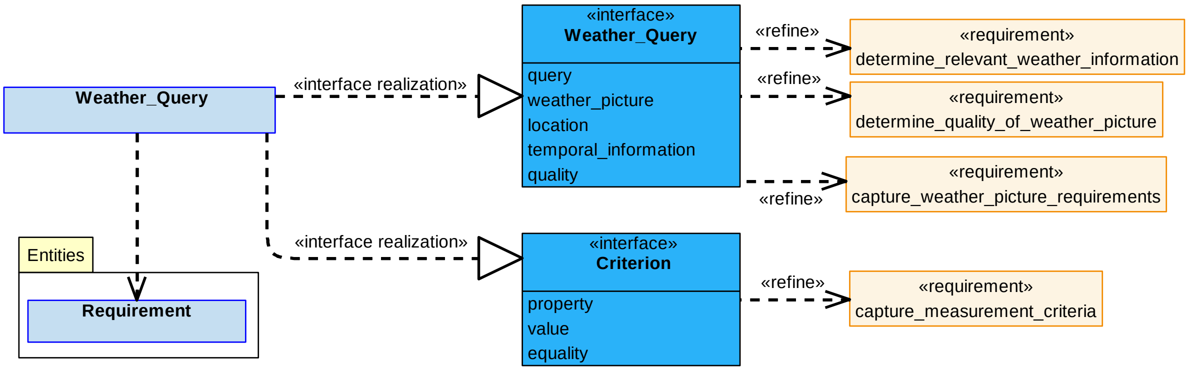


Figure 1262: Weather_Query Service Definition

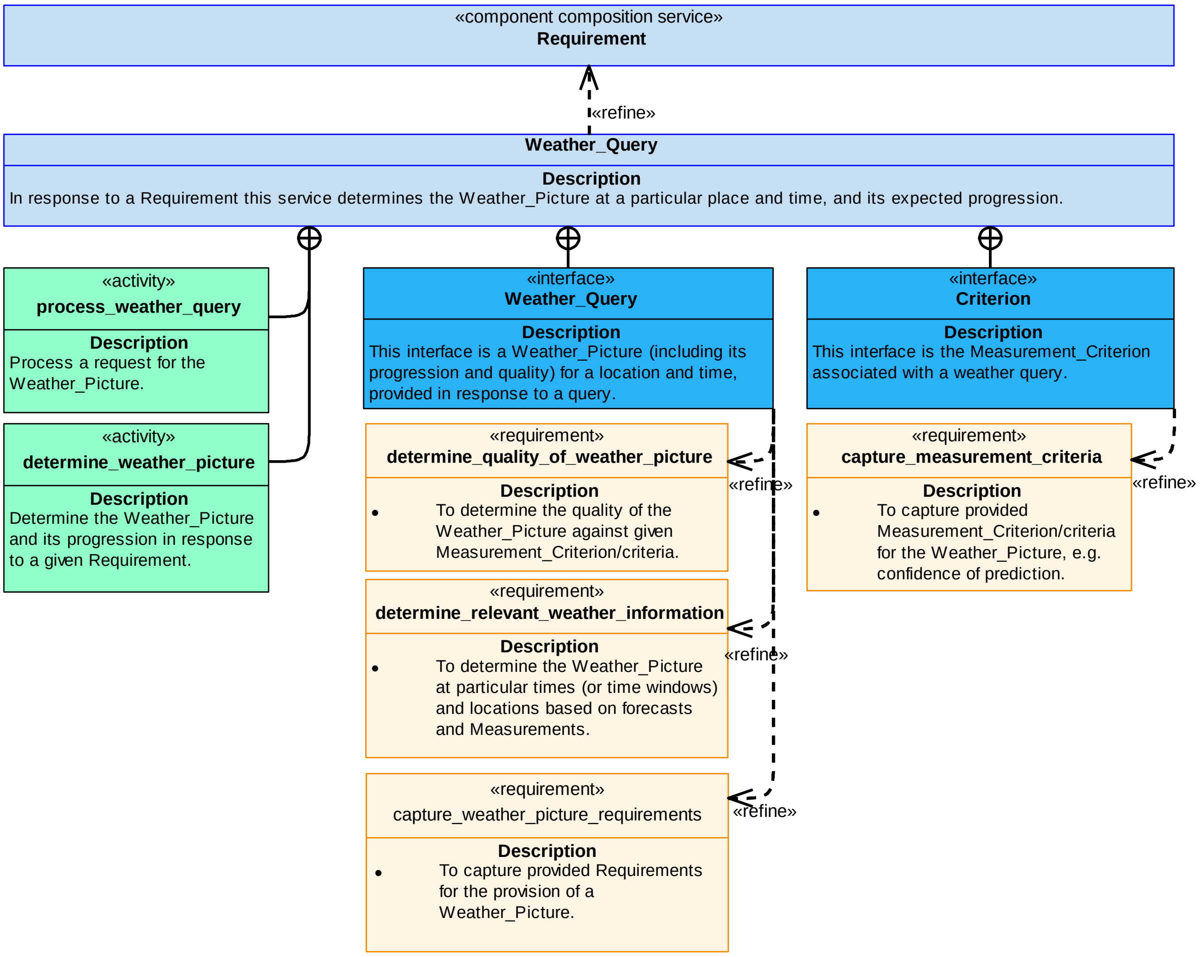

«component composition service»
Requirement
«refine»
Weather_Query
Description
In response to a Requirement this service determines the Weather_Picture at a particular place and time, and its expected progression.
«activity»
process_weather_query
Description
Process a request for the Weather_Picture.
«activity»
determine_weather_picture
Description
Determine the Weather_Picture and its progression in response to a given Requirement.
«interface»
Weather_Query
Description
This interface is a Weather_Picture (including its progression and quality) for a location and time, provided in response to a query.
«interface»
Criterion
Description
This interface is the Measurement_Criterion associated with a weather query.
«requirement»
determine_quality_of_weather_picture
Description
To determine the quality of the Weather_Picture against given Measurement_Criterion/criteria.
«refine»
«requirement»
capture_measurement_criteria
Description
To capture provided Measurement_Criterion/criteria for the Weather_Picture, e.g. confidence of prediction.
«refine»
«requirement»
determine_relevant_weather_information
Description
To determine the Weather_Picture at particular times (or time windows) and locations based on forecasts and Measurements.
«refine»
«requirement»
capture_weather_picture_requirements
Description
To capture provided Requirements for the provision of a Weather_Picture.
«refine»

Figure 1263: Weather_Query Service Policy

Weather_Query

In response to a Requirement this service determines the Weather_Picture at a particular place and time, and its expected progression.

Interfaces

Weather_Query

This interface is a Weather_Picture (including its progression and quality) for a location and time, provided in response to a query.

Attributes

query	The question being asked (e.g. information on the expected weather at a particular time and location, or where the wind is expected to be less than 50kts at a certain time).
weather_picture	The Weather_Picture returned in response to a query.
location	The location that the Weather_Picture applies to.
temporal_information	Timing information, such as the time, or time period, for which the Weather_Picture is required or provided.
quality	The quality of the provided Weather_Picture and its progression.

Criterion

This interface is the Measurement_Criterion associated with a weather query.

Attributes

property	The property to be measured, e.g. confidence in forecast.
value	The measured value of the property, e.g. 90% confidence.
equality	The relationship between the value and any limit on the measurement (e.g. less than, or equal to).

Activities

process_weather_query

Process a request for the Weather_Picture.

determine_weather_picture

Determine the Weather_Picture and its progression in response to a given Requirement.

5.4.2.73.7.1.2 Measurement

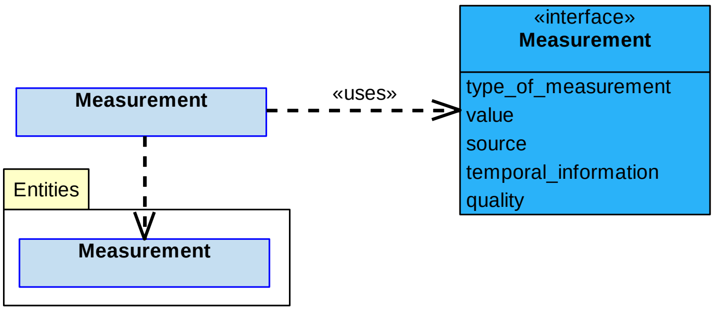


Figure 1264: Measurement Service Definition

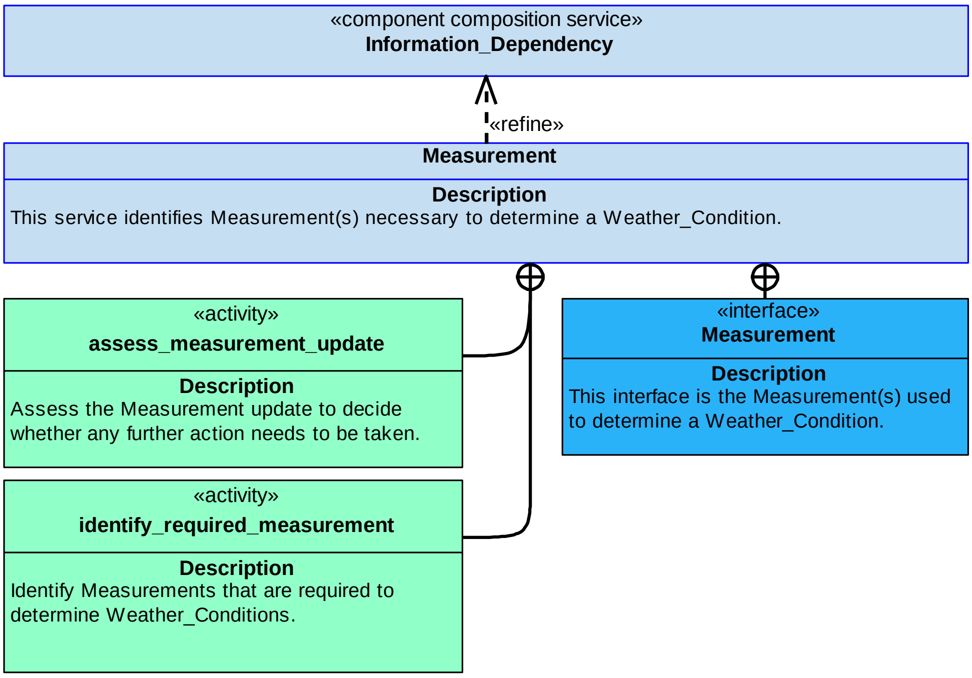

«component composition service»
Information_Dependency
«refine»
Measurement
Description
This service identifies Measurement(s) necessary to determine a Weather_Condition.
«activity»
assess_measurement_update
Description
Assess the Measurement update to decide whether any further action needs to be taken.
«interface»
Measurement
Description
This interface is the Measurement(s) used to determine a Weather_Condition.
«activity»
identify_required_measurement
Description
Identify Measurements that are required to determine Weather_Conditions.

Figure 1265: Measurement Service Policy

Measurement

This service identifies Measurement(s) necessary to determine a Weather_Condition.

Interface

Measurement

This interface is the Measurement(s) used to determine a Weather_Condition.

Attributes

type_of_measurement	The Measurement_Type.
value	The value of the Measurement.
source	The source of the Measurement.
temporal_information	Timing information, such as when the Measurement was taken.
quality	The predicted quality of the Measurement.

Activities

assess_measurement_update

Assess the Measurement update to decide whether any further action needs to be taken.

identify_required_measurement

Identify Measurements that are required to determine Weather_Conditions.

5.4.2.73.7.1.3 Weather_Condition

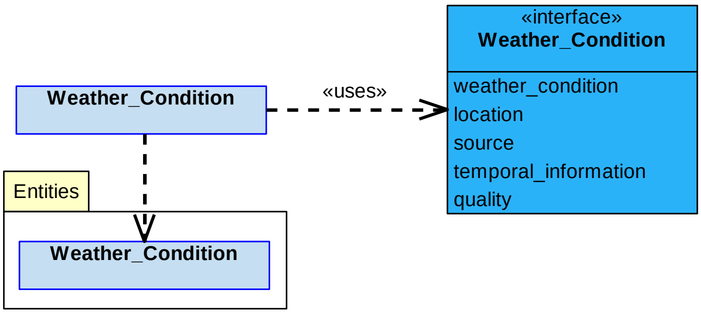


Figure 1266: Weather_Condition Service Definition

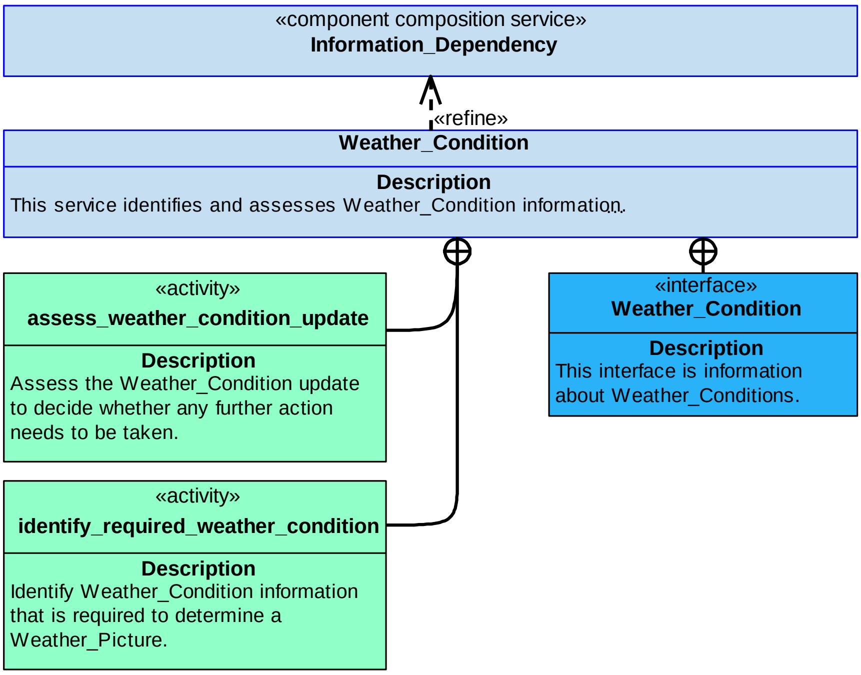

«component composition service»
Information_Dependency
«refine»
Weather_Condition
Description
This service identifies and assesses Weather_Condition information.
«activity»
assess_weather_condition_update
Description
Assess the Weather_Condition update to decide whether any further action needs to be taken.
«interface»
Weather_Condition
Description
This interface is information about Weather_Conditions.
«activity»
identify_required_weather_condition
Description
Identify Weather_Condition information that is required to determine a Weather_Picture.

Figure 1267: Weather_Condition Service Policy

Weather_Condition

This service identifies and assesses Weather_Condition information.

<u>Interface</u>

Weather_Condition

This interface is information about Weather_Conditions.

<u>Attributes</u>

weather_condition	The Weather_Condition.
location	The location at which the Weather_Condition applies.
source	The Source of the reported Weather_Condition.
temporal_information	Timing information, such as when the Weather_Condition was determined, or the time of day at which it applies.

quality	The predicted quality of the Weather_Condition.

Activities

assess_weather_condition_update

Assess the Weather_Condition update to decide whether any further action needs to be taken.

identify_required_weather_condition

Identify Weather_Condition information that is required to determine a Weather_Picture.

5.4.2.73.7.1.4 Constraint

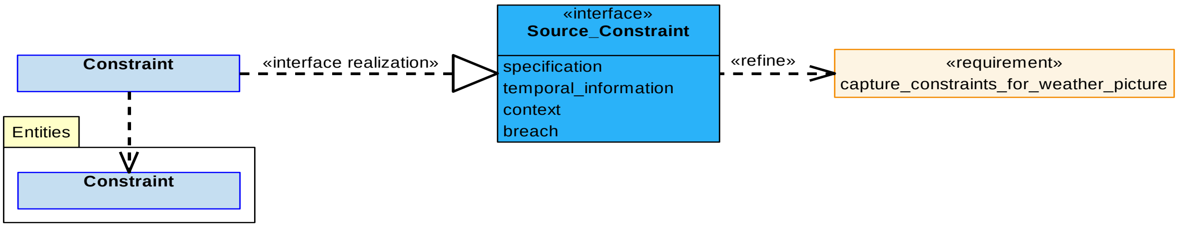
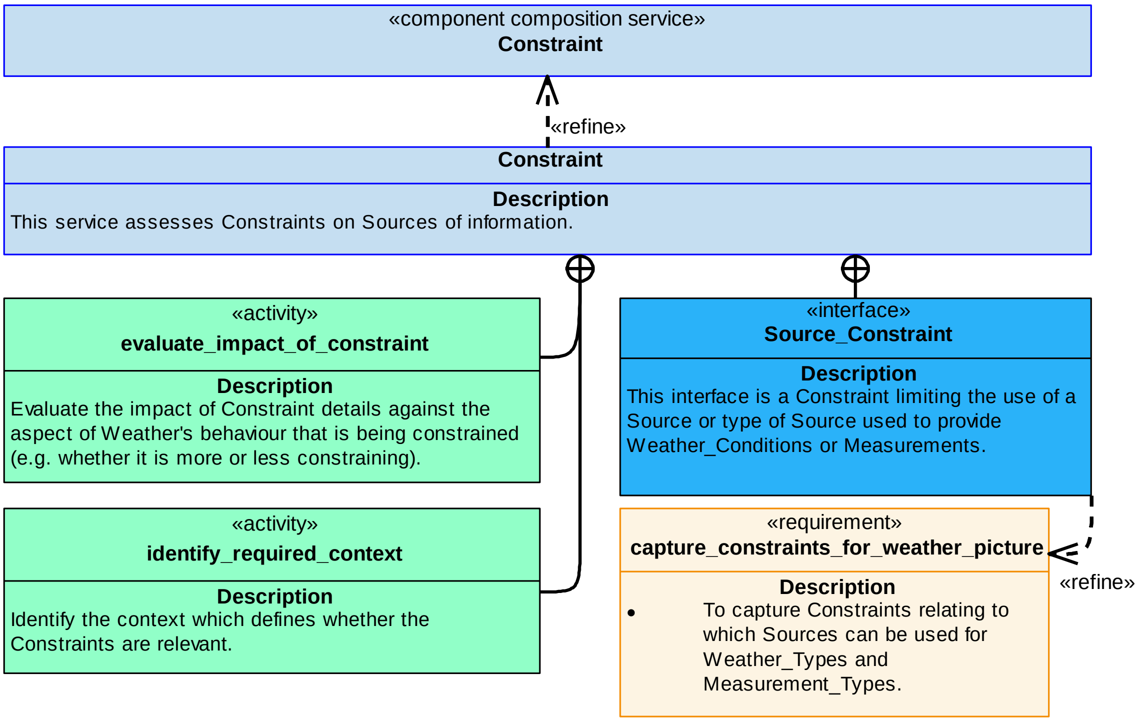


Figure 1268: Constraint Service Definition


«component composition service»
Constraint
«refine»
Constraint
«activity»
evaluate_impact_of_constraint
«activity»
identify_required_context
«interface»
Source_Constraint
«requirement»
capture_constraints_for_weather_picture
«refine»

Constraint

Description

This service assesses Constraints on Sources of information.

evaluate_impact_of_constraint

Description

Evaluate the impact of Constraint details against the aspect of Weather's behaviour that is being constrained (e.g. whether it is more or less constraining).

identify_required_context

Description

Identify the context which defines whether the Constraints are relevant.

Source_Constraint

Description

This interface is a Constraint limiting the use of a Source or type of Source used to provide Weather_Conditions or Measurements.

capture_constraints_for_weather_picture

Description

- To capture Constraints relating to which Sources can be used for Weather_Types and Measurement_Types.

Figure 1269: Constraint Service Policy

Constraint

This service assesses Constraints on Sources of information.

Interface

Source_Constraint

This interface is a Constraint limiting the use of a Source or type of Source used to provide Weather_Conditions or Measurements.

Attributes

specification	Specification of the Constraint. Such as a restriction on the type of information that can be obtained from a source.
temporal_information	Information covering timing of a Constraint, such as start time and duration, or end time.
context	The context in which the Constraint is applicable.
breach	A statement that the Constraint has been breached.

Activities

evaluate_impact_of_constraint

Evaluate the impact of Constraint details against the aspect of Weather's behaviour that is being constrained (e.g. whether it is more or less constraining).

identify_required_context

Identify the context which defines whether the Constraints are relevant.

5.4.2.73.7.1.5 Capability

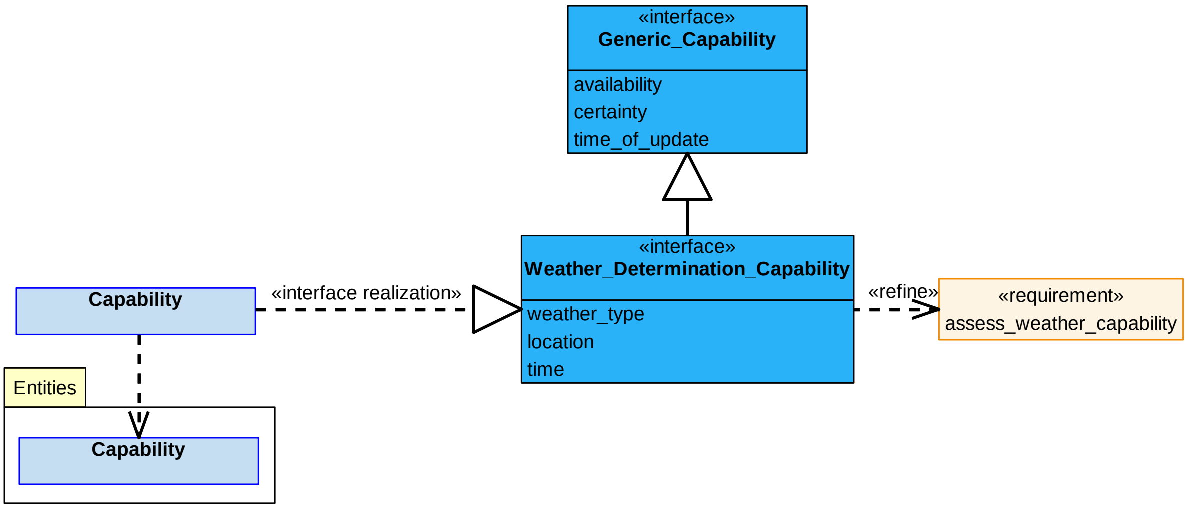


Figure 1270: Capability Service Definition

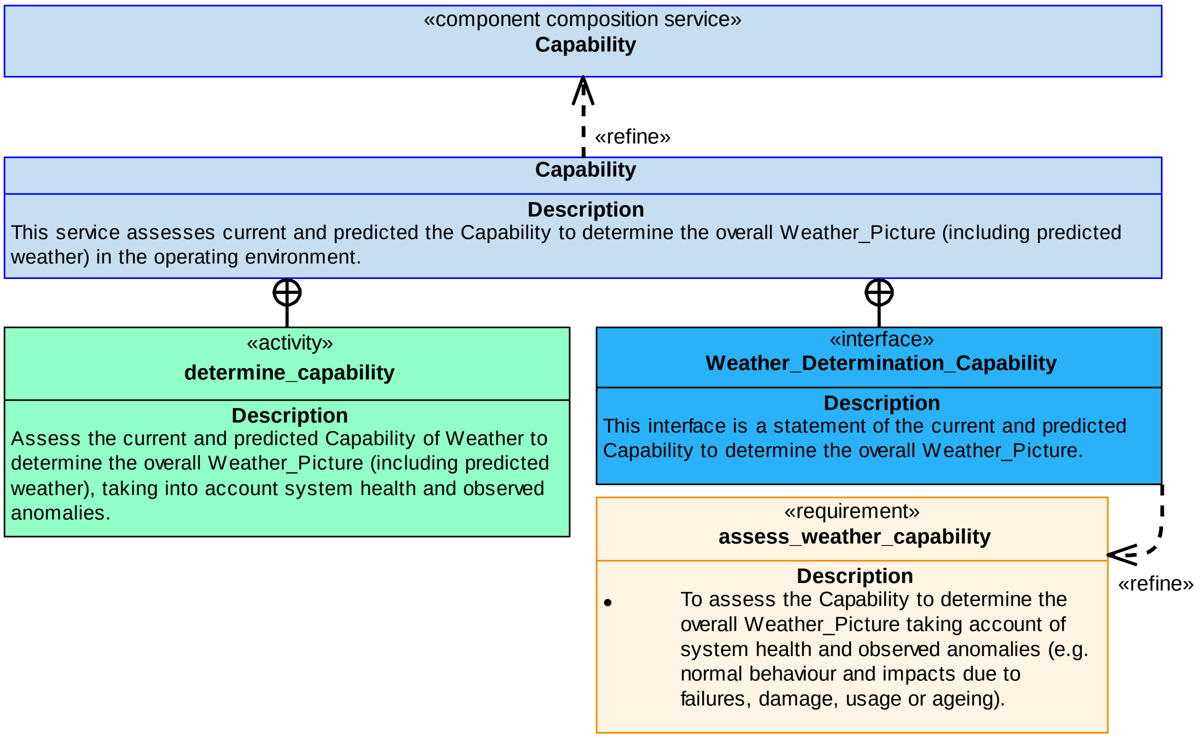

«component composition service»
Capability
«refine»
Capability
Description
This service assesses current and predicted the Capability to determine the overall Weather_Picture (including predicted weather) in the operating environment.
«activity»
determine_capability
Description
Assess the current and predicted Capability of Weather to determine the overall Weather_Picture (including predicted weather), taking into account system health and observed anomalies.
«interface»
Weather_Determination_Capability
Description
This interface is a statement of the current and predicted Capability to determine the overall Weather_Picture.
«requirement»
assess_weather_capability
Description
To assess the Capability to determine the overall Weather_Picture taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).
«refine»

Figure 1271: Capability Service Policy

Capability

This service assesses current and predicted the Capability to determine the overall Weather_Picture (including predicted weather) in the operating environment.

Interface

Weather_Determination_Capability

This interface is a statement of the current and predicted Capability to determine the overall Weather_Picture.

Attributes

weather_type	The Weather_Types that can be determined and provided.
location	The locations for which a Weather_Picture can be provided.
time	The time for which a Weather_Picture can be determined and provided.

Activity

determine_capability

Assess the current and predicted Capability of Weather to determine the overall Weather_Picture (including predicted weather), taking into account system health and observed anomalies.

5.4.2.73.7.1.6 Capability_Evidence

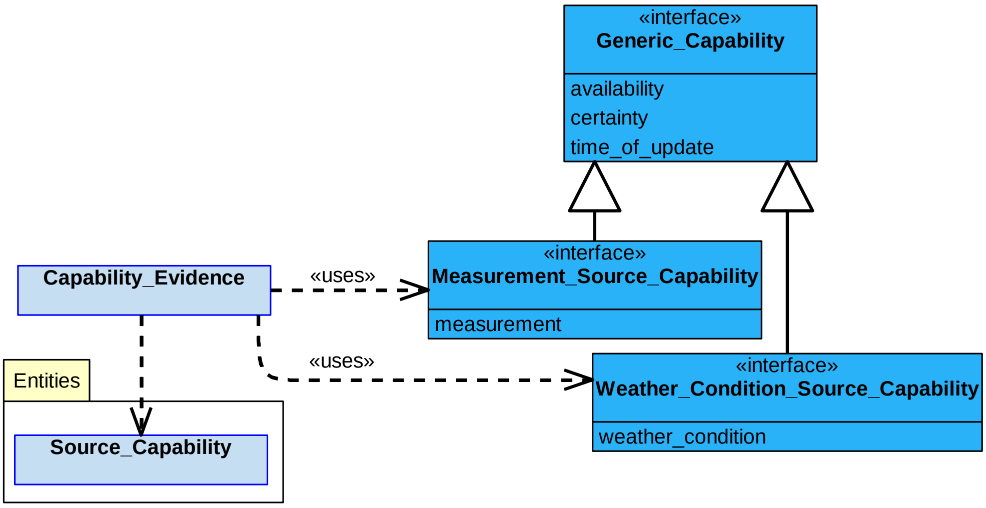


Figure 1272: Capability_Evidence Service Definition

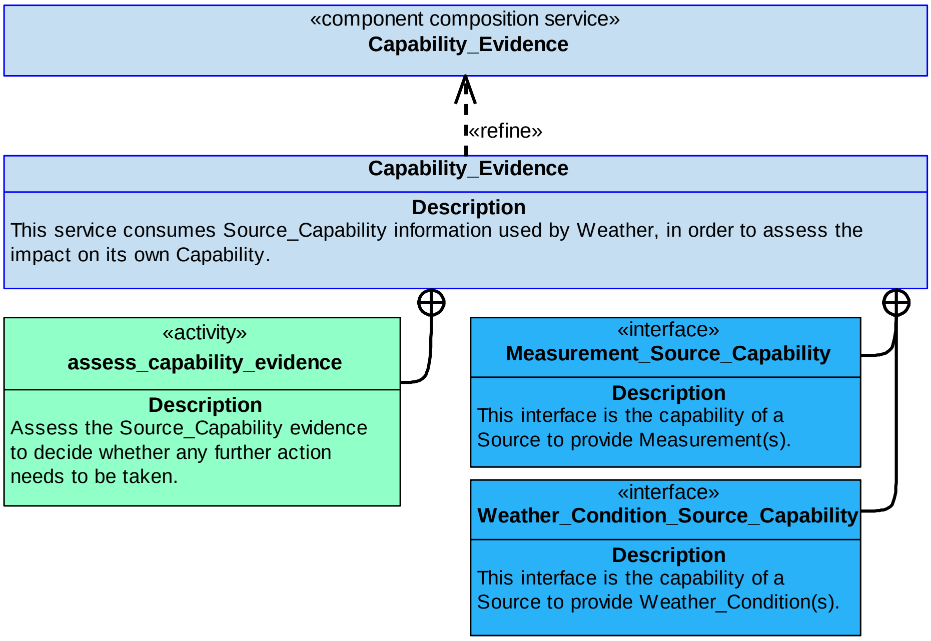

«component composition service»
Capability_Evidence
«refine»
Capability_Evidence
Description
This service consumes Source_Capability information used by Weather, in order to assess the impact on its own Capability.
«activity»
assess_capability_evidence
Description
Assess the Source_Capability evidence to decide whether any further action needs to be taken.
«interface»
Measurement_Source_Capability
Description
This interface is the capability of a Source to provide Measurement(s).
«interface»
Weather_Condition_Source_Capability
Description
This interface is the capability of a Source to provide Weather_Condition(s).

Figure 1273: Capability_Evidence Service Policy

Capability_Evidence

This service consumes Source_Capability information used by Weather, in order to assess the impact on its own Capability.

Interfaces

Measurement_Source_Capability

This interface is the capability of a Source to provide Measurement(s).

Attribute

measurement	A Measurement that can be provided by a source.

Weather_Condition_Source_Capability

This interface is the capability of a Source to provide Weather_Condition(s).

Attribute

weather_condition	A Weather_Condition that can be provided by a source.

Activity

assess_capability_evidence

Assess the Source_Capability evidence to decide whether any further action needs to be taken.

5.4.2.73.7.2 Service Dependencies

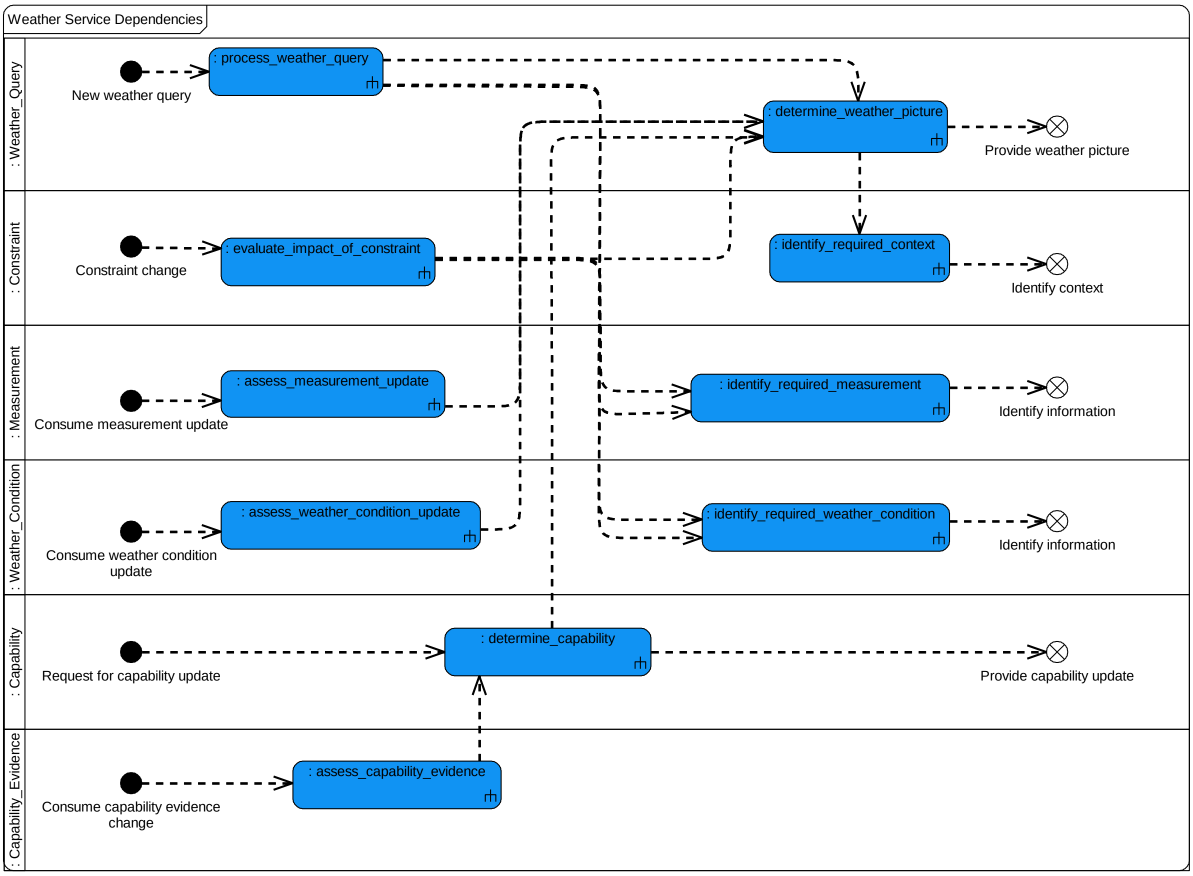

Weather Service Dependencies
: Weather_Query
: Constraint
: Measurement
: Weather_Condition
: Capability
: Capability_Evidence
New weather query
: process_weather_query
: determine_weather_picture
Provide weather picture
Constraint change
: evaluate_impact_of_constraint
: identify_required_context
Identify context
Consume measurement update
: assess_measurement_update
: identify_required_measurement
Identify information
Consume weather condition update
: assess_weather_condition_update
: identify_required_weather_condition
Identify information
Request for capability update
: determine_capability
Provide capability update
Consume capability evidence change
: assess_capability_evidence

Figure 1274: Weather Service Dependencies

Appendix A: Glossary

A.1 Introduction

This appendix provides the definition of a common set of terms and abbreviations relevant to both this standard and the PYRAMID Technical Standard Guidance document, Ref. [2]. There may be terms included within the glossary that are not used across both documents, but are only used in one of the documents.

A.1.1 Definition of Terms

Section Definitions of Terms provides a list of terms and definitions used within this standard, and the PYRAMID Technical Standard Guidance, Ref. [2]. The list includes PYRAMID specific terms and non-PYRAMID specific terms that are used throughout. References for any non-PYRAMID related sourced definitions have been included where appropriate.

A.1.2 Abbreviations and Acronyms

Section Abbreviations and Acronyms provides a list of abbreviations and acronyms used throughout this standard and the PYRAMID Technical Standard Guidance, Ref. [2].

A.2 Definitions of Terms

The table below defines two types of term:

PYRAMID specific:

These terms have a bespoke meaning within the PYRAMID Technical Standard and PYRAMID Technical Standard Guidance document. PYRAMID specific terms are indicated with their term descriptions highlighted in *italic* text.

Non-PYRAMID specific:

These terms are not bespoke to PYRAMID, however their definitions for use within the PYRAMID context are narrower in scope than their dictionary definitions. In addition to their definition, some terms also include additional information within their description to give the context of how the term applies within the PRA.

Term	**Description**
Achievability	*The ability to accomplish a requirement successfully.*
Action	*An activity defined in terms of what needs to be done. Actions are executed by coordinating resources.*
Activity	*Something that a system (or part of a system) does.*
Aircraft System	*An integrated system comprising of only one air vehicle and any number of supporting assets - such as its associated ground control station, mission planning systems and assets that are fully subordinate (such as missiles or support drones) to it or its control station. (Note that some subordinate assists may technically be air vehicles; however, they are an exception to the limit of one air vehicle. Likewise, it is possible for such assets to become or cease to be subordinate during the course of a* mission.*)*
Attribute	*An element of data that forms part or all of an interface on a* service.
Authorised Operator	Any person, user, or operator with validated credentials allowed to interact with a system to carry out a system role.
Availability	Property of being accessible and usable upon demand by an authorised entity. Ref. [30]
Bridge	*A mechanism for connecting a* component *to other system elements and that performs the translations necessary to enable the component to remain independent of the structure and semantics of other system elements.*
Catastrophic	Failure conditions that result in the death of one or more people. Catastrophic outcomes are considered DAL A within the PRA safety considerations.
Certification	The confirmation that the system complies with the applicable regulatory requirements (as agreed with the certifying authority, e.g. for airworthiness this is the MAA).
Component	*A piece of software or modelled artefact used as the basis for a piece of software (e.g. a* PYRAMID component *or* PRA component*).*
Component Behaviour	*Something that a* component *does in order to fulfil its responsibilities.*

Term	**Description**
Component Specification	*The precise requirement of the component as part of a specific* deployment*, described in terms of its* services *and/or entities.*
Confidentiality	Property that information is not made available or disclosed to unauthorised individuals, entities, or processes. Ref. [30]
Conflict	*A state where two or more demands (requirements or* constraints*) cannot be satisfied simultaneously.*
Constraint	*A limitation on the behaviour of a PYRAMID* deployment *at any level (whole system or constituent part).*
Consumed Service	*The means by which a* component *gets something done for it.*
Counterpart	*An abstraction (viewpoint), contained within a component, of a real-world object or concept that has a counterpart relationship with a different abstraction of the same real-world object or concept in another component.* *The abstraction can be for either a specific instance of the real-world object or concept (which may be represented as an object in object oriented software) or the concept from which a specific instance can be derived (which may be represented as a class in object oriented software).*
Counterpart Relationship	*The relationship between counterparts in different components, defining how they are related and how they may interact.*
Counterparting	*A concept whereby the same real-world object or concept is described from the viewpoints of different components.*
Critical	Failure conditions that result in major injury to people, loss of aircraft or a large reduction in safety margins. Critical outcomes are considered DAL B within the PRA safety considerations.
Cyber Attack	A deliberate and malicious exploitation of computer systems, technology-dependent enterprises and networks. Cyber attacks use malicious code to alter computer code, logic or data, resulting in disruptive consequences.
Data Driving	Data driving is the act of applying a data set to a component design, in order to provide a complete definition of how the component is required to behave.
Deployable Asset	*Any physical hardware (e.g. role fit* equipment*) carried on an* Exploiting Platform *that can be deliberately separated from the* Exploiting Platform *during a* mission*.*
Deployment	*A set of hardware and software elements forming a system (or part thereof) that satisfy the overall system requirements.*
Design Integrity	The extent to which the design is free from flaws that could give rise to or contribute to hazards, or to failure modes that contribute to a hazard. Ref. [32]
Dumb Asset	*A* deployable asset *that does not have a data interface with the* Exploiting Platform*.*
Equipment	*Hardware or a combination of hardware & software, that provides a capability or resource to the system under consideration.*

Term	**Description**
Executable Software	A computer file that contains encoded instructions capable of being executed by a processing unit. Executable software can be composed of one or more PYRAMID components.
Execution Platform	*The infrastructure supporting the execution, communication, etc. of application functionality, e.g.* ECOA, ARINC *653, Linux, Windows, and the computing hardware.*
Exploiter	*An organisation involved in the design and development of* PYRAMID components *or the design of PYRAMID* deployments*.*
Exploiting Platform	*A product (e.g. an air vehicle, ground station, or a test rig) that incorporates a PYRAMID* deployment*.*
Exploiting Programme	*A programme developing or incorporating PYRAMID components or a PYRAMID* deployment*.*
Extension Component	*A developed component that separates out or extends the functionality provided by a single* parent component*, by providing provided* services *that only the* parent component *has access to.*
Extension Point	*A* consumed service *that defines the* parent component*'s dependency upon an* extension component *for a single purpose.*
Extension Set	*A set of one or more* extension components *that satisfy the same* extension point*.*
Feasibility	*The practicality of achieving a solution.*
Flight	A collection of one or more aircraft, potentially of dissimilar types, performing roles and tasks to achieve the overall mission.
Flight Lead	The vehicle responsible for coordinating the activities of a flight to meet the objectives specified for the mission or supplied by the crew.
Flight Member	Any aircraft that forms part of a flight. Each member of the flight acts in support of the overall flight aims and of the other flight members.
Handover	*The process of performing a command and control transfer from one operator to another operator, e.g. between pilots on a twin-seat aircraft, operators on a single workstation, or across control stations.*
Health Data	Health data includes all data that is required as input for assessing the health and capability of the system. It includes, but is not limited to: • Hardware and software configuration data. • Fault and error codes (BIT reports). • Sensor data (including, but not restricted to, specific sensors for monitoring health and structural integrity). • System control, command and mode data. • Consumables data. • Usage data (for life and usage monitoring). • Manual measurements (requested and volunteered).
Integrity	(Safety context) The probability that the system will provide a specified level of safety. Ref. [11] (Security context) Property of accuracy and completeness. Ref. [30]

Term	**Description**
Item Development Assurance Level	The level of rigour of development assurance tasks performed on item(s). Ref. [27]
Logging	*The process of identifying and retaining data items associated with a component's processing that are not specific to its* subject matter.
Major	Failure conditions that result in minor injury to people, major damage to the aircraft or a significant reduction in safety margins. Major outcomes are considered DAL C within the PRA safety considerations.
Managed Resource	*A resource used by the system whose availability is limited and whose use by the system needs to be managed.*
Mission	*One or more aircraft ordered to accomplish one particular assignment.*
Mission Plan	The plan for the particular flight of one or more air vehicles from start-up / turnaround to shutdown / turnaround. It describes the planned flightpath and timings that the air vehicle should follow and the air vehicle’s assigned mission objectives to be achieved in order to meet the tasking. A Mission Plan may be modified whilst an air vehicle is in flight as the result of dynamic re-tasking.
Mission Support System	A system which supports another system that is responsible for carrying out a real life operations (e.g. a mission planning system or a data extraction system).
Non-Repudiation	Ability to prove the occurrence of a claimed event or action and its originating entities. Ref. [30]
Objective	*A high level goal which either defines the purpose of the* mission *(e.g. the requirement to suppress enemy air defences) or is otherwise required of the* mission *(e.g. the requirement for aircraft survivability) that is assigned to the system to support a broader strategic goal.*
Parent Component	*A developed component that supports the use of* extension components.
Platform Independent Model	A representation of a system that is independent of the execution platform.
Platform Specific Model	A representation of a system that incorporates the execution platform.
Protection Domain	A grouping of components within a platform specific deployment context that have similar segregation requirements (e.g. for security, safety or specific functionality reasons) that are separated from other domains such that they are unable to interfere with the resources or processing in that domain. Communications between protection domains is strictly regulated.
Provided Service	*The means by which a* component *is asked to do something.*
PYRAMID Reference Architecture	*The open PYRAMID Reference Architecture is a set of platform independent component definitions for air system application software.*
PRA Component	*A PYRAMID reference artefact, defined by a role, a distinct set of responsibilities, entities and* services*, for a specific, discrete area of* subject matter*.*
PYRAMID Component	*A component that is intended to comply with a* PRA component *definition.*
Recording	*The process of identifying and retaining data items directly associated with a component's* subject matter.

Term	**Description**
Retention	The keeping of important data for future use or reference.
Retention Strategy	*A set of specific* retention *rules and supporting information covering which data is to be captured and retained and the conditions for* retention *(including duration, classification, etc.).*
Sanitisation	(Security context) The process of deliberately, permanently and irreversibly removing or destroying data to make it unrecoverable.
Security Accreditation	The formal, independent assessment of an ICT system or service against its IA requirements, resulting in the acceptance of residual risk in the context of the business requirement and information risk appetite. This will be a prerequisite for approval to operate.
Security Domain	A grouping of elements (e.g. components and data) with similar security requirements that are managed by a defined security policy such that groupings remain separate unless specific controls are in place (e.g. data encryption).
Security Enforcing Function	Function relating to specific controls that provide protection to the system (e.g. providing cryptography). Failure of a SEF could lead to a security breach.
Security Related Function	Functions that support the security activities within the system but are not directly involved in enforcing the separation of security boundaries or preventing cyber attacks. Failure of a SRF will not directly lead to a security breach but may diminish the system's ability to detect or counter a threat (e.g. security event logging).
Service	*The means by which a component is asked to do something, or by which a component gets something done for it. A service is formed of interfaces and activities.*
Situation Awareness	The perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future. Ref. [31]
Storage	The action or method of storing data for future use.
Storage Media	The media used to store data.
Subject Matter	*The definition, semantics and behaviour associated with a topic or subject. This is used to define the bounded scope of a* PRA component*.*
System Integrator	*An organisation or individual involved in the wider integration of a PYRAMID* deployment*.*
Task	*The specification of a goal which needs to be achieved utilising the capability of an* aircraft system *(e.g. the need for an air vehicle to transit to a location, search an area, or for an aircraft system to disable a target).*
Trust	The confidence that a component or other system element will behave as expected.

Table 4: Definitions of PYRAMID Technical Standard Terms

A.3 Abbreviations and Acronyms

Name	Description
A/A	Air to Air
A/S	Air to Surface
AAR	Air-to-Air Refuelling
ACAS	Airborne Collision Avoidance System
ACID	Atomic, Consistent, Isolated, Durable
ADS-B	Automatic Dependent Surveillance – Broadcast
AEK	Algorithm Encryption Key
AMRAAM	Advanced Medium-Range Air-to-Air Missile
API	Application Programming Interface
ARINC	Air Radio Incorporated
ASRAAM	Advanced Short-Range Air-to-Air Missile
ATC	Air Traffic Control / Controller
ATIS	Automatic Terminal Information Service
ATS	Air Traffic Services
BIT	Built In Test
BS	British Standard
C2	Command and Control
CAT	Clear Air Turbulence
CBIT	Continuous Built in Test
CEP	Circular Error Probability
CIA	Confidentiality, Integrity and Availability
CIK	Cryptographic Ignition Key

Name	Description
COMSEC	Communications Security
CPDLC	Controller - Pilot Datalink Communications
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSMU	Crash Survivable Memory Unit
CTT	Controlled-Trajectory Termination
CyDR	Cyber Defence and Risk
DAL	Development Assurance Level
DDS	Data Distribution Service
DEK	Data Encryption Key
DEW	Directed Energy Weapon
DME	Distance Measuring Equipment
DMZ	De-Militarized Zone
DoS	Denial of Service
EAL	Evaluation Assurance Level
ECM	Electronic Countermeasures
ECOA	European Component Oriented Architecture
EED	Electronic Explosive Device
EM	Electro-Magnetic
EMCON	Emissions Control
EMF	Electromagnetic Field
EN	European Standard
EO	Electro-Optical

Name	Description
ES	Electronic Surveillance
ESM	Electronic Support Measures
EUROCAE	EURopean Organisation for Civil Aviation Equipment
EW	Electronic Warfare
FAA	Federal Aviation Administration
FACE®	Future Airborne Capability Environment®
FPGA	Field-Programmable Gate Array
GDPR	General Data Protection Regulation
GMT	Greenwich Mean Time
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSN	Goal Structuring Notation
GUI	Graphical User Interface
HF	High Frequency
HMI	Human Machine Interface
HMS	His Majesty's Ship
HOTAS	Hands On Throttle And Stick
HW	Hardware
IA	Information Assurance
IBIT	Initiated Built in Test
ICT	Information and Communications Technology
ID	Identifier
IDAL	Item Development Assurance Level

Name	Description
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IFF	Identification Friend or Foe
ILS	Instrument Landing System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IP	Internet Protocol
IPS	Intrusion Protection System
IPSec	Internet Protocol Security
IR	Infrared
IRST	Infrared Search and Track
ISO	International Organisation for Standardisation
ISR	Intelligence, Surveillance and Reconnaissance
IT	Information Technology
IV	Interaction View
IVDL	Inter Vehicle Data Link
JPEG	Joint Photographic Experts Group
KEK	Key Encryption Key
LAR	Launch Acceptability Region
LDAP	Lightweight Directory Access Protocol
LDP	Laser Designator Pod
LIDAR	Light Detection and Ranging
LRU	Line Replaceable Unit

Name	Description
MASS	Master Armaments Safety Switch
MB	Megabyte
MBSE	Model Based Systems Engineering
MDA	Model Driven Architecture
MIKEY-SAKKE	Multimedia Internet Keying - Sakai-Kasahara Key Encryption
MIP	Multilateral Interoperability Programme
MITM	Man In The Middle
MSA	Minimum Safe Altitude
MSD	Minimum Separation Distance
MSS	Master Safety Switch
NATO	North Atlantic Treaty Organization
NIST	National Institute of Standards and Technology
O	Official
O-S	Official Sensitive
OMG	Object Management Group
OMI	Operator-Mission Interface
OSD	Office of Security of Defence
OTAR	Over The Air Rekeying
PBIT	Power Up Built in Test
PC	Personal Computer
PIM	Platform Independent Model
PMDH	Post Mission Data Handling
PNG	Portable Network Graphics

Name	Description
PRA	PYRAMID Reference Architecture
PRI	Pulse Repetition Interval
PRIME	Protocol Requirements for IP Modular Encryption
PSM	Platform Specific Model
QFE	Q code - pressure at airfield runway
QNH	Q code - pressure adjusted to mean sea level
QoS	Quality of Service
RA	Resolution Advisory
RCD	Residual Current Device
RCS	Radar Cross Section
RDP	Remote Desktop Protocol
RF	Radio Frequency
RFI	Request For Information
RoE	Rules of Engagement
RTB	Return To Base
RTCA	Radio Technical Commission for Aeronautics
RTPS	Real Time Publish Subscribe
S&RE	Suspension & Release Equipment
SA	Situation Awareness
SAE	Society of Automotive Engineers
SAM	Surface to Air Missile
SAR	Synthetic Aperture Radar
SATCOM	Satellite Communications

Name	Description
SC	Security Check
SCEO	Secret - Coalition Eyes Only
SDP	Session Description Protocol
SEAD	Suppression of Enemy Air Defences
SEF	Security Enforcing Function
SID	Standard Instrument Departure
SIEM	Security Information & Event Management
SIGMET	Significant Meteorological Information
SIP	Session Initiation Protocol
SNEO	Secret - National Eyes Only
SOA	Service Oriented Architecture
S.O.L.I.D.	Single-responsibility principle Open-closed principle Liskov substitution principle Interface segregation principle Dependency inversion principle
SOS	Store On Station
SOUP	Software of an Unknown Pedigree
SRF	Security Related Function
SSUN	Single Statement of User Need
SysML	Systems Modelling Language
TA	Traffic Advisory
TACAN	Tactical Air Navigation System
TB	Terabyte
TCAS	Traffic alert and Collision Avoidance System

Name	**Description**
TCP	Transmission Control Protocol
TDL	Tactical Data Link
TLS	Transport Layer Security
TOA	Terminal Operation Area
TRANSEC	Transmission Security
TS	Top Secret
TTP	Techniques, Tactics & Procedures
UAS	Uncrewed Air System
UAV	Uncrewed Air Vehicle
UC	Use Case
UCS	UAV Control System
UHF	Ultra-High Frequency
UK	United Kingdom
UML	Unified Modelling Language
US	United States
USB	Universal Serial Bus
UTC	Coordinated Universal Time
VoIP	Voice Over Internet Protocol
VOR	VHF Omnidirectional Range
VPN	Virtual Private Network

Table 5: Abbreviations and Acronyms