PYRAMID Exploiter's Pack Version 4.1

Annex A – PRA Description Document Issue 4.1

Ministry of Defence

This document has been prepared, as part of the PYRAMID Exploiter's Pack, in order to set out a generic approach to implementation of the PYRAMID 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 following your review of the PYRAMID Exploiter's Pack V4.1, 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

EXECUTIVE SUMMARY

The PYRAMID Reference Architecture (PRA) is an open, air system, reference architecture aimed at software implementation that is both Exploiting and Execution Platform independent. It will support the realisation of the PYRAMID Key User Requirements (KURs) when fully instantiated within a complete mission system.

This document annex forms part of the PYRAMID Exploiter's Pack and defines, in a more accessible form, the PYRAMID Reference Architecture developed as a UML model. It outlines:

- Policies that assist exploiters and future component developers to understand the principles that underpin the PRA, and how it is intended to be deployed;
- Component specifications from which a deployment of the PRA can be developed;
- Interaction Views (IVs) showing indicative ways in which the components could be combined to satisfy Exploiting Platform requirements.

CHANGE HISTORY

Date	Issue	Description of Changes
November 2019	1	Initial issue
January 2020	1.1	Embodiment of customer comments
December 2020	2	The scope of this document has been decreased (some of the content was moved to the PYRAMID Exploiter's Pack Ref. [1]). The PRA Description Document has become Annex A to the PYRAMID Exploiter's Pack Ref. [1] and so some previous content of this document has moved to the PYRAMID Exploiter's Pack. This document now only includes content for policies, components and interaction views and has been restructured accordingly. The features worked on for Issue 2 of the PRA Description Document were as follows: • Architecture Alignment Queries • Mission Planning & Debrief • Power & Cooling • SMART Equipment Integration • Exploiter's Pack Accessibility • Readability In addition changes have been made to the PYRAMID Exploiter's Pack elements held within this Annex resulting from query answering from both internal and external stakeholders and general maturity improvements. The following are artefacts that have been added, removed, re-scoped or renamed: Policies: New • Data Exchange • Interaction with Equipment • Operational Support • Recording and Logging • Storage Deleted • Interaction with Physical Hardware • Recording and Storage Components: New • Data Distribution Renamed

Date	Issue	Description of Changes
		• *From* Code of Connection *to* Semantic Translation Re-scoped • *From* Simple Effector *to* Effectors • *From* Simple Sensor *to* Sensors Deleted • Communication Service • Communication Transport • Complex Effector • Complex Sensor • Reports • Software Logging Interaction Views: New • Vehicle Performance • Mission Data Load • Generation of Reports • Request for Information • Fuel Management • Power Management • Cryptographic Key Revocation Use Case Renamed • *From* Identify Cyber Attack *to* Identification of Cyber Attack Deleted • Time Management
December 2021	3	The features worked on for Issue 3 of the PRA Description Document were as follows: • Expanded Component Definitions • Component Service Development • Pattern Development • HMI • Security • Architecture Alignment Fixes • Readability and Accessibility In addition, changes have been made to the PYRAMID Exploiter's Pack elements held within this Annex resulting from query answering from both internal and external stakeholders and general maturity improvements. The following are artefacts that have been added, removed, re-scoped or renamed: Policies:

Date	Issue	Description of Changes

		New
		• Dependency Management
		• Component Connections
		
		Renamed
		• *From* Security Analysis *to* Security Approach
		
		Components:
		New
		• Component Composition
		• HMI Dialogue
		• Information Presentation
		
		Renamed
		• *From* Cryptographic Keys *to* Cryptographic Materials
		• *From* Vehicle Infrastructure *to* Asset Transitions
		• *From* Tanks *to* Fluids
		• *From* Spatial Alignment *to* Spatial Correction
		• *From* Offensive Actions *to* Target Engagement
		
		Deleted
		• System Identification
		• HMI Application
		• HMI Compositor
		• HMI Device
		• HMI Presentation
		• HMI Scope
		
		Interaction Views:
		New
		• Time Synchronisation between Nodes
		• Middleware Error
		  ○ Middleware Handled Error UC
		  ○ PRA Handled Error UC
		  ○ Distributed Hardware UC
		Renamed
		• *From* Cryptographic Key Revocation *to* Cryptographic Material Revocation

October 2022	3.1	Issue 3.1 is a UK OFFICIAL version of Issue 3. No changes to the content of the document have been made other than non-technical changes due to the up-issue of the PYRAMID Exploiter's Pack from Version 3 to Version 3.1 (headers/footers, references etc.)
December 2022	4	The features worked on for Issue 4 of the PRA Description Document were as follows: • Component Service Development • Pattern Development

Date	Issue	Description of Changes
		• Architecture Alignment Fixes • Readability and Accessibility In addition, changes have been made to the PYRAMID Exploiter's Pack elements held within this Annex resulting from query answering from both internal and external stakeholders and general maturity improvements. The following are artefacts that have been added, removed, re-scoped or renamed: Policies: Re-scoped • Control Architecture • Dependency Management • Operational Support Components: New • Trajectory Prediction Re-scoped • Release Aiming • Sensor Products • Data Fusion • Tactical Objects Renamed • *From* Time Determination *to* Reference Times • *From* Weapons *to* Destructive Effects Interaction Views: Renamed • *From* Antenna Management *to* Link Selection • *From* Identification of Cyber Attack *to* Defence Against Cyber Attack • *From* Intelligence Gathering *to* Search
September 2023	4.1	The document has been updated due to now being released via Open Government License v3.

List of Effective Pages

1716 pages UK OFFICIAL

1716 pages in total

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TERMS AND ABBREVIATIONS USED IN THIS DOCUMENT

Definitions of project terms, the meaning of acronyms and the meaning of abbreviations used in this document can be found in the PYRAMID Glossary Ref [5].

REFERENCES

The reference numbers are consistent across all the documents in the PYRAMID Exploiter's Pack. This means that in this document, when a reference is not used, the corresponding reference number will not appear in the reference list.

Project Related Document References:

For the avoidance of doubt, all documents referenced below which form part of the PYRAMID Exploiter's Pack are subject to the terms of DEFCON 703.

Reference	Title, Document Number, Issue & Date
[1]	PYRAMID Exploiter's Pack (Main Document), RCO_FUT_23_004, Issue 4.1, September 2023.
[3]	PYRAMID Exploiter's Pack Annex B: Deployment Guide, RCO_FUT_23_006, Issue 4.1, September 2023.
[5]	PYRAMID Exploiter's Pack Annex D: Glossary, RCO_FUT_23_008, Issue 12.1, September 2023.
[60]	Dstl, Security Guidance for PYRAMID Exploiters, DSTL/TR111125, Issue 1.0, October 2021.

Non-Project Related Document References:

Public domain references below contain information which is proprietary to that referenced third party. Any information from this source is subject to separate rights and terms and is not subject to the terms of DEFCON 703 or DEFCON 705.

Reference	Title, Document Number, Issue & Date
[11]	T. Erl, Service-Oriented Architecture: Concepts, Technology and Design, 2005.
[12]	ISO/IEC 18384, Reference Architecture for Service Oriented Architecture (SOA RA), 2016.
[15]	Aircraft/Store Electrical Interconnection System, MIL-STD-1760E, 2007.
[16]	Design and Airworthiness Requirements for Service Aircraft, Def Stan 00-970, Issue 21, 2019.
[17]	Sanremo Handbook on Rules of Engagement, 2009.
[18]	Condition monitoring and diagnostics of machines, ISO 13374, 2003-2015.
[19]	WIUK Plug and Play Reference Architecture Framework, BAES/WIUK/W/721/RP/000037, Issue 1, 2013.
[20]	Cockpit Display System Interfaces to User Systems, ARINC Specification 661-6, 2016.
[21]	Airworthiness Security Process Specification, RTCA DO-326A, 2014.

[22] EU General Data Protection Regulation, Regulation (EU) 2016/679, 2016.

[23] Software Considerations in Airborne Systems and Equipment Certification, RTCA DO-178C, 2011.

[25] Object Management Group, Service oriented architecture Modeling Language (SoaML) Specification, OMG, 2012.

[37] SAE International, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, SAE ARP4761, 1996.

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

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

[46] Safety Management Requirements for Defence Systems, Defence Standard 00-56 Part 2, Issue 5, February 2017.

[47] Requirements for Safety of Programmable Elements (PE) in Defence Systems, Defence Standard 00-055 Part 1, Issue 4, April 2016.

[51] HMG IA Standard No.1 & 2: Information Risk Management, Issue 4.0, April 2012.

[52] Information technology - Security techniques - Information security risk management, BS EN ISO/IEC 27005:2018.

[53] Framework for Improving Critical Infrastructure Cybersecurity, National Institute of Standards and Technology, Version 1.1, April 16, 2018. [Online] Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf.

[54] Information technology. Security techniques. Code of practice for information security controls, BS EN ISO/IEC27002:2017.

[55] Information technology. Security techniques. Evaluation criteria for IT security. Security functional components, BS EN ISO/IEC 15408-2:2020.

[61] Airworthiness Security Methods and Considerations, RTCA, DO-356A, 2018.

[62] Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture, ISO 7498-2:1989

1 Introduction

The MOD's PYRAMID programme introduces a paradigm shift to the current method of avionic systems design and procurement, aiming to make the next generation of air systems affordable, capable and adaptable by the adoption of an open architecture approach and systematic software reuse.

This document is an annex to the PYRAMID Exploiter's Pack Ref. [1].

The PRA exists as a model which has been developed using Windchill Modeler.

1.1 Scope

This PRA description document provides the PRA in a more accessible form than the PRA model. It includes:

- Policies that assist Exploiters and future component developers to understand the principles that underpin the PRA, and how it is intended to be deployed;
- Component specifications from which a deployment of the PRA can be developed;
- Interaction Views (IVs) showing indicative ways in which the components could be combined to satisfy Exploiting Platform requirements.

1.2 Purpose

The purpose of this document is to describe the functional components that constitute the PRA and to provide information on how these components could be deployed in a system.

2 Introduction to Policies

Architectural Policies are intended to assist Exploiters and future component developers to understand the philosophies that underpin the PRA, and how it is intended to be deployed. They cover a broad cross-section of themes that affect air systems such as how system health data should be handled and how the system should interact with the external environment. The PRA architectural policies are provided in Appendix A.

The basic structure of each policy is as follows:

1. **Pre-Reading and Related Policies -** Any key concepts necessary to understand the subject matter
2. **Introduction** - The Introduction introduces the subject matter of the policy, including:
    a. What the policy is
    b. Why the policy is important
    c. Aims of the policy and how it supports the PYRAMID KURs
3. **Overview** - The Overview summarises the key points of the policy.
4. **Detailed Breakdown** - The policy will be explained in greater detail in subsequent sections.

The architectural policies are split into four categories:

- Architecture Wide Policies - These describe ways of using the components for system-wide aspects of a PRA based system in a way that supports the PYRAMID KURs.
- Specific Policies - These describe how the components are intended to be integrated to provide specific functionality in a way that supports the PYRAMID KURs.
- Modelling Principles - These support a deployment of components in meeting the PYRAMID KURs.
- Safety and Security - These describe how safety and security have been considered during the development of the components.

Since policies are concerned with architectural principles, some technical details are left out for clarity, specific technical implementation detail of policies is the responsibility of Exploiting Programmes to define and apply. Diagrams in this section generally abbreviate how service connectivity is illustrated. In particular, although connections between component services are always implemented by bridges, these are not shown except where they are the specific focus of the policy.

The components capture design decisions or specific design statements resulting from the application of these policies.

2.1 Scope Summaries for Architecture Wide Policies

- Control Architecture: Describes a control architecture that has been embodied in the PRA, enabling an exploiting platform to implement system-wide control.
- Constraint Management: Explains how a PRA-based system can keep within the constraints that limit its behaviour.
- Dependency Management: Describes how the PRA has been designed to enable dependencies to be managed so that mission objectives can be achieved.
- Autonomy: Explains what is meant by autonomy within PYRAMID and how it applies to the PRA, including the relationship with authorisation for carrying out actions.

- Health Management: Explains what is meant by health and how the PRA enables a system to manage situations of reduced health.
- Capability Assessment: Explains how a PRA-based system assesses its ability to perform its designed functions as internal system factors change.
- Multi-Vehicle Coordination: Describes how different vehicles can interact in a coordinated manner through the use of components, including instances of the same component, deployed across different vehicles.
- Interaction with Equipment: Explains how the PRA has been designed to enable an exploiting system to interact with equipment, including determination of equipment capability and control of equipment resources.
- Resource Management: Explains what is meant by resources and how resources can be managed when there are multiple demands on them.
- Operational Support: Explains how the PRA can be used for purposes beyond the execution of a mission. It describes a range of such uses, for instance mission preparation, and the components that would support these.
- Storage: Describes the mechanisms provided by the PRA to enable storage of data, and the interactions of the components with storage facilities provided within a deployment of the PRA.
- Recording and Logging: Defines the means by which components identify information that is needed for current or future use and therefore the information that needs to be retained through recording or logging. It discusses how a component knows how long information should be retained for and if it is no longer required.

2.2 Scope Summaries for Specific Policies

- Cyber Defence: Describes how the PRA can be used to provide the system with a level of security monitoring and protection from unauthorised interactions, including how components should work together to protect against different types of cyber attack.
- Human-Machine Interface: Introduces the HMI components and describes how the HMI components are intended to interact.
- Interfacing with Deployable Assets: Deployable assets are hardware which can be deliberately separated from the exploiting platform during a mission. The policy describes interfacing with deployable assets, before and after separation, from a PYRAMID-compliant Exploiting Platform.
- Tactical Information: Explains how the PRA supports the handling of sensor data and associated tactical information.
- Test: Defines the ability to support testing which is provided by the PRA, including how components support different types of test at various levels of system capability. The policy scope is restricted to self-testing of a PRA deployment.
- Use of Communications: This policy explains how communications capability may be used by PRA components and is agnostic to components deployed on the same or different platforms, communicating directly or through the use of the communication infrastructure. This includes how components with differing levels of communications ‘awareness' interact with components which provide communications capability.

- Data Exchange: Explains the exchange of data and information between systems, at least one of which is PYRAMID compliant, and the interactions of the components which provide distribution facilities to support this.

2.3 Scope Summaries for Modelling Principles

- Component Connections: Explains how components can be connected in a deployment of the PRA to produce complex systems.
- Component Extensions: Defines what is meant by a component extension, considers their benefits and provides guidance on their appropriate usage.
- Data Driving: Explains how configuration data could be utilised in a deployment of the PRA to modify component behaviour to cater for different conditions; for example, to cater for different role fit equipment or operational requirements.

2.4 Safety and Security Policies

- Safety Analysis - Explains how safety analysis has been applied to the PRA and why. Note that the safety analysis does not place any requirement on an Exploiting Programme.
- Security Approach - Explains the approach to security aspects, how it has been applied to the PRA and why. Note that the security analysis does not place any requirement on an Exploiting Programme.

3 Introduction to Components

The PRA is composed of components each responsible for a specific, discrete area of subject matter. These PRA components are defined in Appendix B.

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 which layer of the Control Architecture the component is in, the standard pattern of use, and examples of use.
- A Service Summary Diagram, which shows the services and interfaces that have been defined for the component.
- A set of Responsibilities, which describe the behaviour it 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.
- 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 section also includes a semantics diagram, showing the relationships between entities.
- A Design Rationale containing a set of assumptions, design considerations (including exploitation considerations), safety considerations and security considerations to be evaluated during development to ensure that the component is developed, used and maintained correctly. Although they are not requirements for an Exploiting Platform, the design considerations will provide Exploiters with guidance on which policies are particularly relevant to a component (with the exception of the Control Architecture, Capability Assessment and Storage policies that are applicable to all components), areas where extensions may be useful, as well as other factors which should be taken into account (e.g. applicable standards). The exploitation considerations will highlight specific design choices that may need to be made for a deployment. The safety considerations provide a statement of, and rationale for, the component's indicative Item Development Assurance Level (IDAL). The security considerations provide a statement of, and rationale for, the component's indicative security classification.
- Services, which comprises:
    - Service Definitions, which are a detailed description of each service.
    - A Service Dependencies Diagram, which shows how the services depend upon one another.

The different parts of the component definition are different views on its subject matter. The behaviour of a component is defined at a high level by its responsibilities and in more detail by its provided services. The consumed services of a component define what the component depends on to achieve that behaviour.

3.1 Services

The PRA uses a service modelling approach which allows the component services, and therefore its specification, to be defined without reference to the internals of a component.

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 semantics diagram.

3.1.1 Interfaces

Interfaces are containers of information which have no directionality. Information can be contained 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 are themselves also expected to be made specific to an Exploiting Programme, i.e. either the name or description made specific to the programme while remaining within the PRA component's subject matter. The PRA does not specify how write access to an attribute is controlled.

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.

3.1.2 Service Summary Diagram

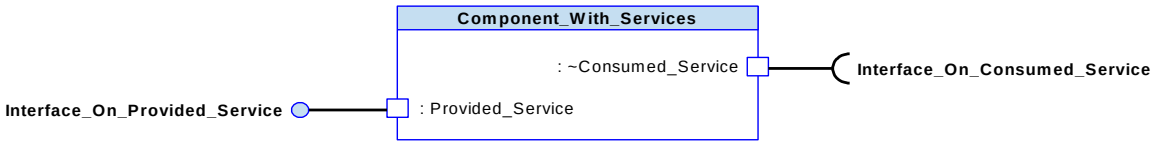

Figure 1: Service Summary Diagram

The Service Summary Diagram is an overview of the services and their respective interfaces on a component. As can be seen on Figure 1: 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.

3.1.3 Service Diagrams

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

3.1.3.1 Service Definition Diagram

A Service Definition Diagram defines each 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 be generalised by 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.

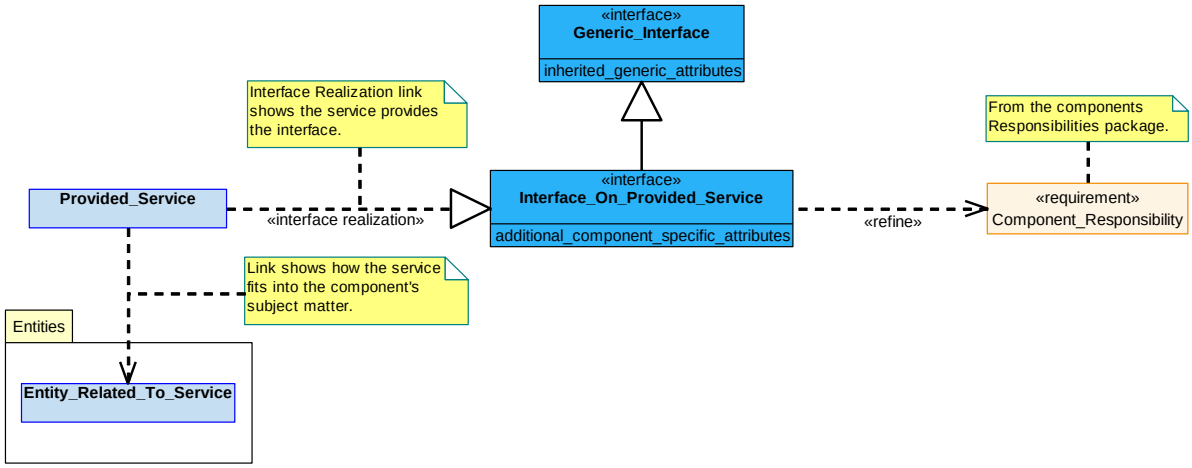

Figure 2: Provided Service Definition Diagram

In Figure 2: 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.

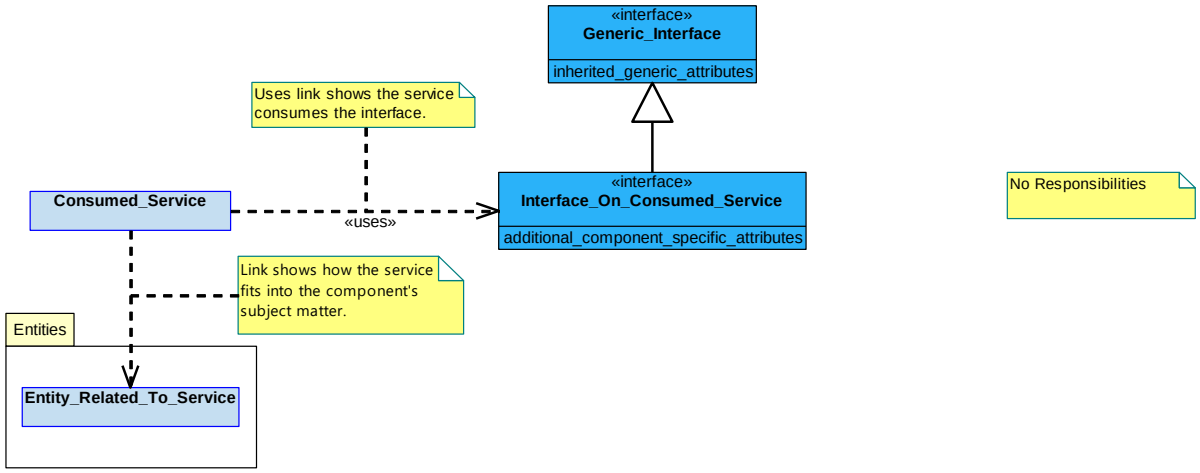

Figure 3: Consumed Service Definition Diagram

In Figure 3: Consumed Service Definition Diagram, the “uses” link shows that this is a consumed service.

3.1.3.2 Service Policy Diagram

A Service Policy Diagram gives an alternative view of a service, focussing on the textual descriptions of the interfaces, activities and of the service itself. In the case of provided services, the related responsibilities and their descriptions are also shown. See Figure 4: Provided Service Policy Diagram and Figure 5: Consumed Service Policy Diagram.

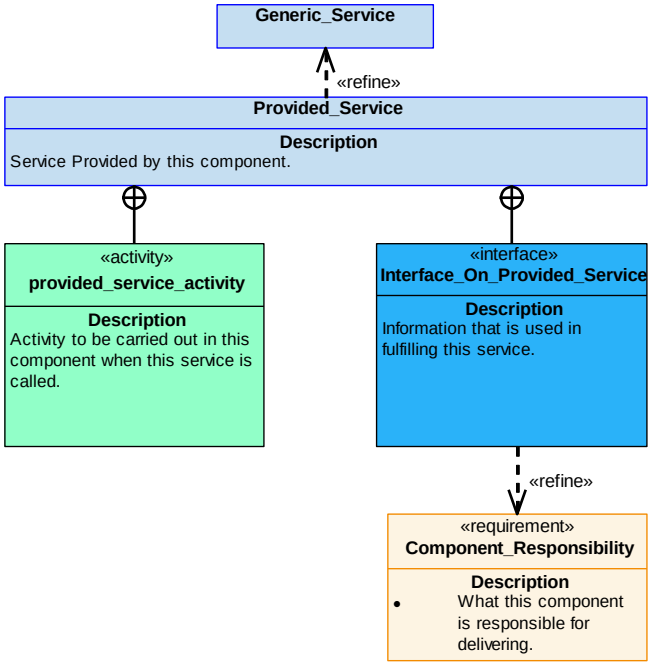

Figure 4: Provided Service Policy Diagram

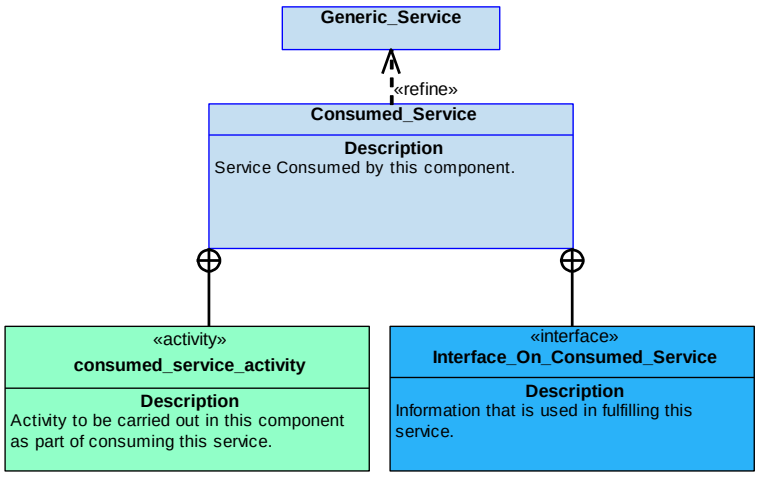

Figure 5: Consumed Service Policy Diagram

3.1.4 Service Dependencies Diagram

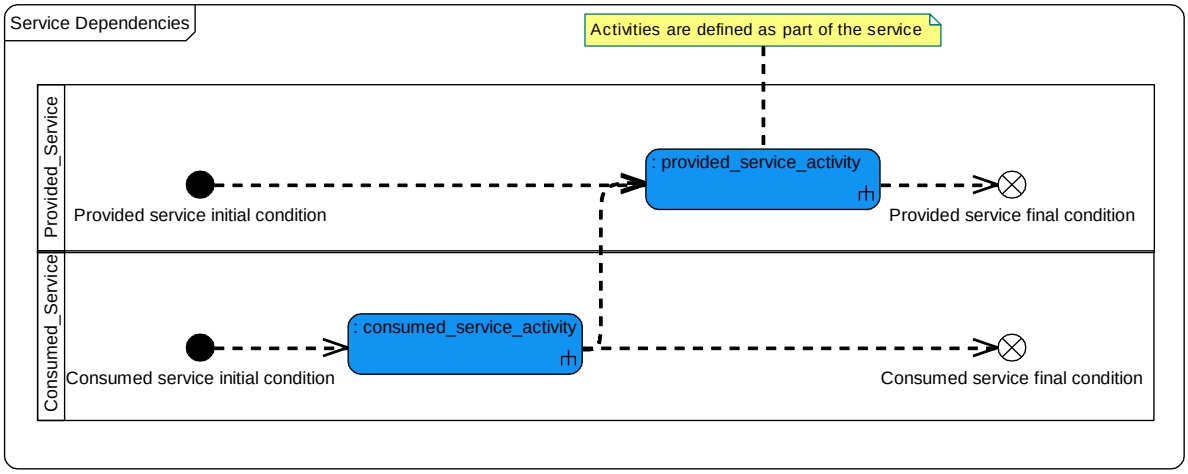

Figure 6: Service Dependencies Diagram

A service dependencies diagram (see Figure 6: 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.

3.2 Services Not Defined in the PRA

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

3.2.1 Services Internal to the Scope of a Component

The services to support interactions with a component and an alternate version (or extension) of the same component are not included in the PRA. This is irrespective of where the alternate version is located, whether within a single deployment or across separate deployments.

Examples of such interactions are:

- **Instanced Components:** Services between different instances of a single component
- **Component Extensions:** Services between a parent component and its extension components

Where the capability of a component is partly provided by non-PRA software, this is equivalent to an alternate instance of that component satisfying its role. As such, these interactions fall under the ‘Instanced Components’ group above, and are therefore not included in the PRA.

3.2.2 Services Crossing the PRA Scope Boundary

In order to achieve their objectives, implementations of PRA components may 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 a resource component and the resource that is managed
- **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

3.2.3 Specialised Functions

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 configuration data 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

4 Introduction to Interaction Views

An Interaction View (IV) is used to illustrate how a set of components could be integrated to achieve the system behaviour described in an IV Scenario. An IV Scenario, together with its accompanying diagram, describes an abstract use of the system and is based on a particular aspect of functionality that an exploiting platform could be expected to provide.

The IVs are intended to add context to the component definitions and provide a level of validation for them, as well as illustrating the intended use of components in PYRAMID deployments. System Integrators can use interaction views as examples of combining components to achieve system functionality. Component developers can use interaction views to give examples of context for a component of interest in order to determine potential uses of the component in deployments.

The PRA IVs are provided in Appendix C.

4.1 IV Scenarios

An IV Scenario is captured as a UML/SysML use case, which contains:

- A textual description providing context for the IV Scenario, including any assumptions or exclusions considered when scoping its coverage;
- The pre- and post-conditions to further define the scoped coverage of the IV Scenario;
- The flow of the scenario, given by a communication diagram and descriptive text, which provides a narrative to support understanding of the IV Diagram and the interactions represented on it.

An IV Scenario is an example of how a core set of components can interact, as such it may not include provision of all of the information required to implement the described functionality. Like the scenario itself, the flow describes a specific possible implementation. Similar functionality may be achieved by using an alternate set of components or flow as determined by an Exploiter.

4.2 IV Diagrams

An IV contains an IV Diagram (in the form of a UML communication diagram) for each use case, which illustrates how the components interact to achieve the behaviour described in the associated IV Scenario(s). Any entities described in the IV Scenario that are outside the boundary of the IV Diagram are represented by actors. The numbers on the IV Diagram events are in the order of the component interactions described in the flow. An asterisk before the name means that event starts at the indicated number and then continues to occur from that point on. An example IV Diagram can be seen in Figure 7: Example Interaction View IV.

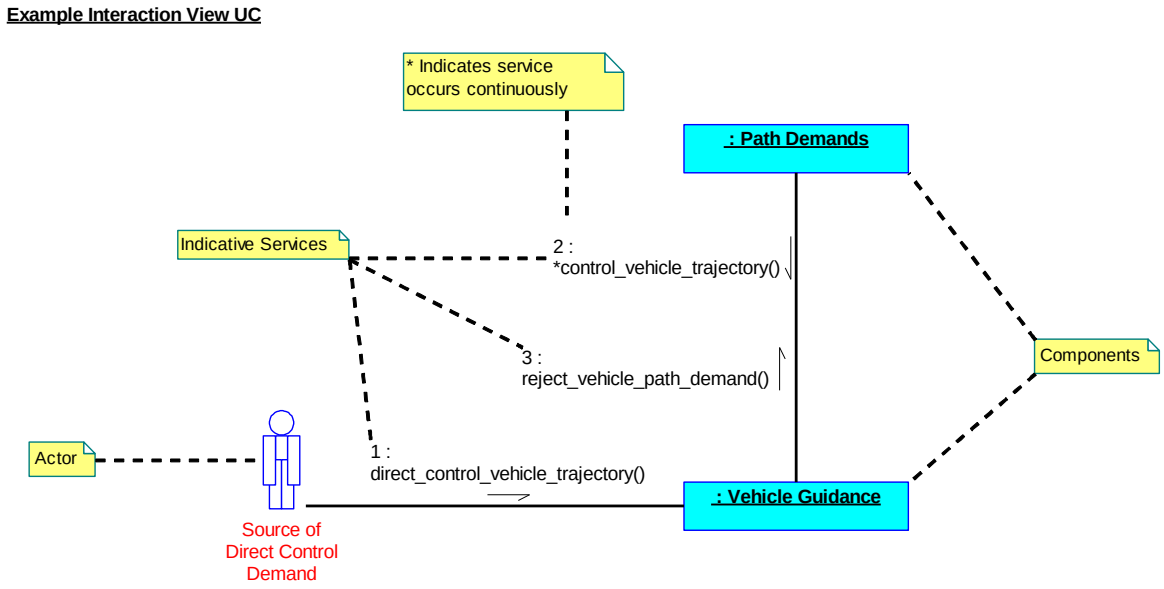

Figure 7: Example Interaction View IV

4.2.1 Components

Only the core set of components required for representing the requirements of the IV Scenario are included; this is intended to reduce repetition and increase readability. For example, the Navigation IV illustrates how the Location and Orientation component interacts with a number of other components to provide positional information to the system. This positional information is required by the Vehicle Guidance component on the Vehicle Movement IV, however that IV only shows the Location and Orientation component - the other supporting components are not shown as the scope of that IV does not need to consider how the positional information is generated, only that it is.

4.2.2 Interactions

Associations between components indicate specific support that one component requires of another. The associations are linked to the responsibilities of the component that fulfils the requirement, and labelled accordingly. These responsibilities will ultimately be realised by the collaborative use of the services on the components. There may be a dialogue between the components involving multiple services: for brevity and clarity this level of detail is not shown on the diagrams.

Within the textual flow, certain behavioural patterns are used, such as:

- One component will ask other components to collaborate in determining a solution to a problem. This represents not just a simple request but an exchange of provided and required services culminating in an agreed solution.
- One component will tell another to perform a predetermined action. This represents the invocation of provided services on the receiving component to implement the determined solution to a problem.

In the example in Figure 7: Example Interaction View IV, the Path Demands component requires the trajectory of the vehicle to be controlled. The Vehicle Guidance component has a defined responsibility to determine the trajectory of a vehicle which satisfies that requirement.

4.2.3 Actors

These are defined in the context of the IV Diagram they appear on. They are used to represent the role of an entity that exists outside of the IV Diagram boundary (including components that are not within scope of the IV) but interacts with the components within the boundary. The actor definition includes a supporting description to clarify its role in the IV Scenario.

A Appendix A: Policies

This appendix defines the PRA architectural policies introduced in section 2.

A.1 Architecture Wide Policies

Architecture wide policies are policies that provide consistent ways of developing the system-wide aspects of a PRA based deployment, in line with the PYRAMID KURs.

A.1.1 Control Architecture

A.1.1.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Dependency Management
- Autonomy
- Human-Machine Interface
- Constraint Management
- Capability Assessment
- Resource Management
- Interaction with Equipment
- Multi-Vehicle Coordination

A.1.1.2 Introduction and Scope

This policy describes how the PRA has been designed to enable the architect of an Exploiting Platform to implement system-wide control through use of a layered architecture.

A.1.1.2.1 What is the Control Architecture?

A deployment of the PRA will be given mission objectives that it needs to achieve. The control architecture shows how components could be arranged to work together to achieve these objectives.

The Control Architecture policy defines layers of components which have different roles in the achievement of mission objectives. It shows how a deployment of the PRA can achieve such objectives by the successive delegation of requirements through the components to coordinate the use of resources.

A.1.1.2.2 Why is the Control Architecture Important?

The PYRAMID KURs require the PRA to be configurable and to support a range of operational scenarios. The control architecture policy plays an important role in meeting the PYRAMID KURs by allowing resources to be used, and coordinated, in the best way to achieve a mission objective, both by allowing a wide range of configurations and by allowing flexible communication between components within a configuration.

The control architecture increases readability of the architecture for potential PRA exploiters by giving context to the components.

A.1.1.2.3 Aim of this Policy

This policy explains how the PRA has been designed to provide a flexible architecture to enable the planning and execution of mission objectives or specific mission requirements (i.e. tasks or actions input directly).

The PYRAMID KURs which are applicable to this policy are:

- **Configurable**: This policy allows a deployment to be configured in support of different types of task, action and resources.
- **Incorporate Future Growth**: This policy provides a framework within which different types of task, action and resource can be added with minimal impact.
- **Scalable**: This policy provides a framework which allows different numbers of components to be deployed flexibly. It particularly supports differing levels of autonomy.
- **Utility Across A Range Of Missions**: This policy shows how the PRA can be deployed for different mission objectives or requirements that require different mission behaviour in different operational scenarios.

A.1.1.3 Overview

The control architecture policy can be summarised as follows:

- Each component is allocated to a layer which helps bound its role in meeting mission objectives.
- Requirements are delegated through each layer such that decisions can be made at the most appropriate layer, and problems resolved at the lowest possible layer (e.g. conflicting requirements) while providing overall control.
- The responsibility for and control of resources occurs at the lowest appropriate level of the architecture.

A.1.1.4 Objective Breakdown

By definition, a mission objective is achieved by carrying out a number of tasks. A task is an activity to be carried out in support of a mission objective and is executed via specific actions that coordinate the use of resources of the system. This breakdown is shown in a simple form in Figure 8: Objective Breakdown. For more complex mission objectives more levels of breakdown may be necessary to obtain the solution. For example, it is possible for tasks to be derived from other tasks, actions from other actions, and steps from other steps, see Dependency Management policy.

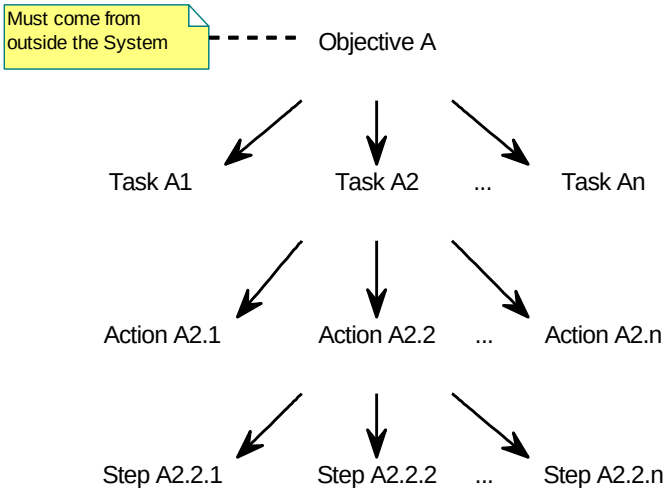

Figure 8: Objective Breakdown

A.1.1.5 Layers

The layers of the control architecture are shown in Figure 9: Control Architecture Layering. The control architecture is arranged into four layers. These layers are based on the principle that mission objectives are met by tasks, which are achieved by carrying out actions, using resources. A deployment with low levels of autonomy is likely to focus on lower layers of the architecture, with simple, or absent, components at the higher layers. Not all components fit into a layer: many components provide information or general services in support of the control architecture. These service components provide services to all layers of the architecture.

Interaction with an external system, standalone equipment, or an operator can be performed with any PRA component. The Interaction with Equipment policy explains how capability and resources provided by equipment external to the system can be integrated at different levels of abstraction under a PRA deployment and accessed by the components at different layers within the architecture. Therefore, in some cases an external system, standalone equipment, or operator may be selected to perform the role (in full or in part) of a PRA component. For example, in less autonomous deployments the operator may fulfil the role of the Objectives component and Tasks component.

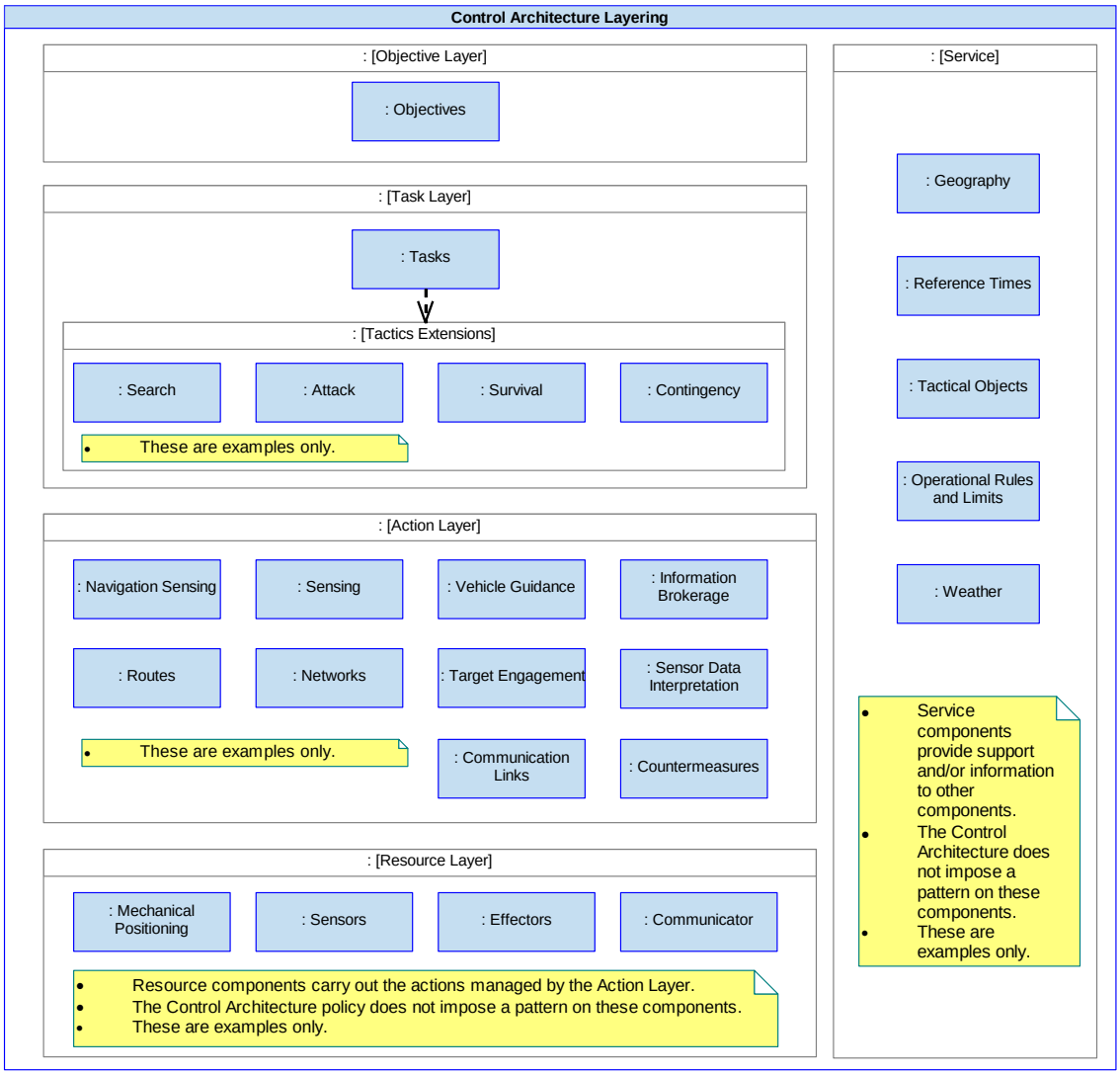

Figure 9: Control Architecture Layering

A.1.1.5.1 Objective Layer

The Objective Layer contains a single component called Objectives. This component will be provided with requirements from outside the system which define the mission objectives and form the mission context. The role of the Objectives component is to manage these mission objectives through the execution of tasks.

A.1.1.5.2 Task Layer

The Task Layer contains a single component called Tasks. The role of the Tasks component is to execute tasks through the coordinated execution of actions. The Tasks component is supplemented by a set of tactics extensions which provide the tactics associated with different types of task, i.e. different ways to break down a task into actions. Tactics extensions interrogate action components, either via their Tasks parent or directly to the action components (see Component Extensions) to identify possible sets of actions to achieve the task. The Tasks component has overall responsibility for achieving tasks placed on it by co-ordinating the execution of actions (see the Dependency Management policy).

A.1.1.5.3 Action Layer

The Action Layer contains multiple components (see Figure 9: Control Architecture Layering for examples). The role of these components is to perform actions by managing the resources available to them (see Resource Layer section). Each action component addresses actions within a particular area of subject matter. An action component is responsible for identifying a solution which can achieve the requirements given to it. These requirements vary widely - Appendix C: Interaction Views gives many examples. Action components are responsible for identifying any dependencies (e.g. activity or information) that they have in order for their actions to be satisfied. These dependencies could be satisfied by the Tasks component or other action, resource or service components (see the Dependency Management policy for further details).

A.1.1.5.4 Resource Layer

The Resource Layer contains components which control resources in support of the actions that are managed at the Action Layer. This policy does not impose any specific conditions or pattern on components in the Resource Layer.

A.1.1.5.5 Service Components

Service components are those components which do not fall into any specific layer of the control architecture. Service components provide information or general services in support of the control architecture, and can be used by other components from any layer of the control architecture requiring that information or support. This policy does not impose any specific conditions or pattern on service components.

A.1.1.6 General Principles

The following are general principles of the control architecture:

- Control is distributed through the system. There are no "management" components: each component manages the activities which are within its subject matter.
- There is no rigid control hierarchy. The definition of layers does not restrict communication between components, or the way they can delegate requirements to each other.
- Not all deployments will implement all layers of the control architecture, or all components within the PRA.
- Resources are available for any action or task which can make use of them.
- Each component is responsible for satisfying its own requirements.
- Each component is responsible for identifying its dependencies and deriving requirements to get them satisfied. See the Dependency Management policy.
- A component is responsible throughout the lifecycle of a requirement: for its planning and execution.
- The exploiting platform can evaluate and track multiple solutions within a solution space.
- A Tasks component can interact with other instances of Tasks components for purposes such as resolving inter-vehicle task demands in multi-vehicle scenarios (see Multi-Vehicle Coordination policy), or for intra-vehicle hierarchical task coordination where required.
- Action components can coordinate multi-vehicle actions (see the Multi-Vehicle Coordination policy).
- The Constraint Management policy applies at all layers of the control architecture.
- Safety interlocks may be employed in the Action or Resource Layer to ensure that unsafe actions do not take place.
- Requirements placed on a component to perform an activity in support of an objective, task or action is assessed on its achievability. For a requirement to be achievable, a component must first have a feasible solution (or plan). However, the infeasibility of a particular solution does not mean that the requirement is unachievable.

A.1.1.6.1 Breakdown of Objectives by Components

Figure 10: Component Interactions to Breakdown Objectives shows how component interactions correspond to the objective breakdown shown in Figure 8: Objective Breakdown. The Objectives component place task requirements onto Tasks; Tasks place action requirements onto components in the Action Layer; and the action components place requirements on other action components to coordinate and command associated activities (see the Dependency Management policy). The action components can also issue commands to resource components to carry out the steps required to fulfil the objective. Whether something a component requires to happen is treated as a command or coordination is dependent on the deployment design.

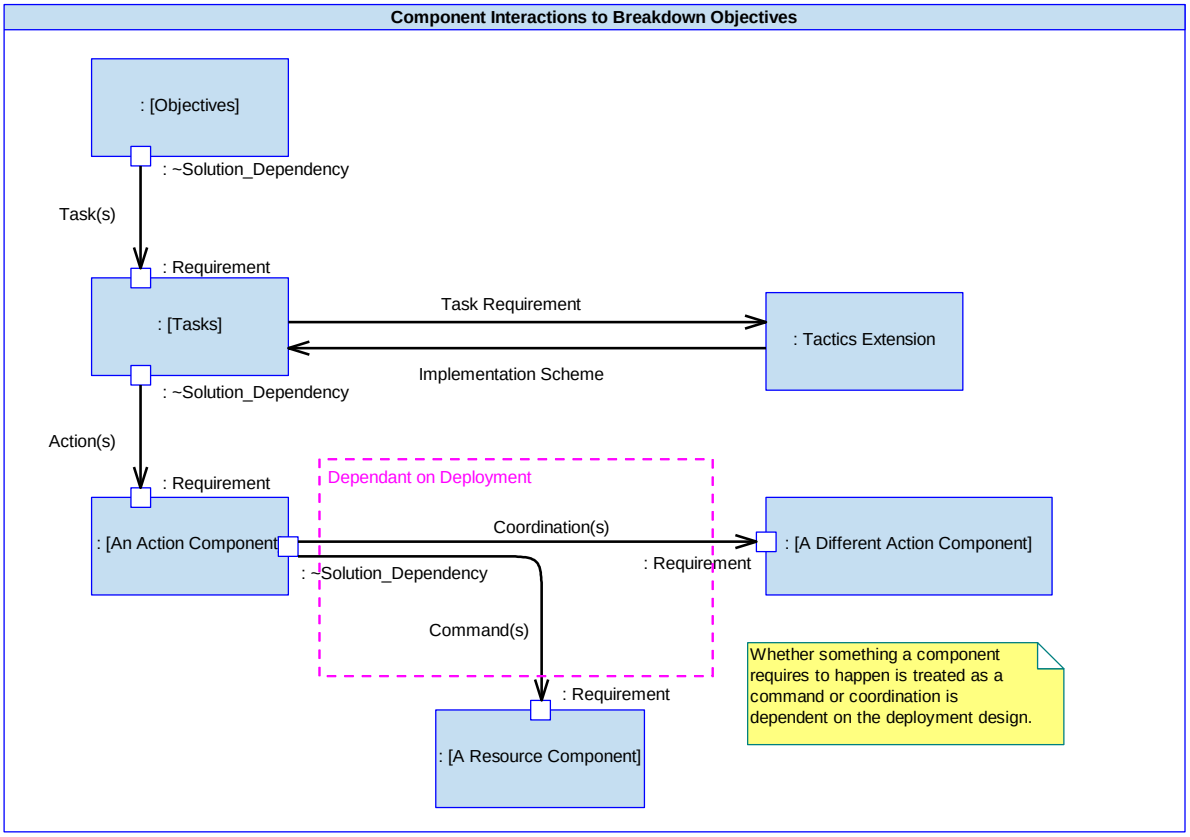

Figure 10: Component Interactions to Breakdown Objectives

The following concepts (covered in detail by other policies) are also critical to how objectives are suitably broken down by components, and turned into requirements at all levels of abstraction, to allow objectives and requirements to be met in a way that is both operationally viable and supports safe operation:

- Constraint Management - explains how constraints can apply to and be applied by any component in the control architecture and used to limit how an activity (e.g. activities required to satisfy an objective) can be achieved. (See Constraint Management policy.)
- Capability Assessment - explains how changes in the capability of systems can impact the ways to achieve activities (e.g. activities intended to satisfy an objective). (See Capability Assessment policy.)
- Resource Management - explains how different activities broken down from the same objective or different objectives can compete for finite resources. The Resource Management policy provides a framework that can be adopted to help ensure that finite resources are used in a way that most appropriately satisfies an objective or set of competing objectives.
- Autonomy - explains how Tasks and different action components acting together can provide a system capable of supporting different levels of operator interaction and authorisation associated with tasks, actions or steps that form part of a solution. (See Autonomy policy.)

A.1.1.6.2 Operator Interaction

Authorised operators may interact with components from any of the control architecture layers. For a highly autonomous system, interaction will be mostly at the Objective Layer, but where operators have more fine-grained control they will interact directly with the lower layers of the system.

Figure 11: Mission Planning Interactions shows how a mission planner may interact with any layer of the control architecture, defining objectives, tasks, actions, steps, or even the use of specific resources as required in the context of a mission. The mission planner shown on the diagram could be either a human operator or output from a separate mission planning system. A mission planning tool, for example, could provide a detailed plan that includes the timing of actions and the use of resources. A component that receives requirements from an operator will carry out those requirements by fulfilling its responsibilities in the usual way. It must maintain consistency with other components, negotiate resource use, handle dependencies and use other components as necessary to carry out the requirements. Except for those components that explicitly represent operators as part of their subject matter, components will not explicitly know about an operator but they will need an understanding of the requirements being placed upon them along with knowledge about their source or origin. This is needed to support prioritisation and the application of rules to resolve any issues with fundamentally conflicting requirements from different sources. It is up to the Exploiting Programme to determine whether a component needs to consider the source of the information that is provided to it.

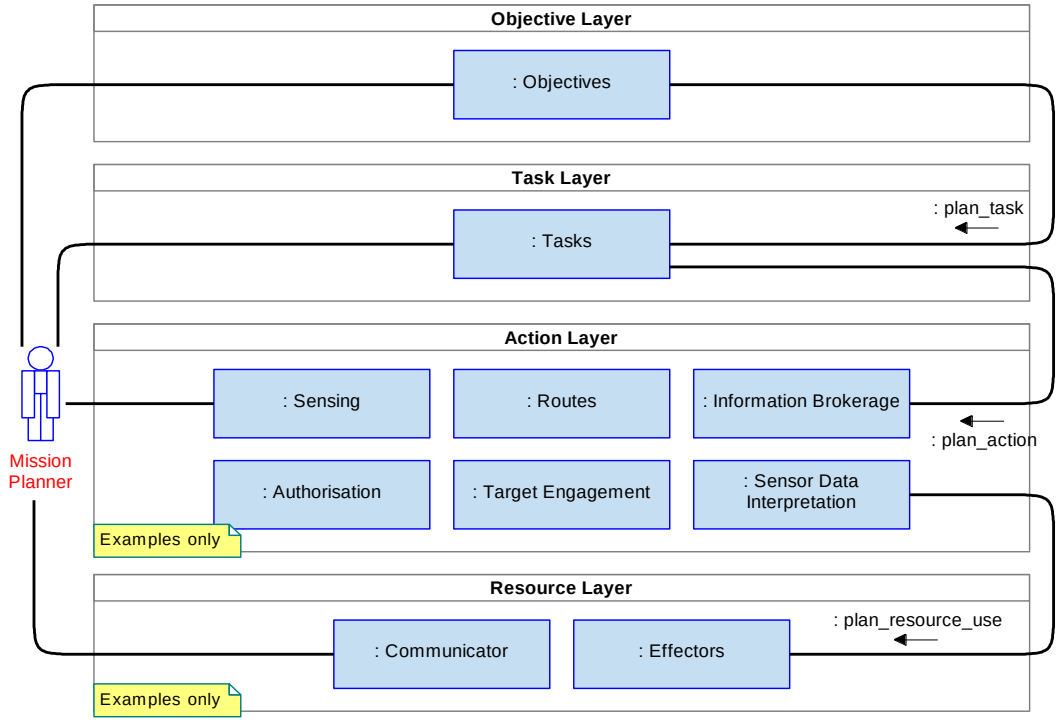

Figure 11: Mission Planning Interactions

Although Figure 11: Mission Planning Interactions shows interactions for mission planning, the same principle would apply in other cases. During the execution of the planned mission, for example, the level of interaction could vary as the operator chooses to increase or decrease the amount of autonomy that the system is allowed. See the Autonomy policy.

All PRA deployment interactions between an authorised operator and the system are mediated by HMI components, as described in the Human-Machine Interface policy.

A.1.2 Constraint Management

A.1.2.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Dependency Management

A.1.2.2 Introduction and Scope

This policy explains how the concept of managing system constraints has been applied to the development of the PRA.

A.1.2.2.1 What are Constraints?

In the context of the PRA a constraint is a limitation on the behaviour of a PYRAMID compliant deployment at any level (whole system or constituent part). A constraint is the actual limitation rather than the rule or policy from which that limitation is derived; e.g. a rule might prohibit the use of electronic warfare measures, however the associated constraint will be on the use of a particular piece of equipment. Some of these limitations are fixed, e.g. the aircraft's structural limits, and require no system management. Others are more flexible and need to be managed so that the exact extent of the limitation imposed on system behaviour at a point in time is understood. Thus constraint management can be considered to be the structured imposition or relaxation of limitations based on rules and policies. Note: As far as the PRA is concerned, external rules and external policies are equivalent in that they will both lead to constraints. For clarity, this architectural policy will only refer to rules, therefore a reader considering an external policy such as an Emissions Control (EMCON) policy should apply the concepts identified here for rules to the external policy under consideration.

A.1.2.2.2 Types of Constraints

There are many types of constraints, but for the purposes of the PRA, at a system-level they can be categorised into two broad categories:

- **Fixed:** Fixed constraints are those imposed by the physical capabilities of the vehicle. Although a PRA deployment could be configured to make varying levels of allowance for such a constraint the underlying constraint itself cannot be managed by a deployment of the PRA. Therefore they are considered to be outside of the scope of this policy. Rules adopted to account for fixed constraints are within scope and fall within the category of flexible rule-based constraints outlined below.
- **Flexible:** Flexible constraints are those imposed as a result of some form of rule, policy or variable consideration. These are subject to management by the system based on the rules from which they are derived and so are within the scope of this policy. Flexible constraints can be further subdivided into two sub-categories:
    - **Rule-Based:** Rule-based constraints are derived from a fixed set of rules and apply to all activities; there may be conditions attached to these rules to determine when a rule is applicable, e.g. vehicle location. Examples of rule-based constraints are those derived from rules of engagement or emissions control policies.
    - **Solution-Based:** Solution-based constraints are derived as part of a problem solution; they can only constrain the system within the bounds of the currently applicable rule-based constraints (see Figure 12: Effect of Constraints on PRA Deployment Decision Making). Examples of solution-based constraints are those derived from tactics.

A.1.2.2.3 Why are Constraints Important?

All actions undertaken in the world are subject to some form of constraint, ranging from the physical capabilities of the person or system undertaking the action, through to legal or moral concerns. A system acting autonomously (on behalf of a human) or semi-autonomously (in co-operation with a human) should act in a manner consistent with the desires of the human who is ultimately responsible for the actions of the system. Equally a system acting in an advisory role should provide advice that is valid and therefore can be accepted: advice which is invalid, and therefore would never be accepted because it breaches rules is worse than no advice at all because it is a wasteful distraction. To achieve these goals the system needs to have its range of actions bounded in some way.

A.1.2.2.4 Aim of this Policy

This policy explains how the PRA has been designed to support the management of constraints in a manner consistent with the PYRAMID KURs.

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** The currently active rules to which a deployment of the PRA is subject will evolve to accommodate changes in the tactical and strategic environment. This policy aims to allow rapid modification of the currently active rules.
- **Exploitable:** The rules from which rule-based constraints are derived may vary between different deployments of the PRA, therefore this policy allows for the use of different rules and policies without significant system modification.
- **Incorporate Future Growth:** Rules evolve to accommodate new technologies and capabilities. This policy allows new functionality to be introduced without affecting the interpretation of the rules and allows new rules to be introduced without affecting the existing functionality of a system.
- **Scalable:** A single rule could constrain multiple areas of the system. This policy allows aspects of the system to be omitted by either design or role fit without requiring modification of either the rules or their interpretation.
- **Security Accreditable:** Some of the rules (e.g. rules of engagement) from which constraints are derived are highly sensitive and classified accordingly. To ease security accreditation this policy seeks to minimise the areas of the system which hold and understand the rules.

A.1.2.3 Overview

The policy for constraint management can be summarised as follows:

- Rule-based constraints are managed in a centralised manner: a single component is responsible for the interpretation of rules and the determination of which constraints the system is consequently subject to.
- Solution-based constraints are managed in a more distributed manner. They form part of the solution to a requirement and may be identified either in determining a solution to satisfy that requirement or as a consequence of addressing the requirement whilst taking in to account an imposed rule-based constraint. In the case where a general rule-based constraint that has been imposed, a more specific constraint may be derived as needed to support the solution. From the perspective of any one component planning a solution, the component does not know whether a solution-based constraint would be interpreted either as a constraint or a requirement on a different part of the system.

A.1.2.4 Applicability of Rules

The rules with which a system must comply are generally defined using a structured approach which progressively applies more detail at each stage of definition.

- First a 'library' of rules is established; this library generally covers a broad spectrum of users and hence not all rules in the library will be relevant to a particular usage of a system. Nevertheless any rules to which a system is subject would be drawn from the library and it is the library that provides the set of rules which would need to be related to system constraints.
- Next, the rules in force for a particular set of circumstances are defined; these are the 'currently active' rules. Examples of these might be the rules of engagement in force in a particular theatre of operations or the EMCON policy in force in a particular geographic area.
- However even the 'currently active' rules will have conditions applied to them leading to the final stage; the determination of which rules are 'currently applicable' (i.e. the rules applicable to the system in 'situation now'). An example of this is a rule of engagement which prohibits certain activities when certain conditions are met, thus while a rule may be active when the system is in a certain area the associated constraints will only be applied when the defined conditions are met. Another example is a rule regarding the torque limit on the engines of a twin-engine helicopter. In the event of a loss of an engine the conditions associated with the rule may now permit over-torque of the remaining engine in order to get the helicopter home. This is sometimes referred to as the concept of 'soft limits', constraints which can be relaxed when certain conditions are met.

A.1.2.5 Effects of Constraints on PRA Deployment Decision Making

The same constraint can be in place for both rule and solution-based reasons; from the perspective of the constrained area of the system the reason the constraint is in place is not important. Take the following example rules and tactics:

- **Rules of Engagement (**RoE**):** Rule 130C - Use of electronic warfare measures is permitted (example taken from Sanremo Handbook on Rules of Engagement Ref. [17]).
- **EMCON Policy:** Use of Electronic Countermeasures (ECM) in region 1 is prohibited.
- **Tactic:** While within 50km of potential threat, emissions in the direction of potential threat are prohibited (potential threat located in Threat area).

All of these can lead to a constraint on the use of a wingtip Electronic Countermeasures (ECM) emitter. Figure 12: Effect of Constraints on PRA Deployment Decision Making illustrates how at different points the examples above cause imposition of the same constraint. Within the entire area of the diagram, RoE permits the use of wingtip ECM emitters, however, at point A, the port wingtip emitter is constrained by the EMCON policy while the starboard emitter is constrained by both EMCON and tactics. When the aircraft moves into region 2, at point B, the port wingtip emitter is no longer constrained (as the aircraft is outside region 1 the EMCON policy rule no longer applies) but the starboard emitter continues to be constrained.

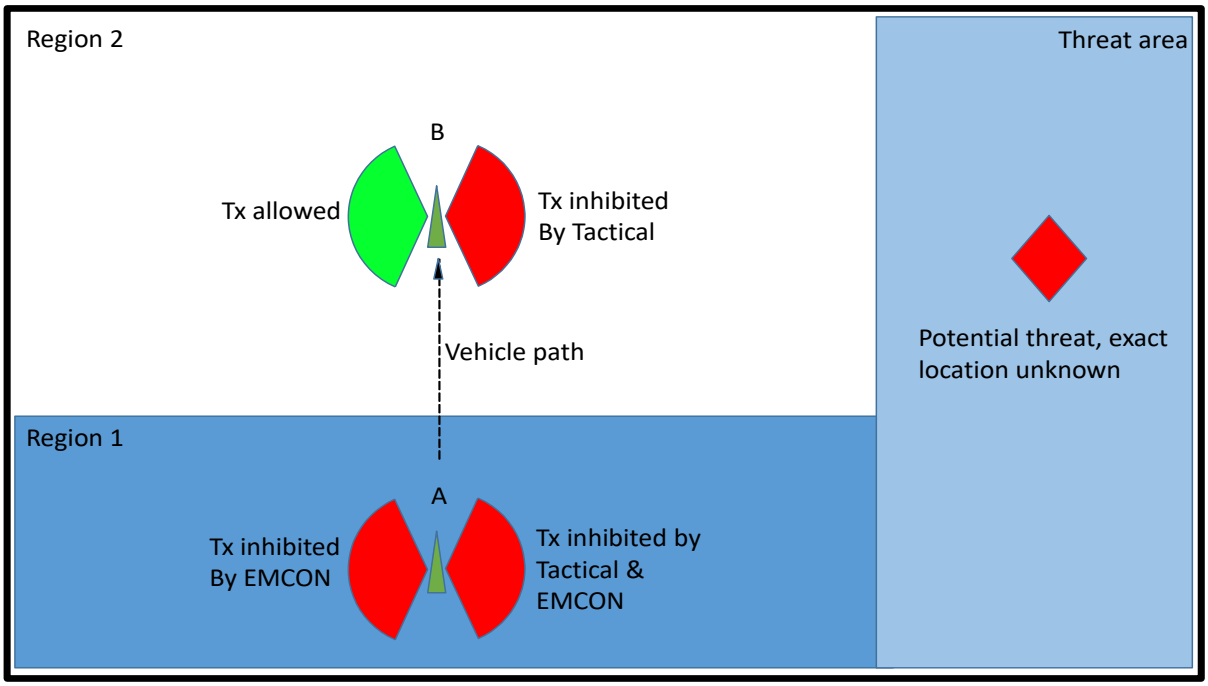

Figure 12: Effect of Constraints on PRA Deployment Decision Making

A.1.2.6 Management of Constraints

A.1.2.6.1 Management of Rule-Based Constraints

Rule-based constraints are managed by the Operational Rules and Limits component. This component's function is to understand the rules and the criteria for rule applicability; it will also be aware of the constraints that can be imposed on the system and their relationship to the rules. It will not be aware of the implications of the constraints it determines or what areas of the system are affected by them. The Operational Rules and Limits component will establish which rules are currently active and assess the current conditions to determine if a rule is applicable. If it determines that the criteria for rule applicability have been met it will set the associated constraint. It will also forecast changes to rule applicability.

Restricting knowledge of the rules and their applicability to this one component minimises the number of components exposed to potentially highly classified rules and ensures they are applied uniformly. A limit identified by Operational Rules and Limits is imposed upon another component as a constraint which it must comply with. If events beyond the component's control cause the limit to be breached inadvertently this will result in the component's need to modify the solution and respond appropriately to prevent the breach from continuing to occur. The component's constraint service will report the breach; Operational Rules and Limits will then determine which rule has been broken.

A.1.2.6.2 Management of Solution-Based Constraints

Solution-based constraints are managed across the architecture; they are imposed by whichever component develops the 'problem solution' which is dependent on that constraint being in place. Solution based constraints may emerge directly in the course of generating a solution to meet a requirement and are identified as constraints that must be observed by other components. As well as resulting from a requirement, solution-based constraints may also result from other constraints imposed on a component. Generally it is expected that solution-based constraints would be imposed by Objectives, Tasks, and Action components (see

the Deriving a Solution to a Requirement section in the Dependency Management policy). As with rule-based constraints, it is expected that the components determining the need for a solution-based constraint would have no knowledge of what areas of the system are affected by that constraint.

A.1.2.7 Component Composition

This policy requires certain behaviour and services from the components that are actively involved in constraint management. The interfaces to support this policy are included in the Component Composition. There is a Constrain Capability use case (see Use Cases) and two services, Constraint and Solution_Dependency. The examples below show how the pattern supports the management of constraints within the system.

A.1.2.8 Constraint Examples

This section gives examples of how components might work together when constraints have been placed upon them.

Note that some of the services shown on the components in Figure 13: Rule-Based Example and Figure 14: Solution-Based Example are generic, as defined in the Component Composition section. The actual services may have been specialised to the subject matter of the specific components, shown in the Component Set section.

A.1.2.8.1 Rule-Based Constraints Example

Figure 13: Rule-Based Example shows an example of rule-based constraints applying to a search task. Operational Rules and Limits has already determined which rules are active. From the active rules, it has also determined the rules that are applicable, and therefore need to be enforced, based on the relevant operating conditions. In order to meet these applicable rules, Operational Rules and Limits identifies the associated limits that need to be adhered to by other components. For example the limits may restrict the use of certain electromagnetic spectrum frequencies due to EMCON rules being enforced within the current operating region.

These limits are interpreted as constraints by the components that they apply to. In the unexpected case that one of these components fails to observe one of the constraints, it reports this (through the constraints service) and Operational Rules and Limits determines the associated rule that has been breached. This can be reported as necessary, such as to make a user aware of the situation.

Note that, whilst not shown, Operational Rules and Limits can also be requested to identify the limits that would be applicable in a hypothetical scenario rather than the current operating conditions. For example this could help determine if a planned future action will be feasible.

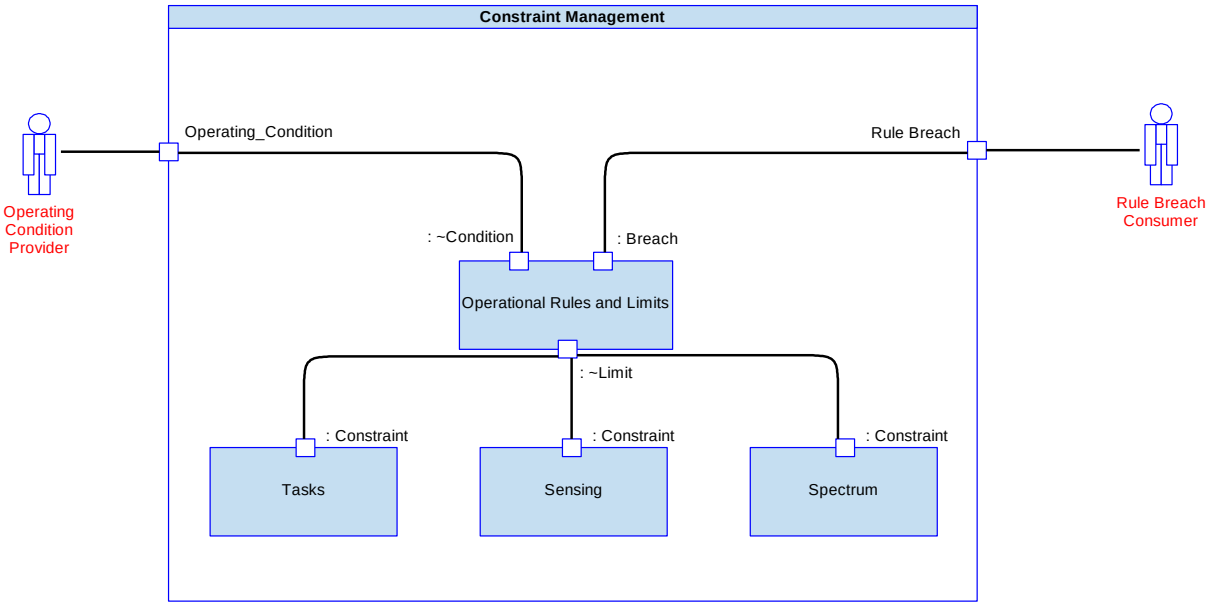

Figure 13: Rule-Based Example

A.1.2.8.2 Solution-Based Constraints Example

Figure 14: Solution-Based Example shows an example of solution-based constraints applying to a search task. The Tasks component determines any solution-based constraints associated with the task. For example, searching should not be carried out in a particular region. Sensing determines solution-based constraints associated with the sensing action, such as the exploiting platform must be a certain distance from the object of interest. A solution to the task is determined, taking into account all the solution-based constraints.

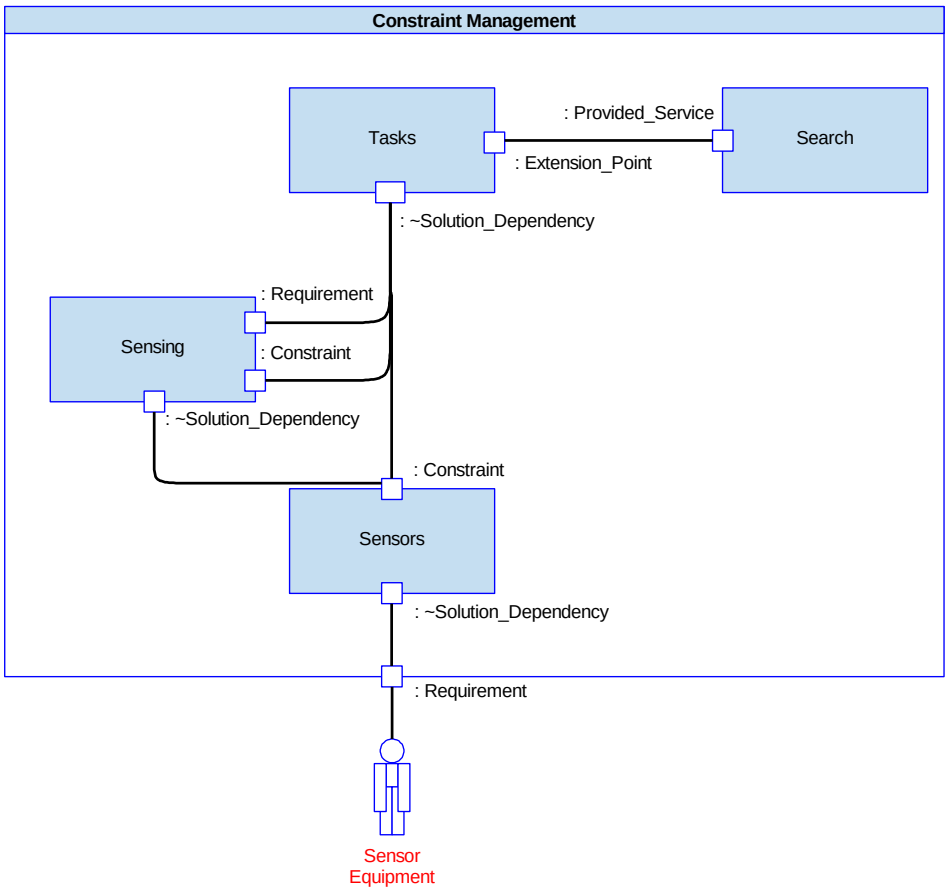

Figure 14: Solution-Based Example

A.1.3 Dependency Management

A.1.3.1 Pre-Reading and Related Policies

Prerequisite

- Control Architecture

Read in conjunction with

- Constraint Management
- Resource Management
- Component Connections

A.1.3.2 Introduction and Scope

Components can be allocated requirements through their services. When a component cannot fulfil the requirements that are assigned to it directly, it may need to request assistance from other components. This assistance is requested by generating a dependency. This policy describes how these dependencies are managed.

Through these dependencies, components can work together to fulfil the requirements that are placed on the system, along with any derived requirements that are generated internally. When the requirements cannot be met, mechanisms are provided to handle the shortfall and report back to the source.

Where a conflict exists in the use of a resource, the principles described in the Resource Management policy can be used to resolve conflicts as they arise.

Behavioural dependencies between components can be expressed in a component interaction pattern. Component interaction patterns are described in the Component Connections policy.

A.1.3.2.1 What is Dependency Management?

PRA components cooperate to achieve the requirements imposed on the deployed system. A major element of this cooperation is to manage the dependencies between components. These dependencies may have different properties:

- **Rigid Dependencies**

  There may be rigid dependencies between components created as part of executing a solution, such as a specific order of actions (e.g. transit to the search area before starting the search). Such dependencies can be implemented by setting up coordination points between actions.

- **Dynamic Dependencies**

  Components may also have dependencies that need to be managed more dynamically. For example, when sensor data is processed it may be discovered that the data was of insufficient quality to fulfil the requirements of the data processing action being undertaken. Consequently it may be necessary to use different sensing techniques, or to fine-tune vehicle manoeuvres, to obtain better data. This kind of dependency can be implemented by establishing a link between cooperating components. As long as the requirements and constraints of the original task are not violated, these links can be maintained until the activity is complete.

Rigid and dynamic dependencies between components require traceability to their resource demands to facilitate resource management and conflict identification. This is described in the Demand Traceability section of the Resource Management policy. Rigid and dynamic dependencies are illustrated in the examples in this policy, see Figure 25: Setting up a Sensing Task and Figure 26: Carrying out a Sensing Task.

A.1.3.2.2 Why is Dependency Management Important?

The identification and management of dependencies enables components to cooperate to achieve the requirements placed on them. It is important that dependencies are managed in an explicit and flexible way if a deployment is to be able to adapt to changes in its capability and to external events.

A.1.3.2.3 Aim of this Policy

This policy explains how the PRA has been designed to provide a flexible architecture to manage dependencies between components.

The PYRAMID KURs which are applicable to this policy are:

- **Configurable**: This policy shows how a deployment can be configured to support different types of interaction.
- **Incorporate Future Growth**: This policy provides a framework within which different types of dependencies can be added with minimal impact.
- **Scalable**: This policy provides a framework which allows different numbers of components to be deployed flexibly.
- **Utility Across A Range Of Missions**: This policy supports the PRA being deployed for different mission scenarios requiring different mission behaviour.

A.1.3.3 Overview

The dependency management policy can be summarised as follows:

- Components identify their dependencies when establishing a solution to the requirement placed upon them.
- To address these dependencies components will generate derived requirements which must be satisfied in order for them to achieve their solution.
- Components monitor the progress of their own solution and determine whether it remains feasible to achieve the requirements it was devised to meet.
- Components report their achievement and progress against their assigned requirements.
- Performance analysis can be used to monitor and improve the quality of solutions as they are being executed.

A.1.3.4 Component Composition

This policy requires behaviour and services provided by the Component Composition pattern. The sections below (with the exception of Planning Contexts and Performance Analysis) show how the pattern supports the management of dependencies within the system.

A.1.3.5 Deriving a Solution to a Requirement

When a requirement is assigned to a component, the component needs to provide a solution that can satisfy it. Each solution includes all the steps that the component can satisfy itself, along with solution dependencies that need to be fulfilled by other components. These solution dependencies are assessed by other components which determine whether or not they can be achieved.

By managing this chain of dependencies between the components, a complete solution can be generated in response to the originating requirement.

A.1.3.5.1 Solution Planning

When a component has been tasked to provide a solution to a requirement, there may be multiple solutions available. This set of solutions will be identified and evaluated in terms of cost, resource conflicts and quality. Once a solution has been selected, the requirement provider will be notified and the selected solution can be communicated to all recipients of solution dependencies. At this stage the requirements become part of the solution plan, resources are allocated, and the selected solution can be executed. Figure 15: Solutions to a Requirement shows a simple example where two solutions are considered.

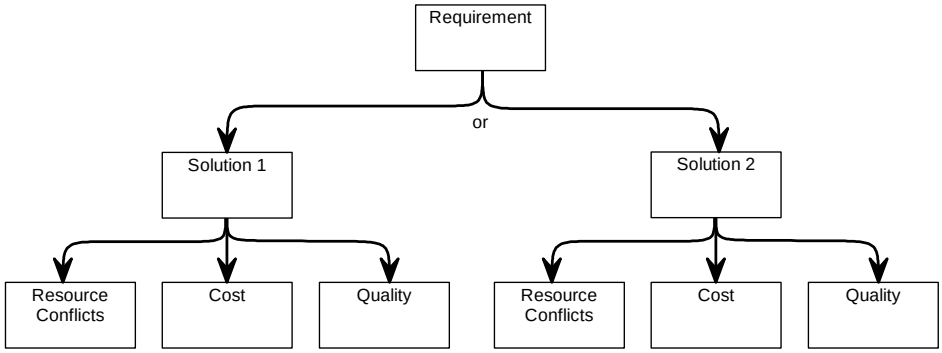

Figure 15: Solutions to a Requirement

A.1.3.5.2 Pattern Overview

The Determine Solution use case (see the component composition Use Cases) shows how a component derives a solution to a requirement. This service pattern uses a three phase exchange, where a requirement is placed, expected achievability is returned, and then if the requirement can be achieved the requesting component triggers enactment of the solution.

Since a component only knows how to satisfy requirements within its subject matter area, components must act together to find available solutions and select an overall solution. Two services enable this cooperation:

- The Requirement service allows a component to accept incoming requirements.
- The Solution_Dependency service allows a component to place derived requirements.

In this way, a high level request is broken down through the system to produce lower level requirements (see Figure 10: Component Interactions to Breakdown Objectives in the Control Architecture policy).

It is also possible that a solution dependency could result in a constraint being placed on another component (see the Constraint Management policy).

A.1.3.5.3 Use of the Pattern

Figure 16: Solution Derivation Example shows a simple example where a Requirement Provider places a new requirement on a Tasked Component. The Tasked Component uses two Contributing Components to achieve the solution.

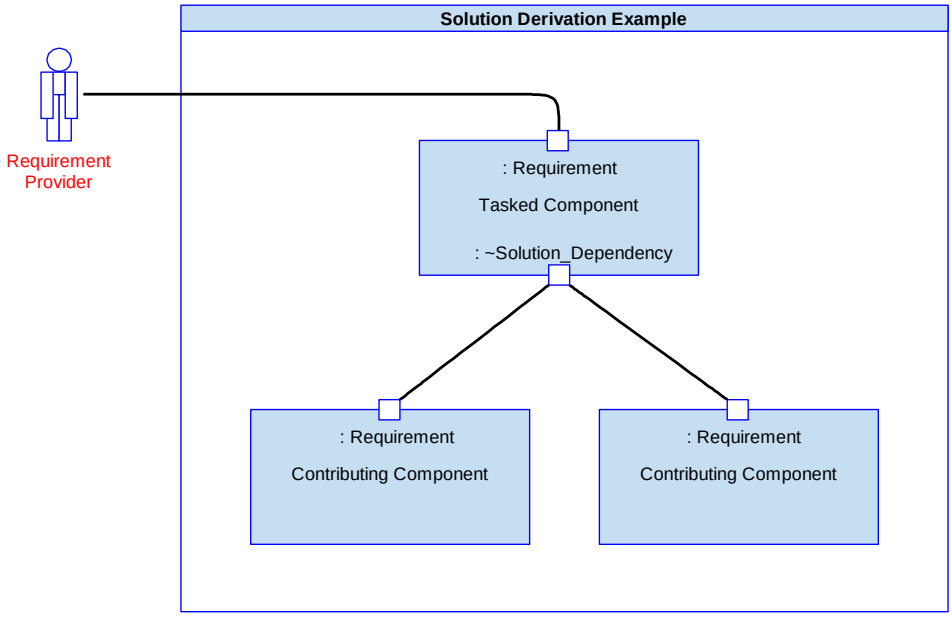

Figure 16: Solution Derivation Example

Figure 17: Solution Derivation and Execution shows the behaviour associated with the example. The Tasked Component receives a requirement change. It finds a suitable solution that includes solution dependencies, which it places on the Contributing Components. The Tasked Component is responsible for ensuring the solution dependencies are being satisfied while the solution is being executed. If the solution dependencies cannot be satisfied, the Tasked Component will seek to adjust the solution to continue to satisfy the requirements (see Feasibility Assessment).

Where there are dynamic dependencies, the Tasked Component may need to adjust the requirements it places on the Contributing Components during execution of the solution.

In order to comply with the operational rules of the system, it may be necessary to seek authorisation (e.g. from an external system or authorised user) to proceed with a given activity. This is managed by including a solution dependency on the Authorisation component. Any activities that require authorisation are prevented from being carried out until appropriate authorisation has been obtained.

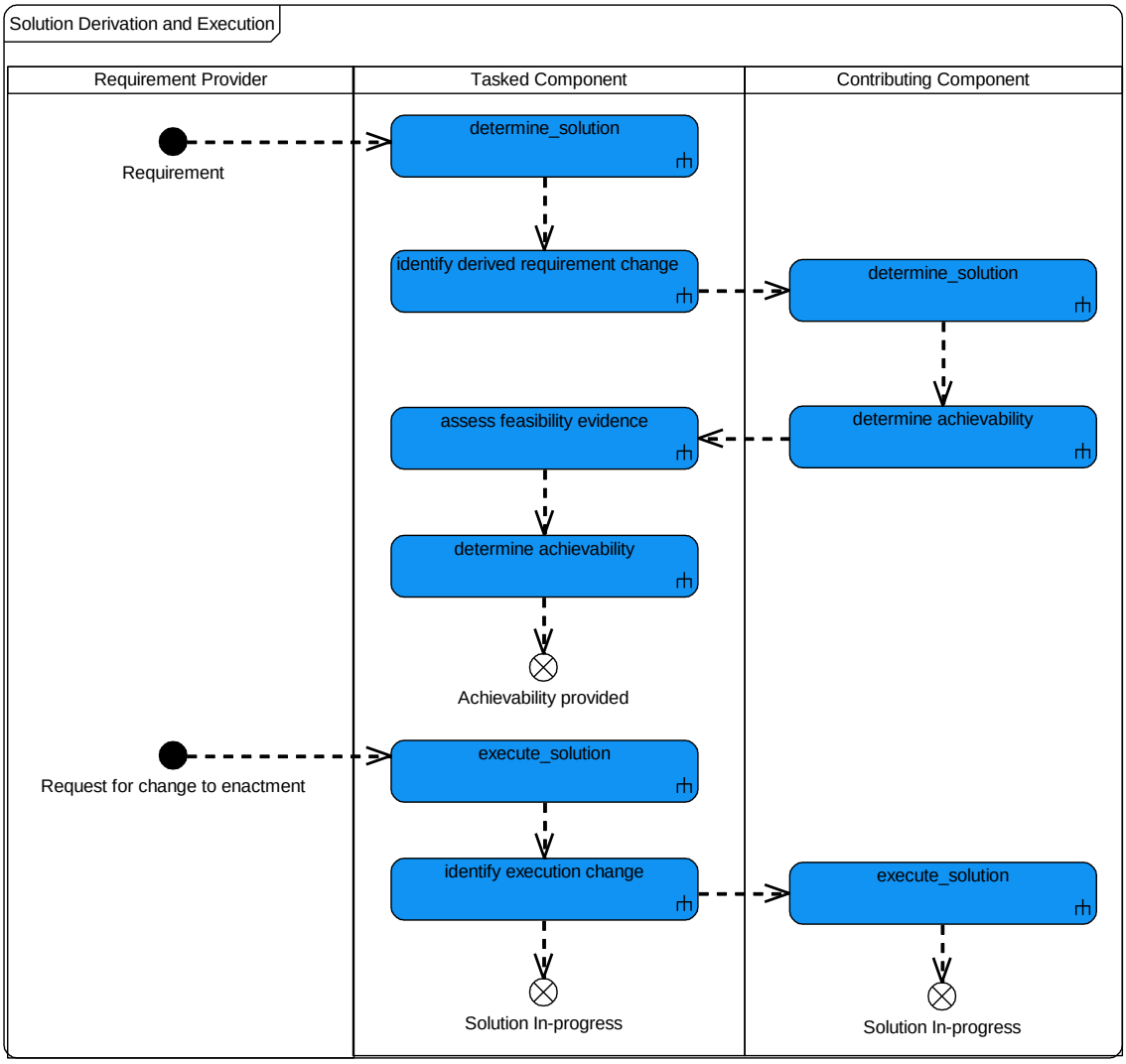

Figure 17: Solution Derivation and Execution

A.1.3.6 Planning Contexts

Where requirements can be achieved in multiple ways, different available solutions may exist at the same time. These solutions will be compared for cost and quality, before a preferred solution is selected. The possibility of there being more than one solution generates a new planning context. These different planning contexts will need to be captured to achieve the planning for alternate solutions when required. The PRA does not specify how a planning context is generated, handled or the format of it. There is, however, a requirement for a planning context to be uniquely identifiable from other planning contexts in order for it to be traceable. The mechanism to identify a planning context could be an attribute added to a service interface.

A planning context may define different conditions which may lead to different requirements being derived by the involved components. When planning a solution within a planning context, components derive requirements in their own planning context that influence the solutions of other planning contexts (see Figure 18: Nested Context View for an example). There is a need to track which requirements belong to each planning context to avoid repeated iterations to produce a valid solution.

The Exploiting Platform should be able to execute existing plans at a future point in time. This should be allowed without needing to generate a new planning context provided there are no changes other than the point in time for execution.

Figure 18: Nested Context View shows the different planning contexts that exist depending on the level of abstraction. Context 1 has a requirement to scan a target whilst avoiding detection.

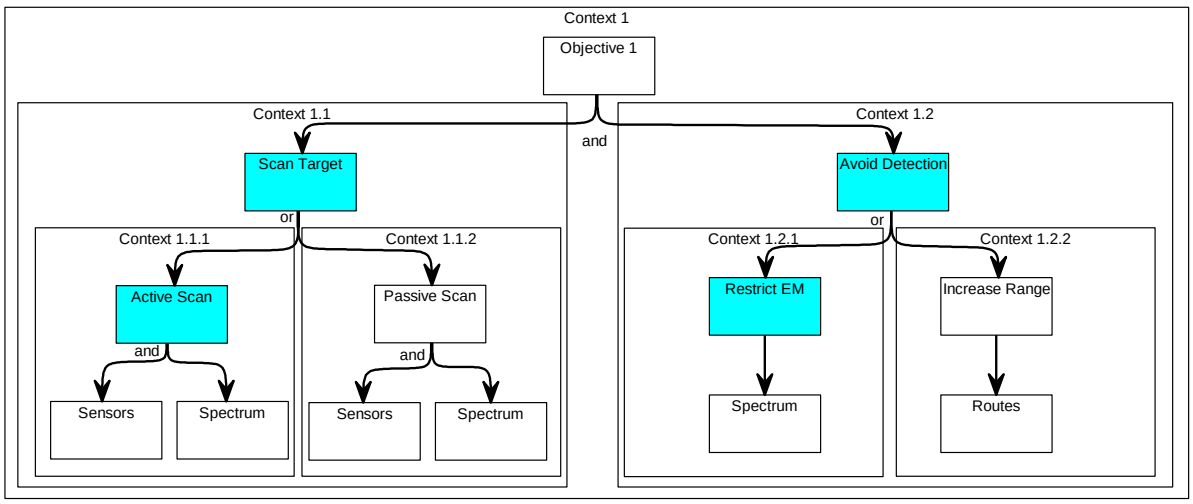

Figure 18: Nested Context View

Context 1.1 defaults to an active scan and the effectiveness of this solution is measured. Context 1.2 defaults to restrict EM and the effectiveness of this solution is also measured. Within each of these planning contexts exists a further two planning contexts relating to the chosen solution to scan the target and avoid detection. The planning context itself may also be a quick way of identifying the solution that should be executed if only one valid solution exists.

A.1.3.7 Feasibility Assessment

Components develop and execute solutions to the requirements provided to them within their given constraints. A component will monitor its own solution (both planned and during execution) to determine whether it remains feasible to achieve its requirements.

After having planned the solution, available resources may change, equipment failures may occur, or there may be changes in the external environment. Such changes could alter the feasibility of a solution within the agreed constraints. Components are allowed (even expected) to adapt their solutions as long as they remain within the parameters defined by the requirements.

If at any point it is discovered that a solution is no longer feasible and no alternative is available, this must be reported to the originator of the requirement. This enables the non-achievable requirements to be re-considered.

At the point a solution is no longer feasible, a new planning context will be generated to allow a new solution to be planned. The new planning context will link requirements, solution dependencies, constraints and resource usage. However, there may come a point when the system has to declare the requirement cannot be met.

A.1.3.7.1 Pattern Overview

The Determine Feasibility use case (see the component composition Use Cases) shows how a component will determine the feasibility of carrying out the solution.

Two services are used in feasibility reporting:

- The Requirement service accepts incoming requirements and can report feasibility back.
- The Solution_Dependency allows the placing of requirements and captures the reported feasibility of them being fulfilled.

Additionally, the following services are needed to support the feasibility assessment:

- The Constraint service accepts incoming constraints.
- The Information_Dependency service provides contextual information.
- The Capability_Evidence service captures the current and predicted capability of other areas of the system.

A.1.3.7.2 Use of the Pattern

Figure 19: Feasibility Assessment Service Dependencies shows the evidence that components may monitor to determine whether it is feasible for the solution to achieve the requirement it was devised to meet. If new evidence is received, it will result in the feasibility of the solution being reassessed and a new solution sought if the original is no longer feasible or the solution can no longer deliver a quality that meets the required measurement criteria. To determine the new solution, a component may assess all contributing factors before identifying a preferred solution (see Figure 108: Change in the Achievability of Dependent Component Trigger, Figure 110: New Requirement Trigger, and Figure 109: Change in the Constraints Trigger).

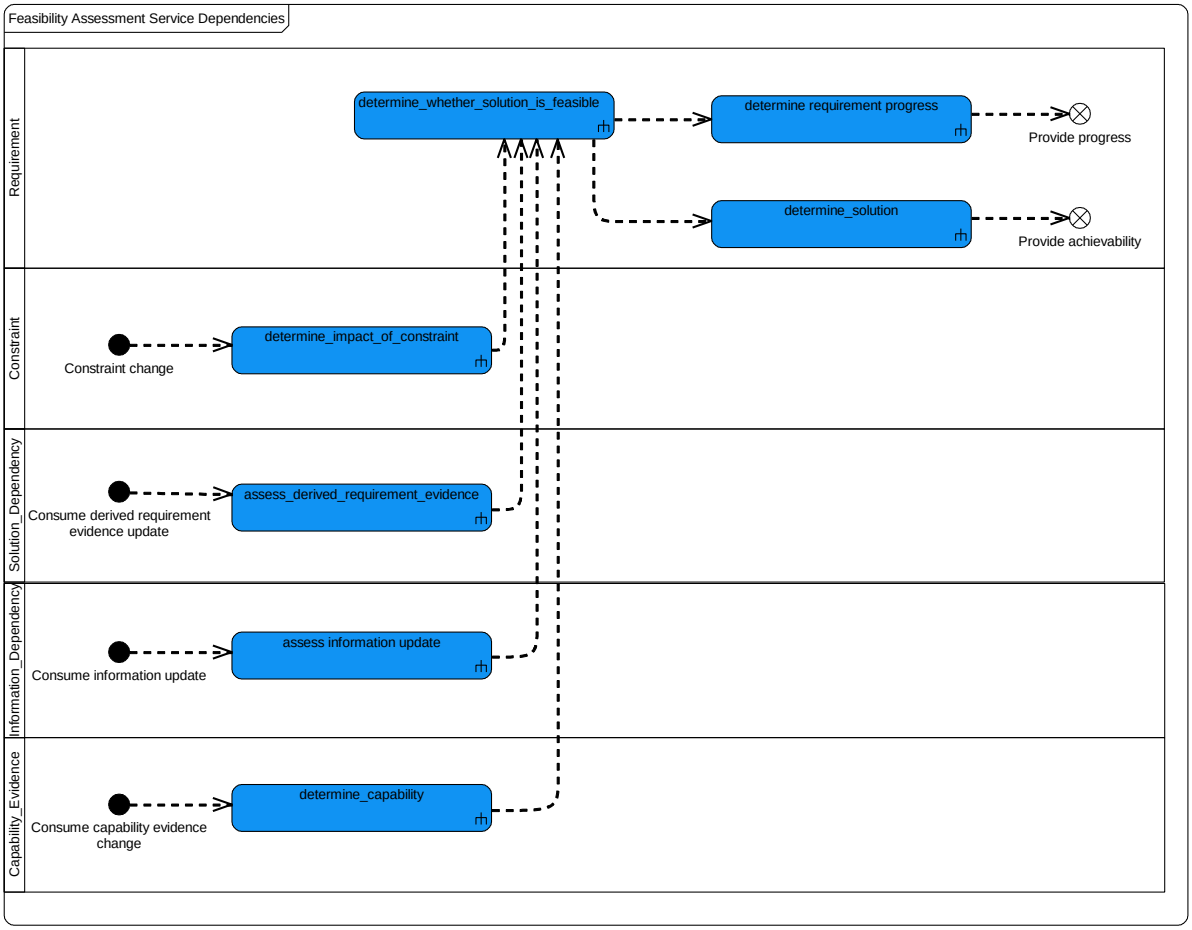

Figure 19: Feasibility Assessment Service Dependencies

A.1.3.8 Achievement and Progress Reporting

It is desirable to assess the satisfaction of requirements in greater detail than a simple ‘complete’ or ‘not complete’ state. By monitoring progress and achievement of each solution, the system can provide more information on the completion of the requirements.

A.1.3.8.1 Pattern Overview

The Fulfil Requirement use case (see the component composition Use Cases) addresses achievement and progress reporting, particularly the Figure 113: Report Achievement scenario. Achievement and progress are reported against a requirement, so the Requirement service is used, supported by Solution_Dependency service where derived requirements are involved.

The Figure 133: Achievement Data Model expresses a generic definition of progress and allows achievement reporting to be defined consistently between components.

A.1.3.8.2 Use of the Pattern

Reporting is against a requirement and falls into two categories:

- Achievement: an indication of the achievement of the requirement against a given measurement criterion. An achievement report can indicate to what extent the requirement is being met. It may include information such as the status (e.g. not started, in progress, complete) and quality, i.e. how well the deliverables are satisfying the requirement. See the Achievement section for more information.
- Progress: an indication of progress against the requirement measured against a given measurement criterion. A progress report may identify what has been achieved against the requirement, e.g. 50% complete.

The following examples show how progress reporting can be achieved. Although achievement reporting is not shown in the examples, it operates in the same manner as progress reporting.

Figure 20: Progress Reporting Structure shows a simple example where a tasked component has interfaces with two contributing components, which in turn can use two further components to progress their requirements.

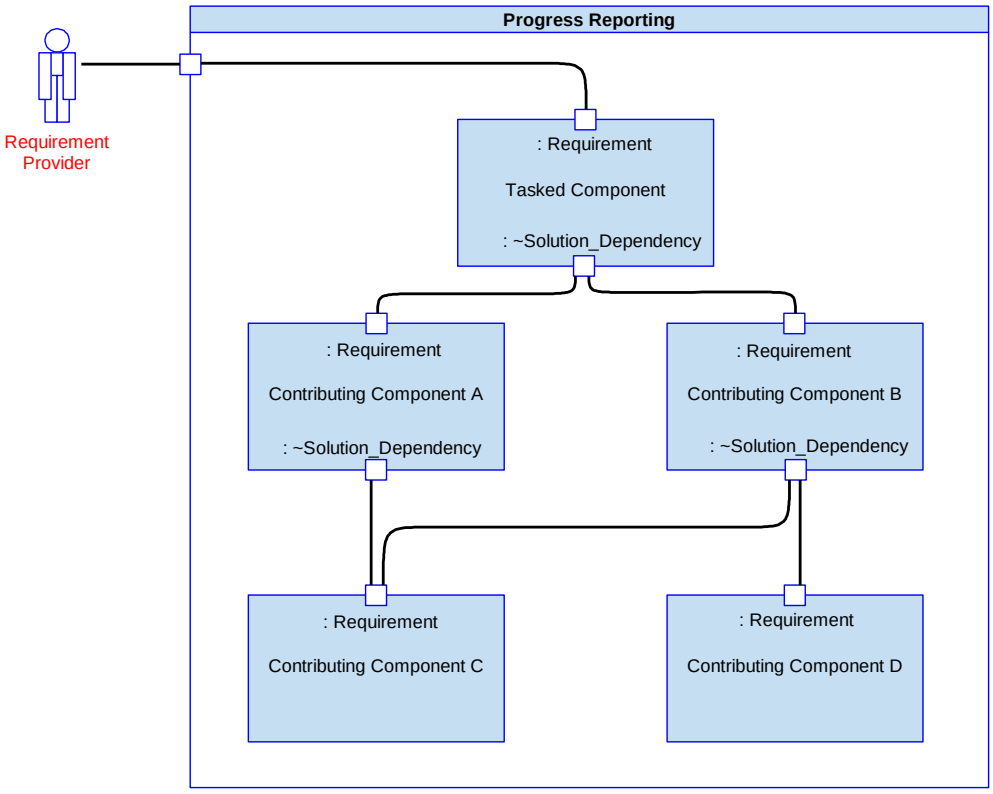

Figure 20: Progress Reporting Structure

Figure 21: Progress Reporting Behaviour shows the behaviour associated with the example. The Requirement Provider asks for a progress report on a requirement that the Tasked Component has been instructed to fulfil.

The Tasked Component has identified derived requirements which have been passed on to two contributing components which, in turn, depend on two more components.

Each component, when asked about its progress, first checks the progress against requirements it has passed on to the components it depends on. This chain is complete when the request reaches a component which can fulfil all of the requirements it is allocated without further assistance and is able to report its own progress.

Progress reports are passed back up the dependency chain, each component interpreting the reports from its dependencies in terms of its own requirements.

Note that the details of the solution (how the components achieve their requirements) are not part of the progress report.

Also note that progress reporting could also be implemented as a component outputting progress on a cyclic basis, rather than only upon a request.

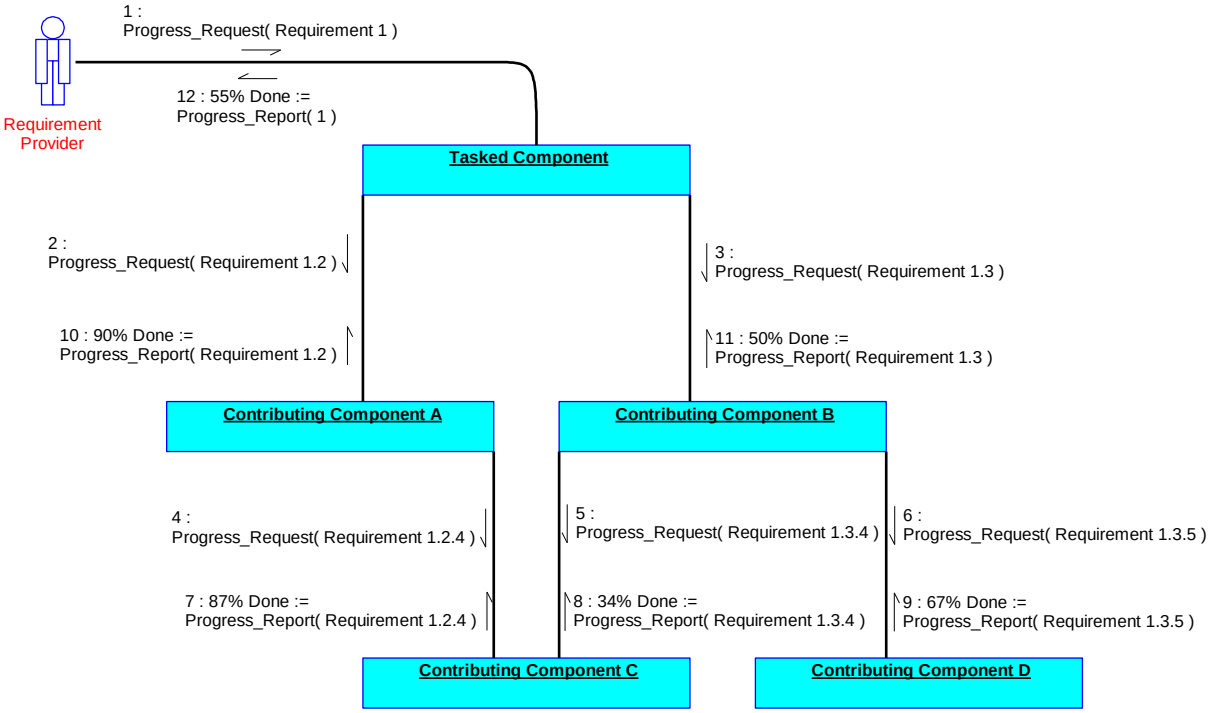

Figure 21: Progress Reporting Behaviour

A.1.3.9 Performance Analysis

A component can specify a required quality as part of a solution dependency. Additionally, components can report their predicted and achieved qualities of any measurement criteria relating to the achievement of a requirement (such as a track position accuracy) as well as the progress against achieving a requirement and the current feasibility of the solution. This reporting allows analysis and assessment of performance against a requirement.

Since reporting occurs through a series of components with solution dependencies on one another, analysis can be carried out at different levels of requirement abstraction. Performance can be assessed by each component in an operational system. This approach also captures changes in performance due to changes in the external environment, such as an enemy platform employing a new tactic after its previous tactic was defeated.

Performance analysis during operation allows for underperforming solutions to be dynamically improved, or (if compatible with other requirements) for satisfactory solutions to be dynamically optimised. This may include generating qualitatively different solution dependency requirements as part of re-planning rather than refining existing ones.

The results of the performance analysis may also be recorded to aid non real time analysis (see the Operational Support policy).

In some circumstances the reason for the system underperforming may be due to an anomaly, such as a health issue or cyber attack, resulting in a reduction in capability which will need to be taken into account when re-planning a solution (see the Capability Assessment policy).

A.1.3.10 Solution Replanning

When a solution that has been selected for execution or is in progress becomes unfeasible, then a solution replan becomes necessary. The following sections present two example scenarios requiring solution replanning, one for a solution that is replanned following a change in the feasibility of the solution actions, the other is for a solution that suffers a loss of resource.

A.1.3.10.1 Action Replan

Figure 22: Solution Action Replan shows a simple example where the currently selected solution to a requirement is Solution 1. Solution 1 is made up of three actions, of which Action 3 is using Action 3.1 of the two available sub-options. When Action 3.1 becomes unavailable, it requires the solution to be replanned for Action 3 to use Action 3.2.

Action 3 can be updated for the replanned solution, without needing to involve the Requirement Provider, providing Solution 1 still meets the requirement and has not changed the cost or quality.

If Solution 1 no longer meets the requirement, then the alternative solution (i.e. Solution 2) will be considered.

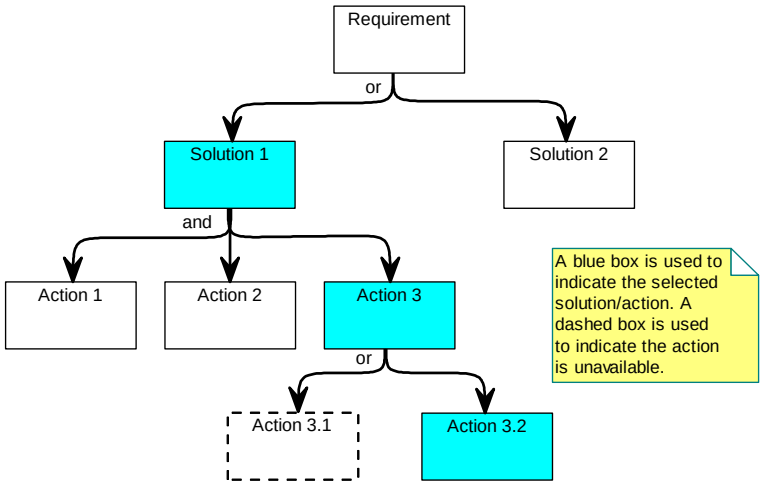

Figure 22: Solution Action Replan

A.1.3.10.2 Resource Replan

Figure 23: Solution Resource Plan shows an example of a requirement that cannot be solved by a single component, where the current solution to a requirement is made up of two solutions, one from Component X and another from Component Y. Both Components are using resources, Component X and Y are using Resource A and C respectively. Both Component X and Y have alternative solutions that can use other resources, i.e. Resource B and A respectively.

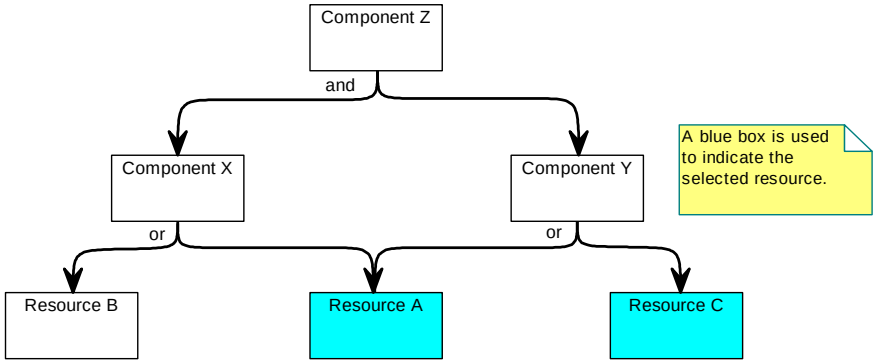

Figure 23: Solution Resource Plan

In this example, Component Y can no longer fulfil its requirement with Resource C which has become unavailable. The desire of Component Y to use Resource A is not directly expressed to Component X, but the allocation is linked through the solution dependencies. The Resource Management policy shows how conflicts can be identified allowing a replan of Component X's actions so that it doesn't require the use of Resource A.

Figure 24: Solution Resource Replan shows how the resource replan has developed with Component X no longer using Resource A, and has replanned its solution to use Resource B to achieve its own requirements. Component Y is now allowed to use Resource A, the new solutions from both components X and Y are accepted by Component Z and the required resources allocated accordingly.

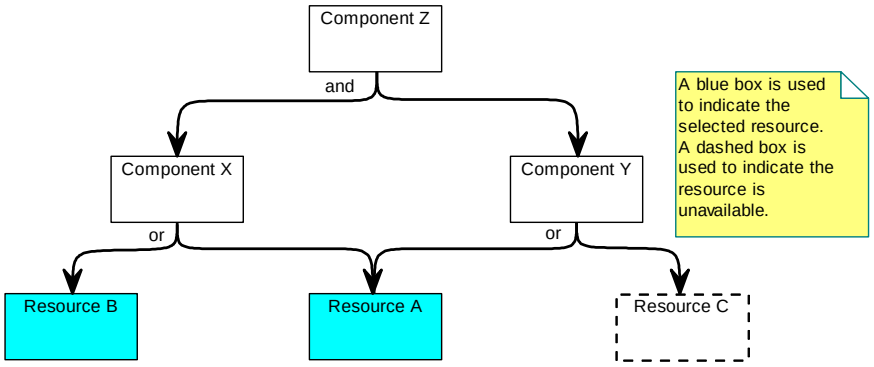

Figure 24: Solution Resource Replan

A.1.3.11 Examples of Rigid and Dynamic Dependencies

A.1.3.11.1 Setting Up a Sensing Task

The system is given the mission objective of searching an area for a particular class of objects. The objective is broken down into two tasks: make a transit flight to the area, and perform a search. The search involves flying over the area, using available sensors to scan the area, and processing sensor data to identify objects. This use case covers the setting up of the two linked tasks.

A.1.3.11.1.1 Pre-Conditions

The search objective has been assigned to an aircraft.

A.1.3.11.1.2 Flow

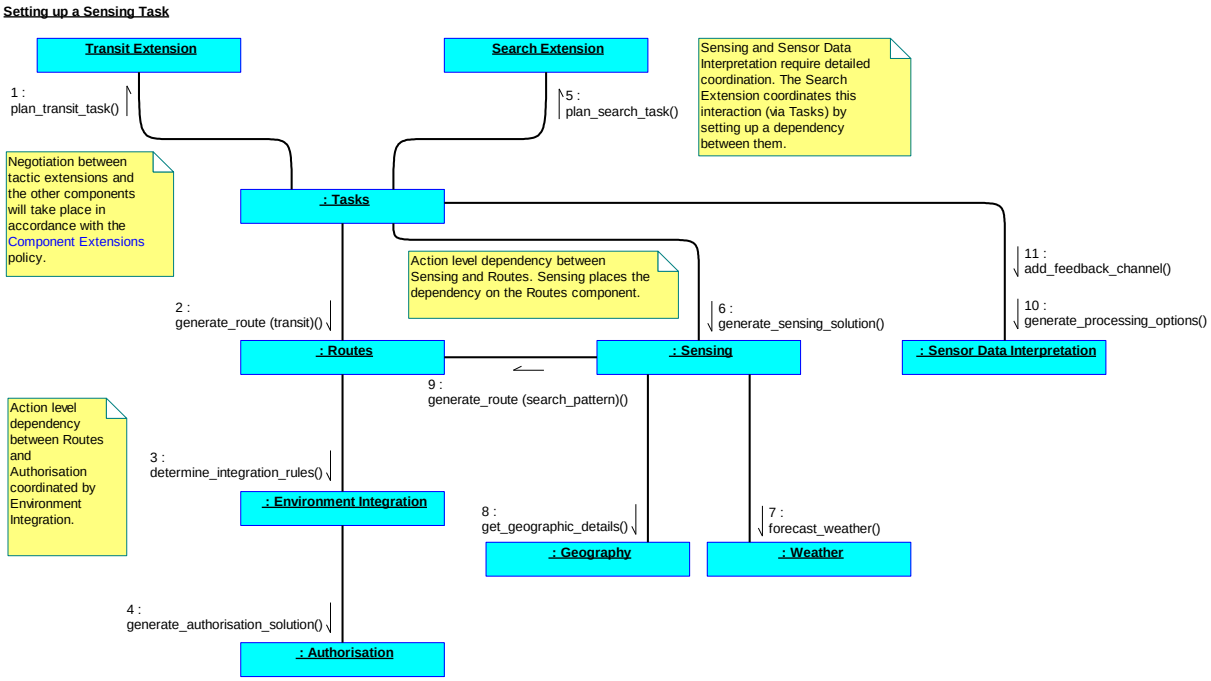

Figure 25: Setting up a Sensing Task

Tasks instructs its Transit tactics extension to plan the transit. The Transit tactics extension uses Routes to find a transit solution. A route is proposed which would require authorisation to fly through a restricted zone (determined by Environment Integration). This is identified as a dependency for the transit, and Authorisation determines that it will be possible to request clearance.

The Tasks parent component instructs its Search tactics extension component to plan the search task. The Search tactics extension component uses Sensing to find a sensing solution. Based on the details of the search task, together with knowledge of the search area (from Geography) and the weather forecast (from Weather), a sensing solution is proposed. This involves flying a particular search pattern, which is placed as a dependency on Routes.

The Search tactics extension component also uses Sensor Data Interpretation to plan the tactical processing required to analyse the outputs from the sensing actions. A dynamic dependency is created between Sensor Data Interpretation and Sensing to manage the search as data is received.

A.1.3.11.1.3 Post-Conditions

The tasks are ready for execution.

A.1.3.11.2 Carrying Out a Sensing Task

The system is given the mission objective of searching an area for a particular class of objects. This is broken down into two tasks: make a transit flight to the area and perform a search. The search involves flying over the area, using available sensors to scan the area, and processing sensor data to identify objects. This Use Case covers the execution of the two tasks.

A.1.3.11.2.1 Pre-Conditions

The tasks have been set up and the mission is ready for execution.

A.1.3.11.2.2 Flow

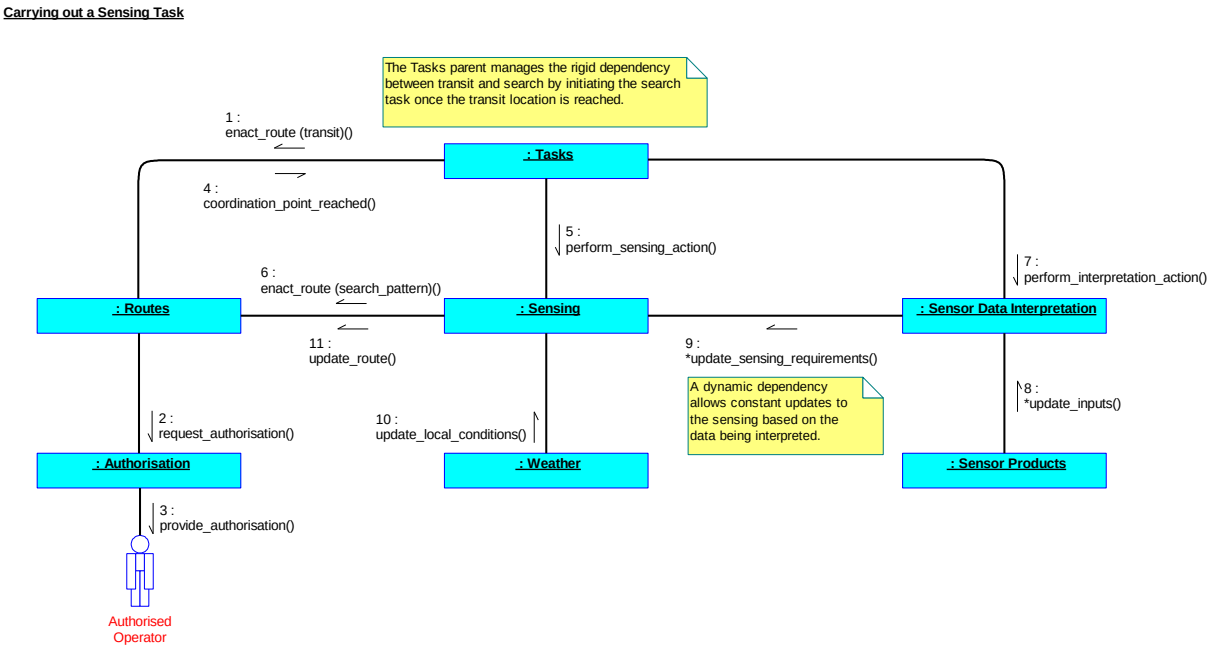

Figure 26: Carrying out a Sensing Task

Tasks manages the dependency between transit and search by initiating the search task once the transit location is reached. Tasks requests that Routes execute the planned transit route. Routes can begin execution of the route as the aircraft is not in the restricted area, but relies on Authorisation obtaining permission to transit through the restricted zone. Authorisation obtains this approval from an Authorised Operator. This allows Routes component to continue through the rest of the planned route.

Tasks receives an update when the search area has been reached. At this point the search task is initiated. This involves flying the planned search pattern (via a request to Routes), using the sensors as planned, and analysing the sensor data to identify the objects of interest.

As the sensor data from Sensor Products is analysed, Sensor Data Interpretation adjusts the requirements on Sensing (without exceeding the original requirements and constraints) to make sure that the sensors are used in a way that will support the data analysis. This is an example of a dynamic dependency between components as defined in What is Dependency Management?.

Changes reported in Weather conditions are reported to Sensing to determine if this will impact the capability to carry out the search. As the search progresses and sensor data is processed, the parameters fed to Routes can be refined to update the search path.

A.1.3.11.2.3 Post-Conditions

The tasks are complete.

A.1.4 Autonomy

A.1.4.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Control Architecture
- Constraint Management
- Human-Machine Interface

A.1.4.2 Introduction and Scope

This policy explains what is meant by autonomy in the context of the PRA. It introduces the Authorisation component and describes how this component, acting together with other components discussed in the Authorisation section, can be used to develop and configure a system capable of operating at varying levels of autonomy.

A.1.4.2.1 What is Autonomy?

The PRA regards an autonomous system as one that can develop and implement problem solutions in accordance with pre-defined rules without direct crew control, following the receipt of any necessary authorisation.

A.1.4.2.2 Why is Autonomy Important?

The PRA supports the design and configuration of a system incorporating autonomy. This gives PRA Exploiters the ability to design for, and control, the levels of autonomy required by an exploitation.

A.1.4.2.3 Levels of Autonomy

A system may be able to operate at varying levels of autonomy, ranging from fully autonomous to fully manual. The level of autonomy may be constant for a particular Exploiting Platform or mission, or it may change depending on, amongst other things:

- The operating environment.
- The specific task or action being carried out.
- The state of the infrastructure at the time. For example, if communication is reduced or lost the vehicle may be permitted to operate with greater autonomy.
- The state of the vehicle at the time. For example, an automatic response to internal events such as faults may be permissible.
- The workload of the authorised operators. For example, enabling more autonomy during periods of high operator workload.

An activity or decision which is autonomous can be considered to have received authorisation. However, there are situations where, although an activity or decision is autonomous, there may be a benefit in the operator being informed, for example to maintain their situation awareness and enable them to make decisions or take action when required. Other actions or decisions may require greater operator involvement, for example to choose between a number of potential solutions generated by the system, and to enable this the system would need to provide the operator with necessary information. Benefits of this operator involvement may include:

- A greater ability to respond to unforeseen circumstances and a rapidly changing environment.
- The ability to apply judgement and interpretation for non-deterministic rules (e.g. rules of engagement) or scenarios.
- Potentially reduced development costs of some aspects of the system (where automation technology is prohibitively expensive).
- Improved engagement/situation awareness for the operator.
- A safe alternative during early stages of development to allow automation technology to be matured.
- Mitigation for some failure conditions.

The level of autonomy is therefore linked to authorisation and interaction with the operator. There are a number of models and frameworks available that provide methods of quantifying and measuring the level of autonomy at which a system can operate, and it is down to the Exploiting Programme to determine which is appropriate. An example of how levels may be applied to a search task is described in the Styles of Interaction section.

A.1.4.2.4 Introduction to Authorisation

Every action carried out by the system requires some form of operator authorisation. This may be built into the system by design (in the form of general rules), or as an explicit authorisation for a task or specific action. The authorised operator's act of placing a requirement (and accepting the cost of the solution) may constitute authorisation for the task as a whole. However, even a task being carried out with general authorisation, or under a high level of autonomy, may need separate authorisation for a specific action such as jettisoning fuel, depending on the rules that are in force. For such an action, the system must ensure it has authorisation before carrying out the action.

A.1.4.2.5 Aim of this Policy

This policy explains how the PRA has been laid out to support the control of autonomy in a manner consistent with the PYRAMID KURs.

The principle PYRAMID KURs associated with autonomy are:

- **Configurable:** A deployment of the PRA will wish to control the levels of autonomy allowed to the system. This policy allows for levels of autonomy to be controlled at design time. It allows constraints to be defined that limit the levels of autonomy of the system at run-time.
- **Flight Certifiable:** The flexibility of the autonomy policy allows actions to be performed by whichever of the authorised operator or the system is best equipped to do them, which can help make the system certifiable.
- **Utility Across A Range Of Missions:** Levels of autonomy granted to the system will vary according to the mission scenario. This policy allows for levels of autonomy to vary at run-time to meet the needs of authorised operators for a particular mission and at different mission phases, while keeping within defined constraints.

A.1.4.3 Overview

The policy illustrates how the components enable a deployment of the PRA to be configurable for different levels of autonomy either at design time or during operation. The policy illustrates how the different styles of interaction can be achieved, and how authorisation can be associated with either whole tasks, or specific actions that form part of a task solution.

A.1.4.4 Context

Context is information that is likely to change the interpretation of observed facts or behaviour in some way. For example, the presence of a tank in a particular area may or may not be interpreted as a threat depending on the understood posture of the tank's owner.

Context is not just an issue for operators; the more complex and autonomous a system is the greater importance context has for the system as well. Systems which develop solutions (either for automated implementation or proposing to an operator) need to understand context so that the solution is appropriate for the current context.

Within the PRA, each component is responsible for a particular subject matter, and associated contextual information belongs to that component and will be provided when necessary to support decision making throughout the system. For any given component its context is established by its connection with other components, as illustrated in Figure 27: Establishing Context. Human-machine interoperability is key to autonomy as it must be ensured that the human and machine have access to the same data, and that their interpretations of that data are made in the same context. This is illustrated in Figure 28: Sharing Context.

Figure 27: Establishing Context and Figure 28: Sharing Context illustrate how components can support both sides of a system and operator interaction. The independence of components means that the same services which support the interacting component can also support the involved HMI components.

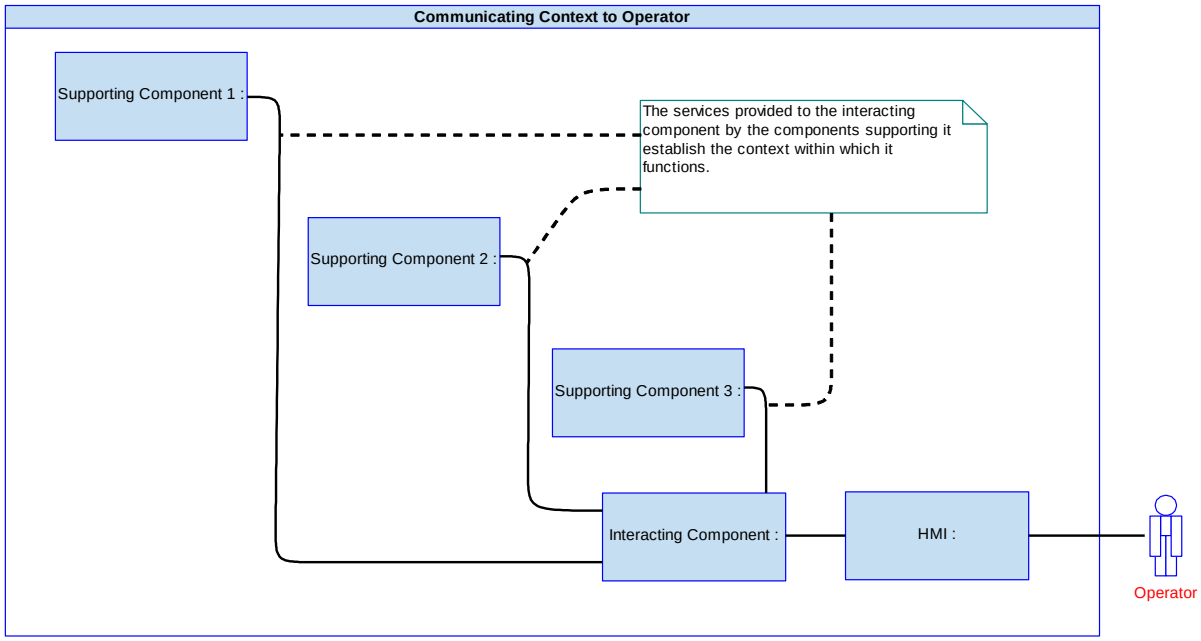

Figure 27: Establishing Context

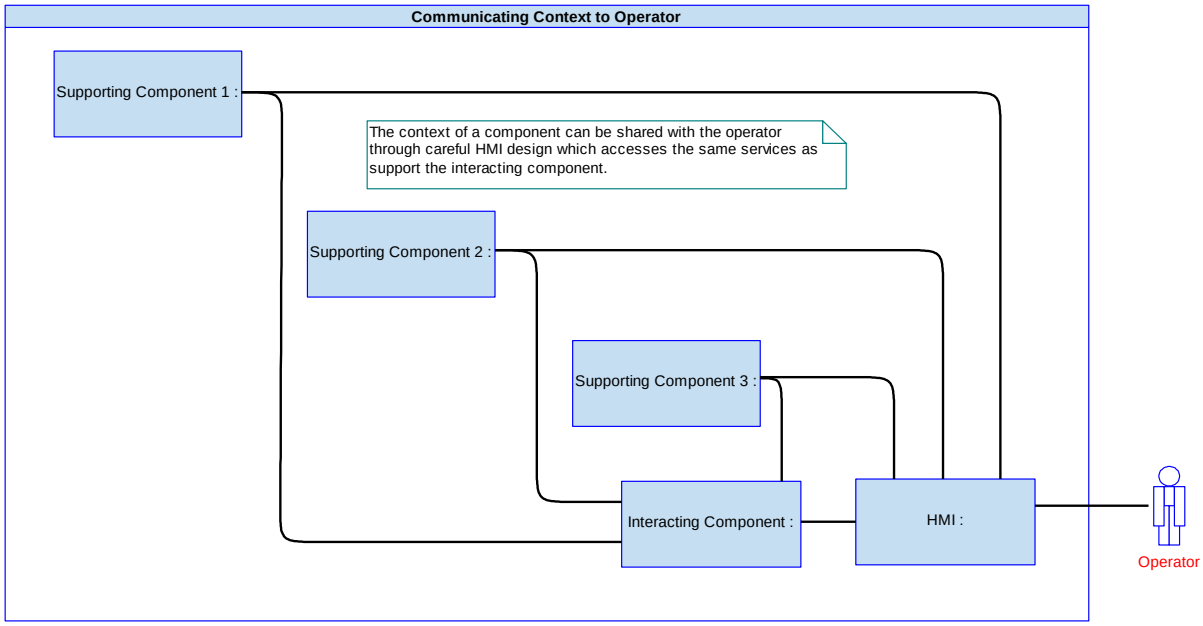

Figure 28: Sharing Context

A.1.4.5 HMI and User Roles

The system Human-Machine Interface (HMI) will need to support the requirement for authorisation of any actions, and any associated interactions with users. For example, a fully autonomous system, without a requirement to provide feedback or present options to an authorised operator, or obtain authorisation for a specific action (such as jettison fuel), will not require any HMI to support autonomy. The system may of course require HMI for other reasons.

However, if there is a requirement for operator involvement, for example to obtain authorisation, inform an authorised operator of an action, or provide solution options, the HMI must support this interaction, enabling the appropriate information to be presented to the authorised operator and, where necessary, translate their response into instructions to be enacted by the system.

A.1.4.6 Authorisation

The need for authorisation of an activity is satisfied by a number of components acting together:

- The Authorisation component is responsible for obtaining authorisation when it is explicitly required. What constitutes authorisation will depend on the Exploiting Platform, but the authorisation must be made in a meaningful and informed way, in line with the Human-Machine Interface policy.
- The Operational Rules and Limits component captures general rules which can constrain a component's operational ability, which may result in the component having to obtain authorisation to continue.
- Components are responsible for identifying when authorisation is required before they can carry out or request specific actions. A components specific actions may:
    - Be inherent to a components design and be naturally allowed as part of its normal behaviour.
    - Require authorisation to be obtained before performing the activity, e.g. this could involve seeking authorisation, checking that pre-authorisation is in place or previous authorisations are still valid.
    - Be assumed possible until the component is informed that they are prohibited. This restriction would be typically indicated by the application of a constraint, possibly provided via the Operational Rules and Limits component or from another component.

  Where a component has identified that authorisation is required, this authorisation would be achieved by the component liaising with the Authorisation component. Whose responsibilities lie with capturing requirements for authorisation from another source and developing an authorisation solution.

Authorisation for a specific activity may be granted through virtue of authorisation having been granted to a broader activity that it contributes to. In this case either authorisation may be implied for a component when it receives a requirement to be implemented, or confirmation of authorisation may be required to be obtained from the Authorisation component - where in this case the Authorisation component would already have the relevant authorisation available. The latter option may, for example, be desirable to increase system safety integrity. However, in some cases, whilst authorisation for a broader activity may have been granted, dedicated authorisation for specific activities may still be needed. For example, authorisation may have been granted to carry out activities contributing to attacking a target; however, separate authorisation may still be required prior to committing to weapon release against the target. In this case the Authorisation component would need to obtain this separate authorisation.

It is likely that the requirement for authorisation will vary depending on conditions. A potential solution to a requirement placed on a component may capture this variability as a measure of the quality of the proposed solution. For example, if the solution to a search task runs a risk of requiring authorisation to enter a restricted zone, this can be captured against the solution as a low robustness to change.

When granting authorisation for a mission plan, or a particular action within a plan, an authoriser may wish to place conditions upon the authorisation such that it is only valid if the specified conditions are met. For example, it would only be valid within a particular time window or only with target data quality above a

defined threshold. Within the PRA, conditions associated with an authorisation will be held with that authorisation, however it will be a responsibility of the Authorisation component to compare the current state with the boundaries of the conditions.

A.1.4.7 Styles of Interaction

In the example where the system has the task to make a transit to reach an area to be searched, the interaction with the operator is governed by the way the HMI has been set up, how the operator is choosing to use it and by behavioural rules that are active at the time.

Depending on the context within which the task is being solved there may be different solutions to the task according to the criteria that are considered. The Mission Plan may require the fastest route, or the shortest. It may focus on probability of success or robustness to change. If the Mission Plan does not include these criteria, the operator may be left to decide. Measurement of generated solutions against such criteria will allow the operator, or the system, to select the most appropriate solution. Operating at the highest levels of autonomy the system will look for a solution that best matches the criteria, possibly informing the operator and enabling them to reject the solution. At lower levels of autonomy the system may look for individual solutions for each criterion to allow the operator to select between them. These interactions between the operator and the system will usually be coordinated by the high-level components Objectives and/or Tasks but can take place with any layer of the control architecture, as shown in Figure 11: Mission Planning Interactions.

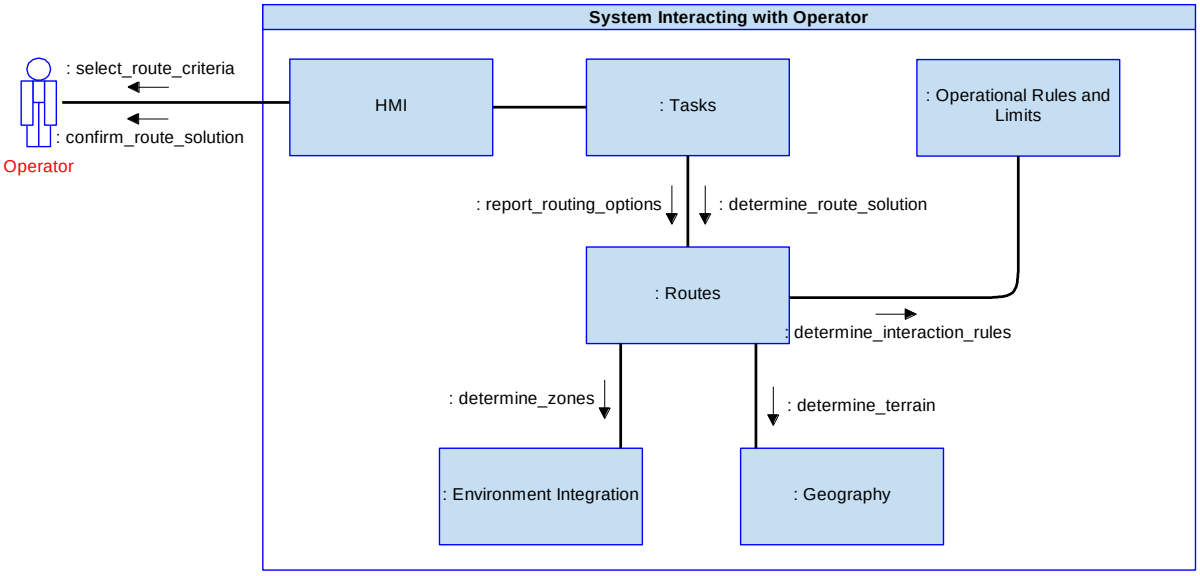

Figure 29: Example of System Interacting with Operator

Figure 29: Example of System Interacting with Operator shows how the operator and the system may cooperate to determine a route to transit to a search area.

The system will behave differently depending on the requirements for authorisation and interaction with an authorised operator, and how the HMI is set up. Some examples of how the system and operator may interact are given below:

- **Full autonomy:** The system generates the best solution it can find and enacts it, providing feedback to the operator as required.
- **Action unless revoked:** The system generates the best solution it can find and presents it to the operator. Unless the operator objects, the system enacts the planned task.
- **Advice and action if authorised:** The system generates solutions, presents them to the operator and highlights the recommended one. The operator approves a solution and the system enacts the planned task.
- **Advice:** The system generates solution options by using different criteria. The operator selects a solution (after refining the criteria, if required) and the system enacts the planned task.
- **Advice only if requested:** The system generates solution options to meet criteria given by the operator. The operator selects a solution (after refining the criteria, if required) and the system enacts the planned task.
- **None:** The system plays no part in generating or selecting solutions.

A.1.5 Health Management

A.1.5.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Capability Assessment
- Control Architecture
- Cyber Defence

A.1.5.2 Introduction and Scope

This policy explains the mechanisms provided by the PRA for the management of system health. The policy introduces the Anomaly Detection and Health Assessment components and describes their interaction with capability assessment at all levels of the architecture.

A.1.5.2.1 What is Health Management?

In the context of the PRA, a system's health describes its ability to perform its intended function within specified limits.

Health can be described for a complete system or for any of its constituent parts. If any of the hardware elements that make up the system have failed or been damaged, the system's health is reduced because it will be unable to perform, or to perform fully, the functions that rely on that hardware element. Looked at more broadly, the overall system may still have full capability because, by design, built-in redundancy means it can achieve the same function using alternative resources. Nevertheless, at the local level, a part of the system has reduced health, and overall, the system has become less robust.

Health management is the process of detecting an anomaly, identifying the element (or elements) that gave rise to it and then providing health information to support capability assessment. Health management also reacts appropriately to a health problem to keep the system safe and/or allow it to fulfil its mission and by providing life and usage information, supports maintenance and system availability.

A.1.5.2.2 Why is Health Management Important?

Incorporating health management into a system may contribute to the safety of the system by reducing the chance that failures will lead to avoidable hazards. For mission-critical functions, good health management may allow the system to achieve its mission goals in the presence of failures, leading to a higher mission success rate. Health management also contributes to the availability and maintainability of a system by providing health data to inform maintenance decisions and activities.

Health management applies at different system levels:

- At the fleet level, by changing working practices, maintenance procedures, manufacturing standards, or other systematic changes.
- At the asset level, by performing or triggering maintenance actions to restore the health of the asset.
- At the mission level, by finding alternative ways to deliver a mission capability using different resources.
- At the resource level, by reconfiguring the infrastructure to use redundant or backup resources. This kind of response may be automatic and performed without extensive analysis of the problem.

A.1.5.2.3 Aim of this Policy

This policy explains how the PRA has been designed to support health management in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Resilient Against Obsolescence:** This policy allows obsolete components to be replaced with minimal impact on the system. It allows deployments to be executed on different hardware with minimal rework.
- **Scalable**: This policy makes it straightforward for systems with different numbers of components to be able to assess and manage their health.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** This policy allows the PRA to be used to build systems that are maintainable and reliable, with a lower risk of mission failure.

A.1.5.3 Overview

The policy shows how the components contribute to health management by:

- Detecting that an anomaly (system not being in an expected state or exhibiting unexpected behaviour as a result of not being in that state) has occurred that may indicate a health problem.
- Identifying the system element that gave rise to the anomaly or anomalies as a consequence of its failure.
- Providing health information to support capability assessment as discussed in the Capability Assessment policy.
- Reacting appropriately to a health problem to keep the system safe and allow it to fulfil its mission goals if at all possible.
- Providing life and usage information to support maintenance and system availability.

A.1.5.4 Health Assessment

Except for automatic responses to a failure, which may be carried out instantaneously where a very fast response is essential, the first step in good health management is to understand the failure. This is health assessment.

The system is monitored, at every level of detail, to determine whether it is behaving as expected. Any unexpected behaviour needs to be explained. It could be caused by:

- An external condition such as weather. This may require a controlled response to accommodate the condition.
- A software problem. Unknown software artefacts (including but not limited to bugs) may exist within the software, or software may be compromised by cyber attack.
- A hardware failure.
- Recovery from a failure. In some hardware there is the potential for failures to recover (e.g. blockages sometimes clear and leaks may get plugged), software can be reset and some systems may accommodate failures internally.

Health assessment is the process of distinguishing between these cases, i.e. determining whether the unexpected behaviour is related to the state of system hardware. For hardware failures (and recoveries) it identifies:

- The change in health of the system, including how it can be expected to progress. This may then be used to determine the effect on the capability of the system. The Capability Assessment policy covers this.
- The hardware element (or elements) that has incurred the change.

This information allows appropriate decisions to be made at resource, mission, asset and fleet levels. Where the health of a system element is difficult to assess directly, its life (time since initial operation) or usage (how, or how much it has been used) may be used as a proxy for the health of a system and can inform maintenance requirements.

A.1.5.4.1 Detection of Anomalous Behaviour

Anomalous, or unexpected, behaviour results in the system entering an unexpected state. An unexpected state may be as simple as an error message, or it may be an inconsistency between two values. Such inconsistencies may only be apparent when different parts of the system are compared: each part of the system may appear to be consistent when viewed locally. Anomalies therefore need to be considered at every level of the system, not just at the interface with hardware.

For some hardware elements it may be possible to detect discrepancies that are an early sign of degradation that could get worse. In such cases it will be desirable to monitor for such discrepancies, where the element they relate to is still operating within its specification, but not in the expected range. Local small discrepancies may also be part of a wider pattern that could be a sign of a cyber attack or a physical problem such as overheating.

The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about the importance of detecting anomalous behaviour in ensuring the continued security of the Exploiting Platform.

A.1.5.4.2 Failure Identification and Prognostics

In order to make maintenance decisions, and also to have a more precise knowledge of the effect of a health change on its capability, the system needs to identify the hardware element that has failed or been damaged.

In an integrated system, the failure of a hardware element may cause problems in any part of the system that relies on that element, however distantly. Ideally, each hardware element would be monitored closely and any failure identified immediately and unambiguously. This may indeed be the case in highly instrumented safety-critical systems, where an anomaly is clearly related to a specific functional failure. In other cases, however, it may not be possible to get the specific information that would make health assessment easy.

Where health assessment is unable to provide an answer that is sufficiently certain or specific to inform either maintenance or operational decisions, it may identify additional information that could improve the answer. If this includes requesting a BIT to be run, the test will be managed by the Test component as shown in the Test policy.

In health management, prognostics is the use of health data to predict the progression of a failure or damage over time and with use. In the PRA, prognostics is not treated separately, but is regarded as a forward-looking aspect of health and capability assessment. Prognostics may also be done as part of operational support when the prediction is too long-term, or too uncertain, to inform decisions within a mission. However, many failures progress either too fast or too slowly, or unpredictably (or not at all) to be considered; so in many cases the best assumption will be that the current hardware condition, and hence capabilities that rely on it, will continue unchanged.

A.1.5.4.3 Health Data

Health data includes all data that is required as input for assessing the health and health related capability of a system. It can include, 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).

The collation of health data that needs to be available for exploitation outside the system is coordinated by the Information Brokerage component in accordance with the Storage and Recording and Logging policies.

The exploitation of health data for fleet management and maintenance is discussed in the Operational Support policy.

A.1.5.5 Component Composition

Health management services belong on the Anomaly Detection and Health Assessment components and are defined in their component service definitions. Other components support health management using the general services described in the Component Composition, or using their own subject-matter-specific services: they don't require additional services to support health management.

A.1.5.6 Health Components

This section describes a framework for the use of components to support health management. It will help component developers and implementers understand the context in which the health management components are intended to work.

A.1.5.6.1 Health Management of a Single Resource

Figure 30: Health Management of a Single Resource shows a generalised example with a single instance of the Health Assessment component assessing the health of a single resource.

The Resource component represents any component that manages a hardware resource, such as a tactical sensor. The Resource User is any component that will make use of that resource to carry out actions such as sensing or opening a door to release a store. Larger and more complex systems will generally require many instances of Health Assessment working together, this is discussed further in Hierarchy of Health Components.

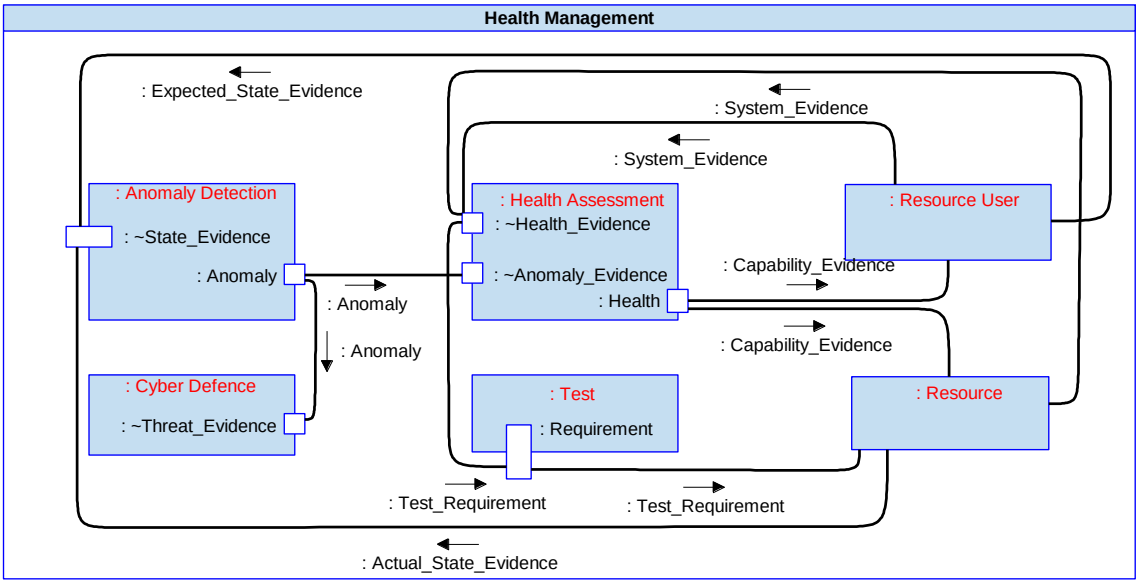

Figure 30: Health Management of a Single Resource

The Anomaly Detection component receives information about the expected and observed property of an element of the system. This could be, for example, the command received by an actuator, the physical position of an actuator as determined by a sensor, the privileges associated with a system user or the estimated velocity of a detected aircraft. Where hardware is concerned, under most circumstances a fault or error message from the hardware, or from the software controlling the hardware, constitutes evidence of an anomaly. Figure 30: Health Management of a Single Resource shows the Resource User providing the expected state of the resource, following issue of a command to change the state of the resource, for example to move an actuator. The actual state of the actuator comes from a separate sensor. In the example of the estimated velocity of a detected aircraft, a single component (i.e. Tactical Objects) may provide the evidence.

The Anomaly Detection component uses this information to determine if there is a departure from normal or expected behaviour, or when two states are inconsistent with each other.

The Health Assessment component is responsible for identifying which, if any, hardware element has failed or degraded, thereby giving rise to the observed anomalies. There are two goals here: one is to inform maintenance so that the affected part can be repaired or replaced without extensive diagnostic activity. This requires the failure to be isolated to a line-replaceable unit. If a hardware element needs BIT to be run, the test will be managed by the Test component as shown in the Test policy. For operational purposes, however, the main goal of the Health Assessment component is to allow Resource Users to understand what capability has been affected by the failure. This may require either more or less specificity in the answer: the focus is more on the failure mode of the equipment than on the unit to be repaired or replaced.

Each Resource User is responsible for assessing how its capability has been affected by any failures. The assessment will be informed by local anomalies and/or failures of the hardware they control. Components further from the hardware will base their assessment on the change in capability of other components that they rely on. See the Capability Assessment policy.

The Cyber Defence component will also consider, in parallel, whether the anomaly could be a sign of a cyber attack. See the Cyber Defence policy.

Some resource components may have the ability to reconfigure their resource in response to an identified failure in their hardware so that the resource remains available.

Many health issues are of concern to authorised operators, and may require warnings or alerts to be raised. The HMI components involved in this interface do not feature here, but the principle to be followed is that interaction with authorised operators is handled by Resource User components on the basis of the capability they provide and the tasks and actions they are trying to achieve. This principle allows alerts and other information to reflect the implications of failure.

A.1.5.6.2 Hierarchy of Health Components

Although it is possible to architect a deployment of the PRA such that there is a health assessment solution involving a single instance of the Health Assessment component with knowledge of the entire system, this solution means that every change to any part of the system will require that component to be updated. This is usually only practical for small, simple systems. For larger systems a distributed solution is more flexible, with a hierarchy of Health Assessment components working together in a coordinated way to identify where in the system the problem actually lies. There should be no assumption that the real cause of the problem is in the component where an anomaly was detected. A distributed solution will require the Health Assessment components in the system to have a consistent approach to carrying out fault diagnosis (reasoning about cause and effect), for example by having a generic implementation of health assessment. Individual instances can be tailored by data driving, if safety concerns allow, with information about the hardware element(s) they are concerned with. Health Assessment components at higher levels will have a more abstract view of the system and will coordinate the reasoning of lower level Health Assessment components. The dependencies between hardware elements, which determine how failures will propagate, can be modelled by the way Health Assessment components are connected to each other.

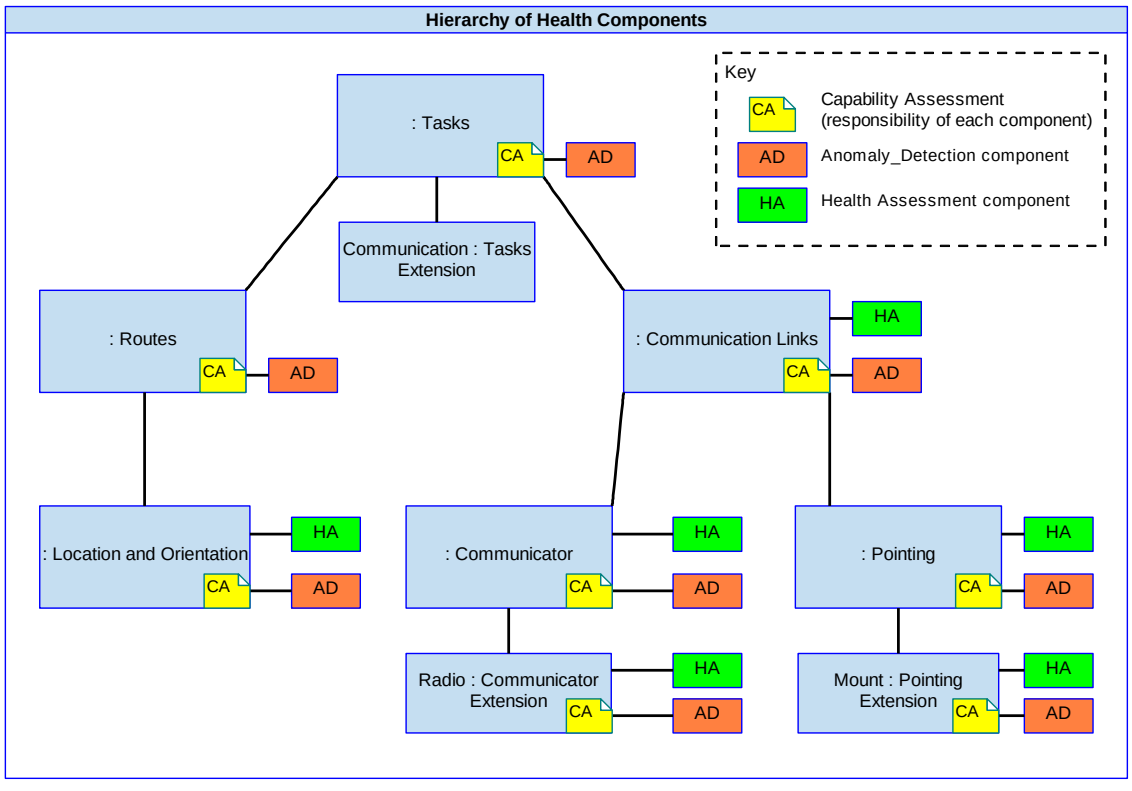

Figure 31: Hierarchy of Health Management Components

Health is relevant at all levels of the Control Architecture. Health problems in hardware resources may be observed at any level as a change in the ability to carry out tasks and actions. Components at each level rely on capability provided by lower-level components. Every component is responsible for assessing its own capability, as explained in the Capability Assessment policy.

In the example shown in Figure 31: Hierarchy of Health Management Components, the Communication Links component relies on the capability provided by the Communicator component (using a radio) and Pointing component, which uses a mount to direct the radio antenna. Each of these components has an Anomaly Detection component associated with it, because anomalies may become apparent at any point. The radio and the mount may each report correct behaviour, for example, but if the link is not achieved an anomaly will be detected. In fact, anomaly detection is relevant throughout the system, since anomalies (violated expectations) may be detected at any level.

Health assessment, by contrast, has more relevance for lower-level components that deal directly with hardware resources. In the example, the Communicator component (which deals with the radio) and the Pointing component (which deals with the mount) are clearly relevant. The Communication Links component is also involved, because there is a relationship, in hardware, between the radio and the mount which needs to be understood in order to diagnose the cause of the problem. Higher level components in the Objectives and Task Layers (see the Control Architecture policy) are not typically concerned with health assessment because dependencies between tasks are not determined by hardware.

Health Assessment components are intended to communicate with each other up, down and across the hierarchy when necessary, in order to identify the root cause of an anomaly, if that anomaly is the result of a health change. However the hierarchical relationship between each of the Health Assessment components is largely dependent on the relationships between physical items of equipment. It is expected that any single health assessment can contribute to the health assessment in another instance of the Health Assessment component. An example of this could be where two different pieces of equipment are physically co-located, such as a radio box and a GPS/INS in a nose cone. An anomaly indicates that the GPS/INS health has started to deteriorate but it may be a direct result of the radio overheating (root cause). In this instance both items may be handled by individual instances of Health Assessment components and may need to share health information in order to correctly diagnose the root cause.

Components in the PRA are unaware of the computing infrastructure of the implementation and the Health Assessment component is not responsible for diagnosing failures in the computing infrastructure. However, errors in the infrastructure can be treated as anomalies and may have implications for the system's capability and are thus reasoned about by Health Assessment.

A.1.5.6.3 Cooperation of Components

Figure 32: Cooperation of Health Management Components shows an example of how health management components work together to determine the cause of an anomaly which is not apparent in small parts of the system viewed locally.

A jettison solution has been identified which requires a door to be opened so a store can be jettisoned. The Asset Transitions component is responsible for regulating the door opening by commanding the Effectors component interfacing with the door. No anomalies are detected by the Door Effector Anomaly Detection component, nor the Sensor Anomaly Detection components; when viewed locally they appear to be working as expected. The Asset Transitions Anomaly Detection component receives the actual state of the door from the Door Closed and Door Open Sensors. The Door Closed Sensor reports that the door is closed, and this is consistent with the report that the door is not open from the Door Open Sensor component. However, the Asset Transitions Anomaly Detection component is also aware that the door was commanded open.

This anomaly in the state of the door is passed to Asset Transitions Health Assessment for assessment. In an attempt to determine if the cause of the anomaly is health related, Asset Transitions Health Assessment requests the Door Effector and Door Sensor Health Assessment components provide information on the health status of the Door Effector and Door Sensors. The Door Effector Health Assessment component determines that the necessary hydraulics and power to the Door Effector have been supplied, and that no fault is found. Similarly, the Door Sensor Health Assessment components determine that, with the evidence available, no fault with the Door Sensors is identified. When Asset Transitions Health Assessment assesses the evidence it determines that it is likely that either both Door Sensors have failed or that the Door Effector has failed. Using its subject matter knowledge of Door Sensor and Door Effector failures (including likelihood), Asset Transitions Health Assessment determines that the most likely cause of the anomaly is a failure of the Door Effector. Although not shown on this diagram, this would result in the Door Effector Health Assessment component being informed that the door effector is not working, and the health reported to interested parties for capability assessment and maintenance purposes.

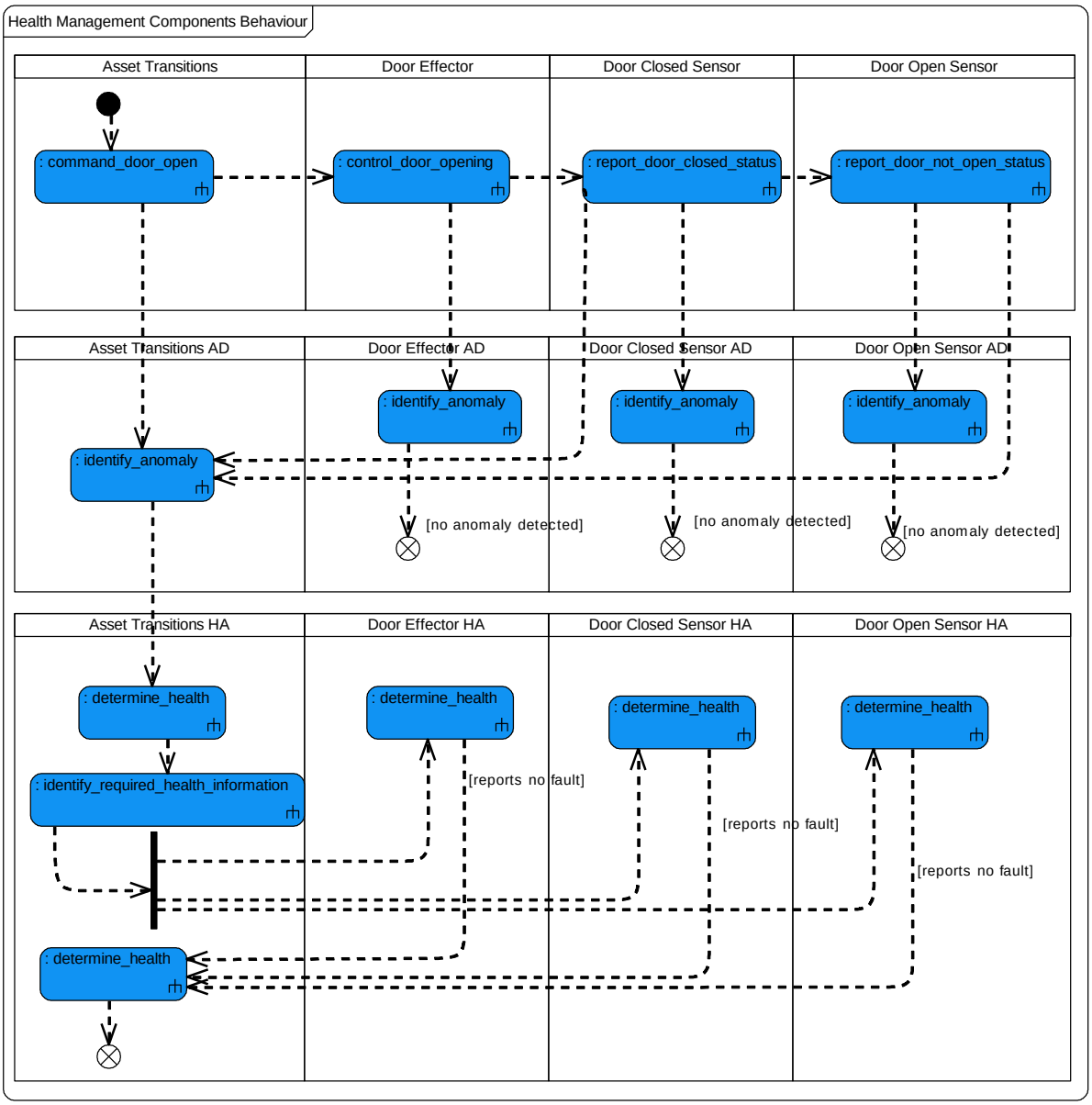

Figure 32: Cooperation of Health Management Components

A.1.5.7 Understanding of Capabilities

Capability assessment, as explained in the Capability Assessment policy, is the responsibility of each component. When an anomaly is detected, associated components are notified. The anomaly may be the result of a failure somewhere else in the system, but it still represents an unexpected state as far as the operation of the component is concerned. The component may have already observed the unexpected behaviour as part of its usual monitoring of its actions. Either way, the anomaly may be directly related to the capability of the component, especially where there is a high degree of monitoring in the hardware, as well as its own behaviour, as is often the case in safety-critical areas.

This is shown in Figure 30: Health Management of a Single Resource where an anomaly in the state of a Resource has been identified. Notification of the anomaly is received by both the Resource and the Resource User via Health Assessment and they revise the determination of their capabilities as a result. This information is then available in the component hierarchy. Components that rely on that capability can understand that their own capability has been changed as a consequence.

Figure 30: Health Management of a Single Resource also shows that the detected anomaly is simultaneously reported to the local Health Assessment component, which will identify the hardware element that has caused the problem. That information will feed back into the associated component. In the example, this is the component where the anomaly was detected, which is already aware of the problem. It may be, however, that the information from Health Assessment is more specific, or more certain, about the problem and the effect on capability can be determined more precisely or with more confidence. If the failed hardware is associated with a different component, the update from the Health Assessment component could be its first indication of the problem. Simultaneously, the Cyber Defence component assesses the anomaly to see if it is a sign of a cyber attack.

Capability assessment is the responsibility of each component whereas Health Assessment and Anomaly Detection are individual instances of components in their own right. When viewing Figure 30: Health Management of a Single Resource it is important to view it in this context. An unambiguous view of how the capability assessment functionality is being implemented can be viewed in Figure 34: Capability Example: Structure in the Capability Assessment policy.

A.1.5.8 Insulating the System from Hardware Dependencies

One of the goals of the health management components is to insulate the other parts of the system from having to understand the details of their dependency on hardware, especially as it relates to failure and damage.

Exploiting Programmes will want the freedom to replace equipment with minimal impact on the system. Unless the replacement is identical, it may have different failure modes, even if it is designed to provide the same functional behaviour and interface. In that case, the changes to accommodate various failure modes can be restricted to the Anomaly Detection and Health Assessment components, allowing other components that deal with the equipment to be unchanged.

Similarly, only the health management components need to understand the dependency of component software on the computing infrastructure. This allows the software to be redistributed without affecting the bulk of the components.

This separation makes exploitations of the PRA more portable and resilient to obsolescence.

A.1.5.9 Implementation

For a specific deployment of the PRA, health components will need tailoring, but they must also work together to diagnose failures. A desirable approach is to have a generic implementation of Anomaly Detection and Health Assessment which can be data-driven to accommodate the specific details of the implementation, including types of hardware and how they are connected. Bespoke implementations may also be required in some specialised areas. In all cases, the connectivity of the implementation will determine which components Anomaly Detection and Health Assessment report to.

ISO 13374 Ref. [18] covers condition monitoring and diagnostics of machines. It is a widely-recognised industry standard for fault diagnosis, and the PRA must support implementations that conform to it. Components have been designed to correspond to the six processing blocks identified in the standard, as shown in Table 1: Correspondence between ISO 13374 and the PRA. This correspondence enables a deployment of the PRA to conform to the data processing and communication protocols required by the standard.

ISO 13374	PRA
Data Acquisition	Not in the PRA: this will be handled by infrastructure.
Data Manipulation	This may be handled by infrastructure or by Anomaly Detection as required by the deployment.
State Detection	Anomaly Detection component.
Health Assessment	Health Assessment component.
Prognostic Assessment	Health Assessment component.
Advisory Generation	Out of scope. This processing block is concerned with the maintenance and use of equipment, which would be part of fleet management.

Table 1: Correspondence between ISO 13374 and the PRA

A.1.6 Capability Assessment

A.1.6.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Health Management
- Cyber Defence
- Dependency Management

A.1.6.2 Introduction and Scope

This policy explains how the concept of capability assessment has been applied to the PRA. It describes what is meant by capability in the context of the PRA and how to achieve consistent capability assessment in a deployment of the PRA.

A.1.6.2.1 What is Capability?

Capability is the ability to do something, or the ability to perform a particular function based on internal system factors only, external factors will affect performance not capability. A system's capabilities are derived from the resources it has at its disposal, and the uses it is able to put them to. Higher-level capabilities such as searching depend on lower-level capabilities such as sensing and data processing, which in turn rely on resources like sensors and power. The concept of capability therefore applies at every level of a system, but its expression changes from one level to another.

A system's capability varies with time and internal circumstances. For example, the designed capability of a resource such as a sensor may be restricted by the way it is installed. A failure can remove or restrict the capability of a hardware element, which may have consequential effects on higher-level capabilities that depend on the resource, or a constraint imposed on a component may mean its capability is restricted, again with possible effects on higher-level capabilities.

Within the PRA, capability is distinguished from achievability and feasibility. Achievability is relative to a specific requirement, whereas feasibility applies to a particular solution; these are described in the Dependency Management policy. The achievability of a requirement is influenced by the feasibility of its planned solution, which relies on the available capability. Capability assessment does not take resource allocation into account; theoretical capability is determined, ignoring how a resource may already be allocated.

A.1.6.2.2 Why is Capability Assessment Important?

Every part of a PRA-based system is responsible for making decisions at its own level. To do that correctly, it needs to know not only the requirements and constraints that are imposed on it, but also the capability that it has at its disposal. Components can be deployed in many configurations, so capability must be assessed in a way that reflects the configuration of an Exploiting Platform. This applies both to the way it is designed and to how it is affected by run-time events such as reconfigurations and failures. A dynamic assessment of capability throughout the system means that every part of the system has an up-to-date view of its capabilities that will enable it to make decisions.

A.1.6.2.3 Aim of this Policy

This policy explains how the PRA has been designed to support capability assessment in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Scalable:** This policy allows systems with different numbers of components to be able to assess and manage their capability.
- **Utility Across A Range Of Missions:** This policy allows the PRA to be used to build systems that can be readily configured to support different missions.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** This policy allows the PRA to be used to build systems that are reliable because they have a precise understanding of the impact of failures which enables more intelligent decision making around maintenance, e.g. delaying a maintenance activity because the impact of the failure on capability is minor compared to the impact of the maintenance task on availability.

A.1.6.3 Overview

The principle behind this policy is that every component in the PRA is responsible for assessing its own capability. This applies to components at every layer in the control architecture (see Control Architecture policy). Each component bases its assessment of its capability on its own inherent capability (which may be influenced by the data loaded into the component, or the lack of such data), and the capabilities that it relies on. The capabilities that a component relies on includes capabilities provided by other components and other capability dependencies, such as the elements of the execution platform which enable the component to operate and equipment controlled by the component.

Reconfigurations which change the dependencies between components may cause the system's capability to be re-assessed. A failure in hardware may lead to a loss of capability reported by a component. A cyber attack may cause a compromised component to suffer a loss in the capability it can provide. These losses of capability will be reflected as a corresponding loss of capability in the components that rely on these services.

A.1.6.4 Component Composition

This policy requires certain behaviour and services from the components that are actively involved in capability assessment. The interfaces to support this policy are included in the Component Composition. The Determine Capability use case in the Use Cases section covers capability assessment. The examples below show how the two services, Capability and Capability_Evidence from the Component Composition support the capability assessment within the system.

A.1.6.5 Capability Assessment

There is no separate "Capability Assessment" component: each component is responsible for assessing its own capability. Components derive their capability from the resources available to them, either directly or via other components. A resource component's capability is likely to be very heavily tied to the health of the equipment itself, but may relate to other factors. This policy expresses a standard approach which allows components to communicate their capability in a uniform way.

When a component is informed of a change of capability of something it relies on, there are three things it must do:

- If possible accommodate the change so that it can continue to provide its capability to other components as best it can.
- Assess how the change affects the feasibility of any current or planned actions and tasks, report back the actions and tasks that are no longer feasible.
- Assess how that change affects its own capability, and make that assessment available to other components.

Only the last of these is covered by this policy. The Dependency Management policy covers the first two.

The reporting of capability assessment is in general not binary; that is, the capability is not just reported as available or not available (although it could be). Instead, different levels of capability (perhaps due to a degradation in capability) can be reported against each aspect of capability that a component reports. This enables a component receiving the results of a capability assessment (i.e. a statement of another component's capability) to determine its own capability in an accurate way, with the level of dependant capability being taken into account at any particular moment in time.

The Health Management policy explains how the capability of hardware resources is assessed. This is an important source of information for capability assessment.

Figure 33: Capability Interactions shows the standard services that are available on a component that assesses its capability (most components are in this category) and how two such components communicate their capability. Each component has two services:

- A Capability service, which makes its own capability assessment available;
- A Capability_Evidence service, by which it receives information about the capabilities it relies on.

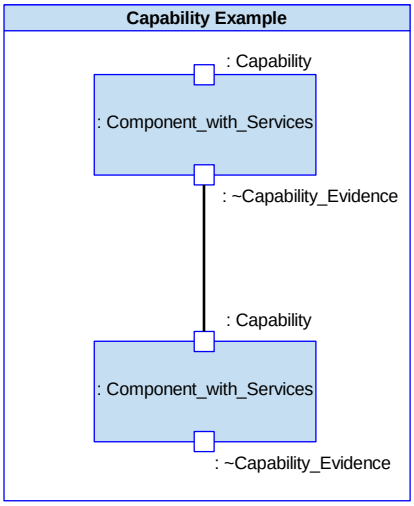

Figure 33: Capability Interactions

Some components (e.g. resource components) will receive information about their resource directly, rather than through the Capability_Evidence service.

Each component assesses its capability in its own terms by understanding how it relies on the components and resources that support it. This pattern is applied repeatedly so that each component can understand, in its own terms, how its capability is affected by changes in the capability of resources.

Figure 132: Capability Data Model expresses a generic definition of capability that can be specialised to specific components.

The Capability_Evidence service is also used for capability information that comes from the Anomaly Detection, Health Assessment and Cyber Defence components. This is shown in Capability Assessment Example and there is more information in the Health Management and Cyber Defence policies.

The Capability service is used to communicate capability information to wherever it is needed: this could include HMI components, for example, as well as components that rely on that capability.

A.1.6.6 Capability Assessment Example

This section gives a typical example of how components might work together to provide capability assessment. The two diagrams (Figure 34: Capability Example: Structure and Figure 35: Capability Example: Behaviour) show structural and behavioural features of the example and should be considered together.

A.1.6.6.1 Structure

In Figure 34: Capability Example: Structure Tasks has access to an action layer component, Action Component W, which can use two resource components, Resource Component X and Resource Component Y, to carry out actions in support of a task. All these components exchange capability information using their Capability and Capability_Evidence services. This is a simple example: in a more realistic case, other action components may also rely on the capabilities provided by Resource Component X and Resource Component Y.

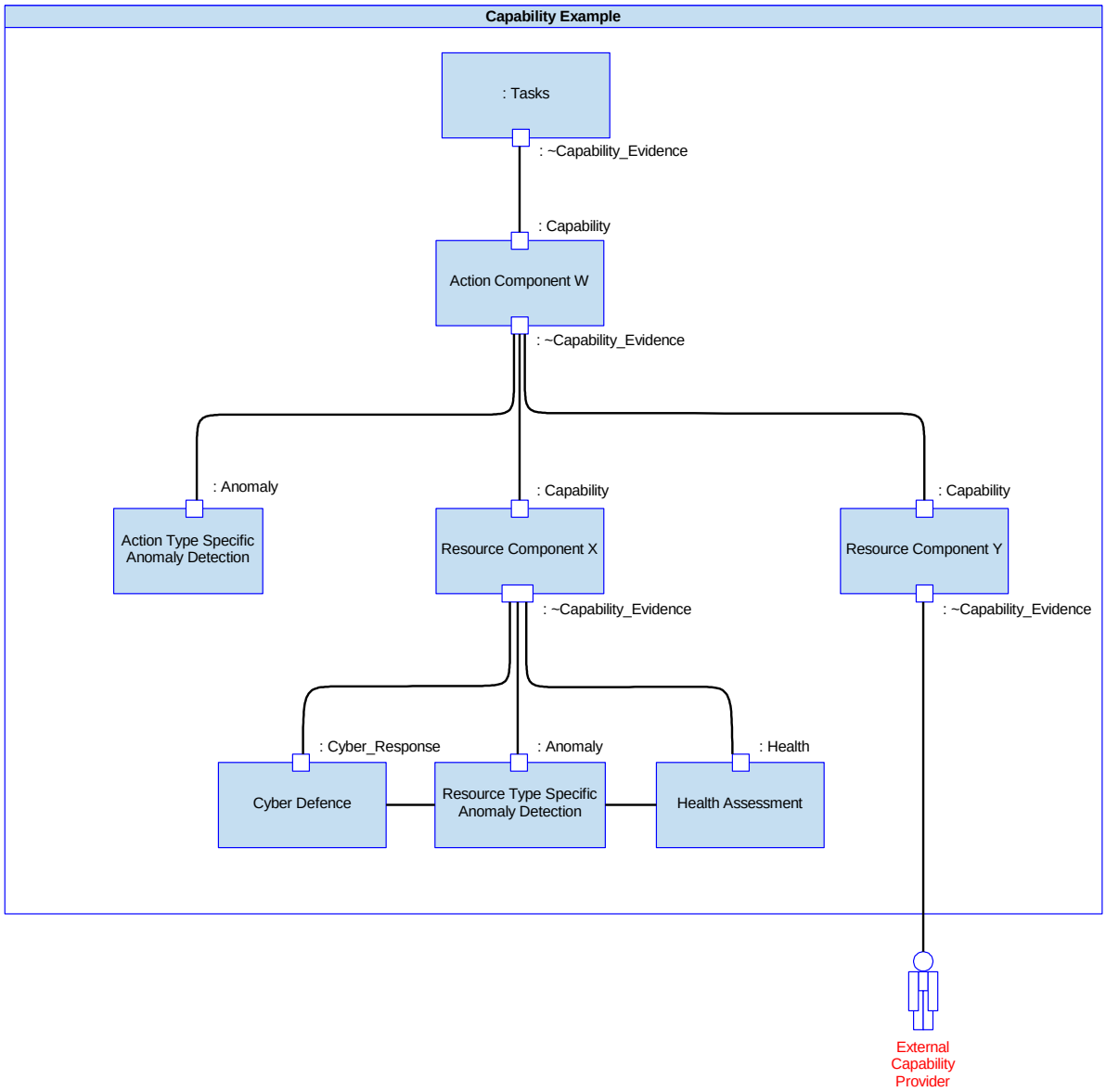

Figure 34: Capability Example: Structure

Resource Component X can receive capability evidence from Resource Type Specific Anomaly Detection, Health Assessment and Cyber Defence. Instances of these components are created to provide capability evidence specific to the resource. This could be:

- From Resource Type Specific Anomaly Detection: an anomaly that maps directly onto a capability. For safety-critical resources, an error message often identifies the loss of a capability unambiguously.
- From Health Assessment: a change in the health status of the resource which directly indicates the loss of a capability.
- From Cyber Defence: a change in the threat status of the resource because it has been compromised by a cyber attack and should be ignored.

Resource Component Y receives capability evidence directly from its resource, which is shown as an actor external to the system.

Also shown is Action Type Specific Anomaly Detection which is tailored to identify anomalies that only become apparent at the action level when information from different parts of the system is compared.

There will also be other interactions between these components as they cooperate to plan and execute a task, but these are not shown.

In this example, Health Assessment and Cyber Defence provide capability evidence only to Resource Component X, but they can and should provide capability evidence to all components, not just resource components. This is because Health Assessment and Cyber Defence are able to provide information about the underlying computing infrastructure, which is relevant to all components' capability evidence. This may also require a component to resolve differences in the capability evidence provided by different sources about the same capability, such as where the infrastructure is healthy but is unavailable due to potentially being compromised by a cyber attack.

A.1.6.6.2 Behaviour

Figure 35: Capability Example: Behaviour shows the behaviour associated with the example above. It shows how the components (represented by swimlanes) behave when a capability change occurs.

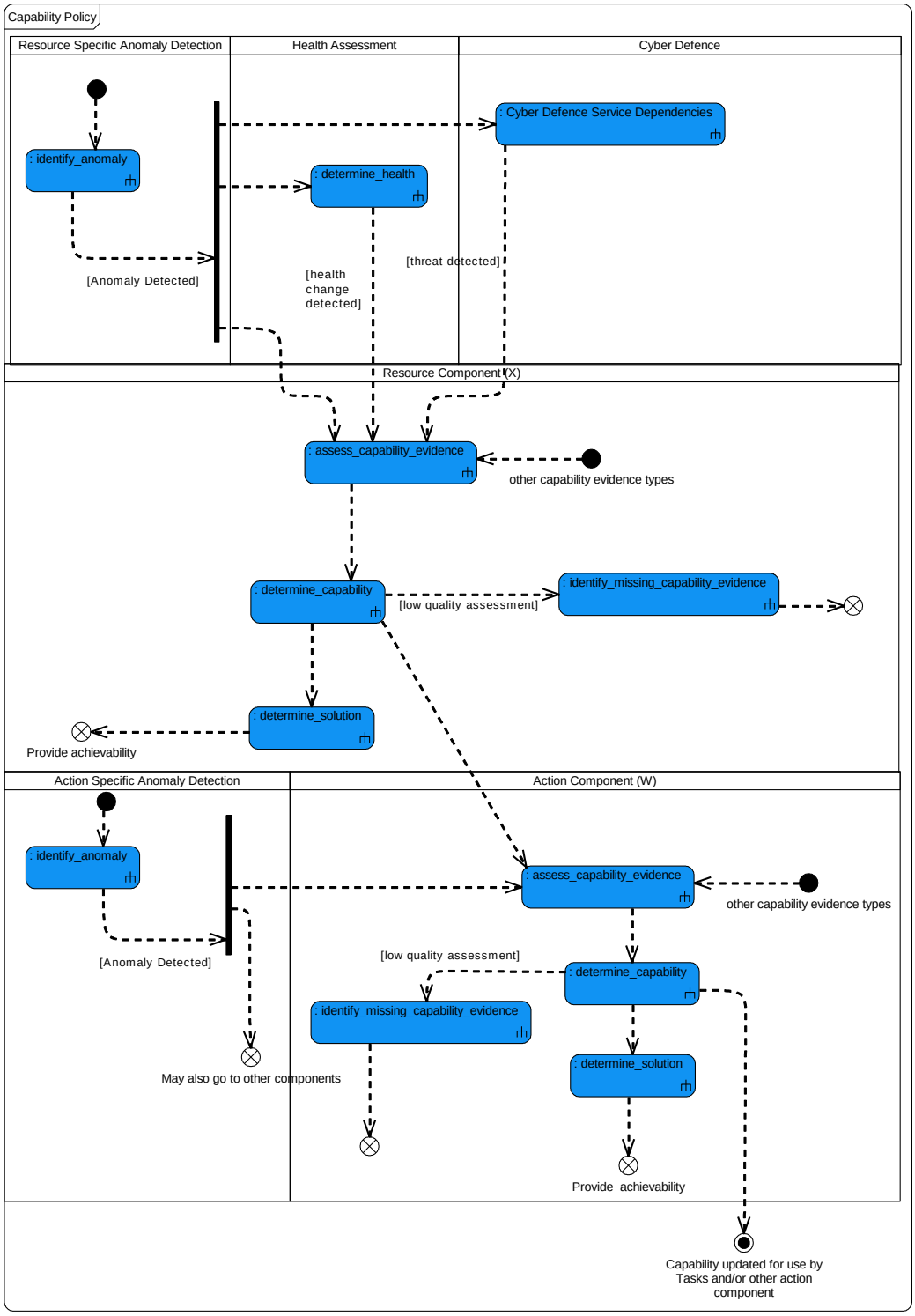

Figure 35: Capability Example: Behaviour

When a problem occurs, Resource Type Specific Anomaly Detection will identify an anomaly and notify Resource Component X, Health Assessment and Cyber Defence (see Health Management policy for more detail). If the anomaly can be mapped to a capability change, Resource Component X will recognise it at the Capability_Evidence service and will react to it by re-assessing its capability and re-considering the feasibility of current or planned activities. Resource Component X updates its Capability service to inform Action Component W (and other components that can use its capability) of the change.

Action Component W re-assesses its own capability based on the new capability evidence supplied by Resource Component X. As with all components, it uses the Dependency_Map (see Component Composition) to make the assessment. This captures the relationship between a component's own capabilities (expressed in terms of its own subject matter) and the capability evidence supplied to it. Not all changes in capability evidence result in a change to the overall capability of a component, if it has other resources available to it.

In the example there is a change, however, which Action Component W reports to Tasks using its Capability service. In this way the change in capability of Resource Component X is communicated, at an appropriate level of abstraction, to all components that need to know about it.

The feasibility of current or planned activities is covered in the Dependency Management policy.

In parallel, Health Assessment and Cyber Defence will assess the anomaly to decide whether it represents a health change or cyber threat. This may result in additional information about the capability change which may be more specific and allow the component to improve its assessments.

Action Type Specific Anomaly Detection identifies anomalies that arise from inconsistencies between different parts of the system that are only visible at the action level. Any such anomalies are reported to Action Component W (and also to action-level instances of Health Assessment and Cyber Defence components, not shown in this example).

A.1.6.7 Missing Information

If a capability assessment is not sufficiently specific or certain, the component may identify an item of capability evidence which could improve its assessment. Such information may be available only on demand (as a test result, for example). Since the requested information may not be available immediately (or at all) a component must not wait for the information to arrive, but should report the results of its current assessment. The component may also need to request improvements to the quality of the capability evidence received, for example due to a time delay between updates, or to increase the confidence in the capability evidence. When (if) the additional information arrives, in the form of capability evidence, it may result in an improved capability assessment, the result of which is provided on the Capability service in the usual way.

A.1.7 Multi-Vehicle Coordination

A.1.7.1 Pre-Reading or Related Policies

Prerequisite

- Control Architecture

Read in conjunction with

- Dependency Management
- Capability Assessment

A.1.7.2 Introduction and Scope

This policy presents the general principles of multi-vehicle coordination that can be applied at different layers in the PRA Control Architecture. Different scenarios require that different interactions be exercised at the appropriate layer.

A.1.7.2.1 What is Multi-Vehicle Coordination?

Platform Independent Model (PIM) components, including instances of the same component, can be deployed on different vehicles. This policy describes principles for how these components should be deployed to interact between vehicles in a coordinated manner, for the purpose of achieving a greater operational effectiveness than if the same vehicles were deployed to operate independently from each other. An example of this might be where a weapon deployment and target illumination are provided by separate vehicles, while they are both being tasked using intelligence from a third, stand-off vehicle.

In multi-vehicle coordination the concept of a flight is important. Multi-vehicle coordination within the PRA is only supported at the level of a flight; other vehicles must actively join a flight in order to cooperate. Multi-vehicle coordination provides the facility to plan and execute tasks and actions in a way that pools the capabilities of all the vehicles within a flight.

A Platform Specific Model (PSM) must take into account issues like latency. Although these issues are not addressed in the PRA, the architecture does allow the necessary interactions to enable PSMs to address them, such as the synchronisation interactions used in Figure 39: Multi-Vehicle Action Execution Example. Multi-vehicle coordination is an area where such interactions are particularly important.

A.1.7.2.2 Why is Multi-Vehicle Coordination Important?

The PRA is designed to support coordination between different vehicles under various circumstances. This enables different coordination approaches to be adopted and different coordination behaviour employed on different types of mission. It gives potential PRA exploiters the ability to coordinate without significant system modification.

A.1.7.2.3 Aim of this Policy

This policy explains how deployments of the PRA can enable multi-vehicle coordination in a manner consistent with the PYRAMID KURs.

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** Different deployments of the PRA may require different methods of achieving coordination between vehicles. This policy enables deployments of the PRA to be designed so that they can be configured to use different coordination strategies or coordinate between different vehicle types.
- **Exploitable:** Different deployments of the PRA may be required to coordinate on the same mission, therefore this policy allows for this coordination between non-homogeneous systems.
- **Scalable:** The ability to coordinate during a mission allows capability to be split between participating vehicles. This policy enables smaller, more specialised vehicles, e.g. one with a laser designation capability and another with a weapon release capability, to come together for greater effect.
- **Utility Across A Range Of Missions:** Different mission scenarios may require different methods of achieving coordination between vehicles. This policy enables different coordination behaviour on different types of mission.

A.1.7.2.4 Exclusions to this Policy

The following areas of functionality, although related to multi-vehicle coordination, are provided within the PRA but are not discussed by this policy:

- Flight composition management.
- Unmanned Air Vehicle (UAV) control handover.

A.1.7.3 Overview

The principles of multi-vehicle coordination addressed in this policy can be summarised as follows:

- The PRA only considers deployments with multi-vehicle coordination between vehicles in the same flight.
- Dependencies identified by a PRA component can be satisfied by components hosted either on the same vehicle or on other vehicles within the flight.
- This is coordinated as needed at whatever level of the control architecture is appropriate.

The levels of multi-vehicle coordination considered are as follows:

- **Planned Objective Coordination** is coordination of planning at the Objective Layer, where different vehicles within a flight are assigned distinct tasks that collectively deliver an overall mission objective. This is described in Planned Objective Coordination.
- **Planned Task Coordination** addresses coordination of planning at the Task Layer, where the vehicle responsible for coordination provides tasks to its own Tasks component which then generates a series of multi-vehicle actions. Different vehicles are then assigned actions that collectively contribute to a task. These actions may be unique to vehicles, or may involve shared synchronised activity between vehicles. This is described in Planned Task Coordination.
- **Adaptive Action Coordination** addresses coordination of execution at the Action and Resource Layers through the flexible distribution, across multiple vehicles, of synchronised activities that contribute to the same complex action, based on dynamic capability regardless of vehicle role. This is described in Adaptive Action Coordination and may involve data sharing directly between Resource Layer components.

These levels of coordination can be combined as required. Some multi-vehicle activities may be achieved by flight members following their own rules, without explicit coordination by the flight lead.

A.1.7.4 Planned Objective Coordination

The highest level of multi-vehicle coordination is carried out at the Objective Layer, where different vehicles are assigned distinct tasks that collectively contribute to an overall mission objective. Each co-operating vehicle performs a received task in accordance with the Control Architecture policy, breaking down the task into an appropriate set of actions.

An example where this type of high-level coordination is appropriate would be a planned Suppression of Enemy Air Defences (SEAD) mission, where vehicles have pre-assigned roles. A particular vehicle may be assigned as a weapon deployment vehicle, another may be a reconnaissance or primary sensing vehicle and another may be allocated to the role of stand-off jamming. Tasks are then distributed by the flight lead's Objectives component based on abstracted capability knowledge being dynamically reported by the various flight members.

This level of coordination is illustrated in Figure 36: Objectives Coordinating Multi-Vehicle Tasks Example. Objectives are processed on the flight lead vehicle and distributed to the Tasks component on the same vehicle and the Tasks components on other flight members. Each Tasks component then generates its own actions for execution on the same vehicle.

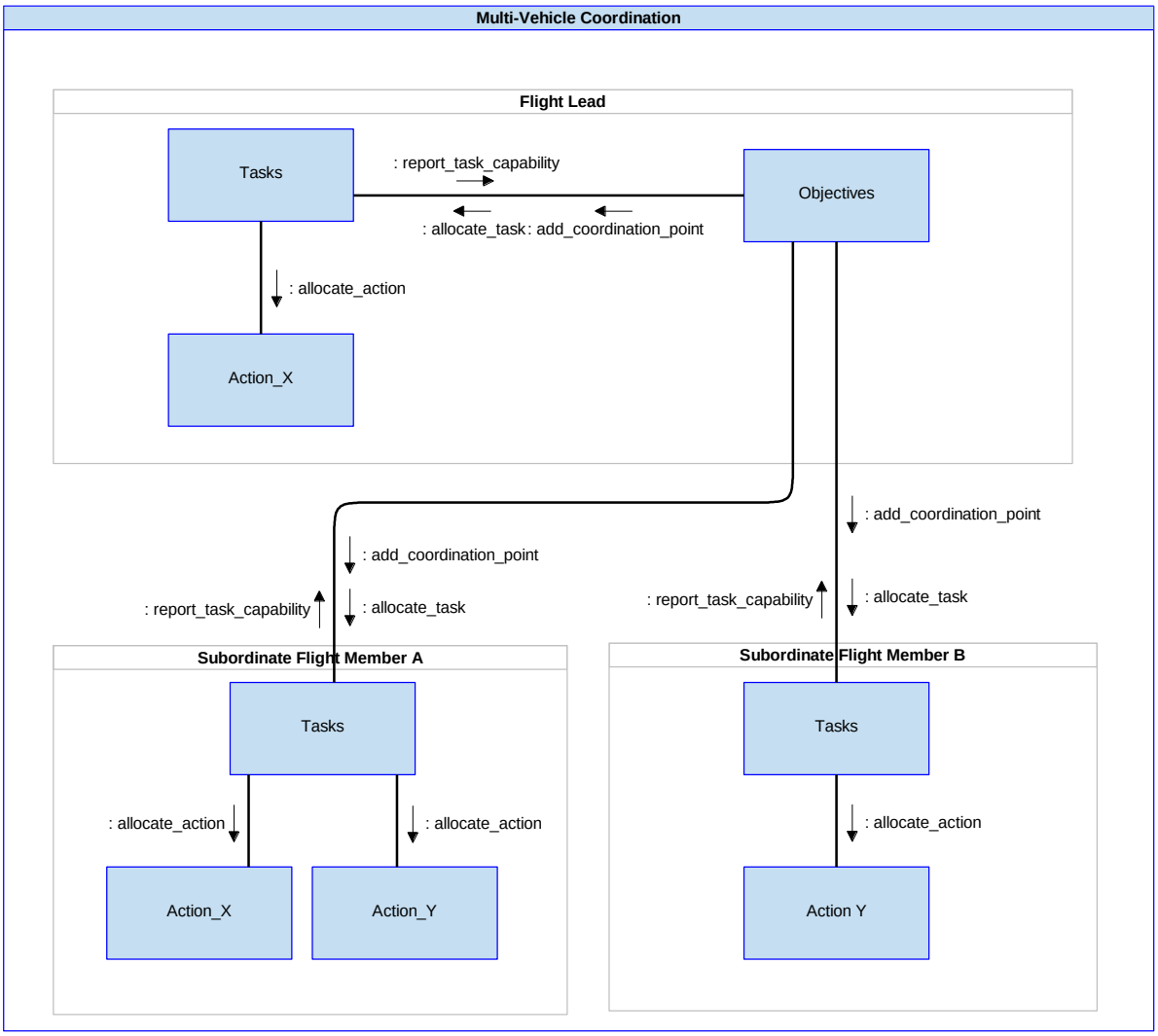

Figure 36: Objectives Coordinating Multi-Vehicle Tasks Example

A.1.7.5 Planned Task Coordination

This level of multi-vehicle coordination occurs between the Task and Action Layers. In this case, the flight lead's Objectives component provides a breakdown of the tasks to its own Tasks component. The flight lead's Tasks component then breaks down tasks into a series of multi-vehicle actions allocated across flight members. This allocation is made on the basis of dynamic capability discovery as assessed and provided by the action components from vehicles across the flight.

Feedback channels are set up as part of this planning and allocation of multi-vehicle action capability, where dynamic interaction between actions will be required between flight members at execution time. The exercising of these feedback interactions for coordination purposes at execution time is addressed in the Adaptive Action Coordination section.

A key aspect of this type of coordination is that the flight lead's Tasks component is required to understand the multi-vehicle action capabilities of flight members for any actions it is coordinating, before it can request actions of another vehicle's action component. Therefore in order to achieve this type of multi-vehicle coordination whilst adhering to the Control Architecture policy, the following policy principles apply:

- Flight members must have the appropriate Action component present if they are to contribute capability of a given action type (in line with the Capability Assessment policy) for a multi-Vehicle task (e.g. a flight member must have a Sensing component if it intends to contribute sensing actions).
- In order to participate in multi-vehicle tasks where they are coordinated by a flight lead, flight member Action component capabilities, resources and action constraints must be made available to the flight lead Tasks component. This enables the flight lead to capture capability of individual vehicles and the flight as a whole.

When these conditions are met, a flight lead's Tasks component can action any flight member Action component to fulfil their requirement. This is illustrated in Figure 37: Tasks Coordinating Multi-Vehicle Actions Example. In this example, the capability of each flight member's Action components is communicated to an Action component of the same type on the flight lead, giving knowledge of both single vehicle and multi-vehicle techniques that are available across the flight, in one place for a given action type. This capability is then provided to the flight lead's Tasks component for use in planning on the flight lead vehicle. Tasks planned on the flight lead vehicle can then utilise all available capability from across the flight through the use of actions allocated to flight members. Not all permitted interactions are shown for clarity, for example the flight lead's Tasks component may interact with another instance of Tasks components to resolve conflicts between vehicles where a subordinate vehicle has specific survivability demands that impinge on multi-vehicle tasking demands from the flight lead vehicle. This would also allow deployments where a mapping to existing military command structures is important.

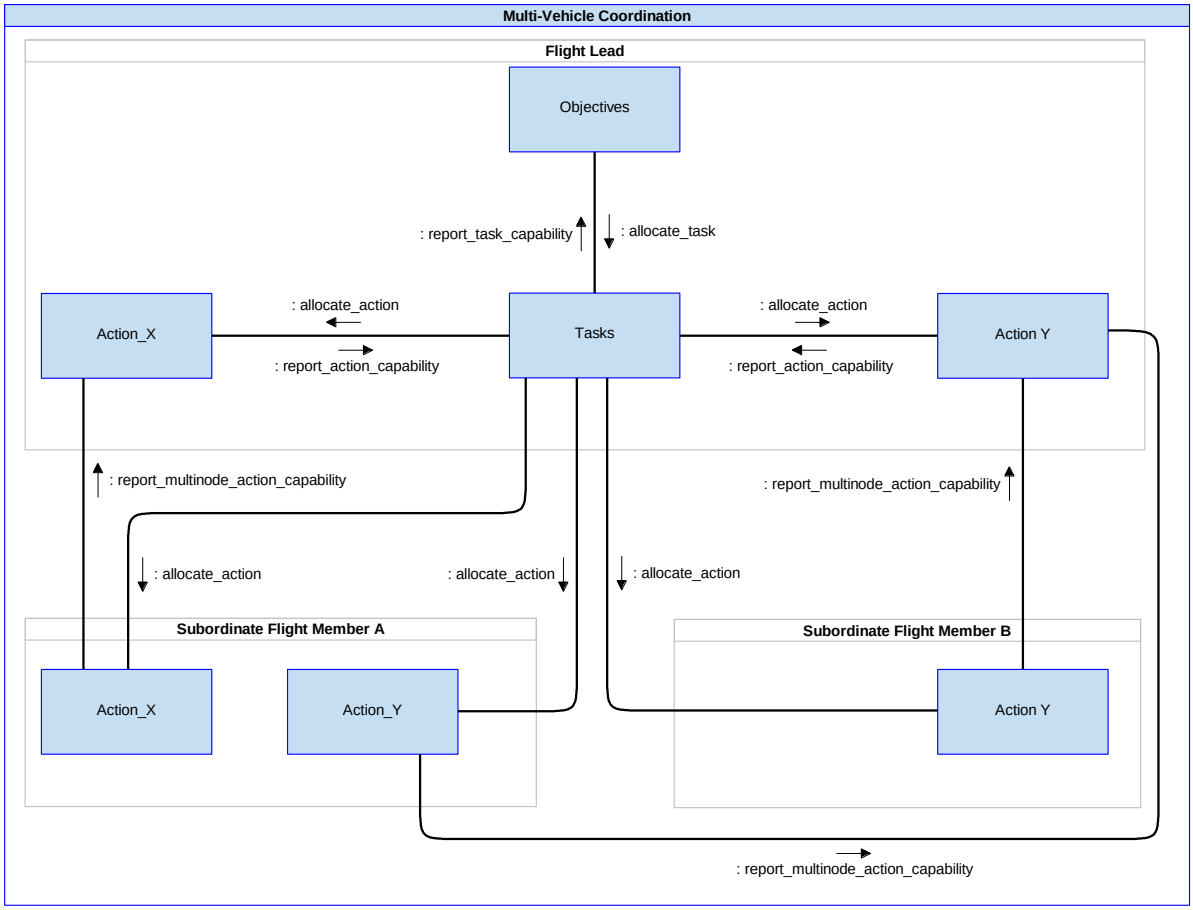

Figure 37: Tasks Coordinating Multi-Vehicle Actions Example

Objective Layer coordination can be combined with Task Layer coordination as shown in Figure 38: Multi-Vehicle Tasks and Actions Combined Example. In this example the flight lead Objectives component distributes tasks to its own and other flight member Tasks components, as well as the flight lead Tasks component distributing actions to its own and other flight member action components, all contributing to the same overall objective. This illustrates the combining of the key elements from both of the previous diagrams, namely that multi-vehicle coordination can occur from Objectives to Tasks components, as well as from Tasks to action components, in the planning of a single objective. Note: to aid clarity, Action Layer capability reporting from subordinate to lead vehicle is not shown on Figure 38: Multi-Vehicle Tasks and Actions Combined Example.

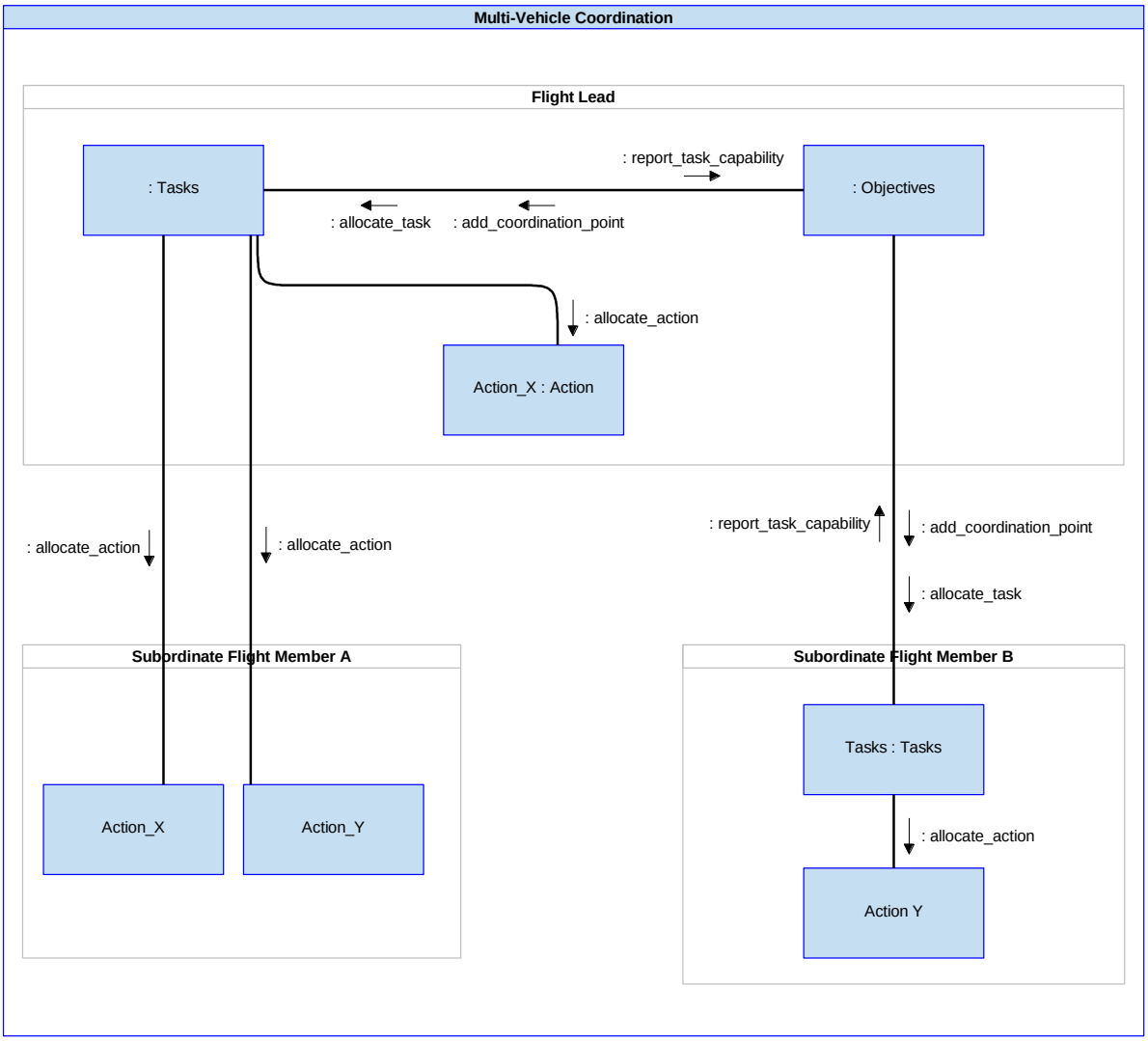

Figure 38: Multi-Vehicle Tasks and Actions Combined Example

A.1.7.6 Adaptive Action Coordination

Execution of some multi-vehicle actions requires a highly dynamic approach that is adaptive in nature. An example application that falls into this category is multi-vehicle passive geo-location, a technique that requires close coordination, synchronising and data sharing between different vehicles at the Action and Resource Layers. The relevant resource components on each participating vehicle must be able to synchronise detection of Electro-Magnetic (EM) energy and combine results for analysis.

Adaptive cooperation between vehicles at execution time requires PRA interactions at the Action Layer and below that were not shown in Figure 36: Objectives Coordinating Multi-Vehicle Tasks Example and Figure 37: Tasks Coordinating Multi-Vehicle Actions Example.

General principles of operation required at execution time for adaptive coordination of multi-vehicle actions are:

- Coordination between like action components on different vehicles within the flight.
- Synchronisation and data sharing directly between Resource Layer components on different vehicles within the flight.

Figure 39: Multi-Vehicle Action Execution Example illustrates how the components could be combined to cater for dynamically adaptive coordination of this type. In this example, the flight lead vehicle itself is shown as participating in the execution of the multi-vehicle action. This is to reduce the complexity of the diagram and is not a restriction of the PRA (i.e. the flight lead Tasks component could be on a third vehicle). A task has already been broken down by the flight lead and the resulting actions have been planned and allocated across multi-vehicle flight members, using dynamic capability discovery and solution planning by action components from vehicles across the flight, as shown in Figure 37: Tasks Coordinating Multi-Vehicle Actions Example. At execution time, the task is coordinated by interactions directly between the equivalent action components on different vehicles, as well as interactions between action components on the same vehicle in the usual way as per the Dependency Management policy. Action components then command execution using resource components on their own vehicle. Equivalent resource components on different vehicles interact directly to synchronise their activity as required for the multi-vehicle technique being performed, facilitated by a Resource Brokerage component.

Prerequisites for this type of low level adaptive multi-vehicle coordination are:

- Dynamic discovery of multi-vehicle action capability, taking into account each vehicle's current inventory, system health and available resources such as fuel.
- Feasibility planning based on the above discovered multi-vehicle capability to match against tasked action requirements.

These were addressed in the Planned Task Coordination section and illustrated in Figure 37: Tasks Coordinating Multi-Vehicle Actions Example.

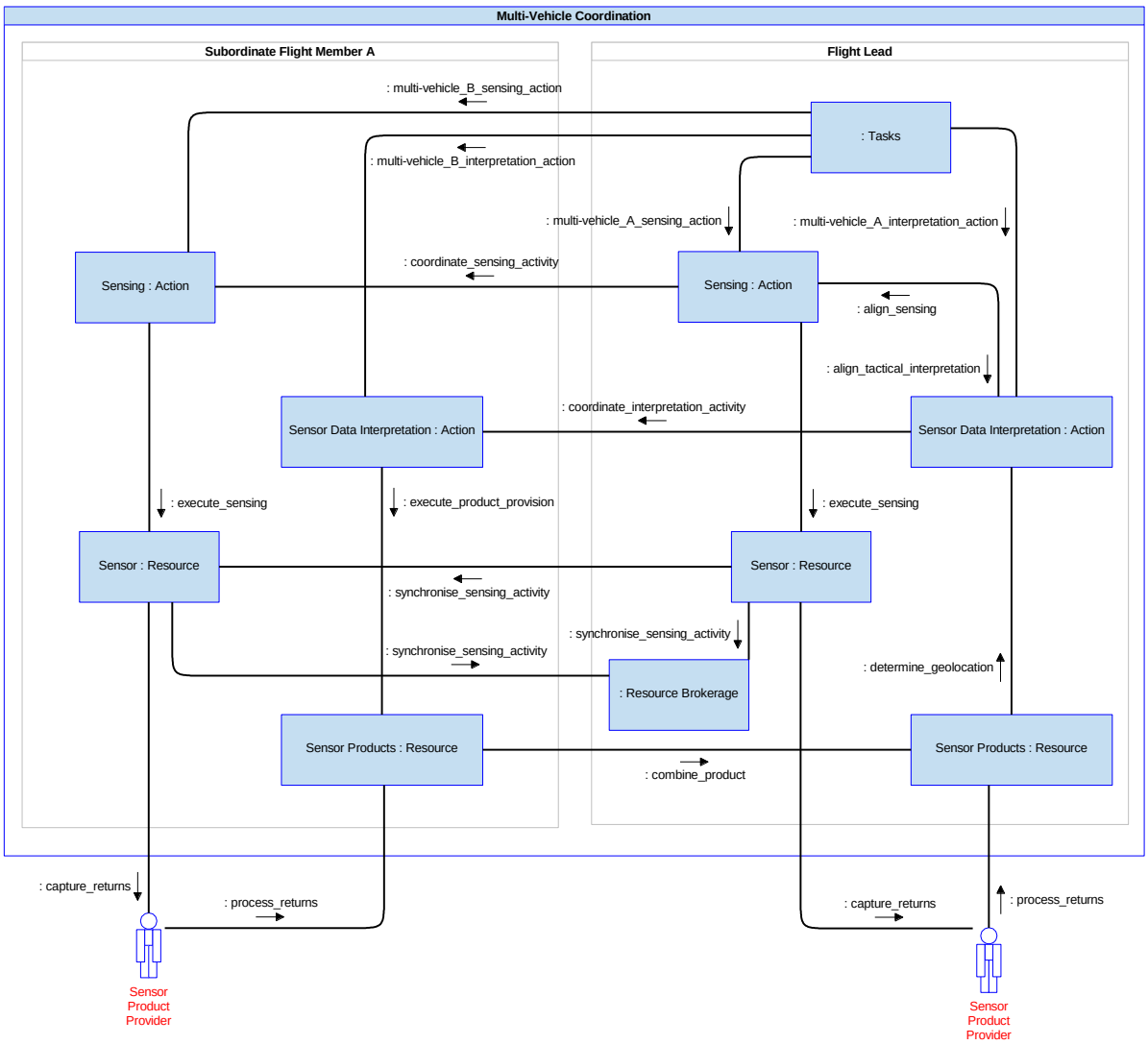

Figure 39: Multi-Vehicle Action Execution Example

A.1.8 Interaction with Equipment

A.1.8.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Resource Management
- Control Architecture

A.1.8.2 Introduction and Scope

This policy explains how PRA components can be used to interact with equipment. It describes how components throughout the architecture are involved and how the PRA has been designed to accommodate interactions with the widest possible variety of independent equipment. It also explains how adding a new item of equipment to an Exploiting Platform can be managed.

The PRA allows for interfacing with equipment of any complexity level.

A.1.8.2.1 What is Equipment?

Equipment could be either hardware (for example, antennas or control surfaces) or a combination of hardware and software that provides a capability or resource to the system under consideration. The complexity of a specific piece of equipment could vary from a simple device such as a thermistor to complex equipment with multiple data interfaces, such as a Laser Designator Pod (LDP).

A.1.8.2.2 Why is Interaction with Equipment Important?

The PRA supports the design and configuration of a system capable of interacting with equipment in order to meet its requirements. It gives PRA Exploiters the ability to incorporate legacy or new equipment into a system with minimal system modification.

A.1.8.2.3 Aim of the Policy

This policy explains how the PRA has been designed to support interaction with equipment in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Exploitable:** By allowing equipment to interface to any layer of the control architecture, the PRA allows exploitation of existing equipment and platforms.
- **Incorporate Future Growth:** By each aspect of an equipment interface being supported by the appropriate components, the PRA is tolerant of changes to equipment and infrastructure.
- **Scalable:** By allowing equipment to connect to different layers of the control architecture, the PRA supports scalability by allowing alternate designs where component functionality (or its equivalent) is held within the PRA system or in external ‘smart’ equipment.

A.1.8.3 Overview

The principles of a PRA deployment interacting with equipment are:

- The PRA components used in a PRA deployment to interact with equipment will be chosen to support the functional interfaces of the equipment.
- The interface of a PRA deployment with an equipment could involve one or more PRA components.
- The components used in the interface can be from whatever layer of the PRA Control Architecture is appropriate for the specific equipment interface. Figure 40: Component Interfaces at the PRA Boundary shows an example.
- Some aspects of an equipment's interface may be supported by the infrastructure rather than by PRA components.

A.1.8.4 Component Interfaces at the PRA Boundary

The components used in the interface can be from whatever level of the PRA Control Architecture is appropriate for the specific piece of equipment. Figure 40: Component Interfaces at the PRA Boundary illustrates this concept.

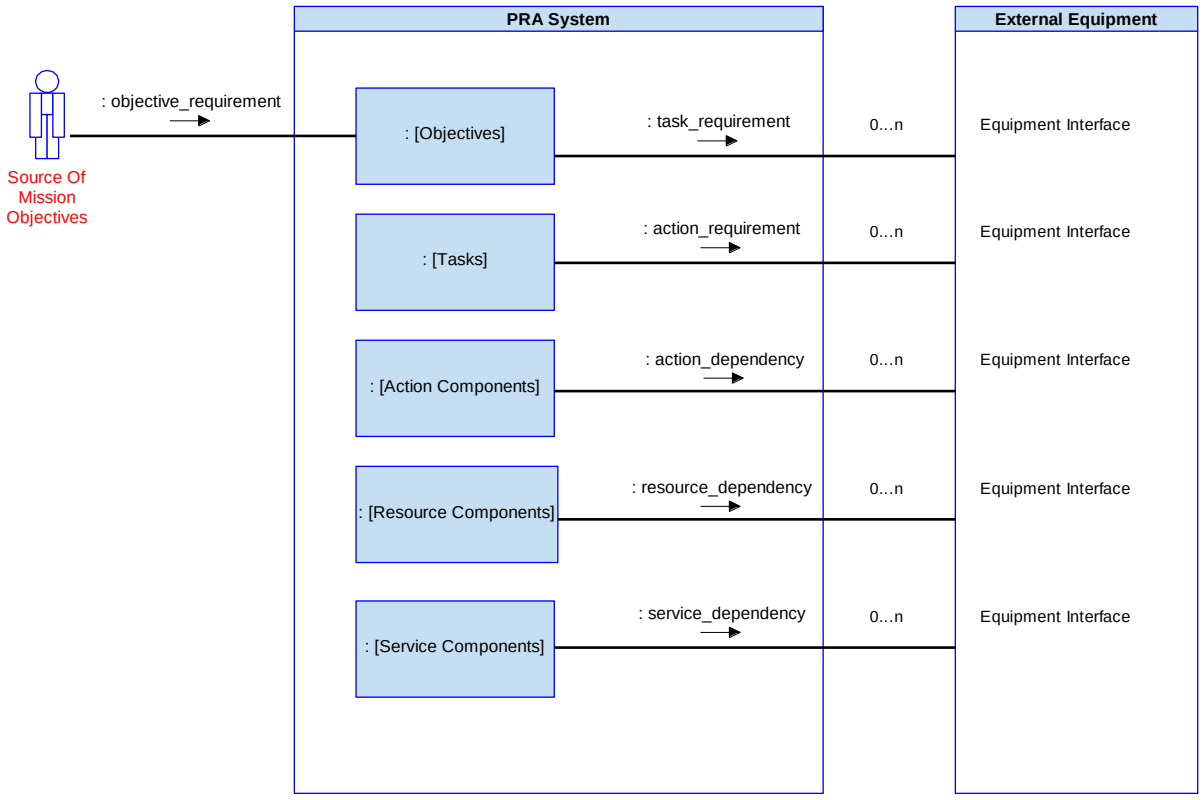

Figure 40: Component Interfaces at the PRA Boundary

A.1.8.5 Components Interfacing with Equipment

A.1.8.5.1 Examples of Resource Layer Components

The components which interface directly with an equipment's basic hardware features will reside at the lowest layer of the control architecture, the Resource Layer. Examples of Resource Layer components are shown in Figure 41: Resource Layer Example.

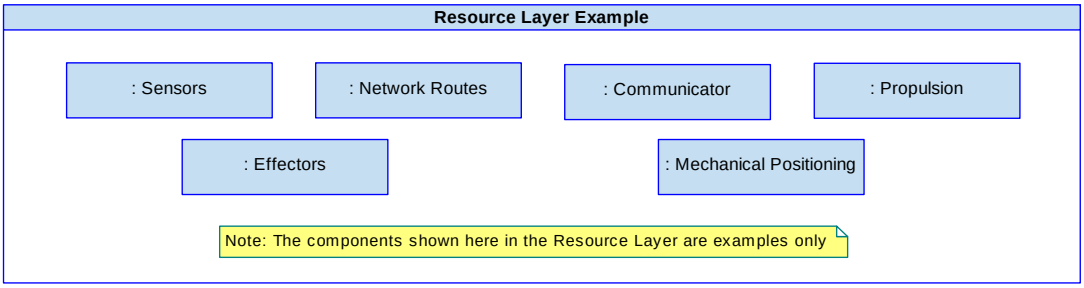

Figure 41: Resource Layer Example

A.1.8.5.2 Sensor and Effector Components

The Sensors and Effectors components are used where there is a need for direct control or monitoring of equipment. The components interact with equipment of varying degrees of complexity, e.g. a solenoid valve (effector), a jamming pod (effector) or an Electro-Optic camera (sensor). Complex equipment may, however, still provide a relatively simple interface as the component does not need to fully understand its internal complexity (which may drive capability), so may use simple instructions such as ‘jam object x’ or ‘provide imagery for area y’.

More complex items of equipment (but which still require direct control) might be considered as multiple instances of simpler equipment (e.g. individual actuators rather than ‘the flight control system’).

An example of this is shown in Figure 42: Vehicle Primary Flight Control Build Example.

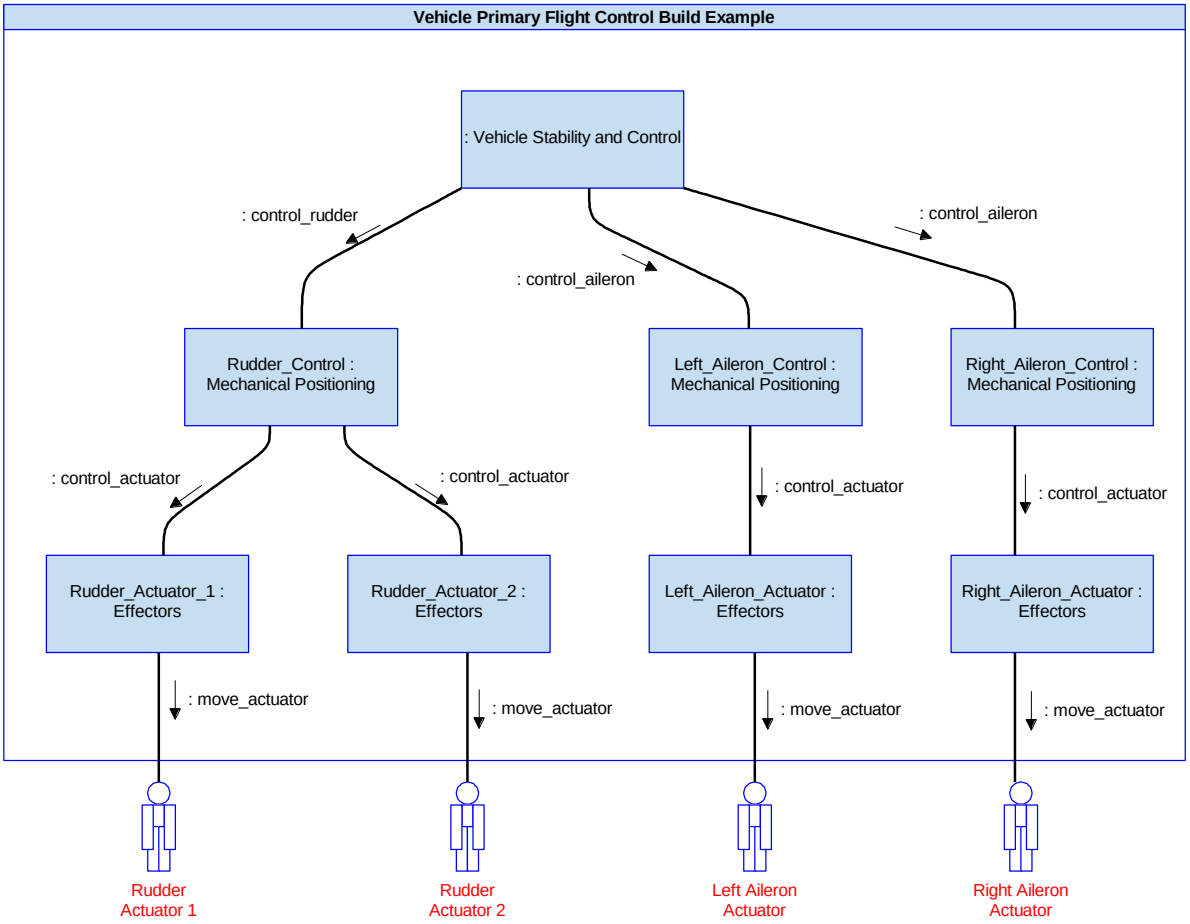

Figure 42: Vehicle Primary Flight Control Build Example

Figure 42: Vehicle Primary Flight Control Build Example shows how the Mechanical Positioning and Effectors components can be used in the primary flight control of an air vehicle. In this case, because the ailerons and rudder are specific to the system and are unlikely to change, the Mechanical Positioning component has been data-driven to create three component instances, two that represent the left and right ailerons which will be controlled in the same way and a third to represent the rudder that can be controlled differently and provide different capabilities. The Effectors component has been data-driven to create four component instances, one for each of the physical actuators.

A.1.8.6 Equipment

A.1.8.6.1 Simple and Complex Equipment

From the perspective of the PRA, the complexity of a piece of equipment refers to the number of components which are required to provide the control interface.

Note that the level of internal complexity of the equipment is not considered by the PRA - for example, a sophisticated jammer might have a simple ‘on/off switch’ from the perspective of the PRA because all further decision making about how to employ the jammer occurs within the equipment. Despite its internal complexity, in the context of the PRA it remains a simple, Resource Layer controlled equipment, as shown in Figure 43: RF Jammer with a Simple Interface Example.

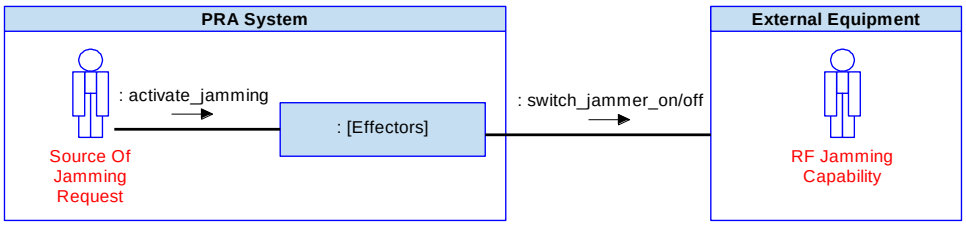

Figure 43: RF Jammer with a Simple Interface Example

A.1.8.6.2 Smart Equipment and the PRA-Equipment System Boundary

The concept of 'smart' equipment refers to equipment that, rather than being controlled from the Resource Layer, accepts requirements from a higher layer in the Control Architecture (e.g. Task or Action Layer requirements).

The components that interface with the equipment may vary depending on the complexity and capability which resides within a specific piece of equipment. 'Smart' equipment is able to interact with higher layers of the PRA Control Architecture because it has a degree of intelligence that enables it to interpret abstract requests corresponding to those layers, and is also able to control some, or all, of its own hardware without needing to be instructed by resource component(s).

Note that 'smart' equipment that receives requirements at an Objective, Task or Action Layer may still also interface with the components in the Resource Layer - e.g. to request power.

Figure 44: The PRA System/Equipment Boundary shows three examples of equivalent pieces of equipment - in all cases the initial requirement and the hardware that will ultimately be used are the same, but in each case additional capability resides within the equipment. This means that the PRA system boundary is drawn at a different point and hence the requirements are placed on the equipment from different components.

Note that the components are shown inside the equipment for illustrative purposes only. The equipment need not contain PYRAMID compliant components, but will contain equivalent functionality.

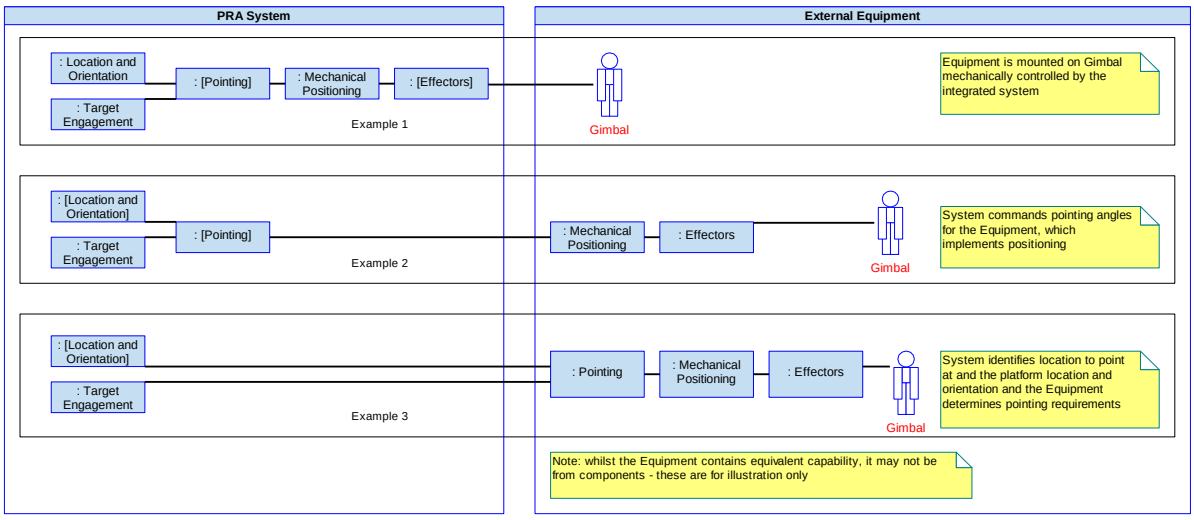

Figure 44: The PRA System/Equipment Boundary

A.1.8.6.2.1 Integrating with Equipment

Each type of hardware item which a particular component will interface with has some commonality in the capabilities it provides and the way it is controlled. For example, a Paveway IV bomb and an AMRAAM missile will both require target and navigation information so a component that understands armaments would have a standard interface to transfer this priming information.

When a piece of equipment is integrated with a PRA deployment, interfacing PRA components need to know its capabilities and the control commands required to deliver those capabilities to the relevant components within the PRA deployment. This could be achieved through the data driving of a resource component to define the capability and control functions of a piece of hardware. Equipment that has been designed to be PYRAMID compliant may have the ability to publish its capabilities directly. For other equipment, e.g. legacy equipment, this type of functionality may not fully exist, if at all, and in that case the relevant interfacing component, e.g. Sensing, will use data-driving to capture and report the equipment capability. Further information on data driving can be found in the Data Driving policy.

Adding a new item of equipment that interfaces with the same components as an existing item of equipment will have no impact on other (non-interfacing) components in the PRA deployment, as the equipment capability is reported to the same components that are responsible for that interface.

If a new piece of equipment provides capability to a component that was not previously involved in an equipment interface, then if that component is already present, there should be no changes required to other (non-interfacing) components in the PRA deployment.

If a new piece of equipment provides capability not understood by the PRA deployment, then the appropriate component needs to be added to the deployment to control the interface.

Note:

- The approach outlined in this policy is a general approach.
- The PRA may not control all equipment or all the interfaces used by equipment. For example, it may be appropriate for the infrastructure to control some equipment interfaces (e.g. processor cooling fans).

A.1.8.6.3 Examples of Equipment

High-level examples of interfaces between a PRA deployment and external equipment are shown in the following sub-sections. For a more detailed discussion of the types of interfaces which may exist with external equipment, refer to the Interfacing with Deployable Assets policy.

A.1.8.6.3.1 SATCOM

The equipment in these examples is a UHF Transceiver on a gimbal assembly, which is assumed to have low and high power output modes for transmission controlled by a safety critical discrete (as high power can cause injury to people), and to require encryption and/or high integrity protection which is performed outside the SATCOM equipment.

In the first example (Figure 45: SATCOM Equipment (Direct Control Example)), the gimbal direction is given to the equipment, whilst in the second example (Figure 46: SATCOM Equipment (High Level Control Example)) the equipment is given satellite location plus vehicle location and attitude from which it is able to calculate gimbal direction itself.

Note that in this example, by moving to a piece of equipment with greater internal capability, the number of

components involved in the interface increases to six components rather than five, as the equipment requires the location of the platform and destination to determine pointing requirements rather than being provided with a pointing angle determined within the PRA system.

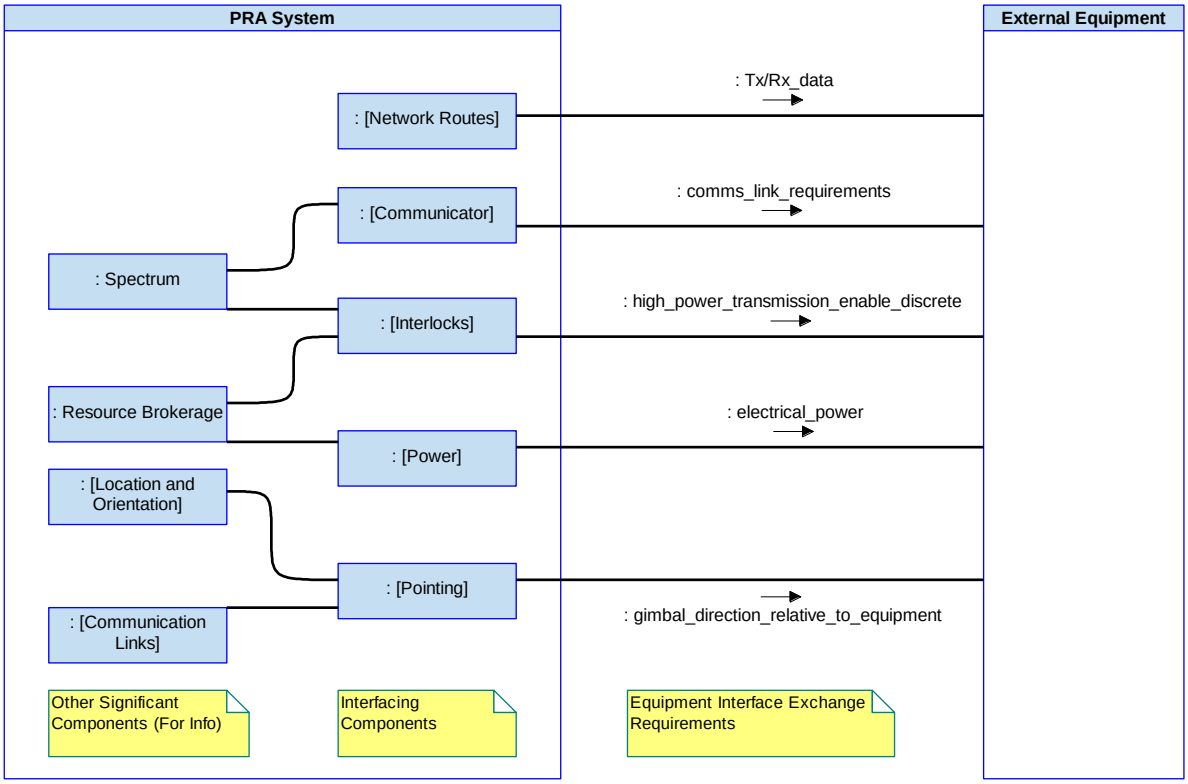

Figure 45: SATCOM Equipment (Direct Control Example)

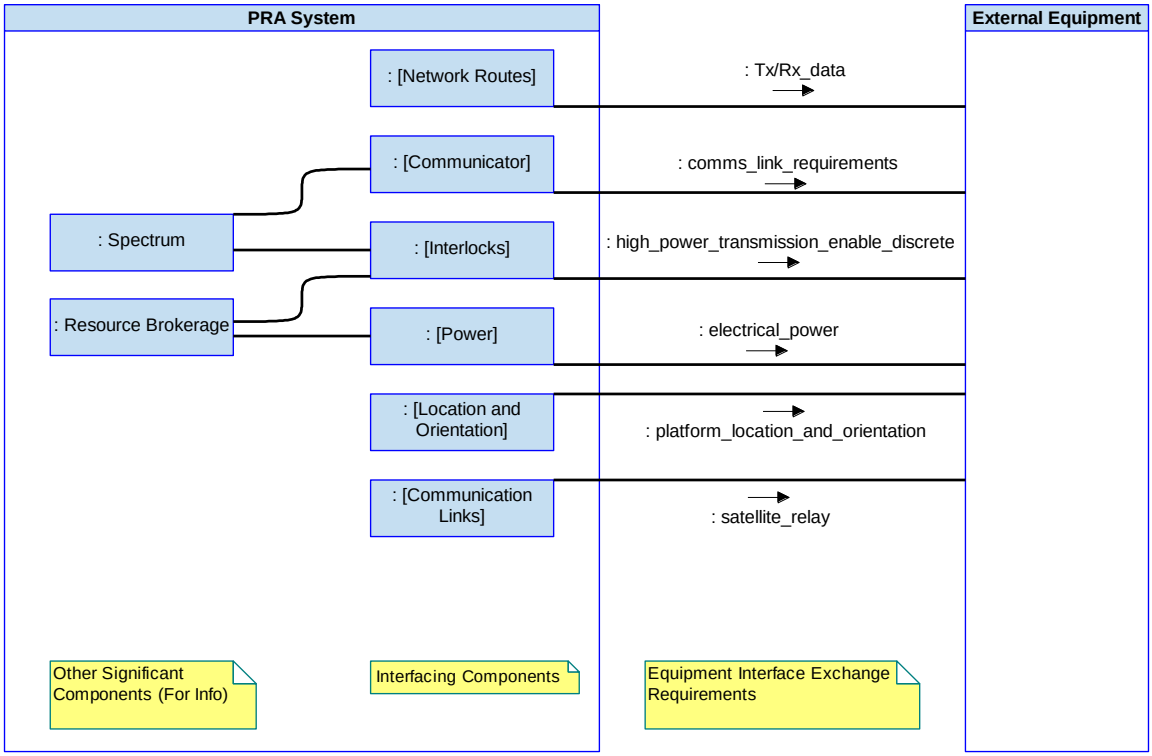

Figure 46: SATCOM Equipment (High Level Control Example)

A.1.8.6.3.2 Laser Rangefinder and Designator

The equipment in these examples is a multifunctional Laser Designator Pod. It incorporates an Electro-Optical sensor, Laser Rangefinder and Laser Designator, all mounted on a gimbal assembly to point it in the required direction. The equipment is assumed to have non-eye-safe lasers controlled by a safety critical discrete (as they can cause injury to people), and the capability to lase with a specific Pulse Repetition Frequency code when commanded.

In the first example (Figure 47: LDP Equipment (Direct Control Example)), the required gimbal direction is provided relative to the equipment until a target is designated and Electro-Optical track mode is selected, and then the Laser Rangefinder is fired when commanded.

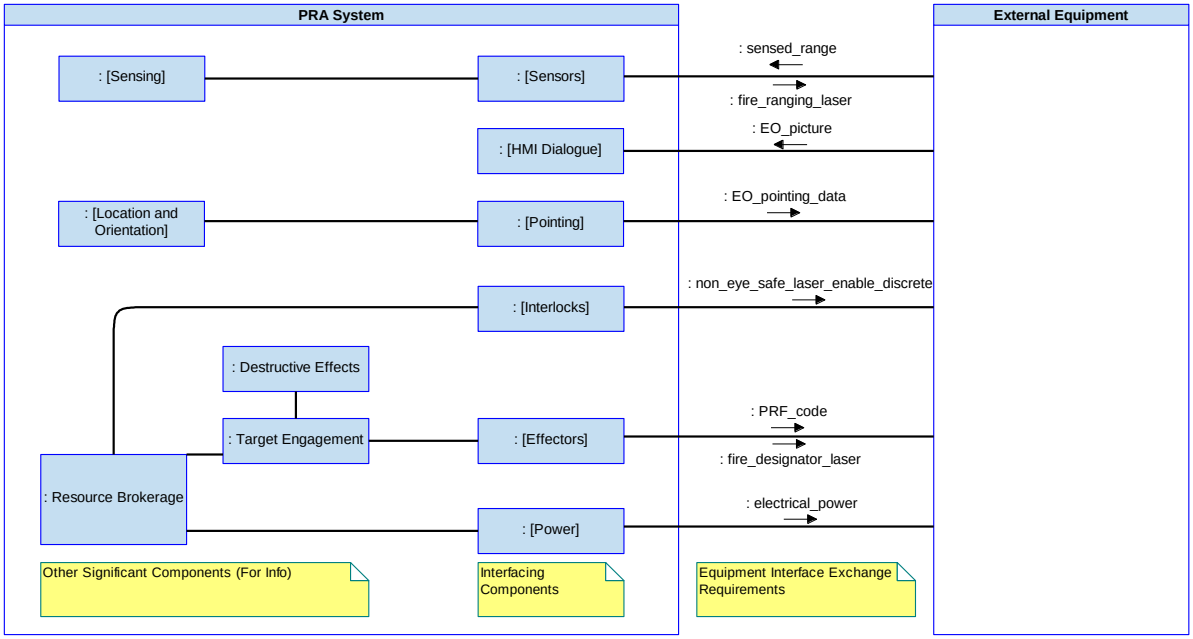

Figure 47: LDP Equipment (Direct Control Example)

In the second example (Figure 48: LDP Equipment (High Level Control Example)) the equipment is set to a search mode and provided with the area of interest and the characteristics of objects of interest (e.g. tanks), relying on the PRA system to provide the location and orientation of the platform. The equipment determines whether it needs to use an IR mode, and negotiates with the PRA system (as per any request for resource, as described in the Resource Management policy) to allow for the increased power demand of this mode. It returns tracks and track characteristics to the PRA system in 'real-world' Earth coordinates. The PRA system can direct it to focus on a particular track and lase it, but the LDP itself determines exactly when to fire its Laser Rangefinder. The HMI Dialogue and Effectors components have been left in this example to show how the equipment could also be controlled at a lower level, including directly by the operator.

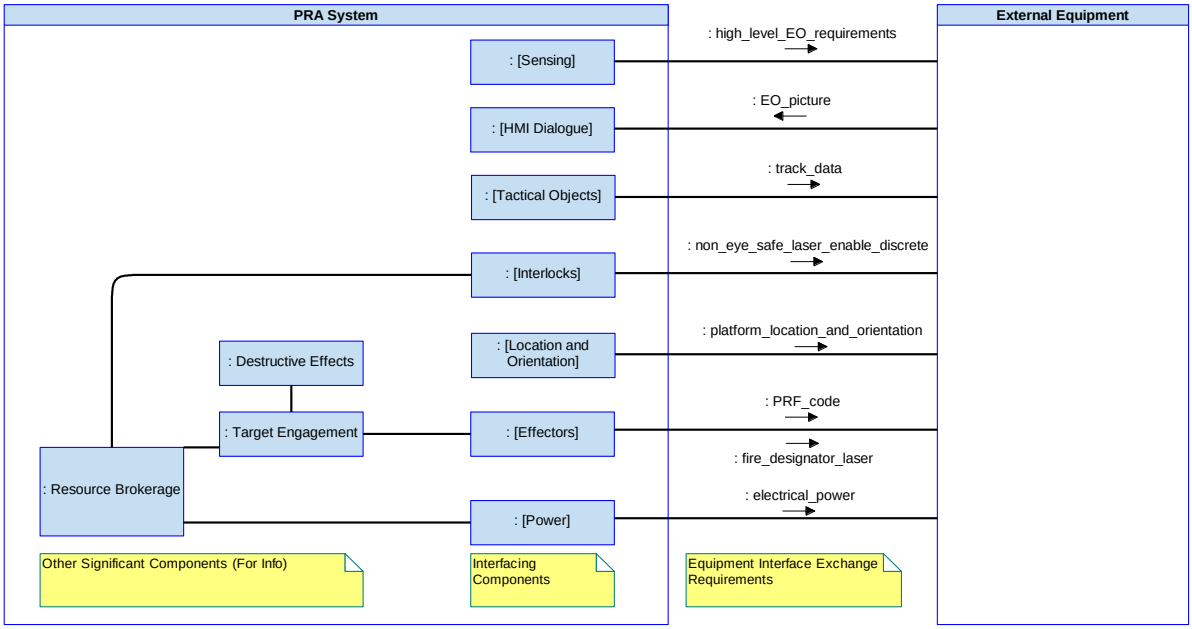

Figure 48: LDP Equipment (High Level Control Example)

A.1.9 Resource Management

A.1.9.1 Pre-Reading and Related Policies

Prerequisite

- Control Architecture

Read in conjunction with

- Interaction with Equipment

A.1.9.2 Introduction and Scope

This policy explains how the principles of resource management have been applied to the development of the PRA. It introduces the Resource Brokerage component and describes how the component can be used to implement those principles.

A.1.9.2.1 What is Resource Management?

In the context of the PRA, a resource is something that is available to use in order to execute an action. It can be a physical commodity that is consumed as a result of performing an action or a service provided to one or more users according to a schedule and cooperative sharing rules.

Resources have some form of cost as they are not infinite and hence must be shared out amongst all legitimate Resource Users.

This makes it desirable to:

- Minimise their use in certain cases (e.g. Fuel usage)
- Restrict their use in cases (e.g. an antenna is a resource that cannot be used by multiple users simultaneously)
- Maximise their benefit while in use (e.g. by considering the placement of a sonobuoy which once in use does not stop until fully exhausted or by prioritising the message transmissions going through a communications device)

Resource management is the process of capturing the resource needs and matching them to resource availability so that an acceptable level of utility is achieved for a bearable cost. This is achieved using the following approaches:

- **Resource Scheduling:** Allocate resource use over time and identify / minimise scheduling conflicts.
- **Demand Traceability:** Maintain a mapping of resource demands to their requirements so that requirements can be changed if resource conflicts cannot be resolved locally.
- **Demand Dependency:** Understand dependencies between the use of different resources to achieve an action.
- **Resource Estimation:** Provide an estimate of the resource cost for an action solution and monitor actual cost against the estimate.

A.1.9.2.2 Why is Resource Management Important?

When resources can be shared or used to perform multiple actions, actively identifying and resolving conflicts provides a flexible and efficient solution to resource usage, making it more likely that mission objectives can be met within an acceptable cost.

- Over-allocation of resources could lead to a very predictable system that is wasteful and inflexible. Sometimes this is the best strategy for high integrity systems.
- Under-allocation of resources could lead to an underperforming system.
- Poor resource estimation could lead to undetected over or under allocation and unpredictable performance.

The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about the role of resource access and allocation in ensuring the continued security of the Exploiting Platform.

A.1.9.2.3 Aim of this Policy

This policy explains how the PRA has been designed to support resource management in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** As the Resource Management policy shows how resource brokering is separated from other aspects of the system, vehicles developed using the PRA can be deployed more flexibly with equipment of different configurations, resource management policies and algorithms applied at different layers or groups, to be decided during operation.
- **Exploitable:** The Resource Management policy explains how the PRA separates out the problem of resource availability, allocation and conflict resolution, contributing to the ability to design and tailor systems that are suited for a wide variety of UK, collaborative and export programmes.
- **Incorporate Future Growth:** Since the problems of resource availability, allocation and conflict resolution are kept apart from other areas of knowledge, components in other parts of the PRA can be deleted or added without requiring changes to the component that deals with brokering resources.
- **Resilient Against Obsolescence:** By showing how resource management is separated out from other concerns, the Resource Management policy supports resilience against obsolescence by allowing different resourcing strategies appropriate to different hardware and technology to be used as necessary, without changing other parts of the system.
- **Scalable:** The Resource Management policy allows different numbers of deployed components to be used in systems that need to manage their resources. This facilitates different levels of complexity.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** The Resource Management policy explains how the PRA separates out the problems of resource availability, allocation and conflict resolution from subject specific knowledge about activity chains. This enables different suppliers to independently concentrate on development and maintenance of the aspects of a system that match their strengths and offerings.

A.1.9.3 Overview

The policy for resource management can be summarised as follows:

- Strategies for resource management have been developed based on pooled and shareable resources.
- Components will implement these strategies by taking on the following roles:

  Action Coordinator: Controls the planning and execution of actions.

  Resource User: Provides resource demands to support execution of actions.

  Resource Broker: Schedules and tracks resource demand allocations and identifies conflicts.

  Resource Provider: Offers up an available resource and provides functions that use it.

A.1.9.4 Strategies for Managing Pooled Resources

Pooled resources are often consumable and represent quantities of a commodity such as fuel, electrical energy or money where the use of the resource will remove an amount from the pool. Some consumable resources may also be replenished. Management of a resource pool consists of maintaining a resource budget by monitoring and controlling the total available quantity of a resource and tracking its usage requests. Management strategies are:

- **Solution Driven:** A proposed solution is generated by the Resource User along with an estimate of the resource cost. It is the responsibility of the Action Coordinator, referring to the Resource Broker for available resource levels, to accept or reject the solution.
- **Cost Driven:** An Action Coordinator can request a proposed solution from the Resource User specifying a permissible amount of resource use. If a solution cannot be proposed within this limitation, the request is rejected. This allows a Resource User to abort the planning if no viable solution is possible.

A.1.9.5 Strategies for Managing Shareable Resources

For shareable resources a Resource User selects a specific resource and negotiates a usage allocation with the Resource Broker. Shareable resources may have constraints on simultaneous use. The uses must be scheduled to avoid conflicting demands in time whilst also conforming to sharing rules. The sharing policy for each resource must define the number, combinations and timing of permitted uses, as well as appropriate transition periods between uses to support any required reconfiguration. Management strategies are:

- **Negotiated Local Resolution:** Many requests can be satisfied by the Resource Broker finding an available slot in a schedule that utilises the flexibility allowed by a demand. Alternatively the Resource User may be able to re-plan locally to change the resource demand: e.g. if the first choice sensor is unavailable, try the second choice and use a longer exposure.
- **High Level Conflict Resolution:** Where a Resource Broker is not able to satisfy a request it can record the nature and timing of the conflicts between demands. The Action Coordinator can then interrogate the Resource Broker to determine how to modify the action requirements and resolve the conflict.
- **Planned Interruption:** If the specific timing of a resource demand cannot be fixed during planning, it would be very inefficient to prevent other Resource Users from using the resource over the period where it may or may not be available. A solution to this is to prearrange an interrupt demand and allow other Resource Users access to the resource with the proviso that access may be interrupted. Note that this strategy is intended to address potential interruptions due to demands which are

known about in advance but whose timing cannot be planned precisely. Unplanned interruptions caused by, for example, a loss of system capability would fall under other policies such as Capability Assessment and Dependency Management.

A.1.9.6 Resource Brokerage Component

The Resource Brokerage component is responsible for tracking the schedule and availability of resources and the status of resource demands. The component may be supported by extensions which define the policy for managing, and the specific characteristics of, a particular category of resource. It addresses the following issues:

- **Total Resource Availability:** The component uses Resource Providers to determine available resources. It can then be interrogated to identify the unallocated resource availability.
- **Scheduling:** The component tracks resource demands such that the schedule and availability determined by the component is considered the 'single source of truth' for permitted resource usage at all times.
- **Conflict Identification:** The component receives resource demands and identifies conflicts with existing resource demands. It provides information on resource demand verses availability which supports the negotiation between resource users and action coordinators to help resolve those conflicts. It can identify which resource demands are in conflict and provide traceability to the actions that requested them and to any higher-level allocations they fall within to the respective Resource Users (or to other actors as required).
- **Dependency:** The component may hold information on relationships between resources, e.g. mutually exclusive resources, or dependencies between resource demands. These relationships and dependencies could occur, for example, where several resources are provided by a single piece of equipment that has inbuilt physical constraints, affecting how its different resources can be used together. Where this is the case, the component will generate a resource demand on the dependent resource, which will be taken into account as part of conflict identification.

The number of independent instances of the component and the choice of which resources are assigned to those instances is an Exploiting Programme decision.

A.1.9.7 Resource Management and the Control Architecture Policy

The Control Architecture policy illustrates how components in the Action Layer make use of components in the Resource Layer to carry out their actions. This behaviour could be carried out without active resource management, especially where resources are dedicated for specific purposes, although that would not always maximise the use of available resources or ensure optimal functional capability.

The roles of the Control Architecture layers in Resource Management are as follows:

- **Task Layer:** The Tasks component acts as the Action Coordinator - defining cost metrics, interrogating the Resource Brokerage component to identify unallocated resources, reviewing resource estimates and choosing between proposed solutions. Where the Resource Brokerage component identifies a conflict, the Tasks component is responsible for resolving the conflict by changing the requirements leading to those resource conflicts. Any requirement to have a resource reserve should be handled by making a reservation in advance thus reducing the quantity available for other uses.
- **Action Layer:** Components in the Action Layer act as Resource Users - responsible for generating proposed solutions. For some resources they can negotiate resource usage with the Resource Brokerage component. For other resources they can estimate the resources required to carry out solutions and provide this to the Tasks component to inform decision making.
- **Resource Layer:** Components in the Resource Layer act as Resource Providers - providing resources to Action components. For a managed resource they will only be asked to provide these resources by an Action component which has a confirmed allocation from the Resource Brokerage component. The Resource Brokerage component is not directly involved in the actual usage. It is also possible for components in the Resource Layer to manage their own resources; e.g. the Sensors component could manage the transmissions of a radar resource based on the actions it is carrying out.

A.1.9.8 Demand Traceability

In order for the Resource Brokerage component to adequately identify and report conflicts, it must maintain traceability between associated resource demands. These fall into two categories:

1. between multiple resource demands that are needed to satisfy a common goal, where either:

   a. The need for one resource demand is a consequence of another resource demand

   b. Independent resource demands contribute to distinctly separate activities that are both needed to satisfy the common goal.

2. between a resource demand and its associated resource allocation.

Objectives are decomposed through the different layers of the PRA. Resource needs are likewise requested and allocated at multiple layers of the Control Architecture and tracked appropriately, with higher level abstractions of resource need (e.g. Action Layer) also being decomposed and represented at lower levels of abstraction (e.g. Resource Layer) by lower level resource demands. Each individual resource demand needs to be traceable back to the requirement that gave rise to the resource demand, and to any higher-level allocation that it falls within (e.g. a fuel reserve allocated for evasive manoeuvres), so that all resource allocations associated with the same overall need, or that fall within the subject matter of a particular component (and hence are in scope for attempts by that component to resolve a resource conflict) can be identified.

A given task will usually rely on multiple actions being carried out across the system, all required for successful completion of the task. The Resource Brokerage component will also need to maintain a trace of the origin of a demand in order to conduct conflict identification and resolution.

If conflicts occur, such that all resource demands cannot be met, the links maintained by the Resource Brokerage component allow components at a higher level of abstraction to understand the nature of the conflict in their own terms to inform their replanning for affected actions.

Replanning is carried out by Resource Users and, where appropriate, Action Coordinators, facilitated by the information provided by Resource Brokerage. During replanning, allocations are provisional while conflicts are resolved. Once all conflicts have been resolved then Resource Users commit to their plans and resource allocations are made firm in the Resource Brokerage component.

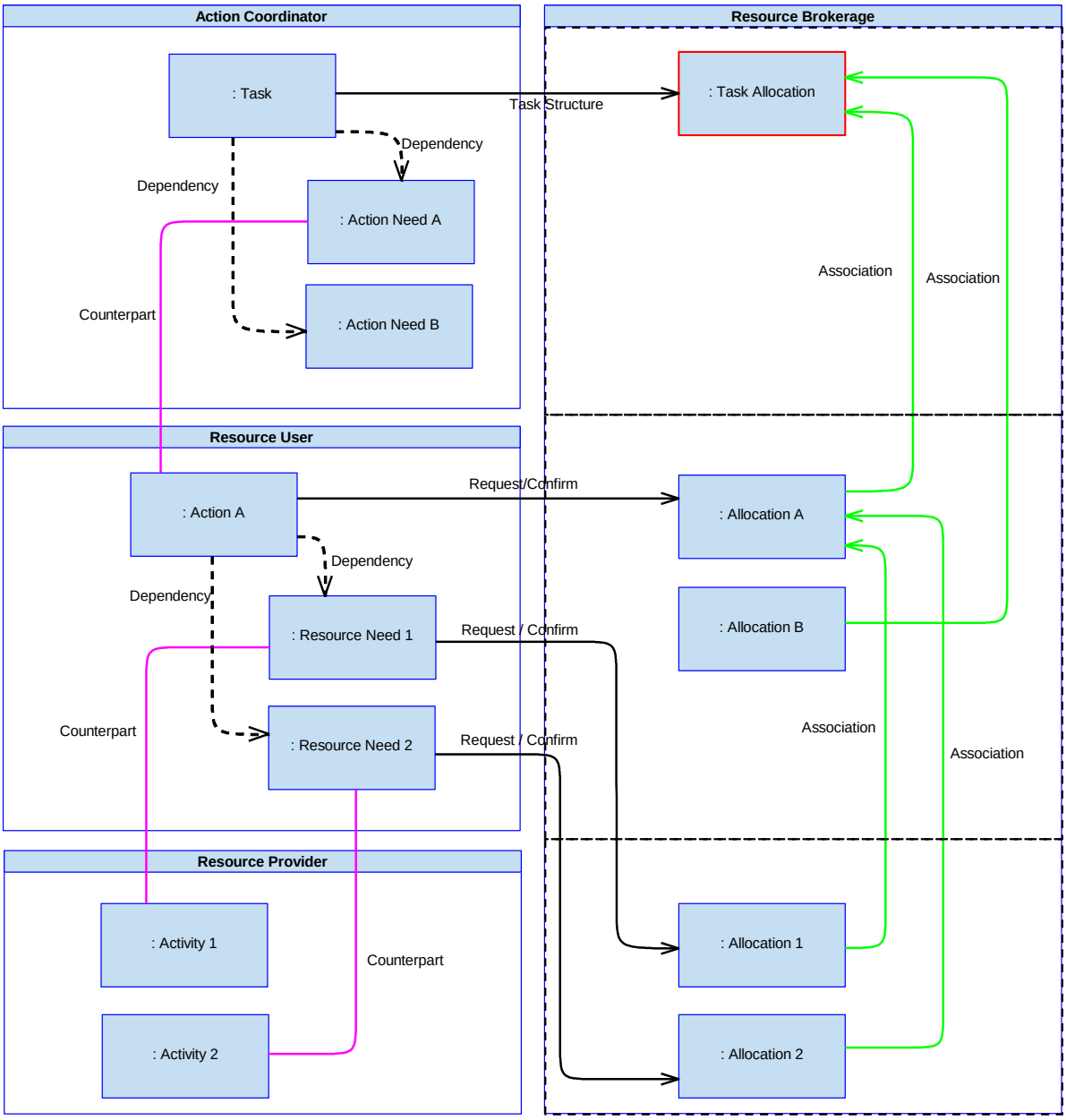

Figure 49: Demand Traceability

For example, in Figure 49: Demand Traceability, a Task is generated that has two Action Needs, A and B, which are created and counterparted to newly created Actions in the Resource User component. An allocation has been reserved for tasks of this nature (e.g. a fuel reserve allocated for evasive manoeuvres), and when resource allocations are requested for these actions they are associated against the overall reserve allocation. Action B itself is not shown in the Resource User frame of the figure to avoid clutter, but Allocation B represents the resources it would need. If Action A and Action B are not both required, but instead belong to alternative solutions for the same task, then they are not associated with one another and only one of the actions would result in an allocation from the overall Task-level allocation within the Resource Brokerage component as part of a final planned solution.

Actions in turn require Resource Needs to be allocated, which are created and counterparted to newly created Activities in the Resource Provider component. In the event that conflicts need to be resolved, a Resource Brokerage component can trace the Resource User (e.g. the action, or agent requesting the allocation) through the request for Allocation A and, if appropriate, the Action Coordinator through the task context information associated with Allocation A.

In the case where Resource Needs 1 and 2 belong to alternative candidate solutions for Action A, then they are not both required and Allocations 1 and 2 would only be provisional until one solution is selected.

A.1.9.9 Resource Management Examples

A.1.9.9.1 Consumable Resource Management

In this example, Figure 50: Consumable Resource Management Example, there is a need to plan a new route within a remaining fuel resource. A significant portion of the fuel budget has already been allocated.

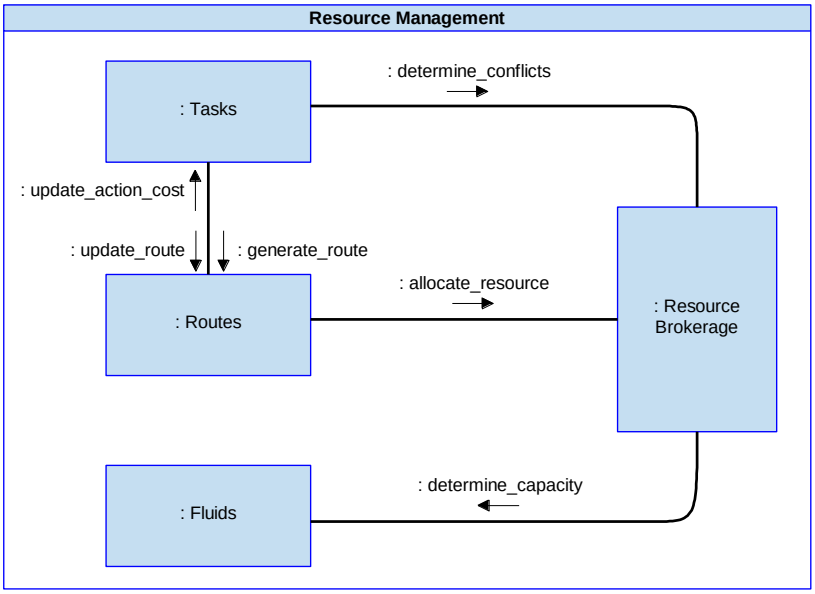

Figure 50: Consumable Resource Management Example

A.1.9.9.1.1 Consumable Resource - Solution Driven

The Resource Brokerage component is aware of the total available quantity of fuel as measured by the Fluids component.

The Tasks component starts to plan the task, requesting that the Routes component find the fastest solution. Routes collaborates with other components to find a solution and estimate the fuel cost for it. The Routes component asks the Resource Brokerage component to determine the unreserved quantity of fuel and allocate the quantity required for its solution. The Resource Brokerage component determines that the quantity requested exceeds the quantity remaining and the Routes component is thus unable to fulfil its route requirement within the task's parameters. Routes alerts the Tasks component and Tasks modifies the requirements of the routing task to prioritise fuel efficiency.

The Routes component re-plans the solution to reduce the fuel use and requests the Resource Brokerage component to allocate the new, lower quantity of fuel. Resource Brokerage confirms to Routes that the required quantity has been allocated. The Routes component updates the Tasks component with the new route cost estimate and the plan is confirmed.

A.1.9.9.1.2 Consumable Resource - Cost Driven

The Resource Brokerage component is aware of the total available quantity of fuel as measured by the Fluids component.

The Tasks component starts to plan another task. Tasks considers its budget and requests the Routes component to find a solution within a maximum fuel cost. The Routes component plans a route and estimates the fuel cost but is unable to find a solution within the cost constraint.

The Tasks component reassesses its budget and increases the maximum fuel cost constraint. The Routes component plans suitable activities, estimates the fuel cost and finds a solution within the constraint. The Tasks component accepts the new solution's cost estimate and confirms the plan. The Routes component requests that the Resource Brokerage component determine the unreserved quantity of fuel and allocate the quantity required for its solution. The Resource Brokerage component confirms to the Routes component that the required quantity has been allocated.

A.1.9.9.2 Shareable Resource Management

In this example there is a need to plan a new sensing task to be performed within a specified time window. This type of task uses a shareable resource - a sensor. The primary events are illustrated in Figure 51: Shareable Resource Management Example. The Tasks component requires the Sensing component to plan a solution. The Sensing component has determined the available capability and attempts to allocate the use of the Sensor Equipment managed by the Sensors component within the parameters of the request from the Tasks component. The Resource Brokerage component makes the allocation if there is no conflict, tracking the overall demands on the sensor. In a situation where a conflict exists with other allocations made on the Sensors component which cannot be resolved by the Sensing component within the parameters placed on it by the Tasks component, then Resource Brokerage rejects the allocation, and the Sensing component alerts Tasks to the conflict. The Tasks component obtains conflict information from the Resource Brokerage component to enable a resolution to be attempted by updating the parameters of the requested solution on the Sensing component, the Tasks component having taken into account all other tasks in the system.

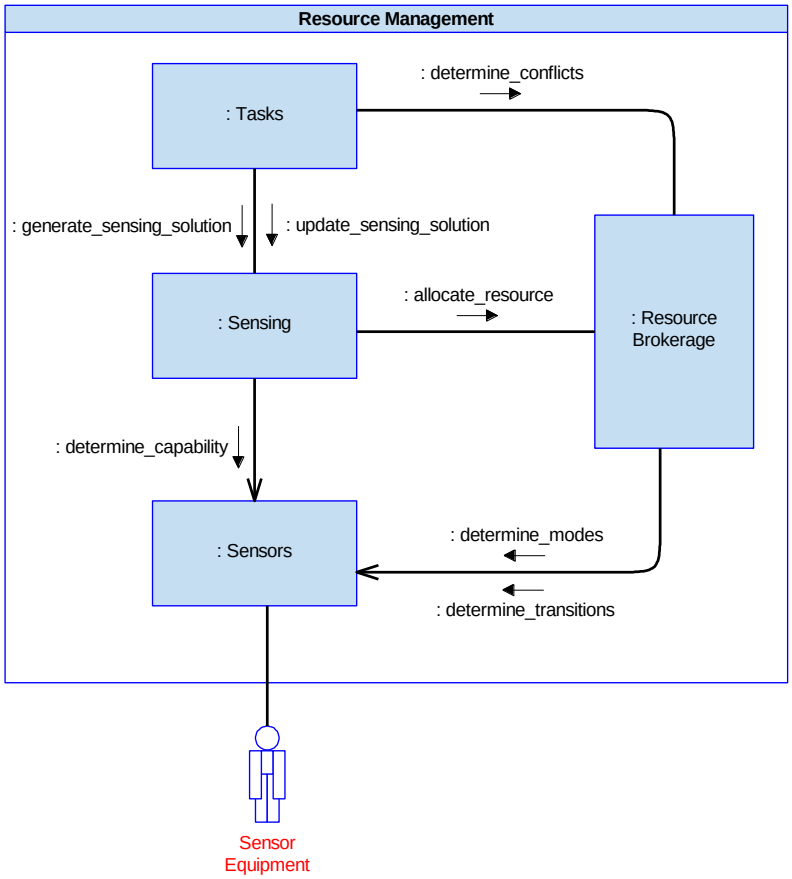

Figure 51: Shareable Resource Management Example

Note that in this example, the sensor equipment is likely to require additional interfaces with components that are not shown in Figure 51: Shareable Resource Management Example. Only the interfaces related to shareable resource management are illustrated.

A.1.9.9.2.1 Shareable resource - without conflict

The Tasks component starts to plan a task. The Sensing component identifies the need to use a sensor for a specific time period. The Resource Brokerage component determines that there is no conflict for the sensor at that time. The Sensing component reserves the use of the sensor with the Resource Brokerage component and informs the Tasks component a solution has been established.

The Tasks component confirms to the Sensing component to action the plan.

A.1.9.9.2.2 Shareable resource - with conflict resolution

The Tasks component starts to plan another task. The Sensing component identifies the need to use a sensor for a specific time period. The Resource Brokerage component determines that an exclusive request has already been made for that sensor at that time (in red in Figure 52: Shareable Resource - Conflict Identified).

Figure 52: Shareable Resource - Conflict Identified

The Sensing component tries to resolve the conflict locally and asks the Resource Brokerage component to determine an alternative time period. The Sensing component is unable to accommodate the proposed time within its constraints so cannot provide a solution to the task. As illustrated in Figure 53: Shareable Resource - Local Resolution Rejected, there is not a sufficient time window available to accommodate both the new requested sensor allocation (in blue) within the required time parameters (blue dashed area) without overlapping the pre-existing sensor allocation. The Sensing component is thus unable to plan a solution within the task's required parameters and alerts the Tasks component to the conflict.

Figure 53: Shareable Resource - Local Resolution Rejected

The Tasks component asks the Resource Brokerage component for details of the conflict associated with this task. The Resource Brokerage component reports the conflict details including relevant contextual information to enable identification of the existing conflicting task(s). This allows the Tasks component to assess the conflicting tasks and modify one or more of the tasks' requirements, enabling the Sensing component to re-plan a solution.

Figure 54: Shareable Resource - Solution Replanned

The alternative allocation initially requested by the Sensing component (illustrated in Figure 54: Shareable Resource - Solution Replanned) still conflicts, but an accommodation can be found by the Resource Brokerage component within the new parameters of the required sensing solution (this may require an iterative process of requests from the Sensing component to the Resource Brokerage component until an acceptable allocation is found). The Sensing component accepts the new allocation (illustrated in Figure 55: Shareable Resource - Solution Modified) and updates its solution accordingly.

Figure 55: Shareable Resource - Solution Modified

The Sensing component informs the Tasks component a solution has been found and the Tasks component confirms to the Sensing component to action the planned solution, resulting in a final sensor resource allocation state illustrated by Figure 56: Shareable Resource - Conflict Resolved.

Figure 56: Shareable Resource - Conflict Resolved

A.1.9.9.3 Interrupted Resource Management

In this example there is a need to plan a new sensing task sometime within a specified time window. This task will need exclusive use of a resource if or when the need arises. Any interrupted users of the same resource need to be accommodated. The primary events triggered to achieve this are illustrated in Figure 57: Interrupted Resource Management Example, using an indicative set of components.

Note that in this example, the sensor equipment is likely to require additional interfaces with components that are not shown in Figure 57: Interrupted Resource Management Example. Only the interfaces related to interrupted resource management are illustrated. In addition, for clarity only one existing allocation to use the sensor resource is interrupted (i.e. that allocated to the Target Engagement component). In actuality the interruption by the Sensing component could potentially affect multiple existing allocations to use the sensor resource, nevertheless the interactions with any existing action component users of the resource will remain the same as those described here for the Target Engagement component.

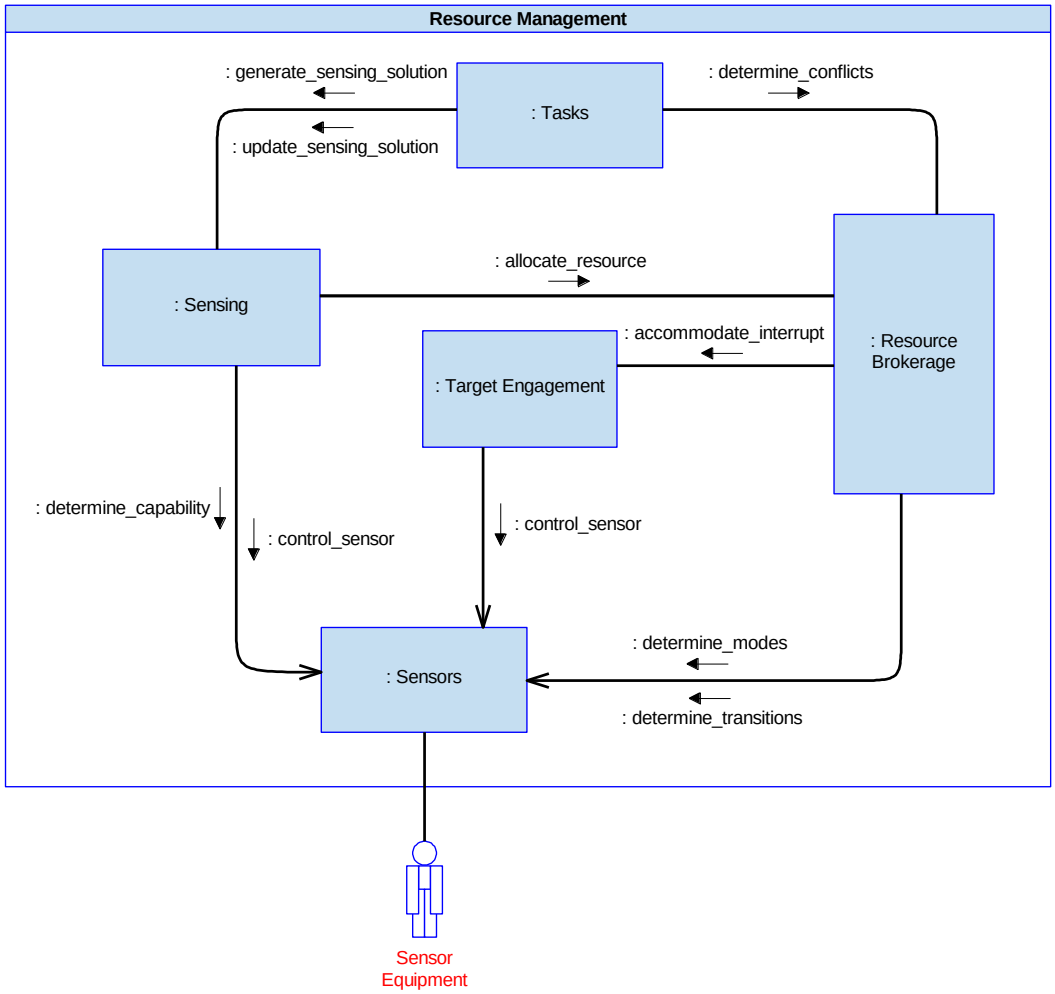

Figure 57: Interrupted Resource Management Example

A.1.9.9.3.1 Example Overview

The Sensing component has determined the available capability and attempts to allocate a time window for the use of a sensor. This time window represents an overall period, within which the Sensing component requires to use the sensor resource for a smaller period that is not yet precisely known. The Resource Brokerage component determines that this time window could conflict with an existing request made by the Target Engagement component to use the same sensor resource within the same time window.

The Resource Brokerage component determines whether the Target Engagement component can accommodate the interruption based on the agreed usage profile already allocated for use of the sensor, and if so makes the allocation with the Sensing component. At the appropriate time, the Sensing component will notify the Resource Brokerage component of its actual use of the sensor, so that the Resource Brokerage component can inform the Target Engagement component of the interruption. Both the Target Engagement and Sensing components will then control their use of the sensor, within the allocated window, so as to avoid conflict with each other, as informed via the Resource Brokerage component. This path is described in more detail under Potential Outcome #1, in the next section.

In the situation where a planned interruption of the Target Engagement component's existing allocated use of the sensor by the Sensing component cannot be accommodated within tasked parameters, then the Resource Brokerage component rejects the allocation, and the Sensing component alerts the Tasks component as to the conflict. The Tasks component obtains conflict information from the Resource Brokerage component to enable a resolution to be attempted by updating the parameters of the requested solution(s), the Tasks component having taken into account conflicting tasks in the system. This path is described in more detail under Potential Outcome #2, in the next section.

A.1.9.9.3.2 Example Detail with Alternative Outcomes

The Target Engagement component has already agreed an interruptible allocation with the Resource Brokerage component to use a resource managed by the Sensors component for a given time. This was the result of the Tasks component requesting an action solution from the Target Engagement component (not shown in Figure 57: Interrupted Resource Management Example, but similar to that illustrated in Figure 51: Shareable Resource Management Example). The nature and degree of allowable attempted interruptions may be specified by the Target Engagement component, depending on Exploiting Programme requirements.

The Sensing component starts to plan a new activity. A sensing action anticipates the need to use a sensor intermittently over a specified time period. The Resource Brokerage component determines that the allocation request range potentially overlaps with the existing allocation (depicted in Figure 58: Potential Interruption Identified), but in accordance with the modes of use published by the sensor, the new allocation can possibly be accommodated as an interrupt. The Resource Brokerage component notifies the Target Engagement component by requesting the interrupt be accommodated.

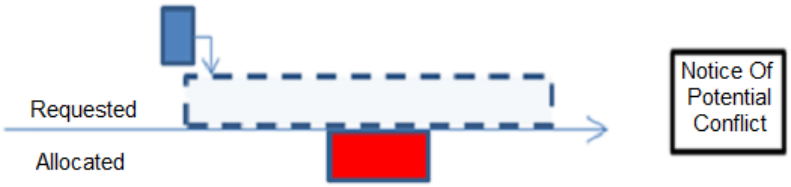

Figure 58: Potential Interruption Identified

Potential Outcome #1

The Target Engagement component determines that it can still complete its actions even with the potential interrupt from the Sensing component. The Resource Brokerage component relays a positive response to the Sensing component accordingly and the new allocation for the Sensing component is reserved with the Resource Brokerage component (depicted in Figure 59: Potential Interruption Accepted). The Sensing component informs the Tasks component that a solution has been established. The Tasks component confirms to the Sensing component to action the planned solution.

Figure 59: Potential Interruption Accepted

After time has passed, during the time window that the Sensing component anticipated using the sensor resource, the Sensing component notifies the Resource Brokerage component that the sensor usage is now required.

Figure 60: Actual Interruption Notified

The Resource Brokerage component determines that the sensor usage by the Sensing component will interrupt the Target Engagement component's current use of the sensor resource. The Resource Brokerage component notifies the Target Engagement component of the necessary interruption, illustrated in Figure 60: Actual Interruption Notified. The Target Engagement component stops using the sensor and the Sensing component starts using the sensor. Once complete, the Sensing component notifies the Resource Brokerage component that the interruption has completed. The Resource Brokerage component notifies the Target Engagement component to continue. The Target Engagement component starts using the sensor again. The interrupted Target Engagement component's use of the sensor (depicted in red) by the Sensing component's use of the sensor (blue) is illustrated by Figure 61: Resource Use Interrupted.

Figure 61: Resource Use Interrupted

Potential Outcome #2

The Target Engagement component determines that its solution would no longer be viable, as it would not be able to complete its own actions and accommodate the potential interruption required by the Resource Brokerage component for the Sensing component. The Resource Brokerage component therefore rejects the resource allocation request from the Sensing component, which in turn is unable to plan a solution within the task's required parameters and therefore alerts the Tasks component as to the conflict. The Tasks component asks the Resource Brokerage component to determine details of the conflict associated with this task. The Resource Brokerage component reports the conflict details including the rejected potential interruption of the Target Engagement component. The Tasks component modifies the requirements of the Target Engagement component's task and/or the Sensing component's task, to enable one or both of these Resource User action components to re-plan their respective solution(s). Consequently, in the example shown by Figure 62: Potential Interruption Replanned, the Resource Brokerage component is now able to allocate a re-planned sensor resource usage for the Sensing component that will not conflict with the Target Engagement component. The Sensing component modifies its solution accordingly and informs the Tasks component a solution has been found. There is no longer any contention for the sensor and no affected users will be found or notified at task execution time.

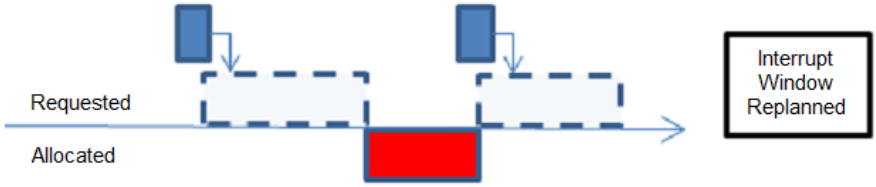

Figure 62: Potential Interruption Replanned

A.1.10 Operational Support

A.1.10.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Control Architecture
- Constraint Management
- Autonomy
- Health Management
- Recording and Logging
- Human-Machine Interface
- Data Exchange

A.1.10.2 Introduction and Scope

This policy explains how the PRA can be used for purposes beyond the immediate operation of a mission. It describes a range of such uses, and explains the ways the PRA could be used and the components that would support such uses.

A.1.10.2.1 What is Operational Support?

"Operational support" includes any activities that support the mission but are not directly involved in carrying it out. These include activities which are:

- In preparation for a mission (planning, rehearsal and briefing);
- Following a mission (debriefing, post-mission data handling and replay);
- To maintain or improve capability (maintenance and support).

Other similar uses are not excluded. Training, for example, is out of scope of this policy, but many of the same solutions would apply.

Support for these activities could be achieved in three ways:

1. By the use of a dedicated, separate system (for example a mission planning or maintenance system) exchanging information with the PYRAMID compliant system;
2. By developing the dedicated, separate system using the PRA;
3. By using the PYRAMID compliant system to support the mission as well as to carry it out.

Figure 63: Separate Planning System shows an example of a dedicated, separate planning system, as in option 1 above. A mission planner would use this to define a Mission Plan that would be uploaded to the system. The planning system could have been developed as a PYRAMID compliant system, as in option 2.

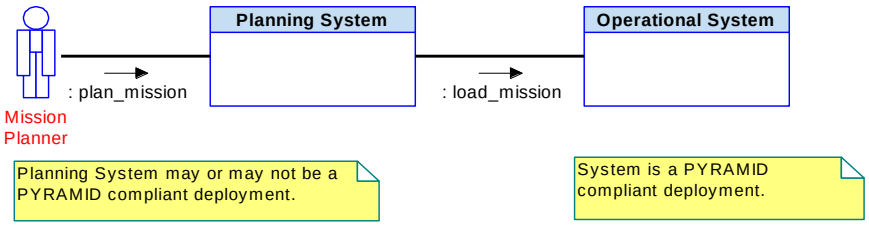

Figure 63: Separate Planning System

Figure 64: PYRAMID Compliant Deployment used for Planning shows how a mission planner could use the operational system directly for mission planning. Mission planning requirements would be placed on the operational system to achieve this. Such a solution would exploit similarities between planning and operational uses.

Other support uses (briefing and debriefing, maintenance, etc.) would have similar options.

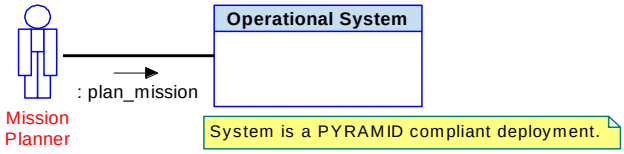

Figure 64: PYRAMID Compliant Deployment used for Planning

A.1.10.2.2 Why is Operational Support Important?

A PYRAMID compliant deployment will not exist in isolation. It will require information to be input prior to the start of a mission. Similarly, following the completion of a mission, information will need to be extracted to support functions such as debrief and maintenance and the collection of gathered intelligence.

This information exchange performed outside the execution of a mission is considered operational support. This policy shows how a PYRAMID compliant deployment and other systems can be combined in a structured manner.

Without a well-defined interface to these other systems, the PYRAMID compliant deployment may not be able to reliably perform the functions required of it.

A.1.10.2.3 Aim of this Policy

This policy explains how the PRA has been designed to enable operational support in a manner that supports achievement of the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** The policy illustrates many possible configurations of PRA deployments.
- **Exploitable:** The policy illustrates how deployments of the PRA can be integrated with each other and with PYRAMID non-compliant systems.
- **Utility Across A Range Of Missions:** The policy shows how deployments of the PRA can be set up to carry out specific missions, and how organisational structures and processes can be accommodated.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** The policy shows how deployments of the PRA can be used for maintenance and support.

A.1.10.3 Overview

Since all PRA components are responsible for the whole mission lifecycle including planning, execution and post-mission analysis, no special components are required for operational support. Some components have particular relevance to operational support, though, including:

- Authorisation and Operational Rules and Limits, which allow the mission plan to constrain the operational system (see the Constraint Management policy);
- Information Brokerage and Data Distribution, which are used to coordinate the exchange of mission information including reports (see the Data Exchange policy);
- Health Assessment and Anomaly Detection, which support maintenance and (by enabling capability assessment) mission planning (see the Health Management policy);
- HMI and communication components, which provide the link with external systems and operators (see the Human-Machine Interface policy).

The policy shows how PRA components can be used to provide a simulation harness for mission rehearsal or playback.

A.1.10.4 Mission Planning and Data Loads

Most deployments of the PRA will rely on information that is changeable and specific for different uses and environments. Such information is contained in the Mission Plan, which comes with a large amount of supporting information needed by the operational system to carry out the plan.

The mission planning process will produce the Mission Plan for a flight of one or more air vehicles that will achieve a particular tasking order. Mission Planners will typically make use of a dedicated Mission Support System to create the Mission Plan, and this is the approach highlighted in this policy.

The information products output from this process then need to be transferred as loadable data into the operational system, to shape how it will operate in a given mission. The PRA facilitates this by allowing for components that are data-driveable, so that the architecture can be configurable, exploitable, and useable across a range of missions.

The provision of mission planning may be supported by or include simulation functionality (see Simulation & Playback Support for further details).

A.1.10.4.1 Mission Planning Support

A dedicated Mission Support System, used by Mission Planners, is external to the operational system (as illustrated in Figure 63: Separate Planning System) and could either be a PYRAMID compliant system itself (as shown in the Mission Data Load IV), or a PYRAMID non-compliant procured solution. In either case, an external Mission Support System would generate the mission planning information products as loadable data, in a form that is tailored for the operational system, to be transferred through a communication channel or some other

transfer mechanism. The Mission Data Load IV describes an example of how the Mission Planner could be assisted to ensure a complete set of Mission Plan data is prepared for loading using a PYRAMID compliant Mission Support System.

A.1.10.4.2 Operational Autonomy and Mission Planning Levels

The PRA supports the definition of desired autonomous behaviour for a system by explicitly representing task authorisation and authorisation limits. In the operational support environment an operator will be seeking to define the desired behaviour or behaviour bounds. By limiting the system's authority to act and constraining the options available to the system, an operator can create a bounded problem space within which a system can operate autonomously without exhibiting undesirable behaviours.

Components will be granted authorisation within specific authorisation limits which control when an action can be performed. More general constraints are handled by the Operational Rules and Limits component which imposes limits on other parts of the system in accordance with the rules given to it. These components are intended to be data-driven to allow them to be configured at run time with the desired limits and rules for an individual mission.

Depending on the level of autonomy an operational system is capable of, and programmed for by the Mission Planner, the Mission Plan may be in the form of objectives, tasks or actions, or a mixture. This is illustrated in Figure 11: Mission Planning Interactions. For higher levels of operational system autonomy, the Mission Planner will interact with higher PRA layers of components. The layers of the PRA are described in the Control Architecture policy.

For example, for higher levels of system autonomy, where vehicles are given tasks to perform, mission planning would develop plans and supporting data to populate the Task Layer components of the deployed system, such that it has the necessary task tactics populated and any required data-driveable content in the Tasks parent component(s). It would be expected that all PRA layers below the highest layer designated for autonomy would also require interaction from the mission planner via the Mission Support System. Regardless of the level of autonomy, the Mission Support System would always be expected to interact with components in the Resource Layer and some service components also, in order to develop the necessary resource specific vehicle fit data and operational level detail that ensures vehicle equipment and services can provide the capability and control needed to support the Mission Plan.

For further details on autonomy see the Autonomy policy.

A.1.10.4.3 Types of Mission Data and Update Frequency

Data to be loaded falls into two broad categories: engineering data and mission data.

Engineering data relates to internal processing employed by a component that is specific to a vehicle platform, or domain of operation. For example, the information required for determining vehicle performance will differ between fast jets, rotary wing, and sub-surface vehicles. Engineering data specific to a vehicle, vehicle type or domain of operation will not change frequently. Vehicle fit will also affect engineering data as different resources and equipment are swapped in and out between missions or even sorties during a mission. The resources and equipment may be ballistic or advanced weapons; EM sensing technology options; or complementary countermeasure resources. This type of data load is carried out whenever vehicle equipment is changed.

Mission data relates to different theatres of operation, requiring, among other things, specific data to represent the geographic and tactical environments in which an exploitation will be deployed. For example, different missions will involve exposure to a range of different threats depending on the territory in which a

mission is performed, as well as the purpose and characteristics of the mission itself. These threats and other entities are local to the mission territory, or the forces operating in the territory. Mapping and terrain data will also be specific to the mission. Mapping and terrain data is not likely to need re-loading for as long as a mission continues to be based in and around the same territorial boundaries and the data remains current, therefore this data is only likely to need re-loading when the mission footprint changes or updates are published. Some more detailed and specific mission tactical elements may need changing on a sortie-by-sortie basis in highly dynamic and evolving scenarios.

A.1.10.4.4 Data Load Interactions

Data loads are produced from mission planning activities that could be performed during a dedicated planning related mission phase or during operational system use, such as to support a re-planning activity. The size and related content of a data load therefore could vary from information to support an entire mission to an incremental update concerned with a single subject matter area, and anywhere in between, based upon the reason for the data load (including synchronisation of data) and information available at the time.

Data Load Manager

A Data Load Manager is responsible for the management and provision of any functionality to support data movement between instantiations of components or systems. This role could be an external actor (e.g. a human, a computer/ autonomous machine or a combination of the pair) or may be implemented using components such as Information Brokerage, dependent upon the implementation. The coordination of data loads is not performed by one particular component, and how this is achieved is Exploiting Programme specific. An example of this using the Information Brokerage component, with interactions from external actors managed via HMI (however the HMI aspects are not shown), is illustrated in Figure 1354: Mission Data Load IV.

During planning, its role is to instruct applicable components to create the associated data load to support mission requirements.

During operational use of a system, its role is to inform components of the availability of new data, and to coordinate the mission data loading process.

The mechanism of data loading is related to the implemented mechanism of data transfer (e.g. comms or via data storage) and will therefore be Exploiting Programme specific.

Mission Data Load Creation

The range of interactions required leading to the creation of a successful data load are illustrated in Figure 1354: Mission Data Load IV. In this specific example the Data Load Manager is represented by the Information Brokerage component and managed via HMI (although the HMI aspects are not shown). This includes the capture of data retention requirements by the Mission Planner, which is in line with the Recording and Logging policy. The Data Load Manager will determine and ensure that all information required for the chosen mission is drawn together to form the loadable package, which will then be transferred to the operational system once authorised.

Mission Data Load Loading

Successful transfer and data integrity during the loading process will be determined by the transfer channel. Once data loading has completed for all components identified across a platform requiring mission data, the receiving components as part of their fundamental set of responsibilities will check the data for insufficiencies and inconsistencies. Errors in component operation will occur if it is acting upon an incomplete or incorrect data set. It is the responsibility of the Data Load Manager to inform a component of a situation where it should suspend loading for a period, or to switch to a new data set.

The PRA supports the use of a variety of data loading mechanisms, including physical communication channels and physical movement of data storage media. The method of transfer will be Exploiting Programme infrastructure specific, however from a PRA perspective the interactions required for collation and movement of data do not differ between deployment design choices. Therefore the channel is considered independent of the transportation method.

Figure 65: Components supporting Mission Data Load Creation and Loading illustrates how component instances can be used in the planning and operational usage of the system, for the creation and collation of the Source Mission Load instructed by the Planning Data Load Manager, and subsequent coordination by the Operational Data Load Manager of the loading and distribution of the Target Mission Load.

Note: The transfer mechanism is not shown as this is transparent to the process being represented.

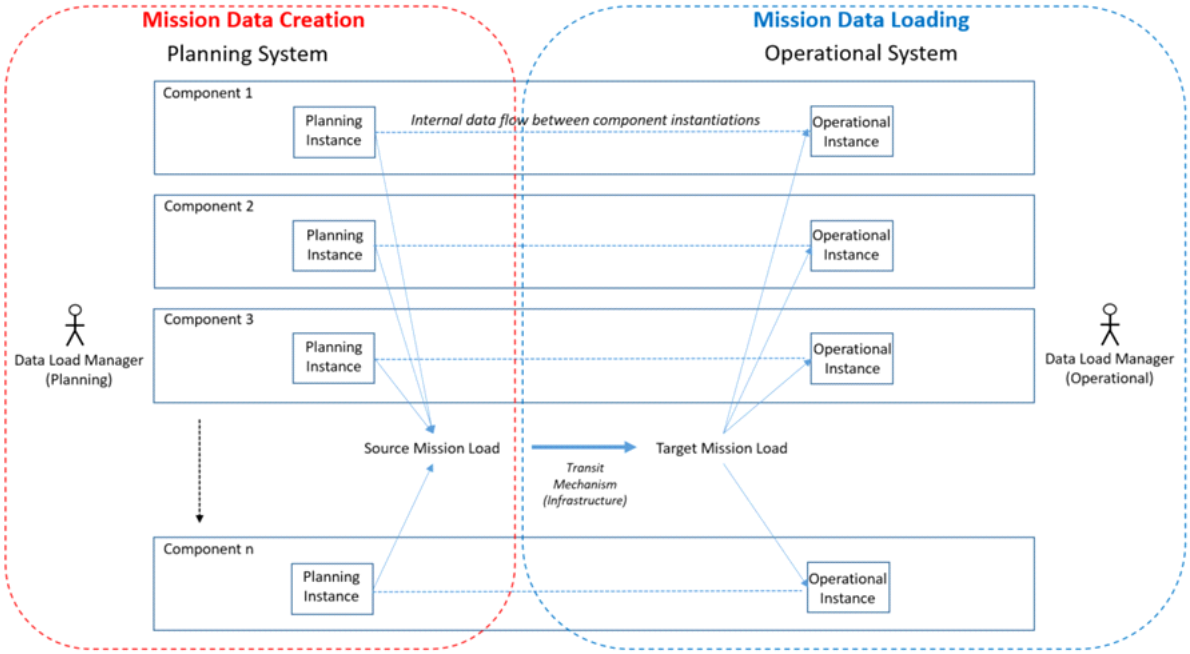

Figure 65: Components supporting Mission Data Load Creation and Loading

A.1.10.5 Mission Rehearsal and Briefing

Mission rehearsal and briefing are both part of the process of preparing for a mission. This section describes how they can be achieved using components as part of a mission planning deployment.

Mission rehearsal involves stepping through a mission plan to familiarise authorised operators with its content. Mission rehearsal may also have a role in validating the plan for authorisation. Since no real actions take place during rehearsal, it will require a simulation harness (see the Simulation & Playback Support section) which replaces those components that interact with the real world with rehearsal versions. Most components will react to events as they are informed about them and will behave normally, unaware that they are in a simulation.

If mission rehearsal takes place within a separate mission planning system, the presence of simulation versions of components is unproblematic. The use of the operational system to provide simulation for mission rehearsal will only be a safe option if the implementation can clearly distinguish (for example by moding) when real and rehearsal versions of component are to be used.

The preparation of a mission report for briefing or other purposes is coordinated by the Information Brokerage component and collated by the Data Distribution component. The approach shown in the Generation of Handover Briefing IV will apply here.

A.1.10.6 Mission Debriefing and Post Mission Data Handling

Mission debriefing may occur during a mission, in support of crew handovers and mission awareness, or post mission for debrief and mission analysis. To support this an operational system may continually provide mission and intelligence reports directly following tasks, or as required, to support situational awareness.

Post Mission Data Handling (PMDH) is the provision of mission generated data (including reports), in response to a query. While the originator of the query could be considered to be an external party, the post mission data analysis facility may be provided by a PYRAMID deployment (for example to support intelligence analysis), in which case the query and data provision would be from and to an ‘internal operator’.

Within the context of these two cases, a PYRAMID deployment must be capable of supporting:

- Varying levels of both information criticality and timeliness (e.g. in support of mission, post-mission and historical information exploitation).
- Both internal and external users.
- The provision of mission reports automatically, e.g. the generation of a mission report on completion of a task.
- The provision of mission information required in response to requests.

The provision of mission debriefing and PMDH can be facilitated by simulation functionality (see section Simulation & Playback Support for further details).

A.1.10.6.1 Levels of Information Criticality and Timeliness

The mechanism for the retention of data, including taking into account its required longevity, priority and criticality is detailed by the Recording and Logging policy.

For the purposes of debriefing and PMDH, information that is required to be available at a later point needs to be defined, its requirements understood and planned accordingly.

Data required for debriefing may be subject to a standard setting (for the operating unit) for criticality and timeliness, as may other mission generated data. Other operational data may be subject to a lower level of retention requirement, e.g. ‘retain if possible’.

A.1.10.6.2 Support for Internal and External Users

While the generation of data required by non-operational users is the same as for operational users, the delivery of such data to the users differs between those that are internal to the operational system and those that are external.

For example a PMDH user (whether internal or external to the system) may request data from the operational system, and receive an appropriately authorised response, in the same way that an operator could request data to gain situation awareness prior to handover. As explained in the Generation of Reports IV, the Exploiting Platform's communications component suite would handle the generation and dissemination of reports and information to external users. In contrast, the Exploiting Platform's HMI component suite would manage internal user interactions. External users will have to be authorised to be able to interface with the system, and their method of interaction would be via their own HMI.

For further detail on use of the HMI components see the Human-Machine Interface policy. For further detail on use of the communications component suite see the Use of Communications policy.

A.1.10.6.3 Provision of Planned and Requested Mission Data

An operator, external user or Mission Support System may be automatically provided with generated data and reports as they are produced, or at pre-defined times as specified by mission requirements.

The Information Brokerage component has the reporting requirements, including what data, when it is required, and how it is to be delivered. These reporting requirements will be predefined, for example by the mission planner or as a standard set, e.g. for a specific mission type.

The Information Brokerage component will use these reporting requirements to marshal data and to inform an operator, via the HMI, what is available, allowing end user selection or automatic report generation as required. The reports will be constructed either in HMI or Data Distribution by collating the information from components that hold the source information required to populate the report.

Reports or mission data may be requested by a user, via either HMI or communications components as appropriate. An example of the request for, and delivery of, such data can be seen in the Request for Information IV.

Mission replay can be considered an example use of retained mission data. All data required for replay must be retained for reuse according to the reporting requirements held by the Information Brokerage component. The retrieval of such data must be achieved in a manner that allows for its use in the required order.

A.1.10.7 Maintenance and Support

Maintenance and support involves a wide range of activities to maintain the capability of a fleet, which are unlikely to be managed by a deployment of the PRA. However, there will be significant communication between the maintenance and support system and the operational system so that the maintenance system will know the condition of the hardware being managed by the operational system, and the operational system can be updated with the post maintenance state of the assets.

It should not be assumed that the maintenance and support organisation will communicate with the operational system via a separate maintenance system. In small or lightweight organisations (including forward operating locations), management of maintenance and support may be manual, and the maintainer may communicate directly with the operational system.

The Health Management policy shows how the PRA can be used to identify anomalies in the deployed system and analyse the causes of such anomalies and (where they are caused by health conditions) to assess the consequences of a health problem. For the operational system, the focus of this activity is on the vehicle's capability and the consequences for the mission.

The support organisation has different needs, however. In particular, it is concerned with the identification of maintenance needs, such as specific Line Replaceable Units (LRU) that need repair or are reaching life or usage limits, and the state of consumables. It is still concerned with capability, but more at the fleet level where the capability of individual vehicles can be altered by carrying out repairs or changing their role fit.

Figure 66: Maintenance and Support shows the operational system communicating with a separate maintenance and support system. In both cases only a few components are shown: the operational system would have many more components concerned with planning and executing a mission, while the maintenance and support system would be responsible for all maintenance activities such as scheduling and logistics. In the diagram, both are shown as deployments of the PRA, but for the maintenance and support system this might not be true.

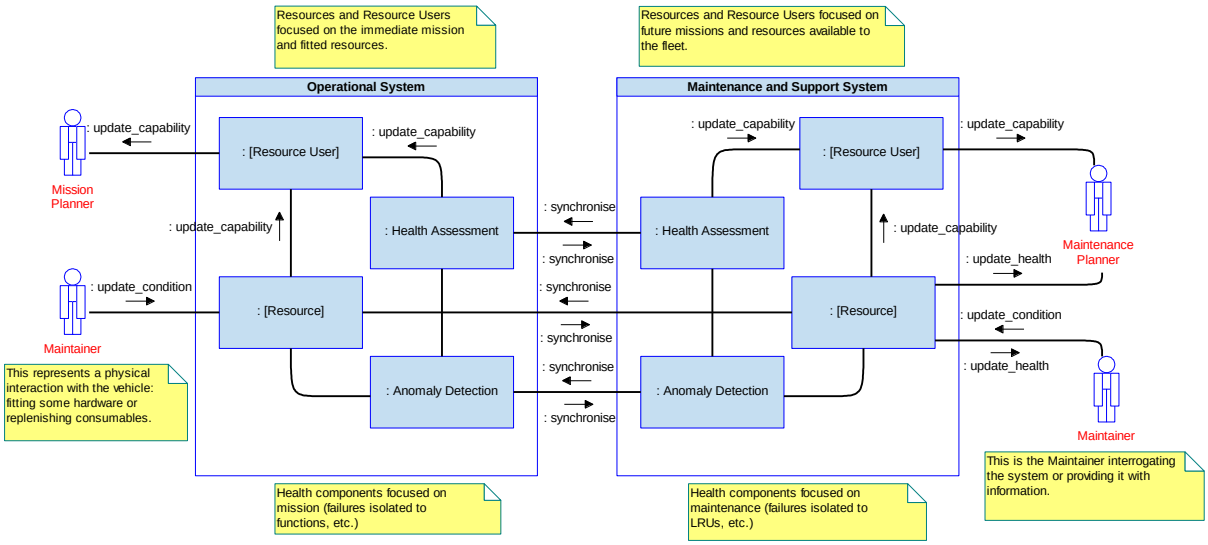

Figure 66: Maintenance and Support

In Figure 66: Maintenance and Support, [Resource] represents any component in the Resource Layer which is responsible for managing hardware. [Resource User] stands in for any component in the upper layers of the control architecture which depends on those resources to provide its capability. Health Assessment and Anomaly Detection components have the roles described in the Health Management policy.

Although superficially similar, the details of the two systems will be different because of their differences in scope and granularity. The operational system is focused on the immediate mission and fitted resources: the primary concern of health components is to determine the system's capability so that the rest of the system can make informed decisions about the mission. The maintenance and support system is focused on future missions and resources that are available to the fleet. Health components need to identify failed or degraded LRUs, and those that have reached the end of their useful life (measured either by time or usage) so that maintenance can be planned. These two goals are consistent (both are descriptions of the same underlying health condition) but may require a different level of detail.

The Mission Planner interrogates the operational system to discover its capability. Other interactions of the Mission Planner are discussed in the Data Load Interactions section and are not shown here. The Maintenance Planner interacts with the maintenance and support system to plan a maintenance schedule which will maximise fleet availability as well as meeting requirements from mission planning. The Maintainer will replace LRUs, replenish consumables, and install role fit equipment as determined by the maintenance plan. The

Maintainer may contribute to fault diagnosis by providing measurements or running tests. Where the Maintainer makes a physical change to the vehicle this will be reflected in both systems. For example, if the Maintainer adds fuel to the vehicle, this will be seen as a maintenance action in the maintenance and support system and also as a direct effect on the vehicle that will detect, via its measurement sensors, that its fuel contents have changed.

A.1.10.8 Mission Analytics and Performance Assessment

Mission analytics and the capture of data for performance analysis allows an operational system to determine its effectiveness and the effectiveness of the platform(s) it controls. Analysis may occur either during a mission or as a Mission Support System activity after a mission. However, the capability to carry out analysis on a platform during a mission allows changes to be made to planned and in-progress activities to incorporate the results of that analysis. It would also minimise the quantity of potentially classified raw data which needs to be retained and transmitted for analysis to a Mission Support System. The identification and retention of data needed is done on a component by component basis, taking into account its required longevity, priority and criticality, and is detailed by the Recording and Logging policy. This means that performance analysis could take place constantly at any level of the Control Architecture and can be captured at whatever level is required by a Mission Support System for a given purpose.

A.1.10.9 Authorisation

Operational support organisations have procedures for authorisation which can be complex and need to take into account factors such as the following:

- What is being authorised.
- When the authorisation is being provided.
- Whether supporting data has been authorised via its own authorisation chain; for example, Techniques, Tactics & Procedures (TTP's) are developed (and authorised) by warfare centres remote from operational units, when authorising a mission plan at an operational unit the authoriser needs to be sure that any TTP's the mission plan relies upon or implements are themselves authorised.
- The type of mission being authorised; some missions have specific objectives with detailed plans to achieve those objectives in which case authorisation is sought after. However a lot of missions are more loosely defined; the operator may be seeking to maximise flexibility in responding to whatever is encountered during the mission.

The Authorisation component understands the mechanisms for obtaining authorisations for proposed courses of action including where to obtain authorisation from (see Action Authorisation IV), it thus supports the need to obtain different authorisations from different sources. It can also maintain a record of the authorisation associated with any data provided to the system. The relationship between authorisations will also be managed by the Authorisation component.

For information on granting authorisation see the Authorisation section in the Autonomy policy.

After a mission has been completed there will likely be a significant amount of data generated; whether and to whom to release this data will be something that needs to be carefully controlled via some form of authorisation process. Preparation of data for release is a combination of Information Brokerage which identifies the data, and Data Distribution which collates it. Whether data is authorised for release is a separate issue which will be handled by the Authorisation component.

A.1.10.10 Simulation & Playback Support

A system may need to contain a simulation facility to support pre-mission related activities such as mission rehearsal, briefing, planning and training, or a playback facility to support post-mission related activities such as data analysis and handling, de-briefing and training. These facilities would allow an operator to progress through a planned or completed mission via the manipulation of parameters such as time or location, and then observe the resultant system behaviour. Such facilities may be realised by simulation-specific instances of components.

A pre-mission facility would interact with components either functionally fulfilling their designed operational role, or a limited simulated role. A post-mission facility would interact with data recorded during a specific mission.

A.1.10.10.1How does the PRA Support this Provision?

The components identified to provide simulated behaviour may be provided as part of a simulation or test harness. The purpose of this harness is to define a boundary between operational components and components which operate differently when providing a simulation of the real world. Components in the harness, in addition to performing their normal functions, may also receive external stimulation to carry out a simulated function. These components could report an enactment while not actually having enacted that step.

A.1.10.10.1.1 Simulation Provision

Figure 67: Figure: Example Simulation Harness Provision illustrates how a pre-mission rehearsal capability could be provided using a set of components working together providing both real and simulated functionality. Most of the components will be operating with no knowledge or awareness that they may be interacting with a component providing simulated behaviour.

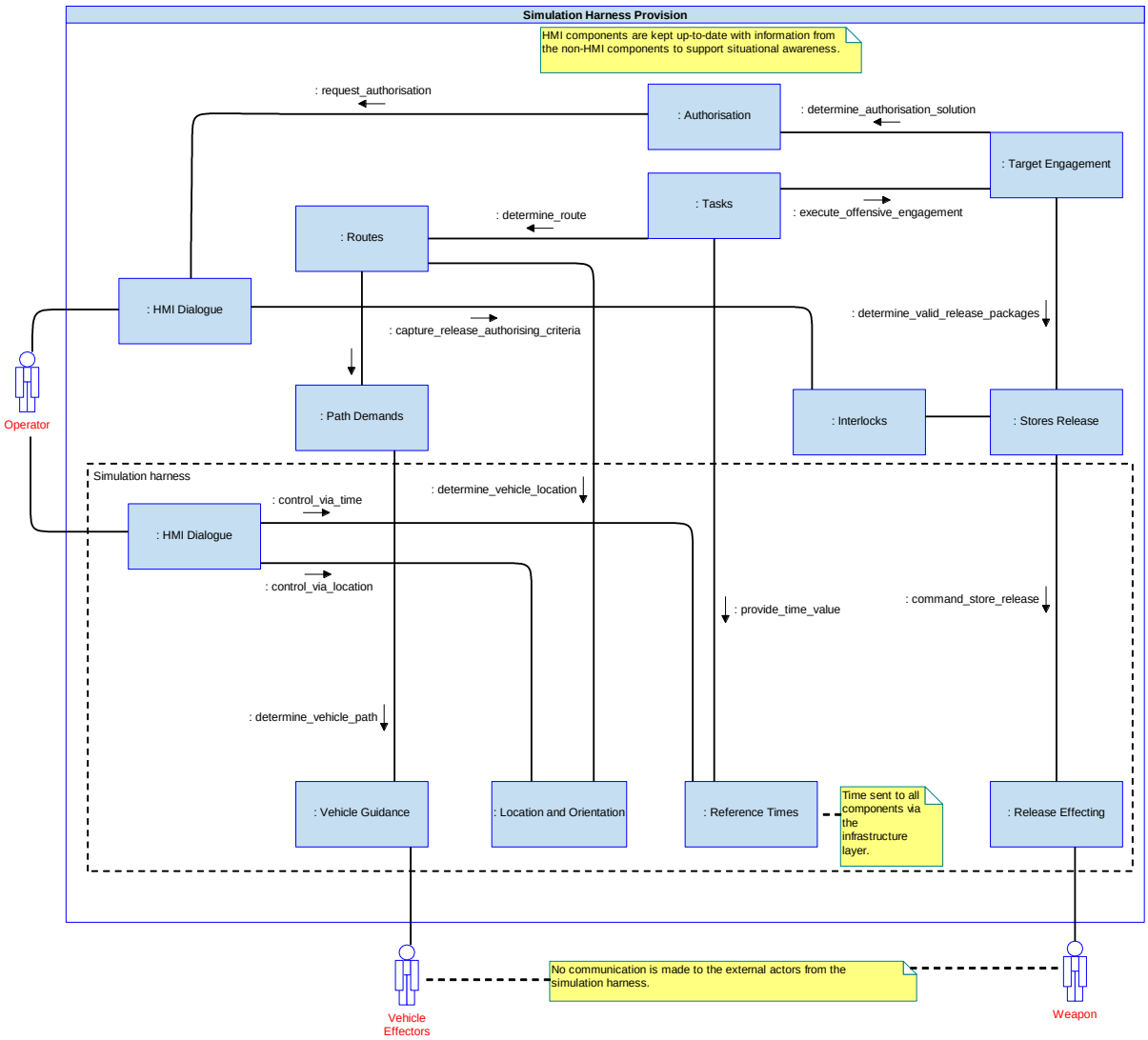

Figure 67: Figure: Example Simulation Harness Provision

In this example, the instances of:

- The Location and Orientation component would provide values associated with locations, and the Reference Times component time could provide deviations from the time source, to artificially stimulate the simulation HMI (examples include jumping to a specific time or location in the planned mission, or playing through a planned mission at double speed).
- The Vehicle Guidance and Release Effecting components report back to the Path Demands and Stores Release components respectively, as well as the operational HMI, that they have enacted any actions asked of them without actually fulfilling them, e.g. report starting of an engine or releasing of a weapon without actually commanding the engine or weapon release.

All other components operate as per their normal operational behaviour.

A.1.10.10.1.2 Playback Provision

To support post mission based capabilities, the simulation HMI could provide a playback facility which would allow operational system recorded data to be retrieved from the system and presented back to the operator via the operational HMI or dedicated simulation HMI.

A.1.10.10.2Deployment Considerations

Depending upon the specific requirements associated with a PYRAMID deployment:

(i) The boundaries of the simulation harness (thus the components encompassed within the harness) would change dependent upon the scope and level of simulation required to be provided.

(ii) Its simulation provision can be provided by defining the components as either:

- **multi-mode** (a component's internal behaviour is moded based upon whether the deployment is being operated in real-time or simulated), or
- **single-mode** (specific separate instances of components being created to perform either real-time or simulated behaviour) operation.

A.1.11 Storage

A.1.11.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Recording and Logging
- Security Approach

A.1.11.2 Introduction and Scope

This policy describes the mechanisms provided by the PRA to control storage of data, and the interactions of components with storage facilities provided by an Exploiting Platform. The policy will address how a deployment may facilitate the retention of the data, and the support that the PRA provides in achieving that.

Whilst the general principles of this policy should apply to the method of storage and the storage media used by the deployment software and other programmable content present within a deployment, the policy has been written from the point of view of the data used for, and arising from, a mission. An Exploiting Platform's software storage, its security requirements and means of protection are subject to the requirements of the Exploiting Programme as defined by the appropriate design authority. Guidance for the implementation of secure storage is available in the Dstl Security Guidance for PYRAMID Exploiters, Ref. [60].

A.1.11.2.1 What is Storage?

Within this policy, "storage" always refers to the action or method of storing data for future use, and not the component (which is always referred to as "the Storage component") or the storage media (which is always referred to as "storage media"). For the purposes of this policy only, "data" can be considered any form of content that can be stored on storage media available to the deployment.

Considerations relating to storage can be separated into the following areas:

- Storage to media - writing the data to the appropriate storage media.
- Security domain partitioning - separation of different types of data items in order to satisfy the security objectives for confidentiality, integrity and availability, etc.
- Data retention - the mechanism by which important data is kept for future use or reference.
- Deletion and sanitisation - the disposal of data from storage media.

Note:

- The determination of the need for data retention by a component is covered within the Recording and Logging policy.
- It is assumed that the system will work on the premise that the data will be passed by value rather than by reference.

A.1.11.2.2 Why is Storage Important?

It is expected that any Exploiting Platform will require and may generate large amounts of data during its operation. Some Exploiting Platforms may have capability constraints for storage such as limited storage capacity and others may have mandated requirements for storing certain items of data during operation, e.g. as required to meet specific legislative needs.

Examples of the need for storage include:

- Retention of data required for mission use.
- Conforming to legislation, e.g. Crash Survivable Memory Units (CSMU). Note that General Data Protection Regulation (GDPR) requirements may constrain what can be stored.
- Retention of data for non-repudiation and auditing purposes.
- Retention of data for analysis of aircraft life and usage data, e.g. structural stress data.
- Retention of mission-generated data, e.g. for mission replay.
- Retention of data for fault investigation, e.g. health and usage monitoring.
- Retention of anomalous behaviour, event and message traffic data for cyber threat analysis.
- Local cache of transient data, i.e. data that is generated within a component as part of its operation (and exists in a local cache).

These requirements may change or evolve during the life of the Exploiting Platform, and it is therefore essential that the PRA is flexible enough to allow the Exploiter to choose the methods of storage, the management of storage media, and the data held.

The different types of data and their use will determine the retention needs, the storage methods, and storage media that will be used.

A.1.11.2.3 Aims of this Policy

This policy explains how the PRA has been designed to support storage in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Flight Certifiable:** This policy provides a framework which enables conformance to legislation, e.g. to enable storage of specific aircraft performance parameters on a CSMU.
- **Incorporate Future Growth:** This policy provides a framework within which storage media can be added or removed to cater for changing requirements for the retention of the data.
- **Scalable:** This policy provides a framework which allows any number of components to store data on any number of storage media available.
- **Security Accreditable:** This policy provides a framework by which the needs for data security can be satisfied.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** This policy provides a framework by which health and threat data, etc. can be retained for later analysis.

A.1.11.3 Overview

This policy can be summarised as follows:

- Components will require access to storage media, with this access being agnostic of the storage infrastructure.
- The Storage component manages the storage media, including the need for separation, encryption, sanitisation, etc., but excluding the actual transfer of data between components and storage media.
- Components are responsible for accessing, storing and retrieving their own data. The Storage component may apply constraints on the amount of data that can be stored.

A.1.11.4 Storage Considerations

A.1.11.4.1 Uses of Storage

A component may need to store data for either processing or retention.

Processing storage covers where a component uses storage in order to facilitate processing and execution of its responsibilities. For processing storage, the component knows why it needs it and for how long. For example any component may have an allocated space in storage media (a memory cache) for discharging their processing requirements.

Retention storage covers where a component holds data for future use, possibly outside of the specific component instantiation's understanding, e.g. for report dissemination or post mission analysis. For retention storage the requirement may be defined externally to the component instantiation. The determination of the retention strategy for a component's data is covered within the Recording and Logging policy.

A.1.11.4.2 Storage Media Types

An Exploiting Programme will choose the types of storage media it needs. A component may have access to several types of storage media to manage processing and for retention storage.

A.1.11.5 Component Interaction with Storage Infrastructure

A.1.11.5.1 Component Storage Transparency

A component's access to data on storage media, for either processing or retention needs, is considered internal to the component. From an operational and a component definition's perspective, storage should be considered part of the component. We can think of its use as being transparent, in that the result of the action of storage can be seen from a systems perspective, but the act is hidden from the component specification. This is illustrated in Figure 68: The Transparent Act of Storage.

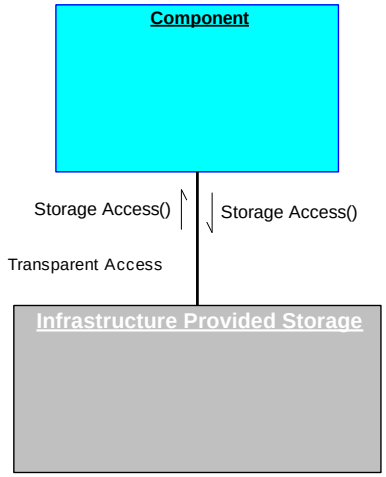

Figure 68: The Transparent Act of Storage

The Exploiting Programme may require stored data to be encrypted, see the Encrypted Storage Media section.

A.1.11.5.2 Retrieval of Own Data by a Component

All data that is generated or used by a component is the responsibility of that component. The relationship between data of different components (e.g. counterparts) is detailed within the Component Connections policy.

If a component has data located on storage media, the component will access that data for use directly. The access to such data is the responsibility of the infrastructure. Likewise it is the responsibility of the Exploiter to ensure that the component is robust and manages the impact of changes to any dataset, e.g. the loss of availability of data held on storage media.

In such cases, the understanding of what data is available and usable by a component is internal to the component, and should be considered as appropriate by each Exploiting Programme.

It is the responsibility of the infrastructure to transfer the data from one storage media to another, if required to do so. Provision of this would be specific to the needs of the deployment, and the provided infrastructure.

The use of storage is not a specific need of each component to fulfil its role, but is a design and infrastructure choice to facilitate its role. Therefore, no component responsibilities or component services are required, or have been included for each component in order for it to use storage.

Note: A component's data may need to be loaded before it can be retrieved. Management of data loading is detailed in the Operational Support policy.

A.1.11.5.3 Access to a Component's Data by Other Components

Components cannot access storage media data belonging to other components directly. If data needs to be communicated from one component to another, this should be provided by a service of the component that owns the data.

A.1.11.6 Management of Storage

While components are not 'aware' of the storage infrastructure that they may be using, it may be necessary for a deployed system to be able to manage the capacity and use of the storage media provided by the infrastructure. Management of the storage infrastructure is the role of the Storage component.

While all components access the storage media transparently, the Storage component is responsible for interacting with the storage infrastructure (i.e. the storage media resource) in order to manage the available storage media capability. This includes ensuring that actions are taken if storage media capability is limited.

The Storage component will maintain a view of the usage of storage media, as illustrated in Figure 69: Storage Component View of Storage Use, and provoke the disposal of items in line with the data storage policy of the exploitation. Identifying the data storage requirements and policy are Exploiting Programme considerations and are therefore the responsibility of that programme.

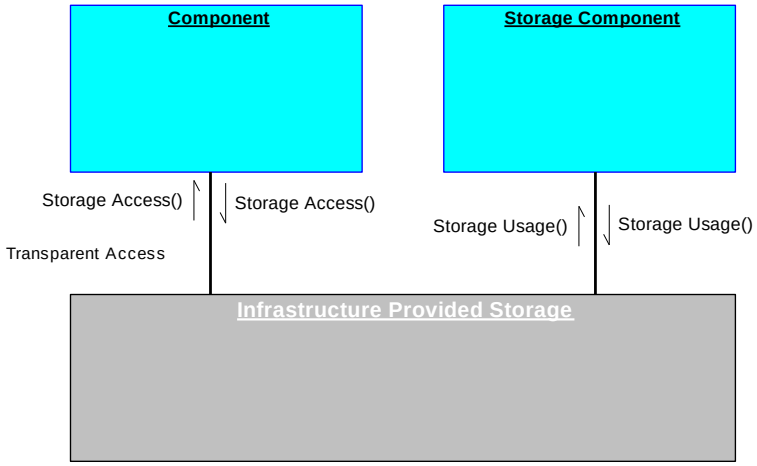

Figure 69: Storage Component View of Storage Use

Corrective action may be via deletion or sanitisation of data, or another storage infrastructure-specific option (e.g. movement, resizing or reallocation). The Storage component holds the knowledge of the data storage policies and will select the most appropriate action based on a policy and several factors such as:

- Overall storage capacity.
- Current remaining storage capacity.
- Storage media usage (e.g. ownership of data).

The Storage component manages the application of security to storage media, if required, e.g. encryption of drives (see the Cryptographic Management IV for an example of how encryption may be applied). The application of a component's data retention policy is covered in the Recording and Logging policy.

A.1.11.6.1 Movement of Data between Storage Media

Where data is required to be moved between storage media devices, this is done by the infrastructure. This may be at the direction of the Storage component or the deployment infrastructure.

Where data is required to be duplicated between two instances of the same component, this is a component internal operation and is not covered by this policy.

Where duplication of bulk data is required between system nodes (e.g. two processing nodes in discrete locations), the act of reading from and writing the data to storage media will be provided by the infrastructure. The transfer of such data is not covered here.

A.1.11.7 Changes Affecting Storage and Data Retention Capabilities

A.1.11.7.1 Types of Change

Certain circumstances may result in the need to remove data from the storage media or to render all data held upon it permanently inaccessible. How this is done in practice is a consideration for the Exploiting Programme to determine, however this policy considers the following options:

- Deletion - removal of data by a component.
- Sanitisation - the process of deliberately, permanently and irreversibly removing or destroying programmable content or data to make it unrecoverable.
- Addition or removal - changes to the availability of storage media.

A.1.11.7.2 Deletion

A deletion is where access to a specific data item or a set of data is no longer retained. Because deletion relates to specific data, this is controlled by the specific component owning that data. Each component manages and performs deletions of its own data, whilst respecting any requirements for security logging, etc. (see the Recording and Logging policy for further details).

This policy considers a deletion to be driven by a change in the retention policy for that data. For example, if the Storage component requires a reduction of storage use as the available space limits are nearly reached, it could set a constraint on other components to instigate a change of the appropriate retention policies, meaning that those components subsequently delete the data no longer required from the storage media. This is shown in Figure 70: Storage Use Triggered Change of Retention Policy. Further information on retention policies can be found in the Recording and Logging policy.

How the components manage the application of the retention policy is internal to the component, whilst how the ownership of data is managed is an Exploiting Programme decision (e.g. through use of metadata or a Table of Contents). Neither are covered in the PRA.

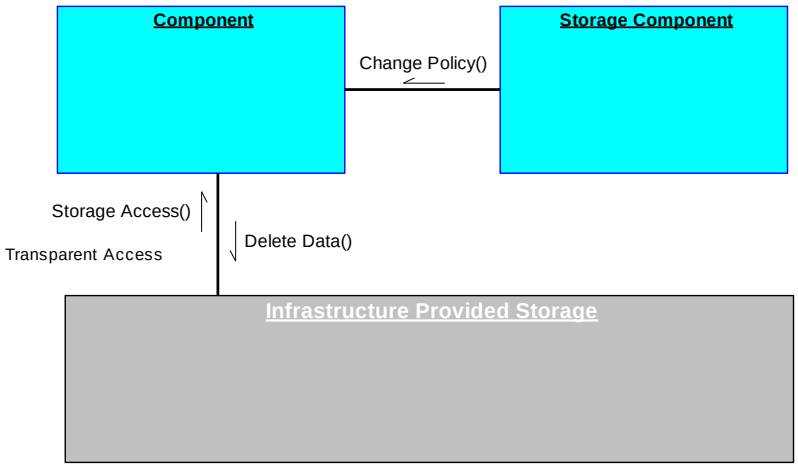

Figure 70: Storage Use Triggered Change of Retention Policy

A.1.11.7.3 Sanitisation

Sanitisation is the process of deliberately, permanently and irreversibly removing or destroying the data held on storage media to make it unrecoverable. Some forms of sanitisation will allow you to reuse the storage media, whilst others are destructive in nature and render the storage media unusable. The scope of sanitisation may depend on the method used, but could be a specific data item, a partition, a single storage media device or some or all storage media used by the system. Within that scope, sanitisation is a blanket ruling, and no exclusions can be applied. For this reason, data should be carefully segregated and there may be specific requirements for ongoing retention of safety or security-related logs.

Sanitisation of data by the infrastructure may be initiated by the Storage component. The trigger for this to occur may be from operator request, or by another component determining a need, e.g. on entering an emergency situation. To ensure speed of destruction, it is recommended there is no component interaction or negotiation that may prevent sanitisation.

The Storage component directly interfaces with the infrastructure to direct sanitisation to occur, as shown in Figure 71: Storage Initiation of Sanitisation. It is the responsibility of the infrastructure to provide a mechanism to sanitise all of the data from the designated storage media.

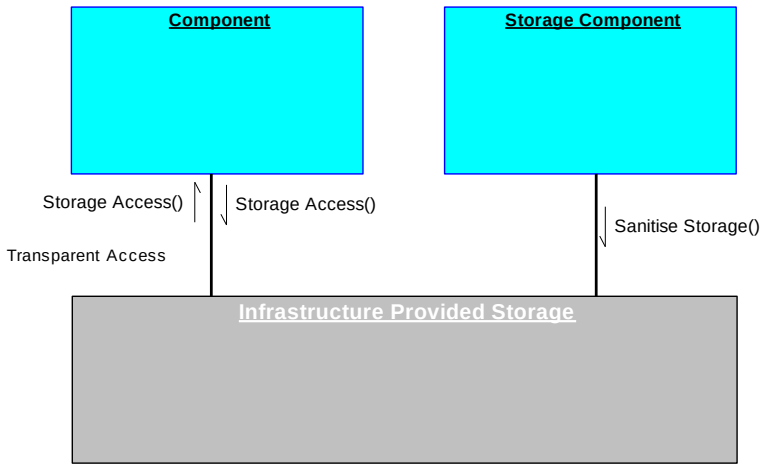

Figure 71: Storage Initiation of Sanitisation

When the Storage component is instructed to initiate sanitisation of the storage media, the Storage component may inform components so that their retention policies can be updated accordingly. This may be required to ensure that no further data is stored, or to manage any potential change of capability that a component may have from a change of data.

A.1.11.7.4 Change in Storage Media

When there is a change to the storage media available to an Exploiting Platform, for example when a new store is connected or an existing one is removed, the Storage component may be informed of this change by the infrastructure (see Figure 69: Storage Component View of Storage Use).

A change in storage media may lead to a change of capability due to data available to the components.

Where storage media is moved, for example node-to-node transfer of data on a USB drive, the impact of the change of availability of data is the responsibility of the components. Examples of data movement, e.g. data loads, can be found in the Operational Support policy.

A.1.11.8 Secure Storage

This section will discuss how the differing methods of storage support for security are provided by the PRA.

In order to achieve accreditation an Exploiting Programme will need to implement physical separation of storage media for at least:

- Original equipment manufacturer/integrator storage (e.g. for application software, in order to protect intellectual property).
- The storing of cryptographic material.
- Bulk storage media for access by the user.

This policy only explicitly covers the latter. Further information on the other types of use can be found in the Dstl Security Guidance for PYRAMID Exploiters, Ref. [60].

The Security Approach policy provides information on security domains and security features that are fundamental to the concept of secure storage.

A.1.11.8.1 Security Domain Partitioning

Security domain partitioning is the separation of data for reasons such as (but not limited to) differing security classifications in order to satisfy the security objectives for confidentiality, integrity and availability, etc.

This is often achieved using the following methods:

- **Physical Separation of Storage**, such as by implementing individual storage for instances of security classification, for example having two separate storage media devices, in order to securely separate data within different security domains.
- **Logical or Virtual Separation of Storage**, e.g. preventing access to parts of the storage media by access controls or encryption.

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 media and to manage gateways, etc.

A.1.11.8.2 Separation of Storage

It is expected that an Exploiting Platform will include a number of storage media devices. This will be determined by the needs of the Exploiting Programme. For example, there may be a requirement to isolate particular data, or to store certain data on a specific storage media device (such as a CSMU) in order to support flight certification requirements.

Separation of storage for security domain partitioning will support an Exploiting Programme's security accreditation. Separation of storage may also be used to support the design for safety, supportability and scalability.

A.1.11.8.3 Encrypted Storage Media

Encryption (and decryption) may be used with storage media, for example to provide access controls. Like the storage mechanism, this is applied at the infrastructure boundary, and thus is likewise transparent to a component, as shown in Figure 72: The Transparent Act of Encrypted Storage.

The management of such cryptography to provide encrypted storage is seen in the Cryptographic Management IV.

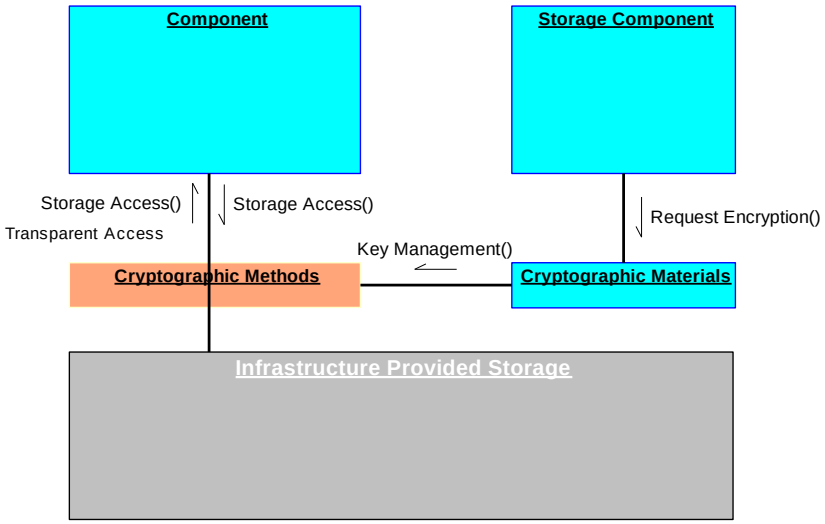

Figure 72: The Transparent Act of Encrypted Storage

Here, if the required encryption/decryption has not been enabled, the storage media, and thus any data therein, would be considered as unavailable for use.

Therefore, where appropriate, a component should be designed for robustness to prevent erroneous or inappropriate behaviour due to (possibly intermittent) unavailability of data on the storage media. This should be considered at the design stages of an Exploiting Programme.

In such cases, protection may be provided by ensuring that the Storage component is aware of the viability of the storage media for use: see the Change in Storage Media section of this policy and the Cryptographic Management IV.

Data stored on encrypted media can also be made inaccessible by revoking the cryptographic keys associated with the storage media (see Cryptographic Management IV). There are different types of cryptographic keys used and different areas of storage media can have different cryptographic keys. Sanitisation of the related cryptographic material will render the encrypted data in the storage media inaccessible (preserving confidentiality) whilst allowing for the recovery of data if the keys are later re-applied.

A.1.12 Recording and Logging

A.1.12.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Storage
- Security Approach

A.1.12.2 Introduction and Scope

This policy provides the framework for a retention strategy by which a component can identify items of data that are needed for future use and hence which need to be retained by recording or logging. It also has guidelines for the kind of rules for data retention within that strategy, such as how long a data item is required for and when a data item is no longer required.

It does not address the specific means by which the storage of these items is managed (this is in the Storage policy), or how it is published by the Information Brokerage component (see the Data Exchange policy).

A.1.12.2.1 What is Recording and Logging?

A.1.12.2.1.1 Recording

Recording is the process of retaining identified data items, received by or generated within a component, that need to be retained for future use (e.g. authorisations received). As the actual storage and publishing of items of data is handled outside each component, the only element of this that a component needs to address is the initial identification of required items of data, as per the retention strategy, which will cover rules for what is recorded and how it is captured, etc.

Note that any data visible to a component can be subject to recording, regardless of whether it was generated by that component or the overall system within which that component resides (e.g. a communications component may record communications which are relayed through a platform as well as those for which it is either originator or recipient).

A component could therefore record data that it cannot make complete use of; for example, a communications component would be aware of data being transmitted and could therefore record the data and its intended destination, but if the data represents a specific graphical or audio format, the component could not actually understand the data it has recorded. (But it would know the intended recipient).

A.1.12.2.1.2 Logging

Logging is the process of identifying and retaining data items received by or generated within a component as part of that component's normal operation that need to be retained for possible later use, but which are not required as a direct result of that component's role. Examples of this are maintenance and usage data such as storing data for debugging, logging errors, updating internal data tables and logging events such as user log on / log off / hand over times for non-repudiation, audit or other security-driven needs. As storage and publishing of data items is handled outside the component, the only element of this which a component needs to address is the initial identification of required items of data, as per the retention strategy.

A.1.12.2.1.3 Recording and Logging Commonality

A single retention strategy, a set of specific retention rules, can be used to cover both the recording and the logging for a specific component. The process of identifying whether a data item has to be retained and applying a set of retention rules to it is identical regardless of the ultimate use of that data.

A.1.12.2.2 Why is Recording and Logging Important?

All Exploiting Platforms will have a general responsibility to recognise and select for retention the certain types of required data, and to know what data has been captured in this manner.

Logging specifically allows a component to record and later articulate data about its current and past internal processes that may not be needed as part of its principle role but may be needed to meet its other responsibilities.

A.1.12.2.2.1 Complying with Legal and Regulatory Frameworks

Data retention requirements may also form part of legal and regulatory frameworks with which a PYRAMID-compliant Exploiting Platform will need to comply, e.g. a Crash Survivable Memory Unit (CSMU, or 'Black Box'). Whilst data retained for such a function is likely to have strict requirements such as accuracy or the inability to delete data, or the use of specific secure and survivable storage media, the fundamental process of a component identifying and capturing data items remains unchanged.

The specific recording requirements needed to comply with a given legal and regulatory framework will be an Exploiting Programme-specific issue, as it depends on platform type and capability and the countries and regions where the platform will be operated.

A.1.12.2.3 Aim of the Policy

This policy explains how the PRA has been designed to support recording and logging in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** This policy describes a framework that allows data driving of the retention strategy.
- **Flight Certifiable:** This policy describes the retention of data, including safety-related data, needed to comply with regulatory requirements.
- **Incorporate Future Growth:** This policy provides a framework which grants ease of modifying which data is to be retained.
- **Resilient Against Obsolescence**: This policy provides a framework that allows the system to be easily adapted to cater for new software logging requirements.
- **Scalable:** This policy describes a framework that controls data retention at a component level, allowing this function to be performed without regard to the structure and complexity of the surrounding system.
- **Security Accreditable:** This policy describes a framework within which the ability of a component to implement handling constraints associated with retained items of information, e.g. for accountability and auditability.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** This policy describes a framework that supports the recording of data about health and capability.

A.1.12.3 Overview

This policy sets out the approach to retaining data within a PYRAMID system and can be summarised as:

- Each component will have its own retention strategy for the data it handles, covering what is retained and for how long.
- Retention of specific types of data may be required for regulatory and audit purposes.
- What is retained and for what purpose will be specific to the Exploiting Programme.

A.1.12.4 Retention Strategy

A.1.12.4.1 Component Retention Strategies

Each Exploiting Programme will have its own requirements for retaining data created through the operation of its systems. Some of this retained data is required in order to perform its system functions (e.g. a value to be held for a short period to enable computations to be performed) whilst other data will be required for much longer, to support post-mission evaluations and audits, etc. This data will be produced by the individual components and each component will be responsible for the retention of all data deemed necessary. This is identified and managed by the component's retention strategy.

The retention strategy needs to be understood 'within' a component, to allow a component to identify data items and their associated context and hence respond to an outside instruction about that data. Note: translation of retained data for use by external systems (e.g. for mission debriefing or analysing health data) is outside the scope of this policy.

In some cases, there is likely to be alignment between the retention strategies of different components in order to capture related data from multiple components associated with a specific concept or operation. In order to facilitate alignment, retained data needs to be synchronised across the system.

In order to meet the Key User Requirement for PYRAMID to be configurable, the retention strategy for a component must be a data-drivable set of rules; allowing rules to be changed between (or potentially during) operations.

A.1.12.4.1.1 Retention Strategy Requirements

The retention strategy consists of a set of specific rules and supporting information about what data the component is required to retain. The strategy also includes how the component should manage the active rules and under what conditions changes to the currently active rules are made.

Retention rules will cover:

- How the component should work out which items need to be retained for subsequent publishing or storage.
- How the data is to be sampled (e.g. the frequency of recording the value of a given parameter and the accuracy to which it is to be recorded).
- The conditions under which the component is to start or cease retaining a piece of data (e.g. to start or stop recording the value of a given parameter). The circumstances permitting this instruction may be an event internal or external to the component itself an example of this is a rule that commences the logging of a measured parameter that is not recorded continuously but only when it falls below a defined value understood by the component, or an external instruction that recording is no longer required. This will likely involve the Authorisation or User Roles components.
- How long a data item of a specific type (e.g. the value of a parameter at a specific instant) is required to be retained for (and hence when it is no longer required and can be disposed of). As with starting or ceasing retaining data items, the authority to dispose of a specific data item may be generated purely internally to the component or may depend on external confirmation that the data item is no longer required by another component or external user. Physical disposal of retained data in response to a change in the active retention strategy is described in the Storage policy.
- The classification associated with storing or transmitting the data item, which could increase the need for encryption.

The retention strategy will need to respect the requirements for mission analysis and safety and security logging, meaning that logged data will not be disposed of if there is a requirement for its ongoing retention and that some pieces of data may be retained that are not normally required by the component.

A.1.12.4.1.2 Retention Strategy Exclusions

A component's retention strategy will not include:

- Which other components or external entities require retained data items, as this is the responsibility of the Information Brokerage component.
- Decision-making covering sanitisation of data. This is covered by the Storage policy.

A.1.12.4.2 Outline Example

Figure 73: Example Use of a Retention Strategy shows an outline example of a component's retention strategy in use. The currently active rules in the component's retention strategy start and stop recording a specific data item in response to an instruction from another component or an external entity.

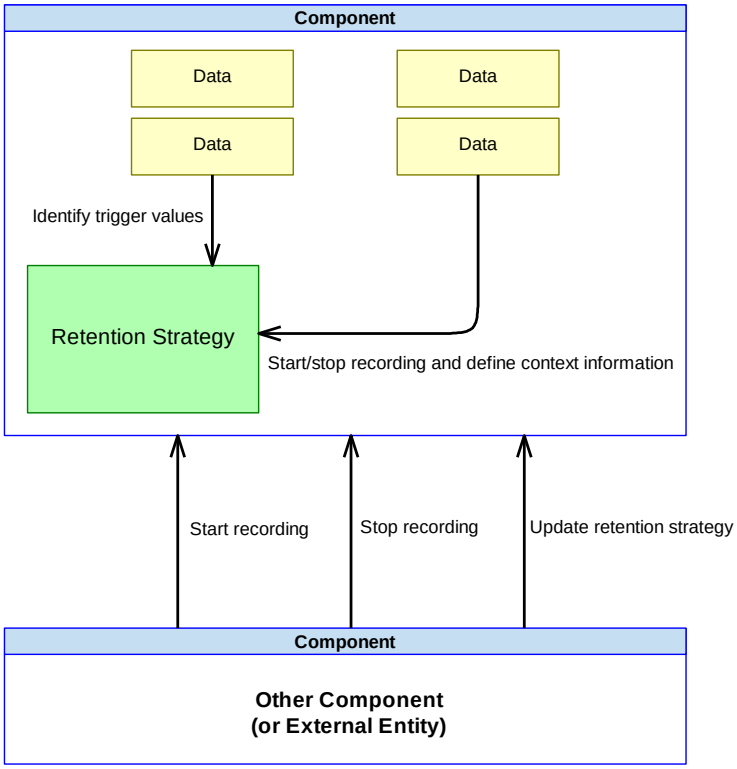

Figure 73: Example Use of a Retention Strategy

Subsequently, a change to the currently active rules is made such that an additional set of rules become active. Under the new retention rules, in addition to recording the data item in response to external instruction, the component will record the data item whenever a different parameter within the component exceeds a specified triggering value.

Note that this is only one example of how a retention strategy may operate. A component can have a retention strategy that requires no input from external entities.

A.1.12.5 Security Logging

In addition to information retained for operational reasons, including any audit purposes, it is expected that each Exploiting Programme will have a requirement to log specific events that arise during its use in order to support the security objectives of integrity, accountability, auditability, non-repudiation and to facilitate forensic examination of cyber incidents, etc. These attributes are described in the Security Approach policy. The data being logged will typically include:

- Anomalous behaviour or states.
- Tamper events.
- Authentication successes and failures.
- Authorisation successes and failures.
- Application and system requests to non-whitelisted resources.
- Application and system start-up, shut-down or pauses.
- Use of higher-risk functionality (certain network connections, changes to user or application privileges, use or change of cryptographic material, etc.).
- Changes to high-value data (including classification).
- Changes to relevant protocols.
- Changes to configuration files.
- Excessive use of certain resources.
- Any other cyber-related data.

The attributes being recorded for each event type are again the responsibility of the Exploiting Programme to define, but as a minimum should detail the who, what, where and when for each event. The Security Guidance for PYRAMID Exploiters, Ref. [60] contains additional information about security logging.

A.2 Specific Policies

Specific policies are policies that describe how the components are intended to be integrated to provide certain specific functionality in a way that supports the PYRAMID KURs.

A.2.1 Cyber Defence

A.2.1.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Health Management
- Security Approach

A.2.1.2 Introduction and Scope

This policy explains how the PRA can be used to provide the system with a level of security monitoring and protection from unauthorised interactions. It describes how components should work together to protect against different types of cyber attack.

A.2.1.2.1 What is Cyber Defence?

Cyber defence is the action of defending against or resisting a cyber attack. The security risks applicable to an Exploiting Platform will need to be analysed by the Exploiting Programme, but may include loss of confidentiality, integrity or availability of systems and/or data through the spoofing of external resources (e.g. ATS or GNSS), infiltration of the ground facilities by hostile forces, intercepting or replicating communications, access to sensitive data in the event of a crash, etc.

The Security Guidance for PYRAMID Exploiters, Ref. [60] contains a section on cyber defence and resilience.

A.2.1.2.2 Why is Cyber Defence Important?

The ability to detect, defend against and recover from cyber attacks is crucial for ensuring the security and continued operability of an exploiting platform.

A.2.1.2.3 Technical Defences in the PRA

The defences that may be required to be supported by the PRA include:

Protecting the boundary

- Validity and integrity checking.
- Application authentication.
- User authentication.
- Identification of spoofing.
- Coordinating infrastructure defences including networks and encryption.

Limiting disruption

- Re-routing traffic following Denial of Service attacks.
- Priority service for safety related data.
- Restricting access and permissions.

Limiting impact of attacks

- Segregation of domains.
- Integrity checking internal data.
- Monitoring system behaviour.
- Anomaly detection and reporting.

Detecting and analysing attacks

- Logging.
- Audit reports.
- Flagging anomalous behaviour.

The Security Approach policy includes information about security functions and different domains that enable the above defences.

A.2.1.2.4 Aim of this Policy

This policy explains how the PRA has been designed to ensure that deployments will be resilient to cyber attack in a manner consistent with the PYRAMID KURs.

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** The policy allows for differing approaches to defence against cyber attacks, depending on the requirements of the Exploiting Programme.
- **Scalable:** This policy accommodates the scaling of cyber defence to cover the necessary elements of detection, defence and recovery for any number of components used in a deployment.
- **Security Accreditable:** The security risks that an Exploiting Platform could be exposed to have to be understood. This policy identifies some of the risks that are applicable at an architectural level and addresses them.
- **Utility Across A Range of Missions:** Some mission scenarios may be more susceptible to certain attack vectors than others. The cyber attack examples covered in this policy apply to a range of operational scenarios.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** A cyber attack can cause part of the system to fail. The policy allows those failures to be identified and addressed.

A.2.1.3 Overview

The policy shows how the components contribute to the cyber defence of an Exploiting Platform by:

- Implementing protective measures that make a cyber attack more difficult to undertake.
- Detecting anomalies (unexpected states or behaviour) that may indicate the presence of a cyber attack.
- Reacting to an identified cyber attack by undertaking an appropriate contingency action.

A.2.1.4 Cyber Attack Vectors and Detection

It is not possible to address all possible cyber attack vectors within this policy, however, some aspects of a proactive approach to defence against cyber attack are covered in the following sections. The Defence Against Cyber Attack IV provides an example of how the PRA reactively responds to the effects of an established cyber attack.

The proactive PRA approach to cyber-resilience calls upon a number of components working together to monitor and control the security border of the system. This includes controlling who (authorised operators) can access the system, and with what role and level of privileges, which external entities are trusted to share information on the internal networks used by the system, detecting any threats against the system hardware, software and data, and responding to any attacks that compromise the service.

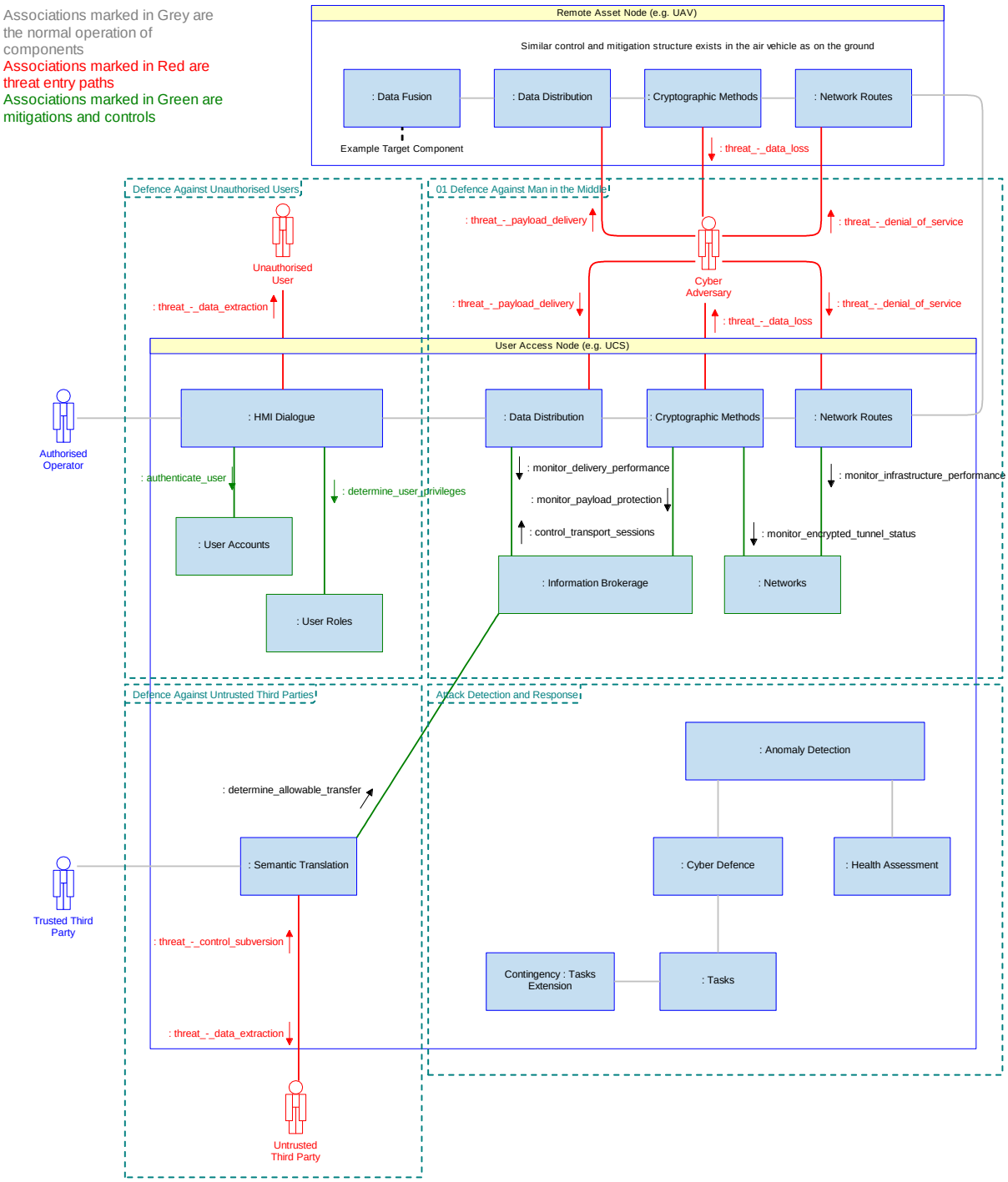

Figure 74: Defence against Cyber in a UAS Example

Figure 74: Defence against Cyber in a UAS Example represents a number of different associations between actors and components:

- Communication flows are illustrated between components that might be present in an Unmanned Air System (UAS), comprising an Unmanned Air Vehicle (UAV) and a UAV Control System (UCS). These are not true interactions as seen in the IVs, but represent (via the grey lines) how data may normally be moved around the UAS infrastructure, in this case between an authorised operator and the Data Fusion component (this could be for the display of track information on a screen).
- Mitigations and controls that can be mapped to responsibilities of a component are coloured green. More detail on interactions with HMI components is provided in the Human-Machine Interface policy and User Management IV, and detail on the interactions between communications components is in the Use of Communications policy and communications-related IVs.
- Attack vectors for a small number of threat types are coloured red. It is not realistic to illustrate all possible threat actions, so examples of possible threats from external cyber-adversaries are provided, including denial of service (DoS), control subversion, payload delivery and attempted unauthorised data access. These are prefixed with "threat" in Figure 74: Defence against Cyber in a UAS Example to further differentiate them from system actions. The main focus of Figure 74: Defence against Cyber in a UAS Example is to illustrate example entry paths available to a cyber-adversary, and where such entry paths can be proactively defended by the PRA.

A.2.1.5 Establishing Trust

Building trust in the system (the confidence that a component or other system element will behave as expected) goes further than what is possible with only the PYRAMID components and will be an Exploiting Programme decision. Typically, this might be achieved through drawing on anti-tamper mechanisms in the infrastructure and software certificates, etc. to ensure that only valid software is running on legitimate equipment. During boot up and at intervals during runtime, the system should authenticate the operation of the system and confirm that trust can be and is still established from the software down to the trust anchors in the hardware (an established authoritative entity for which trust is assumed, not derived). Once established, this baseline trusted state will form the basis by which ongoing cyber defence is performed, with suspicious behaviour such as unauthorised resource usage being analysed by the components shown in the lower right frame of Figure 74: Defence against Cyber in a UAS Example. See Attack Detection and Response for further details.

To minimise attacks through the supply chain, software whose pedigree cannot be guaranteed is expected to be sandboxed to protect the rest of the system.

More information on this is available in the Dstl Security Guidance for PYRAMID Exploiters, Ref. [60].

A.2.1.6 Defence against Unauthorised Users

A common attack vector is for the cyber attacker to pose as a user of the system, or for a user to gain unauthorised access to additional resources or information through escalation of privileges.

The User Accounts and User Roles components would provide a first layer of boundary defence and include security enforcing functions (SEF) to limit the threat imposed by a cyber adversary attempting to gain direct access to the system via an HMI front end, authenticating the user and determining their privileges for the system and therefore protecting against the threats of control subversion, data extraction and payload delivery.

Typically, a user will be required to enter valid credentials associated with their account. It should not be possible to circumvent the login procedure.

The assignment of roles and privileges would prevent a user (including a compromised user) from elevating their permissions in order to access services and/or information to which they are not normally allowed access. It is anticipated that there will be at least three levels of user access:

- Operator user - the authorised operator will have access to different information or functionality in accordance with their operator and command and control roles, e.g. commander, sensor operator or maintainer.
- Admin user - responsible for allocating permissions to the users of the system.
- Root user - provides access to the operating system, application code and data within the system. This level of access will not be available to admin users or operator users, or to application software.

Actual implementation of user access and controls will be dependent on the Exploiting Programme.

A.2.1.7 Defence against Network Attacks

A.2.1.7.1 Defence against Man in the Middle

In a man in the middle (MITM) attack, a cyber-adversary impersonates each endpoint of a communication link, with the purpose of affecting confidentiality, integrity or availability of the data by intercepting and interacting with it en route.

Where the cyber-adversary seeks access to sensitive data being passed from an air vehicle to the ground station via an RF link thus impacting confidentiality or integrity, the Cryptographic Methods component would make this attack vector extremely difficult to enact assuming high grade encryption and authentication methods are utilised, thus ensuring confidentiality of information and preventing snooping on data transfers. Data hashing provides a means of identifying unauthorised modification. Commercial grade encryption keys and tokens may be appropriate for some data transfers, however some data is broadcast unencrypted.

A low tier MITM attack is the replay attack, whereby a previously valid session password is used in an attempt to impersonate a valid user (operator, admin or root). This can be prevented by the use of authentication protocols including "nonce" words, timestamping and unique random session keys, etc.

A.2.1.7.2 Defence against Denial of Service

A DoS attack attempts to restrict the legitimate availability of a communications channel, typically by flooding an unprotected network with spurious data requests or by disrupting connections between nodes. The Network Routes component generates traffic flow performance information over a network link, providing the Networks component with knowledge of a possible attack if the flow is not as expected, and allowing communications to be re-routed, limiting the number of available attack paths and also providing a priority service to any safety related data transfers. As described in the Health Management policy, the Anomaly Detection component may also support the detection of an attack using information from the Network Routes and Networks components.

A.2.1.7.3 Defence against Payload Delivery

Other attack vectors may target the Data Distribution component by including an unauthorised payload (e.g. malware or buffer overflow attacks) within the messages packaged for delivery. The communications components will be required to work together to protect the system borders, including by managing the communication session to reduce the bandwidth available to the adversary to exploit, should other defences be compromised.

Should such a payload manage to make its way into the system, any attempts to install the payload or to make use of resources that are not normally used by the software should be detected. It is expected that only valid, signed software would be permitted to execute, deviations from a valid software configuration would be reported as anomalous behaviour and any software calling on unexpected resources or data would be identified. This is dependent on an understanding of expected “good” system behaviour and established whitelists, etc. See Attack Detection and Response for further details.

A.2.1.8 Defence against Untrusted Third Parties

Further to the information transfer risks identified in Defence against Man in the Middle, information exchange with systems outside the administrative control of the current PYRAMID system poses some additional risks. These additional risks include unauthorised data extraction and control subversion.

The Semantic Translation component manages the semantics of information exchange with an external system (e.g. via Link 16). It manages which external services can interface with the system, thus addressing the threat of unauthorised data extraction and control subversion. It does not, however, control the related devices (e.g. firewalls or gateways). Information Brokerage has an understanding of who the external participants in the exchange are. An example might be for the Semantic Translation and Information Brokerage components to prevent a cyber-adversary accessing tactical information through a valid connection made for the purpose of Air Traffic Control data transfer, such as CPDLC (Controller - Pilot Datalink Communications). The system architecture can be expected to utilise segregated security domains to further protect data confidentiality.

A.2.1.9 Attack Detection and Response

A key indicator of a cyber attack is the system not behaving as expected. Events from all components are collected and analysed within the system. The Anomaly Detection, Health Assessment and Cyber Defence components work together to analyse the different event types, identify anomalies and determine if these may be the result of a cyber attack. Example activities would include:

- System Monitoring - identifying anomalies that are related to changes in the health of the system hardware. From a cyber defence point of view these may lead to additional vulnerabilities not normally present (like the loss of redundant hardware, making an attack against system availability easier).
- Performance Monitoring - identifying anomalies that are related to whether services are operating within expected behaviour profiles. From a cyber defence point of view a significant change in capacity use (e.g. a lot of processor time being used) could indicate a cyber attack.
- Protective Monitoring - identifying suspicious behaviour that can be related to security related activities, for example, an escalation of privileges by a user (e.g. to admin or root), combined with unexpected access to storage data could indicate an attack against the confidentiality of data held in the system.

When a cyber attack is identified, it will be the responsibility of the Cyber Defence component and, where possible, the attacked component to mitigate the effects and to report it appropriately so that the effects of the cyber attack may be addressed, such as disregarding a source of information or sanitising data. The Defence Against Cyber Attack IV provides an example of how the PRA responds to the effects of an identified cyber attack.

A.2.2 Human-Machine Interface

A.2.2.1 Pre-Reading and Related Policies

Prerequisite

- Interaction with Equipment

Read in conjunction with

- Autonomy
- Component Connections

A.2.2.2 Introduction and Scope

This policy explains how components can be used together to support interaction between human users and system (i.e. machine) elements within an Exploiting Platform.

A.2.2.2.1 What is an HMI?

A Human-Machine Interface (HMI) connects humans to a system within an Exploiting Platform, providing users with the ability to control and monitor the system. Its main purpose is to perform translation of:

- Actions performed by users into information and instructions to be enacted by the system.
- Data held in the system into information presented to users.

These translations will occur during HMI exchanges between the system and users, which can usually be divided into a series of constituent HMI interactions. HMI interactions involve bi-directional provision of information, as described below.

In the user input direction, examples include:

- A linear button press (physical or virtual), or lever movement.
- A more dynamically defined user input, such as a voice command spoken into a microphone, a shape drawn on a surface, a hand signal or eye movement in view of a camera, or data entered via keyboard.
- Interaction with the relevant device for iris, fingerprint or speech recognition.

In the system output direction, examples include:

- Immediate illumination of a button to indicate a press was registered.
- An on-screen notification to indicate completion of the ultimate system action requested via a button press.
- A periodic sound or haptic vibration to indicate that a piece of equipment is in operation.
- More complex, integrated outputs, ranging from a dynamic map or weather forecast (which can be interactive via user input) to an augmented reality presentation such as a helmet with its own internal screen, 3D audio, voice input and touch feedback gloves.

The above examples are intended as initial, illustrative examples, and are not indicative of the full scope of potential input and output methods.

The system's fundamental concerns, with regards to its HMI, are:

- The information that needs to be exchanged across the HMI so that it can be understood by the consumer.
- How the information should be conveyed so that it can be understood by the consumer.

A consumer in an individual interaction is either a system element receiving information instigated by a user input, or a user receiving information from the system. An exchange may consist of multiple interactions, meaning that both a system element and a user can act as both an information source and a consumer within one exchange.

A.2.2.2.2 Why is HMI Important?

The HMI is an important part of the PRA because without it users cannot interact directly with the system. By providing the connection between information that is understandable to the system and information understandable to users, HMI enables human oversight and control of the system. This separation between the user and the system by an HMI can support safety and security partitioning.

A.2.2.2.3 Exclusions to this Policy

The HMI supports authorisation, and while it should be considered with respect to user input and security controls, the authorisation process is excluded from further discussion in this policy. The User Management IV specifically covers the topic of user access to the system. The Autonomy policy includes a section on authorisation.

A.2.2.2.4 Aim of this Policy

This policy explains how the PRA has been constructed to support HMI at the boundary between system and user understanding.

The PYRAMID KURs that are applicable to this policy are:

- **Configurable, Incorporate Future Growth, Exploitable, Scalable and Resilient Against Obsolescence:** By distinguishing between system and user understanding while enabling dialogue between data sources and consumers, the HMI and dialogue participants can be changed (e.g. reconfiguration, reporting or scaling) with minimal impact on other components. This allows component reuse in Exploiting Platforms with different HMI requirements.
- **Flight Certifiable and Security Accreditable:** The HMI components support system partitioning that prevents compromise of high priority software and can allow more effective human interaction.
- **Utility Across A Range Of Missions:** By enabling human oversight, the HMI can allow multiple automation levels across different operational contexts.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement that was considered operationally desirable for the PRA to support:

- **Supportable:** The HMI enables human oversight of system automation to help determine maintenance needs, and the HMI allows interaction with humans during maintenance.

A.2.2.3 Overview

The HMI is the interface between the user and the system, and may provide information to, or receive information from a user, in any format that can be sensed and understood (for example light, images, sounds, movement or other means).

The HMI Dialogue and Information Presentation components manage the format and information flows between a user and a system such that system oversight and control can be achieved. These components provide a modular approach to the coordination of information by an HMI and decouple the system from the user interfacing mechanisms. Thus facilitating an HMI implementation in a manner that best meets an Exploiting Platform's requirements.

A.2.2.4 Use of HMI

A.2.2.4.1 Interface between System and User Understanding

The HMI provides a connection between system and user understanding, and establishes the required dialogue that collates appropriate context so that information is comprehensible to each consumer. This prevents functional pollution, and enables systemic behaviour to be reused across multiple Exploiting Programmes with minimal change, even where the programmes have completely different HMI methodologies.

The conversion between system and user understanding is demonstrated in the following example:

- An authorised operator updates an altimeter reference pressure setting using a continuous dial input.
- An appropriate component, such as Environment Integration, will have knowledge of what an altimeter reference pressure setting means within its subject matter, and will know how to use such information (for example as a datum pressure) when it is provided via the HMI.
- The HMI implementation, meanwhile, will understand how input is conveyed via the continuous dial, and can translate dial interactions into requests for the appropriate component to change the reference pressure value. The behaviour as a result of these user inputs is the concern of the enacting component(s).

A.2.2.4.2 Human Oversight and Control

Although many functions of the system may be automated, some system actions will require some degree of human interaction to ensure safe and effective operation. The HMI will need to be able to support system operation at varying levels of autonomy, requiring the presentation of different levels of information content and degree of interaction (see the Autonomy policy). For example, human oversight and control may be mandated by operational regulations (e.g. airspace access) and rules of engagement. Likewise, maintenance activities will also generally require interaction with at least one human maintainer, especially in response to unexpected situations.

A.2.2.4.3 Safety and Security Partitioning

The HMI needs to provide a consistent and usable user interface whilst maintaining robust partitioning between different functional integrity levels and security domains (see Partitioning). The HMI can also contribute to user controls, to ensure that a user can only make inputs and can only access information appropriate to their role or security clearance.

A.2.2.5 HMI Components

The aim of the HMI can be decomposed into six general functions:

- Providing the boundary connection between the system and its users.
- Organising the information needed for meaningful exchange between the system and users.
- Providing consistent HMI across the system, where applicable.
- Conveying the presentation of information in exchanges between the system and users. This includes the generation of subject specific presentations, and the consumption of inputs presented by users.
- Merging of subject specific presentations to produce an integrated output.
- Understanding system infrastructure and the hardware utilised for user interaction so that received and provided information is appropriately processed.

To meet these aims, the PRA includes two components that are specific to the HMI:

- HMI Dialogue - Source data needs to be comprehensible by its consumer. This component enables such comprehension by managing dialogue between HMI interaction participants.
- Information Presentation - This component is primarily concerned with how information is conveyed and perceived in an HMI interaction.

The PRA HMI philosophy is to provide components that can be deployed in different ways to produce interfaces that meet the needs of specific operator roles during particular mission phases, considering the operational environment and system configuration. The HMI components support this philosophy by decoupling the behaviours of HMI dialogue participants from the user interfacing mechanisms, such that they can change or be developed independently of each other in a compatible manner. This will allow Exploiting Programmes to change the HMI with as little impact on the dialogue participants as possible. HMI considerations are contained within the HMI components and not distributed through the system, so changes in other component don't require the HMI components to change, and vice versa.

A PRA component may act as either a data source or information consumer in an HMI interaction: as the origin of system data for presentation, or as a consumer of requests to initiate some form of system behaviour. Similarly, a user and the HMI devices that they utilise may provide source input data or consume presented information. The dialogue between participants enables the system to establish comprehension rules so that a connection can be formed between system and user understanding. The context of a dialogue will inform the comprehension rules, allowing the HMI to be implemented in a flexible and versatile way.

An HMI exchange does not strictly require both HMI Dialogue and Information Presentation:

- For a linear, non-dynamic exchange, HMI Dialogue may not be necessary. For example, a dedicated text display that only presents a certain variable that is constantly displayed or requested simply (such as via a button that is not linked to user permissions), will not require dialogue that needs to be managed.
- Where device equipment contains its own capability to carry out information conveyance processing, Information Presentation may not be necessary. Such devices may range from 'smart' equipment with some degree of intelligence, to simple equipment directly linked to a dedicated information source or consumer.

The HMI components have knowledge of how and when information is presented for output, and may have some logic that can filter inputs from a user in order to reduce other components performing nugatory processing.

Note that no services are defined for components to provide data for use by an HMI (see the Services Not Defined in the PRA section).

A.2.2.6 Exchange across the HMI

A.2.2.6.1 HMI Interactions

An exchange between the system and a user may consist of multiple interactions. These interactions can be categorised into groups in the exchange process, as described below.

Source data provision to HMI components:

- Provision from a user, in which user action information is provided via an HMI device as source data.
- Provision from the system, in which system information is provided as source data.

Presentation conveyance:

- Conveyance to the system, in which input information presented to a device is captured and conveyed to the system. Information Presentation determines the most appropriate manner to convey the information.
- Conveyance to a user, in which cohesive compositions of system information are conveyed to the user via presentation resources. Again, Information Presentation determines the most appropriate manner to convey the information.

Dialogue:

- HMI Dialogue creates and manages dialogue between the system and users. The dialogue determines interaction context, which is used to comprehend the source data as information appropriate for the consumer, and to determine where the information should be sent. This includes, for example, identification of the need for additional context information, or compiling source data and context information together.

To demonstrate how interactions in these groups make up an exchange, consider a scenario where a user requires details on a payload:

- **User source data provision to HMI components:** The user performs a computer mouse click on an "Item Details" graphical element. The computer mouse device then provides data that represents the user action to the HMI components. Information Presentation instances are responsible for all presentation aspects of the Graphical User Interface (GUI) that the mouse is connected to.
- **Presentation conveyance to user (input confirmation):** Information Presentation may convey a confirmation that the user input has been registered, such as a sound or on-screen notification.
- **Presentation conveyance to system:** Information Presentation instances convey the mouse click as a request for details corresponding to the graphical element.
- **Dialogue for system understanding:** HMI Dialogue gathers context and determines that inventory data is required, then manages comprehension rules that enable the system to identify that the element represents a specific payload.
- **System source data provision to HMI components:** Inventory provides payload item data.

- **Dialogue for user understanding:** The dialogue enables the payload item data to be interpreted so that it is understandable for the user. Other data sources may inform HMI Dialogue of supporting data to be compiled into understandable information (for example, that another user has reserved the payload for use, or that the payload should not be released due to the current mission situation).
- **Presentation conveyance to user (satisfying user intention):** Information Presentation conveys the item details in a meaningful way relevant to the circumstances. For example, it may convey the details to fit a display area by choosing a specific size; or it may convey the details by spoken word if, for instance, a user cannot view a display or has chosen an audio conveyance method.

Figure 75: Example HMI Exchange illustrates the main aspects of the above scenario, and includes additional component instantiations to demonstrate a modular HMI implementation. For example, the additional components may be utilised where a user performs an action with the intention of receiving video from a camera monitoring a payload. The video may be presented on the same Display Screen as the payload details, perhaps through its own window. However, the Context Providers actor may include providers of user role information, and that information may indicate that the user does not have permission to access the video. In such a case a Private Monitor Screen may be connected from which an authorised user can access the video while the other user views the Display Screen.

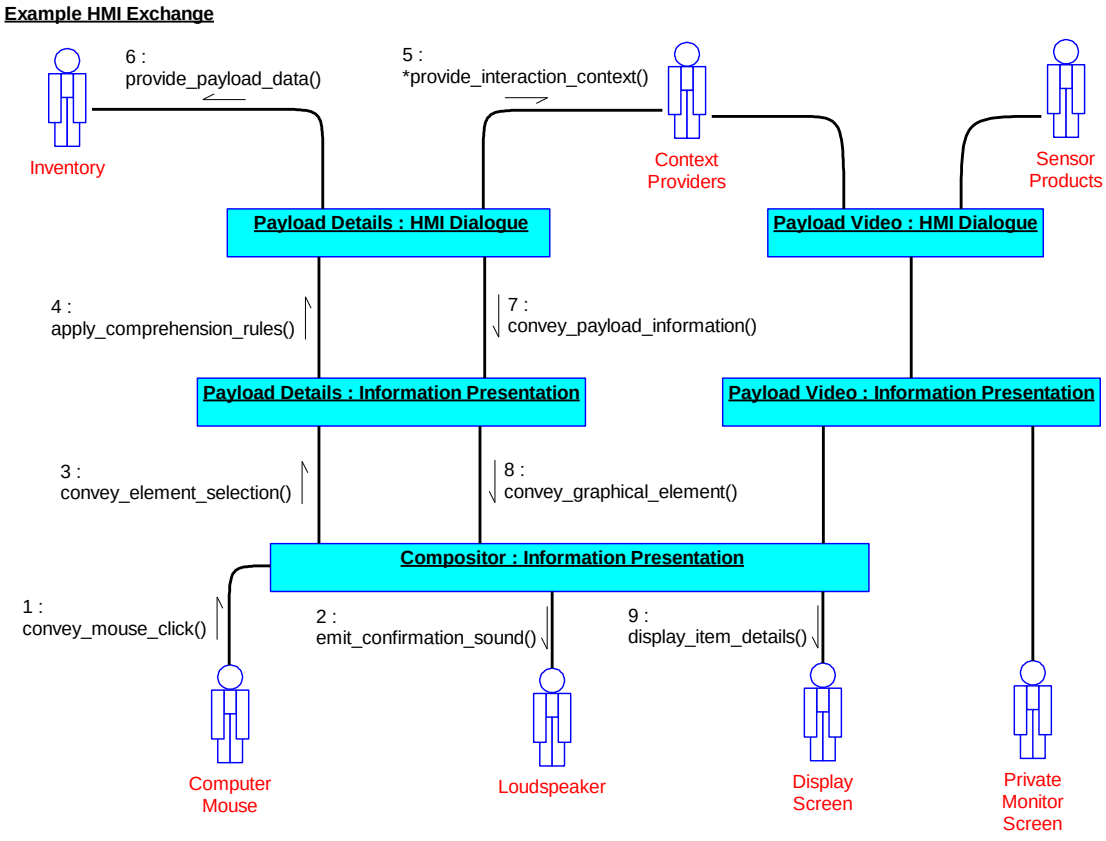

Figure 75: Example HMI Exchange

A.2.2.6.1.1 Input Validation and Filtering

The input validation and filtering performed by the HMI components includes:

- Basic validation of user inputs (e.g. the discounting of non-alphabetic keyboard characters as a result of character entry rules or filtering out background noise). Note: More complex validation, such as a check that an entered altitude setting is compliant with current air traffic services constraints, will be performed by the component that holds this information after dialogue (e.g. Environment Integration).
- Disregarding inputs for which no system component should take action (e.g. a cursor selection that is not on an interactive graphic element, or is on an element that is disabled).

Device equipment may provide an initial filter of user inputs. For example, a keyboard may eliminate erroneous keyboard presses caused by key bounce, or a microphone may reject background noise. However, in these examples, the equipment is external to the system so its filtering capability is not relevant to the HMI components.

A.2.2.6.1.2 Information Conveyance and Dialogue

Information Presentation will determine the intent of user input and system output. For a GUI this may include consideration of aspects such as window layering and transparency, whereas for an audio interface this may include aspects such as speech recognition for voice input or volume considerations for output sounds.

The following examples focus on an HMI with cursor selection and visual display capability. In Figure 76: Information Conveyance and Dialogue Example, Information Presentation determines that a cursor selection event within a visual presentation area is associated with the graphical element representing the switch controlling the undercarriage lights, and conveys this to the system. HMI Dialogue uses interaction context (such as the state of other system tasks) and its predetermined rules to comprehend the input data as a request for the appropriate component, in this case Lights, to perform a setting change. The dialogue may identify further affecting context, such as information that a certain light level is required in the current phase of flight or weather conditions, which may prompt further confirmation of the user intention.

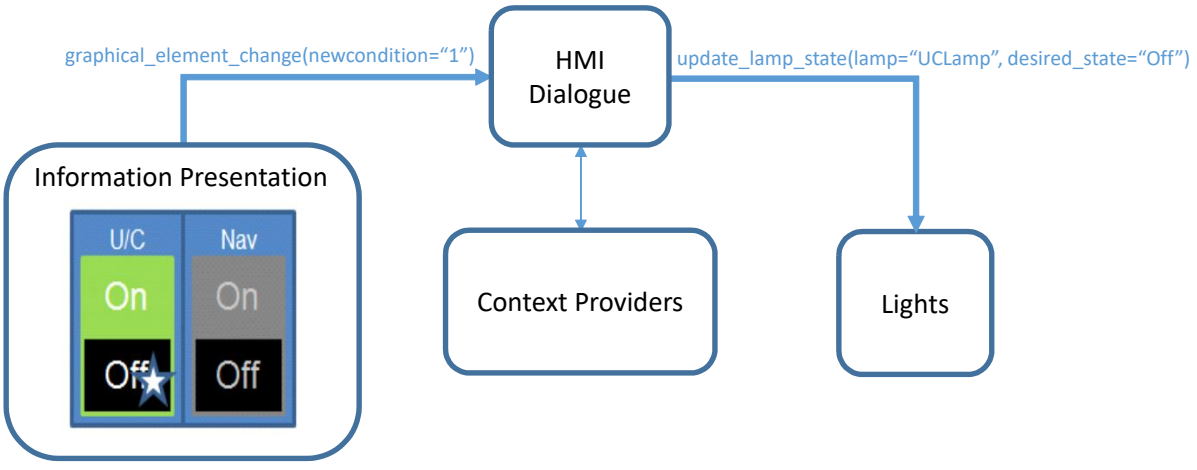

Figure 76: Information Conveyance and Dialogue Example

HMI Dialogue can gather or compile data from multiple user or system sources and map it to parameters that Information Presentation characterises as elements of the HMI.

The PRA does not (and cannot) mandate patterns of behaviour for an HMI Dialogue component as they are determined by an Exploiting Programme. The translations performed by HMI Dialogue will usually be developed to suit the requirements of a specific programme.

In Figure 77: Parameter Mapping, HMI Dialogue maps data obtained from three components to parameters that Information Presentation can use, in order to display a virtual 'switch' output in the 'On' condition (because the Lights component says the lamp is on), with input enabled (because the aircraft is landing) and in green (because the lamp has no fault indications).

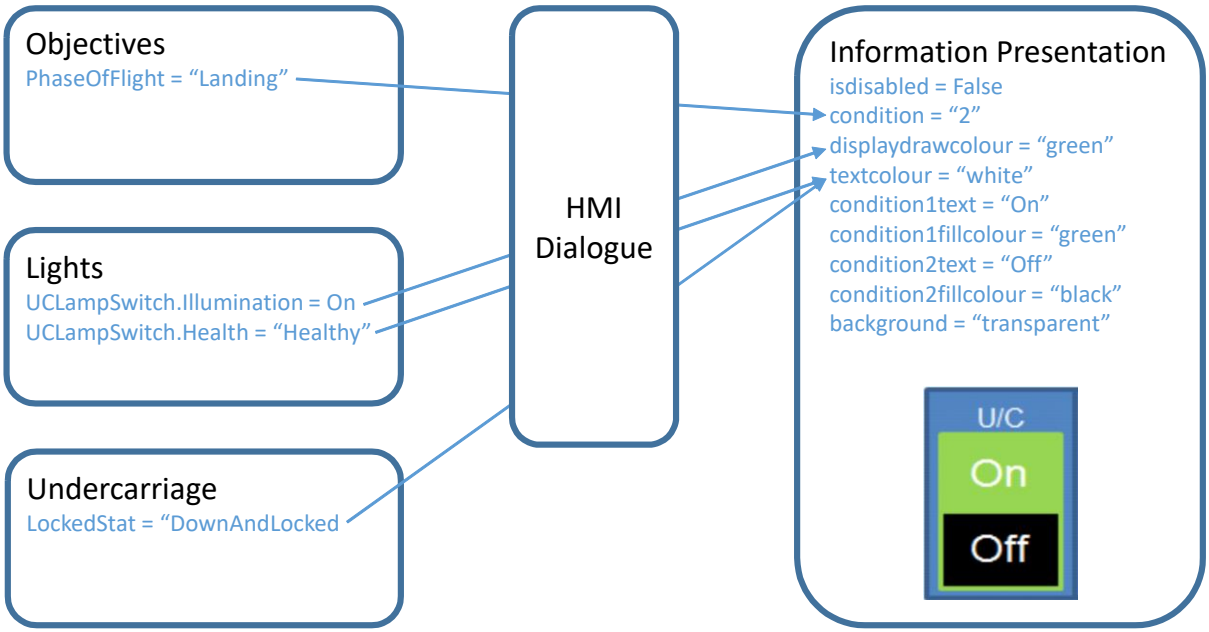

Figure 77: Parameter Mapping

A.2.2.6.2 Presentation Composition

Information Presentation delivers compositions of presentation elements. These compositions can be at a variety of scope levels, from a merging of elements defined within the component, to a combination of specialised presentations that were provided by other Information Presentation instances.

In Figure 78: Presentation Composition, Information Presentation is deployed to understand the rules for presenting two independent items (both switches). In this example Information Presentation understands the graphics commands required to draw a switch, and is provided with a number of conditions that allow it to draw the desired graphics. Settings for the colours and text strings to be displayed are supplied along with more generic higher priority information that globally changes the presentation (such as the HMI being 'disabled' and not selectable by a user).

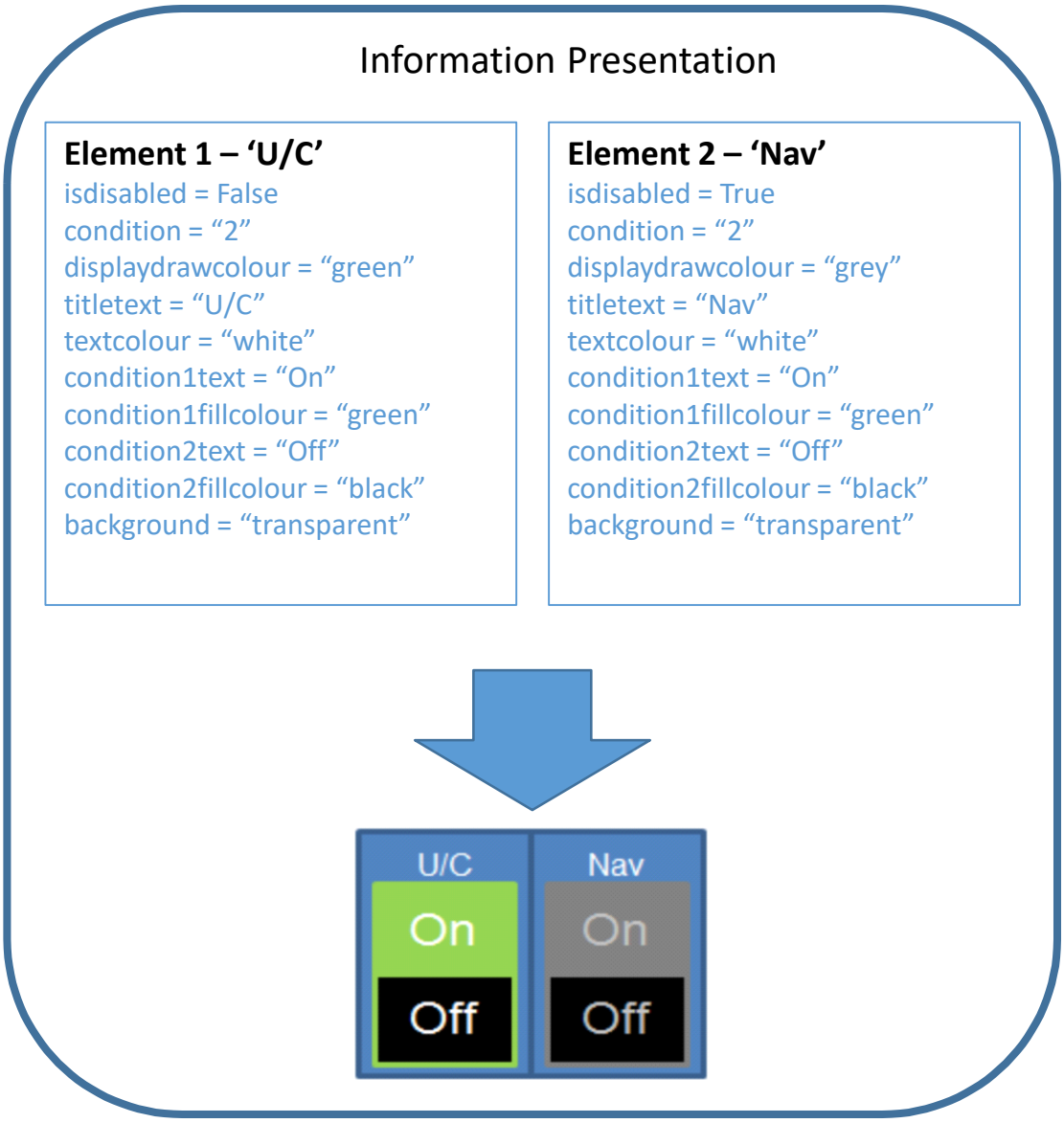

Figure 78: Presentation Composition

Figure 79: Presentation Composition of Multiple Element Types provides an example of a composition consisting of three element types that could each be fed by information from separate HMI Dialogue instances. In this example, Information Presentation uses windowing to layer the information via a display. Note that compositions are not restricted to elements of the same sensory form. For instance, a series of sound alerts could be composited together with visual elements.

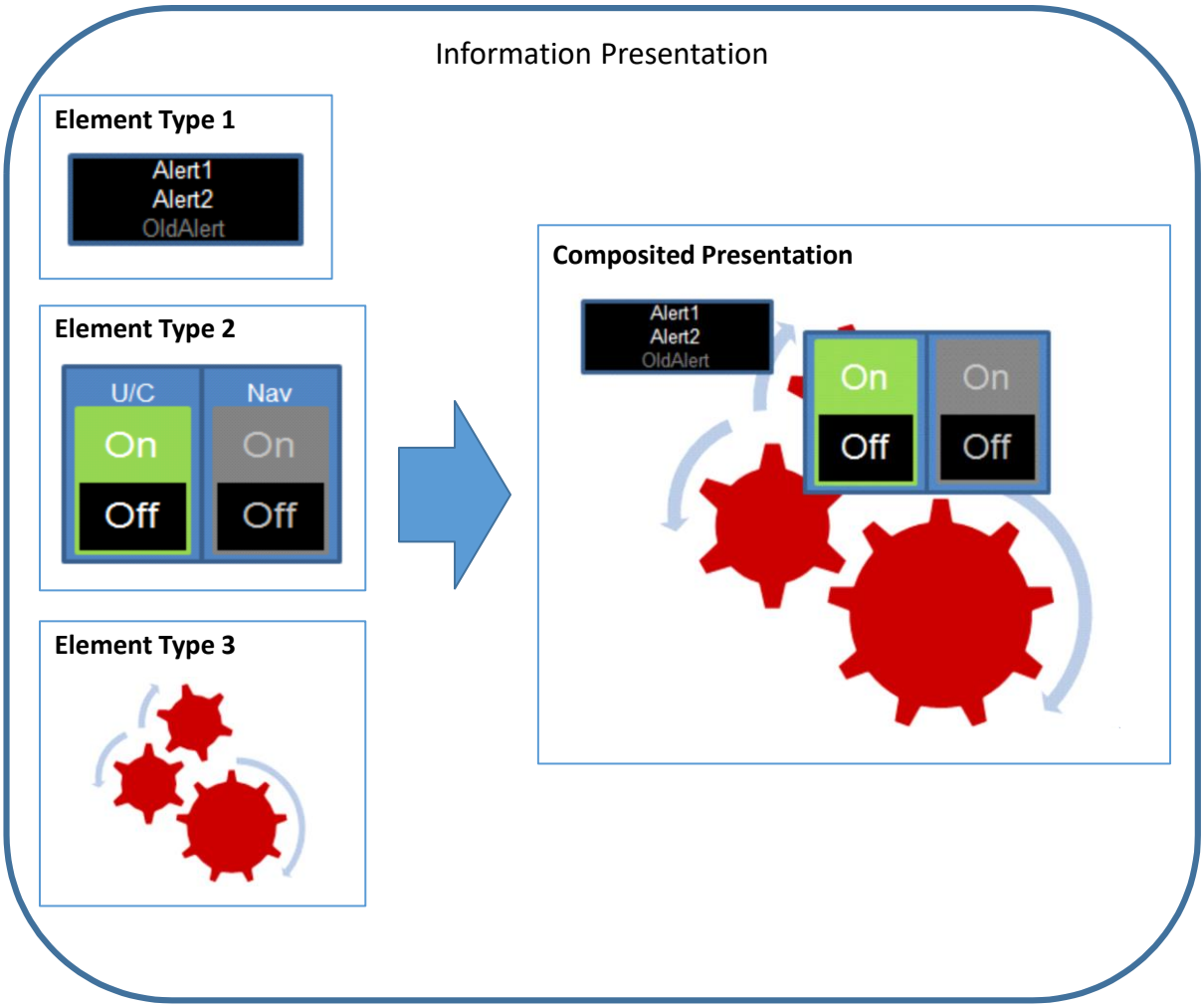

Figure 79: Presentation Composition of Multiple Element Types

A.2.2.7 Structural Considerations

This section summarises design considerations for the HMI components within a deployment. This includes decisions around modularity, partitioning and coordination of HMI-specific information.

A.2.2.7.1 HMI Modularity

The dialogue required to comprehend certain forms of source data will be more effective when the coordinating HMI Dialogue component is highly focused on the HMI exchanges that involve the specific source data types. Similarly, to optimally prepare information so it is available to specific system elements and presentable in certain forms, the dialogue will benefit from tailoring for those HMI exchange participants. This can be achieved by limiting the range of HMI exchange types that an HMI Dialogue instance will manage, and tailoring the instances for those exchanges through data driving. Such modularity and the customisation it allows will enable simpler interfaces, and redevelopment of HMI behaviour will therefore require less effort.

Modularisation of presentation compositions should also be considered, as this will support the design and modification of complex, 'system of systems' presentations. The graphical elements that make up such display compositions are suited for provision by different component instantiations. For example, the windowing systems used by most PCs are highly cohesive, interactive and modifiable, offering the user with the ability to move or resize windows. Modularisation enables these presentation elements to be managed independently

by Information Presentation component instances that can have focused relationships with the most suitable HMI Dialogue instances, and elements like symbology, personalised displays, or widgets can be changed or replaced easily.

Figure 80: Composition from Multiple Component Instances illustrates how the example presentation shown in Figure 79: Presentation Composition of Multiple Element Types could be provided by an Information Presentation instance responsible for a windowing presentation that hosts presentation elements from multiple specialised component instances. Multiple instances of HMI Dialogue could also be deployed to help make the information available to each Information Presentation instance in the required form.

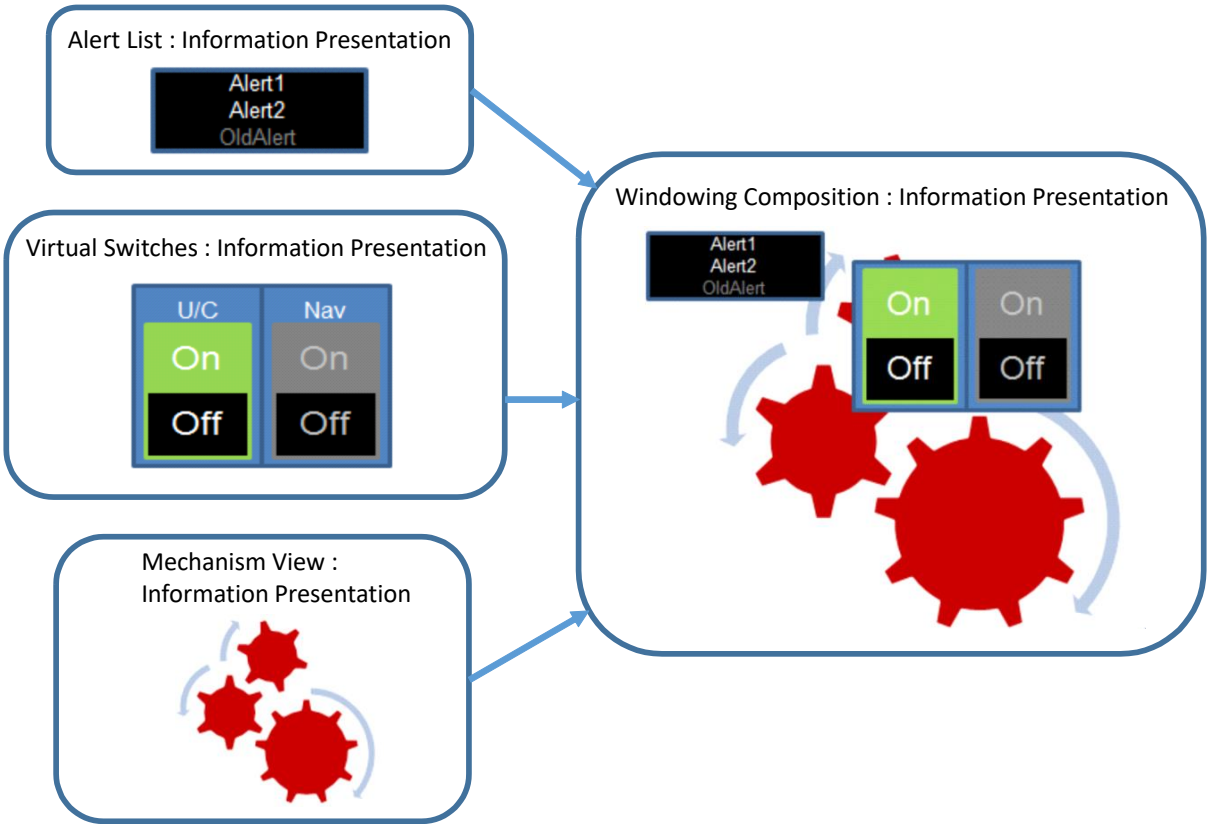

Figure 80: Composition from Multiple Component Instances

A.2.2.7.2 Devices used in HMI

Devices that users interact with can vary significantly in terms of their design, versatility and processing capability, and may be used together to provide a multifaceted interface. For example, the system may react to user input through a control stick by providing haptic feedback through a user's seat, gloves, or through the stick itself. A helmet mounted display may be used which responds to control of the stick, and the helmet may also use head or eye tracking technology. The control stick or helmet may be modifiable during or between missions, perhaps with options for different modes, or through machine learning. The HMI components allow the use of varied device combinations through their flexibility.

The range of potential devices can be grouped by common interfacing mechanisms or specialisations. For example, a touch overlay and pressure mat may be grouped, or there may be a family of keyboard and keypad types. The relationship between devices and modular component instances can allow some devices to be replaced by others from the same family with fewer required changes to the interface.

Where complex information from multiple sources needs to be presented, this drives the need for the Information Presentation component's capabilities. Such capabilities may be provided by the component itself, or by complex devices that can perform a similar role in relation to their area of concern.

A.2.2.7.3 Partitioning

The HMI components support partitioning to meet safety integrity requirements and the security objectives of confidentiality, integrity and availability.

For safety, this means that where a group of HMI components work together to handle information considered to be of high integrity (i.e. high DAL components) they can be assigned to a software partition segregated from lower DAL components. This prevents the lower DAL software from compromising the operation of the higher DAL software, and allows the cost of development to be controlled, with lower DAL software not requiring the same level of verification, etc. For example, an exchange that requires provision of highly consistent information could be conveyed more simply than, or separately from, low integrity information to avoid costly misinterpretation.

Components can also be partitioned based upon confidentiality, integrity and availability requirements with security control mechanisms enforcing separation between security domains. These controls, the nature of which is a decision for the Exploiting Programme, can impact data flow between domains and need careful consideration. For example, where displayed data is required to have high availability, the HMI data source and consumer would ideally be located in the same partition so as to prevent the control, such as a firewall, delaying or even blocking the data. Data must be handled correctly by the HMI to ensure no unauthorised data can leak across the security partition (note that the HMI is not however responsible for maintaining the security partition - see the Safety and Security policy for more detail).

Differing security clearance levels are addressed by the User Accounts and User Roles components, with the filtering and presentation of information to operators provided by the HMI components. More information can be found in the Dstl Security Guidance for PYRAMID Exploiters, Ref. [60].

A.2.2.7.4 Coordination

Information will be provided by users and different system components in numerous forms, so coordination of the presented information is a key consideration for interface consistency and alignment. The only components concerned with user input and understanding are the HMI components: whether a distance is presented in user-understandable units such as metres, kilometres, data miles or nautical miles is an HMI matter, whereas the rest of the system is concerned with the value and internal representation of the distance in question.

Coordination of the forms of presented information is achieved through the design of the specific HMI in question. This coordination is maintained through bridging of HMI component entities to the associated system entities that will ultimately consume information from, or provide information to, the HMI. Consistent data driving of HMI components will also support successful coordination. The Component Connections policy provides further detail on this topic and related areas.

A.2.3 Interfacing with Deployable Assets

A.2.3.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Interaction with Equipment
- Use of Communications

A.2.3.2 Introduction and Scope

This policy covers interfacing with deployable assets before and after separation from a PRA-based exploiting platform.

A.2.3.2.1 What is a Deployable Asset?

A deployable asset is defined as any physical hardware (e.g. role fit equipment) carried on an exploiting platform which can be deliberately separated from the exploiting platform during a Mission. Deployable assets include external assets which can be deliberately jettisoned but which in normal operation are retained aboard the Exploiting Platform (e.g. sensor pods).

Deployable assets can be of a wide variety of types, including non-PYRAMID based stores and can potentially be interfaced with in multiple ways by the PYRAMID-compliant exploiting platform.

A smart asset is an asset which requires a data interface with the Exploiting Platform.

A dumb asset is an asset which does not have a data interface with the Exploiting Platform, but does have utility and/or mechanical connections.

A.2.3.2.2 Why is Interfacing with Deployable Assets Important?

The ability to integrate with deployable assets, both before and after release from the Exploiting Platform, is critical to being able to carry out a multitude of mission types.

A.2.3.2.3 Aim of this Policy

This policy explains how the PRA has been designed to cater for different kinds of asset that can be deployed from an Exploiting Platform.

The PYRAMID KURs which are applicable to this policy are:

- **Exploitable**: This policy allows an Exploiting Programme to integrate any kind of deployable asset onto an Exploiting Platform, regardless of the asset's function or capability.
- **Utility Across A Range Of Missions**: The ability to easily integrate different types of deployable asset allows an Exploiting Platform to be utilised for multiple operational contexts.

A.2.3.2.4 Exclusions to this Policy

This policy only covers the types of deployable asset and the different types of connection required between a deployable asset and an Exploiting Platform.

It is not intended to illustrate the activities required for fitting or deployment of the asset (for example as part of an attack task) or the decision making supporting deployment. Refer to the following IVs for an outline of these:

- Role Fit Discovery IV
- Plan For A/S Engagement IV
- Releasing IV
- Jettison Management IV

This policy does not cover the process of establishing and maintaining communications channels in detail. Refer to the following policies and IVs for further information:

- Use of Communications policy
- Network Initialisation IV
- Connection Management IV
- Data Transfer IV
- Tactical Exchange IV
- Link Selection IV

A.2.3.3 Overview

The policy for Interfacing with Deployable Assets outlines the basic types of interaction between a PYRAMID-compliant Exploiting Platform and various types of deployable asset (such as a weapon or a sensor), and the components that an Exploiter would need to consider in order to implement these interaction types.

In all cases, connections with a deployable asset are managed as connections with a piece of equipment, with data connections being mapped to/from components as required on receipt/transmission (see the Interaction with Equipment policy for further details).

The interactions with a deployable asset will depend on whether or not the asset is PYRAMID based, and what level of intelligence the asset encompasses.

A.2.3.4 Types of Connections

The following potential types of connection exist between an Exploiting Platform and deployable assets:

- **Mechanical**: This covers physical attachment of the deployable asset (e.g. clamps and lugs) in addition to providing mechanical impulse to support safe separation. This is managed by the Release Effecting component.
- **Utilities (non-Data)**: Deployable assets, or their interfaces, may require or provide utilities, such as power (electrical or hydraulic), cooling or fuel. These are treated like any other utility. For example, a power sink is managed by the Power component.
- **Signal (non-Data)**: Discrete signals may be required for control or interrogation (for example a Mated Connection Indication or Release Consent).
- **Data**: Smart assets require transfer of data to and from the deployable asset (e.g. to update plans or settings such as the impact settings of a smart fuse or the target location), or supporting built in test to report readiness prior to deployment. Data transfer mechanisms could include electrical, optical or RF. This will be managed by the Semantic Translation, Information Brokerage, Data Distribution and Communicator components (see the Use of Communications policy and the Data Transfer and Tactical Exchange IVs for more information). The requirements on the interfacing protocol will depend on the physical constraints and operating environment. For example, a protocol designed for direct, close proximity operation in a safety critical airborne system (e.g. MIL-STD-1760 Ref. [15]) would be needed for assets on-board an exploiting platform, whilst a protocol designed for remote, long range operation in air (e.g. TacNet) would be needed for assets after deployment.

When a deployable asset has been attached to one of the stations of an Exploiting Platform, the platform needs to know the location, type, and configuration/status of the asset. This is true for mechanical, signal and/or data type connections. This is covered further by the Role Fit Discovery IV.

A.2.3.5 PRA-Based Deployable Assets

PYRAMID-based assets are those built from PRA components and will use PRA policies to interact with other parts of the PYRAMID system. Whilst no legacy PYRAMID-based assets exist, it is assumed that many external pods and stores specifically developed for a PYRAMID-based Exploiting Platform would themselves be designed to be PYRAMID-compliant. For example, a deployable UAV carried on an Exploiting Platform station as a reconnaissance asset, or communications or targeting relay might well be a PYRAMID-based Exploiting Platform itself.

On board an Exploiting Platform

Whilst connected to an Exploiting Platform, a PYRAMID-based asset that is intended to be deployed to support mission goals will exist as a node in the PYRAMID-based system. The components within the deployable asset can interact with those on the host platform node. Assets that aren't intended to be deployed but can be jettisoned if required (e.g. sensor pods) can be thought of as PYRAMID-based pieces of equipment (see the Interaction with Equipment policy) with the caveat that there would be some differences with regards to initial detection and disconnection.

After deployment

After deployment, if the deployable asset supports remote node interaction the asset will continue to exist as a node in a PYRAMID-based system although it may use alternative interaction mechanisms.

In some cases, node-node connections may only be established after the asset has been deployed.

A.2.3.6 Non PRA-Based Smart Deployable Assets

Smart deployable assets are those which require a data interface with the Exploiting Platform, but which are not themselves PYRAMID-based. Examples of this type of asset are currently most guided bombs and long-range missiles, as well as external sensor and electronic countermeasures pods.

On board an Exploiting Platform

Whilst connected to an Exploiting Platform, smart deployable assets will need a data connection that is compliant with a communications protocol supported by the asset (for example MIL-STD-1760 Ref. [15]). They will also need the ability to identify themselves to the Exploiting Platform to support role fit discovery. This connection will require the PRA:

- To support translation between PRA compatible services and the messaging format used by the deployable asset.
- To support the required transmission technologies, whether it being electrical, optical, RF or other signals in accordance with this format to communicate with the asset.
- To support the required communications protocols to route message traffic between the asset and components within the PRA.

After deployment

After deployment, if the deployable asset supports in-flight communication (for example a Network-Enabled Weapon) it will continue to communicate with the Exploiting Platform via a protocol. Note that the protocol used to communicate via Semantic Translation, Information Brokerage and Data Distribution may change after deployment. For example, MIL-STD-1760 could be used whilst on-board the Exploiting Platform and TacNet whilst in flight.

As with PYRAMID-based assets, in some scenarios connections between the host platform and the asset may only be established after the asset has been deployed.

A.2.3.7 Dumb Assets

Dumb assets are those which do not have a data interface with the Exploiting Platform. Any interface with these assets is purely utility based, e.g. electricity or fuel. Examples of this type of asset are free-fall bombs and external tanks.

On board an Exploiting Platform

No direct data communication is possible with a dumb asset although mechanical, electrical and utility connections may exist. The Exploiting Platform may be able to sense the presence of a store on a station even if it cannot interrogate it. Where data is needed concerning a dumb asset (such as the mass and aerodynamic properties of a free-fall bomb to support ballistic calculations) these are held as data within a deployment of the PRA in appropriate components such as Stores Release, Mass and Balance or Release Effecting (see the Data Driving policy for more information).

Any unusual behaviour exhibited by a dumb asset may still be identifiable by an instance of the Anomaly Detection component on an Exploiting Platform, and reported accordingly (see the Health Management policy).

After deployment

After deployment, no direct communication is possible with a dumb asset.

A.2.3.8 Nested Assets

Dumb deployable assets may be held within a larger asset which itself is smart and may be PYRAMID-based. Examples of this are countermeasure flares within a defensive aids pod, or unguided weapons within a smart carrier (such as a bomb store carrier, launcher, rocket pod or gun pod). In this case, the Exploiting Platform would communicate with the smart carrier as described for smart and PYRAMID-based assets above, which would in turn interact with the asset.

It is also possible for smart assets to be deployed from smart carriers, the communications and control of which would be Exploiting Programme dependent.

Deployable assets may also be nested within smart carriers that are not in themselves deployable, but interact with the host platform as a fitted piece of equipment.

A.2.3.9 Typical Component Usage

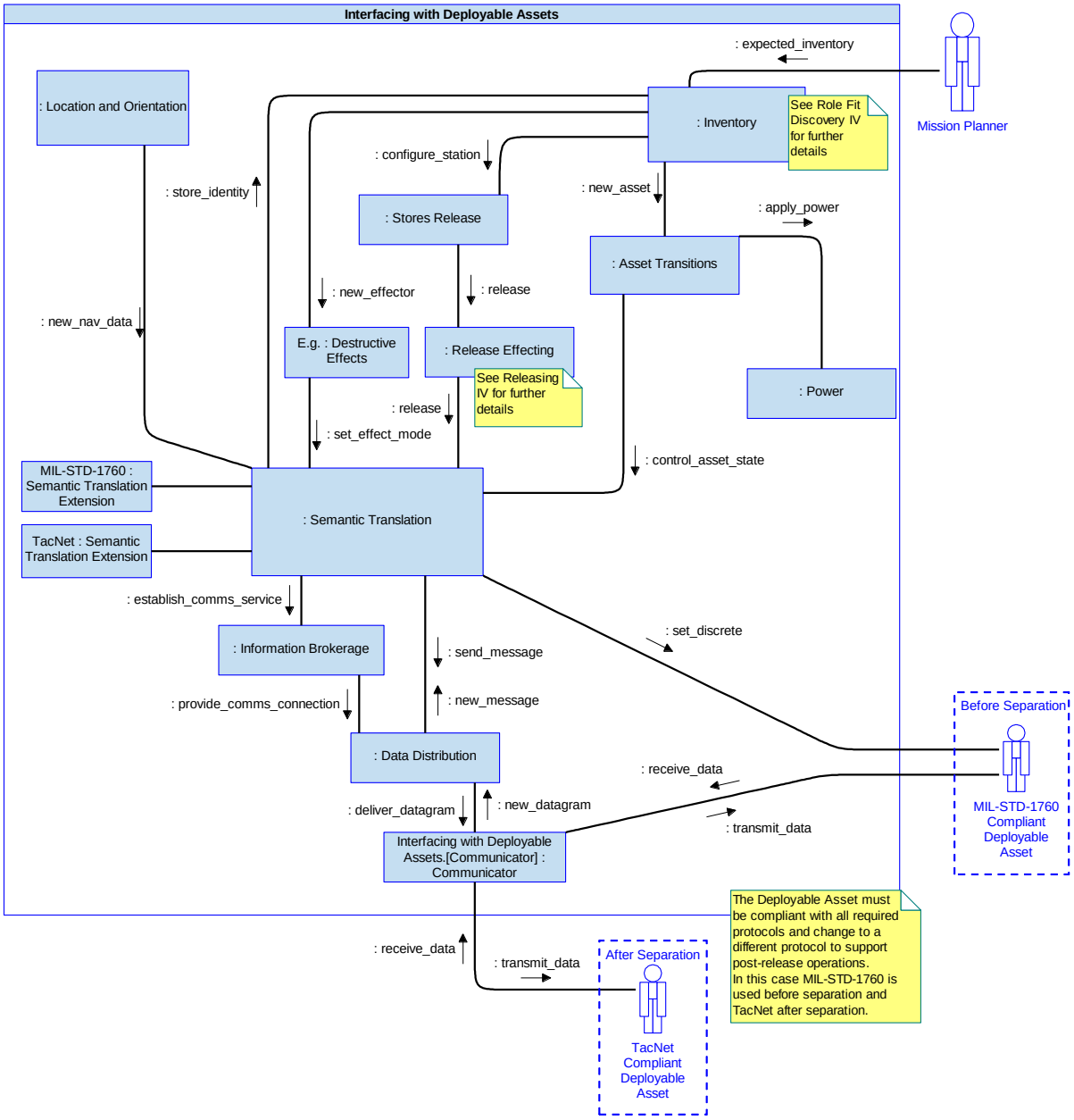

Figure 81: Interfacing with Deployable Assets Example

Figure 81: Interfacing with Deployable Assets Example shows an example of a PYRAMID-based Exploiting Platform interfacing with a smart deployable asset, utilising components pertinent to the functionality required. The diagram focuses on the deployable asset's interaction with the PRA's comms components. In this instance, the Semantic Translation component has been extended with both MIL-STD-1760 and TacNet extensions to cater for the protocol requirements both before and after separation, with the Information Brokerage, Data Distribution and Communicator components setting up the appropriate communication channels.

Note: This example can be applicable to both PYRAMID-based and non PYRAMID-based deployable assets.

A.2.4 Tactical Information

A.2.4.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Control Architecture
- Component Extensions
- Data Driving

A.2.4.2 Introduction & Scope

This Tactical Information policy explains how the PRA supports the handling of sensor data and tactical information.

A.2.4.2.1 What is Tactical Information?

Tactical information in the context of the PRA is defined as information about objects in the battlespace. An object in this context is an entity in the outside world which has relevance to the mission; examples include people, vehicles and buildings. This definition of tactical information also includes information about an object's relationship to the wider battlespace. These relationships are either between objects within the battlespace (such as an object threatening another), or between an object and other measureable parameters of the battlespace (such as an object reacting to a change in the electromagnetic spectrum).

A.2.4.2.2 Why is Tactical Information Important?

Tactical Information is important due to it being a perception of the world that will be used to inform many decisions made by the system or an operator.

A.2.4.2.3 Aim of this Policy

This policy explains how the PRA has been designed to allow flexibility and optimisation with regards to its handling of tactical information.

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** This policy shows how components can be configured to meet the requirements for specific scenarios or Exploiting Programmes.
- **Exploitable:** This policy shows how a single component can be utilised to achieve different goals depending on the requirements of the Exploiting Programme.
- **Flight Certifiable:** This policy provides the ability for Exploiting Platforms to be more easily certifiable by separating the control of sensor data handling and the sensor data handling itself.
- **Incorporate Future Growth:** This policy shows how sensor data handling elements of the architecture can be extended to support many different processing algorithms.

A.2.4.3 Overview

This policy describes how the PRA satisfies a number of PYRAMID KURs through the management of tactical information and the handling of sensor data to derive tactical information. It can be summarised with three key principles:

- The separation of sensor data handling from its high-level coordination and control activities.
- The separation of sensor data handling from the tactical information derived from it.
- The definition of tactical information components to be configurable in terms of the objects they reason about for a given Exploiting Programme.

A.2.4.4 Tactical Information Components

Tactical information is split across a number of components. While they are each responsible for a particular element of tactical information, separating these elements into different components allows a variation in the complexity of the tactical understanding different Exploiting Programmes have. Hence any component not owning any tactical information cannot be explicitly defined as a tactical information component.

Tactical information components exist as service components in line with the Control Architecture policy. The set of tactical information components is as follows:

- Tactical Objects.
- Threats.
- Trajectory Prediction.
- Data Fusion.
- Geography.
- Observability.
- Signature.
- Susceptibility.
- Vehicle Performance.

The key design consideration about tactical information components is that within the PRA they are defined to be configurable in terms of the objects they reason about.

Figure 82: Threat Assessment Deployment Example and Figure 83: Sensing Deployment Example illustrate two different operational concepts utilising the same component to satisfy their requirements. In one case the Observability component is used to provide information used in the determination of threats and in the other to support the planning of a sensing task.

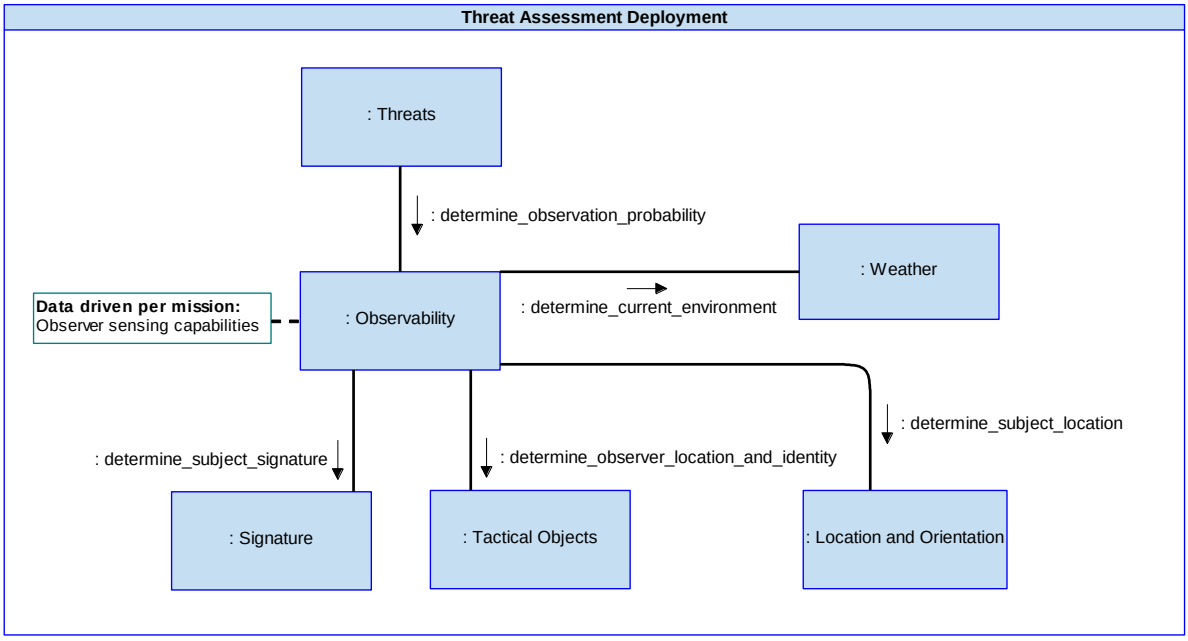

Figure 82: Threat Assessment Deployment Example

In Figure 82: Threat Assessment Deployment Example, Threats tasks Observability to determine the probability of being observed by an object which it has deemed to be a potential threat. Observability then uses the information determined by other components to assess the observability probability which is continually reported to Threats.

In this operational concept, Observability has been configured with operational use data (see the Data Driving policy) that includes the sensing capability of expected observers. Using this data in conjunction with information provided by the other components illustrated on Figure 82: Threat Assessment Deployment Example, the Exploiting Platform's observability can be determined.

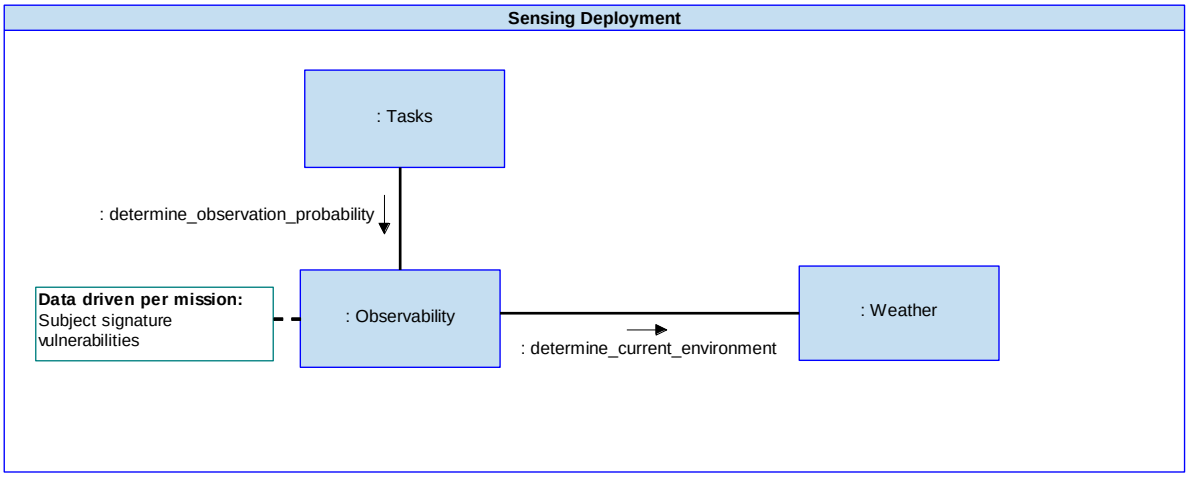

Figure 83: Sensing Deployment Example

In Figure 83: Sensing Deployment Example, Tasks requests that Observability determines the probability of observing an object type within a given region using a given capability. Observability then uses information determined by Weather to assess the probability of the object type being observed.

The difference in this example from the previous example is that the Observability component has been configured with the signature vulnerabilities of subjects, rather than the sensing capabilities of observers. This is because the subjects are external platforms, so knowledge about their signature vulnerabilities is not

determined dynamically by the deployment. The Tasks component provides the approximate location and assumed object type of the subject, as well as the observer sensing capabilities as part of its tasking to the Observability component.

These two examples show that driving a particular component with different data at different times, either through interactions with other components or data configurations, allows that component to be utilised for a range of deployments, supporting the Exploitable PYRAMID KUR.

A.2.4.5 Separation of Data Handling and Control

The handling of raw sensor data is limited to Sensor Products. This component is intended to be developed in a way which allows its capability to be expanded through new and updated algorithmic processes.

An example of this would be updating an image processing algorithm that identifies particular features within an image. The suggested technique to achieve this is extensions (see the Component Extensions policy). Whilst not mandated, application of this technique is shown in the examples throughout this policy.

For Exploiting Platforms where variable control of sensor data handling is required, Sensor Products is controlled by the Sensor Data Interpretation component. For more detail see the Sensor Data Interpretation IV. In this IV a requirement to deliver interpretations is placed upon Sensor Data Interpretation which could have many possible solutions; Sensor Data Interpretation would select the most appropriate solution and enact the relevant control to deliver the solution. Other platforms may not have the need to include variable control in this way. It may be that a fixed solution is always required and is continually being implemented. Sensor Data Interpretation may also coordinate Sensor Products and Data Fusion to extract meaning from stored sensor data, the interpretation does not need to be done in real time when the data is acquired.

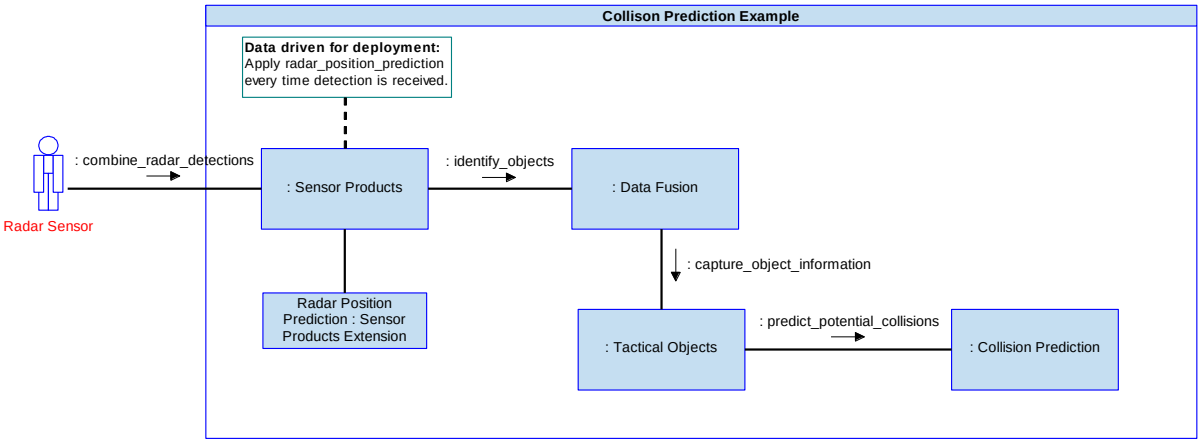

Figure 84: Collision Prediction Example

Figure 84: Collision Prediction Example illustrates a particular use of Sensor Products to support a collision prediction activity. In this particular example Sensor Products is extended with a Radar Position Prediction extension allowing for the combination of non-parametric data associated with multiple radar detections from a single sensor. Collision Prediction is able to react appropriately once it has been supported with any accompanying parametric information via Data Fusion within the same context.

In this example, the PYRAMID deployment has been configured such that this process is applied every time a radar detection is received. These detections are provided by an external Radar Sensor equipment. However, this configuration is not mandated by the PRA and other solutions may be more appropriate.

Separating Sensor Data Interpretation from Sensor Products, Data Fusion and Tactical Objects creates flexibility when architecting deployments which require variable control of processing, while still enabling

Exploiting Platforms which are of higher criticality to be more easily certified. This supports the Flight Certifiable and Exploitable PYRAMID KURs. It also allows the architecture to be extended to increase capability in terms of new processing algorithms, while still allowing flexible control of the processing algorithms to be applied during operation, supporting the Incorporate Future Growth PYRAMID KUR.

A.2.4.6 Separation of Tactical Information from Data Handling

Sensor data handling in Sensor Products has been separated from components that understand and reason about tactical information, with Data Fusion responsible for interpretation of the evidence provided by Sensor Products, and Tactical Objects responsible for maintaining detailed knowledge of objects including their relationships and behaviours.

This approach results in the decoupling of sensor data handling knowledge, such as the lineage of fused sensor data (performed by the Sensor Products component) from an understanding about objects within the battlespace. The fused sensor data could be derived from a multitude of interconnected data sets, comprised of waveforms or images for instance. This is a key separation as it allows the implementation of multiple sensor data handling approaches in a given deployment while still maintaining a single source of truth for the systems view of the battlespace. Data Fusion in the diagram below is operating on the characterisations made by Sensor Products and does not process sensor data. It assesses the sensor product characterisations in a tactical context and supplies objects to Tactical Objects to support the separation of sensor processing from tactical object processing. Data Fusion may be extended to support different fusion algorithms used to interpret the evidence provided by Sensor Products.

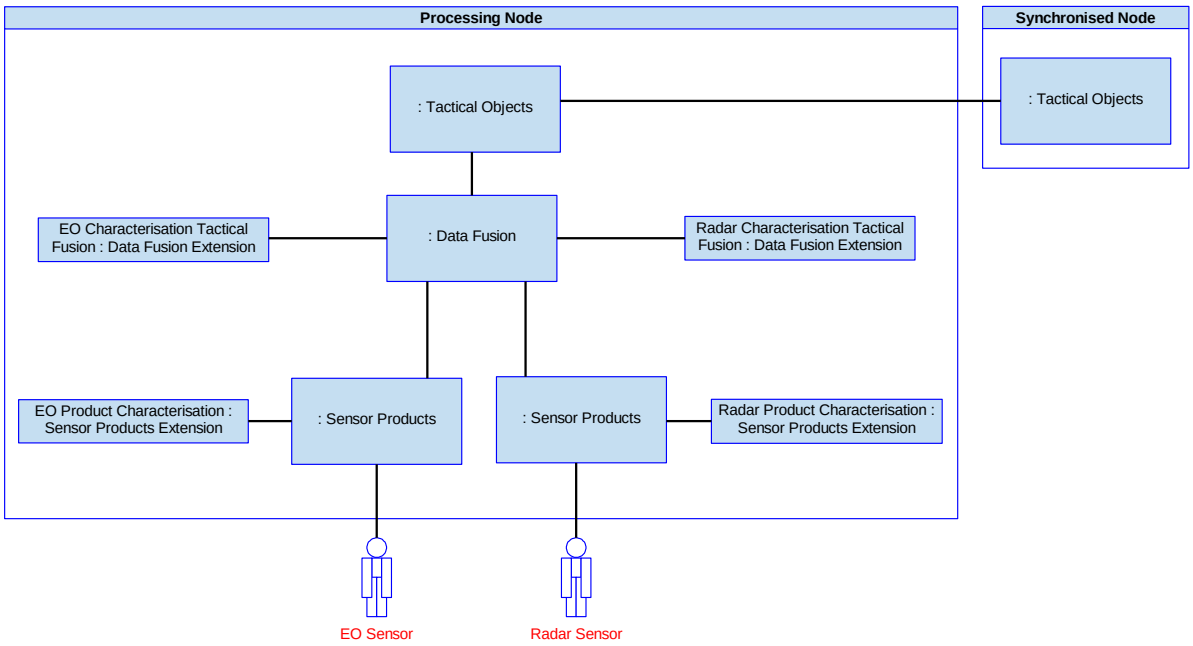

Figure 85: Radar and EO System Example

Figure 85: Radar and EO System Example illustrates a deployment of the PRA with two nodes: a processing node, which is responsible for sensor data handling to maintain a view of objects in the battlespace; and a synchronised node, which is synchronised to the processing node view of the battlespace, but doesn't do any sensor data handling.

The synchronised node only contains a Tactical Objects component. This simplifies this deployment as it only contains a view of objects in the battlespace, rather than a view of the objects and the lineage of their associated sensor characterisations.

The processing node also contains a Tactical Objects component, but this node is responsible for handling sensor data, so it has been deployed with sensor data handling components (two instances of Sensor Products) and an evidence interpretation component (Data Fusion).

The EO processing and Radar processing elements of this could be reused on any Exploiting Programme that utilises those sensor data types. This should reduce the cost of deployments of the PRA, as it allows reuse of an integrated set of components.

Designing the sensor data handling and evidence interpretation components to be extended allows them to be tailored to a particular deployment, whilst separating them from knowledge of object relationships and behaviour allows nodes within a deployment to just consider tactical information, supporting the Exploitable PYRAMID KUR.

A.2.5 Test

A.2.5.1 Pre-Reading and Related Policies

Prerequisite

- Health Management

Read in conjunction with

- Capability Assessment
- Interaction with Equipment
- Resource Management

A.2.5.2 Introduction and Scope

This policy explains how the ability to support testing has been provided during the development of the PRA. It describes what is meant by testing, and how the components support different types of test at various levels of system capability.

A.2.5.2.1 What is a Test, or Testing?

In this policy, Test means a procedure or method intended to establish the state, performance and capability of something.

Testing supports the identification of the current capability of a system in line with the Capability Assessment policy by determining the capability of the hardware or equipment that support the system capability. Although testing can be triggered at any level, the actual test is carried out on a resource. Testing can be used to force the determination of possible capability limits prior to use, or to achieve awareness of a loss of capability or anomaly. Testing may also be instigated in response to the need to effectively determine likely causes of a recognised loss of capability or an anomaly; this is described more broadly in the Health Management policy.

A.2.5.2.2 Why is Testing Important?

This policy is important for explaining how a system can use testing to determine its capability.

Where testing reduces the ability of the system to fulfil other capabilities, the policy shows how testing can be coordinated with other activities.

The contribution of testing to demonstrating continued system integrity and availability is discussed in the Security Guidance for PYRAMID Exploiters, Ref. [60].

A.2.5.2.3 Aim of this Policy

This policy explains how the PRA has been designed to support testing in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Flight Certifiable:** The ability to test during maintenance supports certification. This policy means that exploiting platforms can have the required testing capability.
- **Security Accreditable:** The ability to test during maintenance and during missions supports security accreditation.
- **Utility Across A Range Of Missions:** In order for deployments of the PRA to be used in different types of mission, it is necessary to determine and prove that the limits of system capability meet the requirements of the mission; application of this policy enables such determination.

Additional Considerations

As well as supporting the PYRAMID KURs there was an additional requirement which was considered operationally desirable for the PRA to support:

- **Supportable:** The reliability and maintainability of systems developed using the PRA contribute to their supportability. This policy allows for diagnostic testing during maintenance, as well as the testing of physical elements during a mission.

A.2.5.2.4 Scope of this Policy

This policy covers testing for capability and function; it excludes:

- Testing during software development (verification of requirements).
- Testing during system integration.
- Test harnesses and facilities (e.g. rigs).
- Flight test (including associated ground tests).
- Validation of the successful loading of software.
- Breakdown/build testing (including upgrades).

A.2.5.3 Overview

Testing can be undertaken at any level of a system's hierarchy or abstraction. No single component is responsible for all aspects of testing. Each component is responsible for testing its own area of capability, following a standardised approach; this allows the component's status to be determined independently.

However, the Test component manages and coordinates testing:

- Across multiple components (e.g. at system level)
- In interfacing equipment or infrastructure (including some built in tests) where those tests reduce system capability, consume additional resources (e.g. electrical power) or may pose safety/security risks.

A.2.5.4 Testing of Component Capability

Each component is responsible for assessing its own capability. However, components depend on each other for their capability, or, in the case of Resource layer components, on the resources they control. Components can request better (more specific, or more certain) information about the capabilities they depend on and, for Resource layer components, that may be achieved by triggering a test of the resource. Equipment can also self-trigger testing (e.g. as part of the start-up procedure). The need for information can come from any level of the system, but the test to provide the information is carried out on the equipment.

A component that represents an equipment's function understands how to obtain information about that function, which could include running a Built in Test (BIT). Therefore, BIT, as a test mechanism integral to equipment, will mainly be initiated by Resource layer components, (see the Interaction with Equipment policy), or managed internally to the equipment (for example, running a Power-Up BIT (PBIT) as part of the start-up procedure). This does not exclude the use of BIT as part of a prompted or coordinated test resulting from an Action layer component requesting information; in such a case the initiating component does not know how the required test is being achieved, i.e. it does not know if it is done using equipment BIT or another method.

If a test will interfere with another use of a resource, Resource Brokerage will identify the conflict and the components will decide how to resolve the conflict in favour of the test or otherwise. A test request must be rejected if it would cause the exceedance of a limit.

A.2.5.5 Management of Testing

The Test component knows which tests will provide which information and will manage testing across multiple components (e.g. at system level). This allows a component to support systematic tests without requiring additional effort, test specific software or an understanding of mode of use. Test will initiate (trigger) the different tests or the different aspects of a test, in line with the requirements of the test procedure.

A.2.5.6 Example: Performing an IBIT

Figure 86: Performing IBIT shows how components involved in a test might interact.

Health Assessment has identified the need for additional information to diagnose more precisely the effects on capability and physical assets.

The Test component identifies the tests that will provide this additional information and executes them. The tests need to be coordinated on several resources; such synchronisation may be required to minimise system non-operational time, or to sequence tests to ensure that the effect of one test does not unintentionally impact the results of another.

The tests run by the resource components may be invoked internally or by invoking BIT on an item of connected equipment. Resource Brokerage ensures that tests do not conflict with other activities, in line with the Resource Management policy.

The results of the testing are reported and acted upon as required. Results may also take the form of health data which is used by Anomaly Detection and Health Assessment, as described in the Health Management policy.

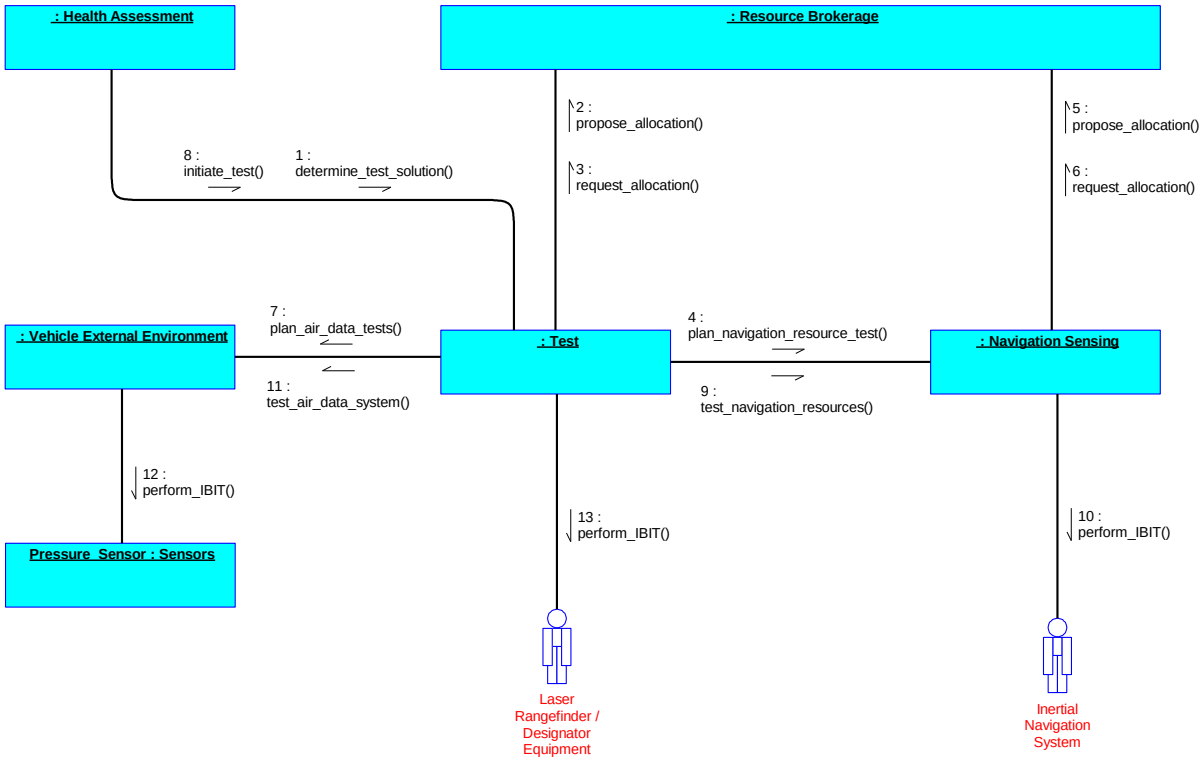

Figure 86: Performing IBIT

A.2.5.7 Example: Equipment Health Test Result Interpretation

Figure 87: Equipment Health Test Result Interpretation shows how multiple components can interpret or process the results of an equipment test. This example shows one way that the policy could be applied. It involves external equipment whose various aspects are managed through multiple resource components, in line with the principles described in the Interaction with Equipment policy.

A test requestor (e.g. the Test component or operator) initiates requirements to perform IBIT on the Sensor Equipment. Resource Brokerage (not shown) has allocated the resource for the test. Sensor Equipment performs an IBIT and generates an error code which is picked up by Anomaly Detection. Anomaly Detection confirms that the error code is related to an anomaly and informs Sensors, Mechanical Positioning and Health Assessment.

Mechanical Positioning reads the error code and assumes its capability is unchanged because the error code is not related to its capability.

Sensors reads the error code and recognises it as related to its capability. Health Assessment analyses the error code along with other health data and determines the cause of the failure (e.g. the lens of an IR sensor is blocked and is not able to provide the required focus). This gives Sensors more details on its affected capability and allows it to take more specific actions.

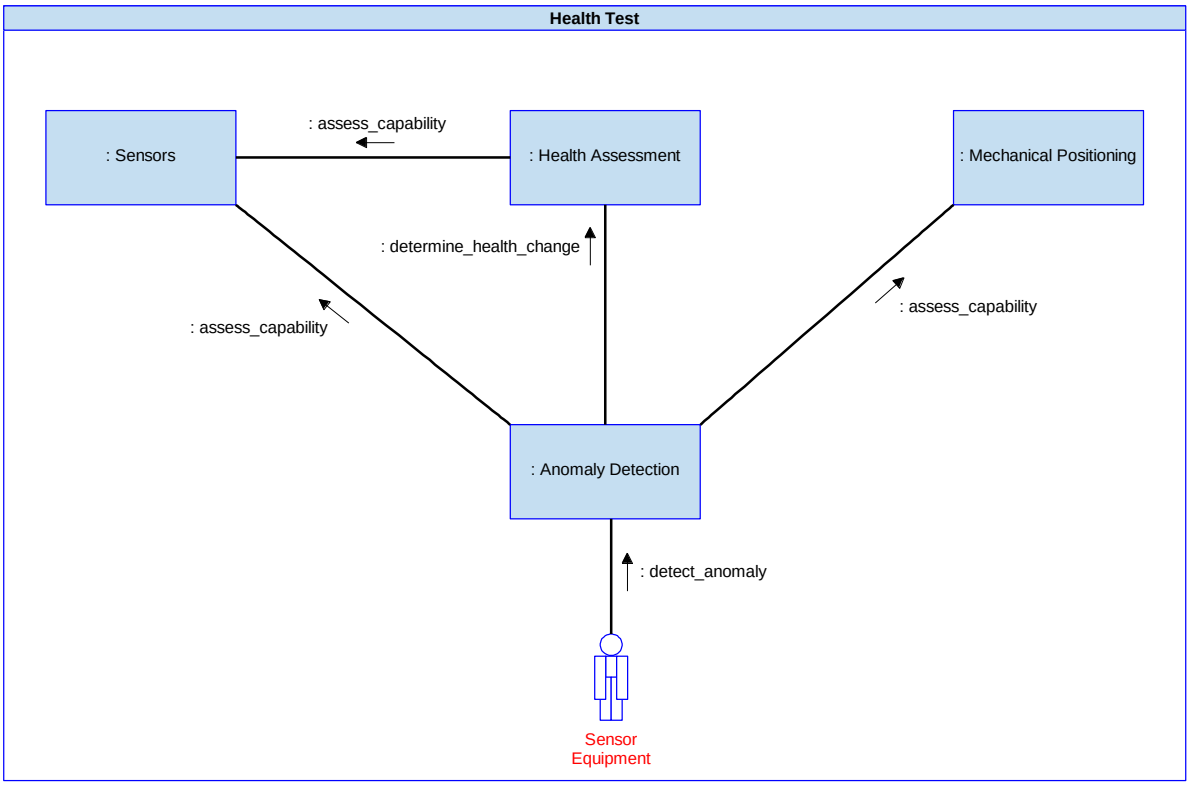

Figure 87: Equipment Health Test Result Interpretation

A.2.6 Use of Communications

A.2.6.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

None

A.2.6.2 Introduction and Scope

This policy explains how communications capability may be used by components within the PRA.

A.2.6.2.1 What is this Policy?

This policy is intended to illustrate how components with differing levels of communications awareness interact with components and systems, on the same or different platforms, both directly and by using communication infrastructure. It is not intended to mandate the underlying communications architecture, specific protocol, or formatting approach which may be adopted, which is to be determined by the Exploiting Programme.

Components are by default agnostic of other components and systems' use, intent and location. They can send and receive signals but in most cases do not know where the signals come from or go to, or the route via which those signals travel. Where necessary, the routing of these signals will make use of communications capability, however by default the originating and destination components will not be aware they are doing so.

The subsequent sections of the policy list the different levels of communications awareness that may exist in components, and outlines the impact on those components' responsibilities. Other than components forming part of a communications capability, the need for components to be communications aware will be deployment specific.

A.2.6.2.2 Why is the Use of Communications Important?

This policy is important because it defines how communications between a component and other components or systems which are on the same or different platforms are managed.

A.2.6.2.3 Aim of this Policy

This policy explains how the PRA has been designed to support the use of communication in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Configurable, Exploitable:** The use of communications policy highlights the ability of the components to communicate with other components or systems which are on the same or different hardware platforms.
- **Incorporate Future Growth:** As components are by default communications agnostic (not aware of where the signals originate or go, or the route via which those signals travel), the PRA allows for a simple and efficient introduction of new capabilities.
- **Scalable:** As components are by default communications agnostic (not aware of where the signals originate or go, or the route via which those signals travel), the PRA allows for systems of varying complexity and number of components to be designed.
- **Utility Across A Range Of Missions:** The use of communication is the foundation to achieve any operational scenarios.

A.2.6.2.4 Exclusions to this Policy

This policy only covers the use of communications between components.

It is not intended to illustrate the activities required for transfer of data, cryptography, management of established communications channels, or real-time communication between humans. Refer to the following interaction views for an outline of these:

- Data Transfer IV
- Cryptographic Management IV
- Connection Management IV
- Network Initialisation IV
- Human Communications IV

This policy does not cover the process of managing multiple nodes or responding to loss of communications. Refer to the following policies for further information:

- Multi-Vehicle Coordination policy
- Health Management policy
- Capability Assessment policy

This policy does not cover the security of communications, some information on this is available in the Security Guidance for PYRAMID Exploiters, Ref. [60].

A.2.6.3 Overview

This policy explains how components with differing levels of communication awareness can communicate and how this communication is managed. Components can send and receive signals but the level of information they have on where the signal goes or the route the signal takes determines the components communication awareness. Components can be

- Communication Agnostic: These components have no information on where the signal goes or the route that a signal takes.
- Communication Aware: These components are aware that communications message traffic needs to be exchanged, for example those involved in the delivery of data.

A.2.6.4 Levels of 'Communications Awareness' within Components

The term communication is used within this policy to mean the process by which a signal is transmitted from a node, within a system, to another node or system. The level of communication awareness can be determined based on the role that component has in the signal being transmitted.

A.2.6.4.1 Communications Agnostic

Components are not aware of each other's location, or of the requirements of the connection between them (see the Component Connections policy). If communications are required to connect two components, the required communications are invisible to the origin and destination components.

A.2.6.4.2 Communications Aware

Whilst components are communications agnostic by default, in some cases a component may be aware that a component with which message traffic needs to be exchanged (either by itself or by another component whose activity it is supporting or managing) requires a communication capability to access it. The component will be able to specify a destination identity to the communications capability, but will not understand how that connection is achieved.

Note that the destination component need not be aware it is being accessed by a component with which it is not physically co-located, and that a component being communications aware does not necessarily mandate that the component understands the full requirements of the connection.

Communications aware components are those that:

- Are aware that an activity being carried out requires communications and/or can identify a remote recipient.
- May ask for communications services to be set up.
- May need information on the state (including availability) of communications service.

Action components, and the Tasks component, are likely to be aware that an activity requires connecting physically remote components, whilst components which are peripherally involved in communications activities, such as an instance of the Resource Brokerage component responsible for allocating the bandwidth of a given communications connection, are also likely to be aware of the requirement for communications.

In some cases, communications aware components may need to understand (and even mandate in detail) the requirements of the connection. This will often be the case with components directly involved in communications activities, which hold responsibility for establishing and monitoring communications services, or for formatting messages, such as Data Distribution. The Data Transfer IV illustrates an example of this.

A.2.6.5 Communications Capability Components

Communications capability components are those components which form part of the communications capability itself, whose primary collective responsibility is to provide communications for other components. Communications capability principally consists of the communication infrastructure and data delivery over that infrastructure. Infrastructure management is provided by Networks (logical) and Communication Links (physical), along with supporting resource components such as Network Routes and Communicator. Similarly the delivery of data is managed by the Information Brokerage component along with the supporting components such as Data Distribution. Note that these components will themselves also be users of communications, in order to communicate and agree communications capability.

An indicative outline of the components forming the communications capability, and the role each plays, is illustrated in Figure 88: Communications Capability. Note that it is not exhaustive to all communications activity (for example, as illustrated in the Human Communications IV, a direct human interaction through communications would draw in the HMI Dialogue and Information Presentation components).

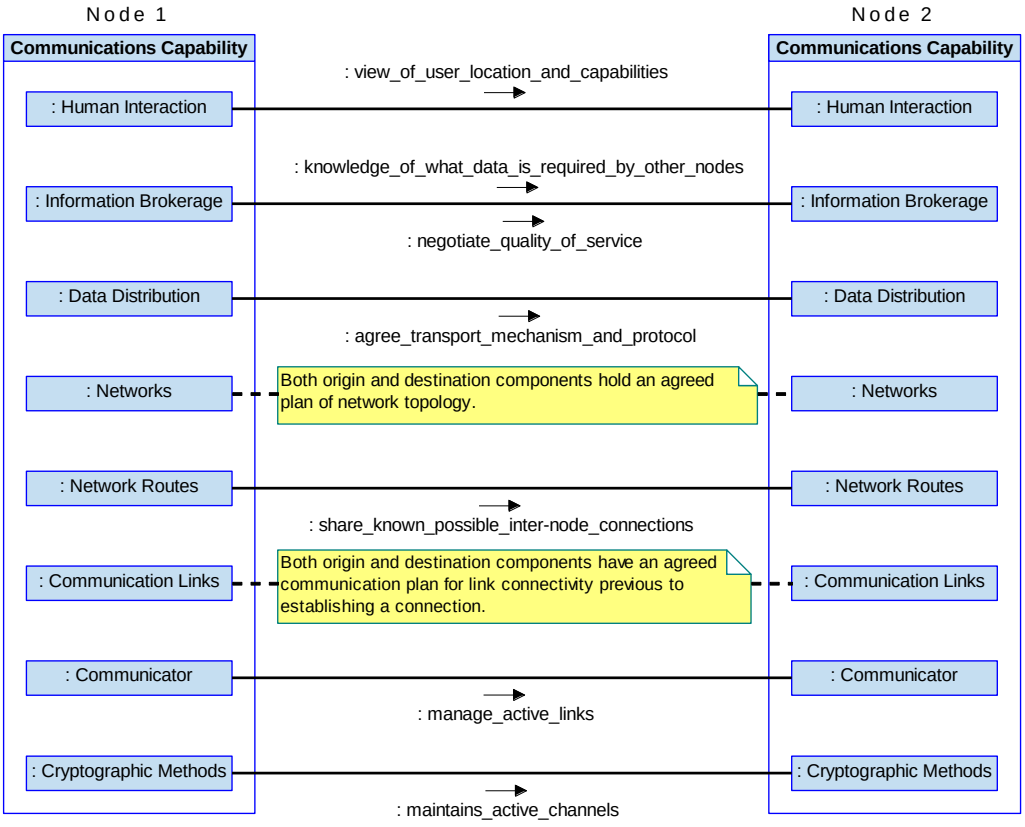

Figure 88: Communications Capability

Note that in practice, a communications flow would pass down the ladder in the originating communications capability before passing back up in the destination communications capability, in accordance with a common plan agreed between the originator and recipient.

The communications capability is responsible for providing communications to other components, by carrying out activities covering the planning, operation and maintenance of communications links.

A.2.6.5.1 Communications Planning

Where an extant communications link is available, the communications capability will be provided by the Data Distribution component. Where it is identified that the Data Distribution component cannot currently meet a communications requirement, a new communications link will be established in accordance with the Network Initialisation IV.

A.2.6.5.2 Communications Operation

See the Data Transfer IV for an example of communications capability in operation.

A.2.6.5.3 Communications Maintenance

The ongoing performance monitoring, evaluation and reconfiguration of communications resources is managed by the Action layer communications components, in accordance with the Capability Assessment and Resource Management policies, and the responsibilities of the respective communications capability components (see the Data Transfer IV for an example).

A.2.7 Data Exchange

A.2.7.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

- Component Extensions

A.2.7.2 Introduction and Scope

This policy explains the exchange of data and information and the interactions of components to provide distribution facilities within a deployment of the PRA. The policy will address how a deployment may facilitate the distribution of data and information, and the support that the PRA provides in achieving this.

It is expected that any deployment of the PRA will require data and information to be delivered to and received from a number of sources using different communications protocols. This policy focuses on the methods that enable systems to communicate with each other when one or more is PYRAMID compliant.

Interactions with a user are excluded from in this policy: see the Human-Machine Interface policy.

The specific methods of delivery of data are not discussed as the exchange of data is independent of the methods, which may include externally provided mechanisms.

As stated in the Storage policy, any component that creates a piece of data will be responsible for its storage and retrieval from storage. Throughout this policy extensions will be discussed and so the Component Extensions policy applies.

A.2.7.2.1 Definition of Data and Information

Within this policy, the terms data and information are distinguished as defined in the following sections.

A.2.7.2.1.1 Definition of Data

Within this policy the term "data" refers to a collection of values (that are being stored or transferred) without connotation to an understanding of the meaning of those values. For example, the term data refers to items that are being distributed, but the distribution mechanism is unaware of the content or use of that data.

A.2.7.2.1.2 Definition of Information

"Information" is digitised knowledge (generated, received, manipulated or stored in the system) that has been derived from data.

A.2.7.2.2 What is Exchange?

Exchange is the method by which data or information is passed between a deployment of the PRA and another system.

It consists of two elements: distribution and receipt. The process of distributing data may include the approval for transfer prior to transmission. For the process of receipt, when data is received by the system there may be additional checks for approval prior to delivery to components for exploitation. Data and information exchanges can only take place between a system capable of distribution and a system capable of receipt.

A.2.7.2.3 Why Is Exchange Important?

The exchange of data and information is important as it could affect the fulfilment of the mission objectives and enables interoperability. There are multiple ways exchanges between systems can be achieved and it is important that the data being distributed (or received) is allowable and will not cause implications for task fulfilment.

At the point of exchange, it may be possible to mitigate for data which is untimely or from a low trust source.

The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about security of data exchange.

A.2.7.2.4 Aim of this Policy

This policy explains how the PRA has been designed for use in data exchange, in a manner that supports achievement of the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Security Accreditable**: Applying understanding to data exchange activities to apply security and data integrity handling rules to important information.

This policy indirectly supports the following PYRAMID KURs:

- **Flight Certifiable**: This policy supports the exchange of mission data required for certification.
- **Incorporate Future Growth, Scalable, Configurable**: Having a central data exchange management component means that limited changes are required when adding or modifying a component to a system or the PRA as a whole.
- **Utility Across A Range Of Missions:** The ability to exchange data in a controlled and structured manner enables flexibility of system use.

A.2.7.3 Overview

The policy outlines how the Information Brokerage, Semantic Translation and Data Distribution components interact in order to provide the distribution and receipt of data. These component interactions are illustrated with examples of Voice over Internet Protocol (VoIP), Data Distribution Service (DDS) and an exchange with a STANAG 4559 information system.

A.2.7.4 Distribution

This section focuses on how data and information is sent from a system. For how information is received and exploited by a system see the Receipt section of this policy.

Distribution could be done via different communications protocols which will follow an adaptation of a standard pattern, see the Distribution Variations section of this policy.

The standard pattern includes three components, Information Brokerage, Semantic Translation and Data Distribution, as illustrated in Figure 89: Standard Pattern.

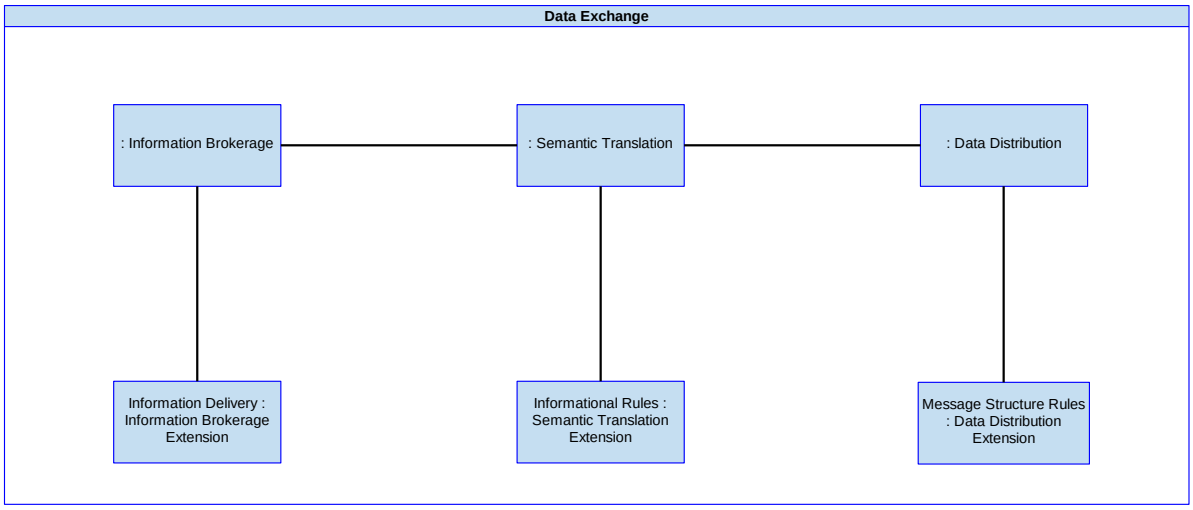

Figure 89: Standard Pattern

Information Brokerage understands what information, produced by the system, has been requested by other systems and tries to fulfil these requests. It will perform checks to ensure the information is allowed to be distributed to the requesting system and identify the need for any reformatting of type, e.g. JPEG to PNG and reclassification. This reformatting will not be done by Information Brokerage but the component which understands the type of data (typically the originating component). Information Brokerage will not handle any information itself. Once Information Brokerage has identified a piece of information is allowed to be distributed it will indicate this.

Semantic Translation understands the needs of exchange participants for a specific piece of information, i.e. informational rules required to be adhered to in the distribution of data.

The informational rules that require adhering to may be captured in extensions. Examples of these can be seen in the Distribution Variations section of this policy. Once Semantic Translation has applied the correct informational rules for the distribution it will pass the appropriate data for delivery.

Data Distribution knows that a piece of data requires distribution to a recipient via a specific distribution method. Data Distribution understands the methods of distribution and how to prepare data for the distribution. The functionality of preparing the data may be performed by extensions which understand the message structure rules for that distribution method, examples can be seen in the Distribution Variations section of this policy. Once the data has been prepared for distribution Data Distribution will initiate the distribution.

For further representations of this template, see the Tactical Exchange IV, the Request for Information IV or the Distribution Variations section of this policy.

The following section shows how this pattern may be implemented using specific protocols.

A.2.7.5 Distribution Variations

A.2.7.5.1 Voice over IP

Figure 90: VoIP Voice illustrates how the standard pattern would be used to represent a VoIP voice distribution system.

Voice over Internet Protocol (VoIP) is a specific method for the delivery of voice communications. In this example, the types of extension used are VoIP delivery for the Information Brokerage component, VoIP Informational Rules (codec) for the Semantic Translation component and RDP/SIP Structure Rules for the Data Distribution component.

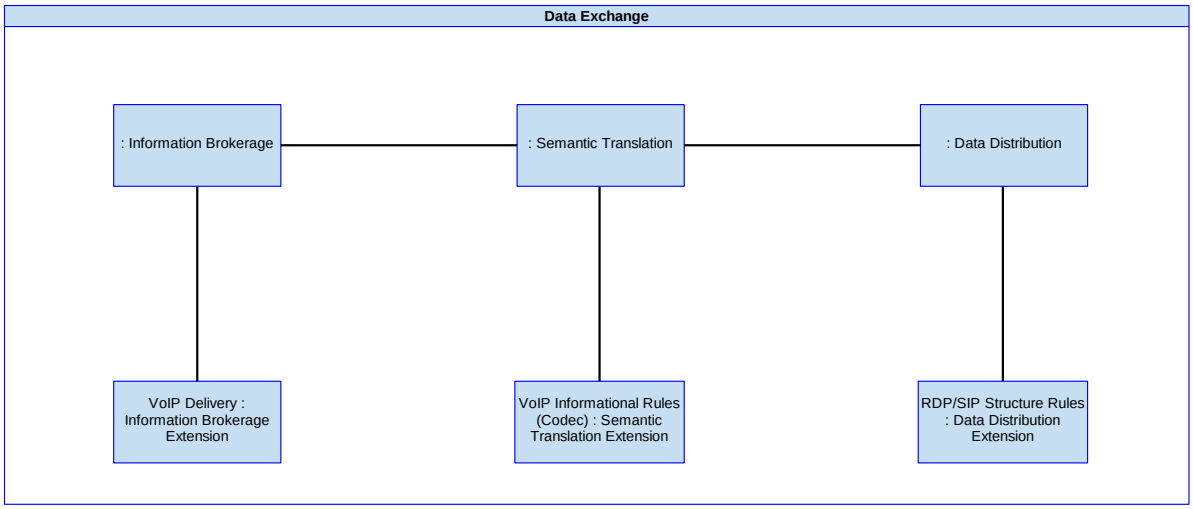

Figure 90: VoIP Voice

A.2.7.5.2 Data Distribution Service

Figure 91: Data Distribution Service (DDS) illustrates how the standard pattern would be used to represent a Data Distribution Service (DDS).

DDS is a system-to-system standard, using a publish-subscribe pattern that allows real-time data exchange. In this example, the Semantic Translation component is not included as it does not perform any functionality. The types of extension used are Information Delivery for the Information Brokerage component and RTPS Structure Rules for the Data Distribution component. A representation of this can be seen in the Request for Information interaction view.

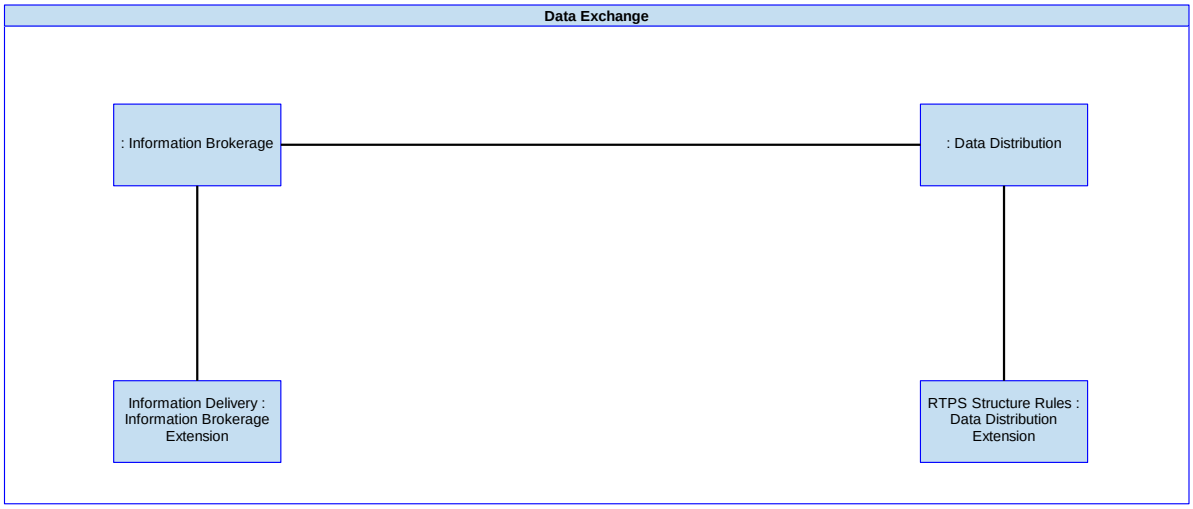

Figure 91: Data Distribution Service (DDS)

A.2.7.5.3 STANAG 4559

Figure 92: STANAG 4559 illustrates how the standard pattern would be used to represent STANAG 4559.

STANAG 4559 is a standardised agreement to promote the interoperability of NATO member states' Intelligence, Surveillance and Reconnaissance (ISR) products managed by product libraries. In this example, the types of extension used are Information Availability for the Information Brokerage component, 4559 Interchange Rules for the Semantic Translation component and Network Structure for the Data Distribution component.

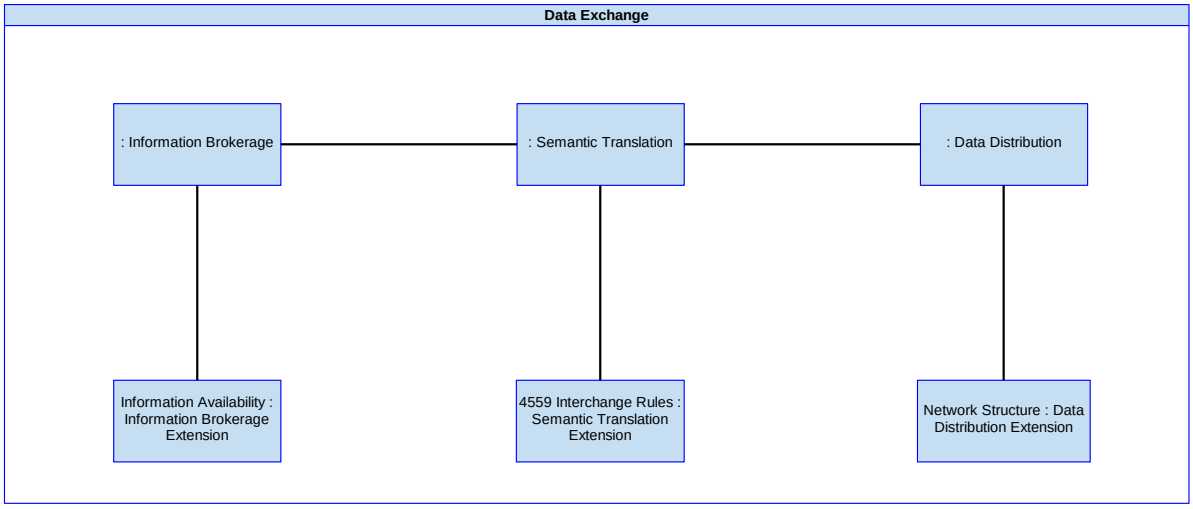

Figure 92: STANAG 4559

A.2.7.6 Receipt

This section focuses on how data and information is received from a system, checked and then sent to the correct component for exploitation. For how a system sends data and information see the Distribution section of the policy.

Similar to Distribution, receipt will follow an adaptation of a standard pattern as illustrated in Figure 89: Standard Pattern. Here the initial stage for receipt is at Data Distribution where the received data is checked for integrity, unpackaged and changed into the message structure for the system by the Message Structure Rules extension. Once it has been formatted into a message structure to be understood by the system, Semantic Translation is informed that a piece of data has been received. Semantic Translation will use its extension Informational Rules to understand what method has been used and how it can be used by the system. If required, Semantic Translation will inform Information Brokerage that a piece of information is available and Information Brokerage will know which component is interested in that piece of information, by using Information Delivery, and therefore be able to ensure the information is exploited within the system.

A further example of this can be found in the Tactical Exchange IV, where a Tactical Datalinks (TDL) Track is received.

A.3 Modelling Principles

Modelling Principles are policies that explain concepts that may be applied to a PRA based deployment in support of the PYRAMID KURs.

A.3.1 Component Connections

A.3.1.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

None

A.3.1.2 Introduction and Scope

This policy explains how component connections can be used between components to produce complex systems.

A.3.1.2.1 What is a Component Connection?

The PRA contains a set of functional components that form the building blocks of a deployment. Component connections define relationships between these components to allow them to work together as a system.

A.3.1.2.2 Why Are Component Connections Important?

Component connections are fundamental to the construction of a system of components. It is important to consider the component connections when describing system behaviour. The way components are connected makes an essential contribution to the overall desired system behaviour, as demonstrated by the examples shown in Appendix C: Interaction Views.

A.3.1.2.3 Aim of this Policy

This policy explains how the PRA has been designed so that it can be used with component connections to support and be consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Configurable** - Component connections can allow for different configurations based on the needs of the system.
- **Flight Certifiable** - Formally modelling the relationship between components can contribute to safety cases.
- **Incorporate Future Growth** - Using system agnostic components and formally modelling the relationship between components allows for easier changing of the components and relationships.
- **Resilience Against Obsolescence** - A clear definition of the relationship between components allows hardware changes to be made and then reintegrated into the system more easily.
- **Security Accreditable** - Formally modelling the relationship between components allows for clearer vulnerability analysis and security case design.

A.3.1.3 Overview

The policy for component connections can be summarised as follows:

- Components are connected together using bridges, which bridge the *semantic gap* between the components.
- Bridges map and translate between component services, but do not contain any functionality that is defined within the scope of a component.
- Counterpart relationships are a common component connection type that captures the relationships between the information of the different components.
- The behavioural dependencies between components can be expressed by specifying these relationships in a component interaction pattern.
- The use of standard service patterns for component connections will ease integration, while still providing a mechanism for predictable, complex and flexible behaviour.

A.3.1.4 Bridges

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, maximising the opportunity for reuse and minimising the impact of changes, easing maintenance overheads, as well as allowing components to be developed in isolation from one another.

Each component represents a distinct subject matter area that is defined in its own language. Consequently, there is no shared interface definition between PRA components. Instead, a deployment can 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 connection that is required by the system's needs.

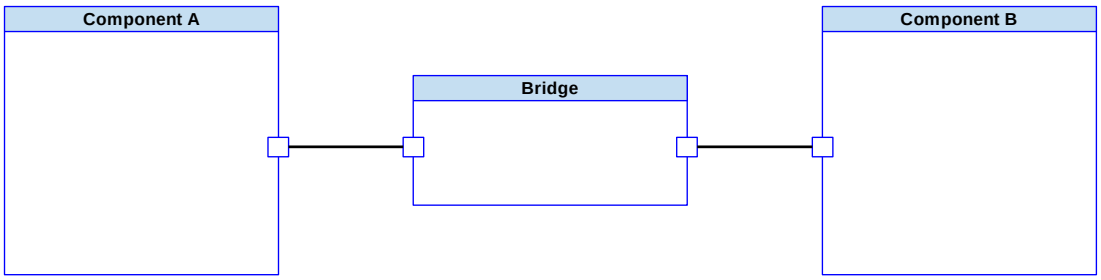

Figure 93: Basic Bridge Diagram

A bridge defines which component services and information are connected to which other component services and information. To achieve this a bridge can perform the following three 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.

Bridges are a means of implementing 'counterparting', described in the Counterparts section. The three functions above are expanded upon in more detail in the Data Relationships and Service Relationships sub-sections, and must be considered and defined specifically for each system design to ensure successful component integration.

Bridges should be kept as simple as possible, but not at the expense of polluting components. Too much complexity in a bridge may indicate that there is a PRA component missing from the design.

Bridges should not contain any functionality that is defined as within the scope of a PRA component.

A.3.1.5 Simple Transactions

The most straightforward of component connections are Simple Transactions where control passes from one component to another, perhaps with some associated data and a simple response. These usually do not require any detailed representation, with the connection often implied from the two service definitions.

There may still be some complexity in the implementation introduced by the chosen messaging protocol. The messaging protocol can have a drastic impact on the service presentation. For example, the same core logic may use a remote procedure call in one deployment and a publish-subscribe process in another. Deciding which connection protocol is appropriate will depend on the deployment.

A.3.1.6 Counterparts

A.3.1.6.1 What is a Counterpart?

A counterpart relationship is created between two entities in different components to show that they represent different views of the same concept in the real world. At a system level, the counterparts may be thought of as being the same thing, but the components consider and act upon them differently.

The two components will each have a service that represents their view of the concept, and these services are connected by a bridge to form the counterpart relationship. Commonly, the relationship will be between a provided service and a consumed service, but a provided-to-provided relationship is also possible, see the Service Relationships sub-section.

A.3.1.6.2 When Should Counterparts Be Used?

Counterpart relationships are to be used when there are two concepts in different components that need to be aligned. For example, a weapon loaded onto a wing is understood in multiple ways by different components in a deployment:

- Inventory sees it as a physical item.
- Target Engagement sees it as a resource type.
- Stores Release sees it as a station ID.
- Destructive Effects sees it as a resource that provides a destructive capability.
- Semantic Translation sees it as an external system type.

A.3.1.6.3 Data Relationships

Figure 94: Data Modelling Relationships Diagram shows two components with a counterpart relationship expressed by a bridge, illustrating the data relationship aspects of counterparts.

The Conceptual_Thing is anything of interest to the system about which information can be known. Such an entity may or may not exist in the physical world - for example, a vehicle that has been detected, or a potential threat that may not materialise. This same Conceptual_Thing may be represented by different data in multiple components, in accordance with their differing roles. This is a design concept, which manifests itself as an element of the component connection (bridging) design.

Other elements shown in the diagram are described and related as follows:

- Information is digitised knowledge generated, received, manipulated or stored in the system.
- Data_Format is the format, units and structure in which Information is held - for example electrical current expressed as a value in Amps that is stored as one element within a larger data structure.
- Data_Understanding is the meaning derived from Information in the context of a mission, a role, or a component - e.g. a current may represent a demand on the electrical power supply.
- Component is a reusable configurable unit with knowledge of a specific Subject_Matter area.
- Subject_Matter is the component specific viewpoint of the information about Conceptual_Things.
- Interface is a point of interaction defined in data terms that allows a component to interact with other components or actors. Interfaces are the common link between this diagram and Figure 95: Service Relationships Diagram
- Bridge is an interconnection enabling a service required by a component to be satisfied by defining how that service maps to other services to satisfy its needs.
- Data_Conversion is the process of translating between Data_Formats used by different components. This process is carried out inside a Bridge.
- Semantic_Conversion mechanises the process of translating the meaning of data from one Subject_Matter area into another. This is another function of the Bridge.
- Counterpart_Relationship is a mapping between packages of data held by different components presented on their respective Interfaces. This mapping is configured in a Bridge, which brings together both Data_Conversion and Semantic_Conversion. For example, converting the quantity of fuel (in litres) that is being used to propel an aircraft into the mass of the fuel (in kilograms), which is relevant to mass and balance calculations.

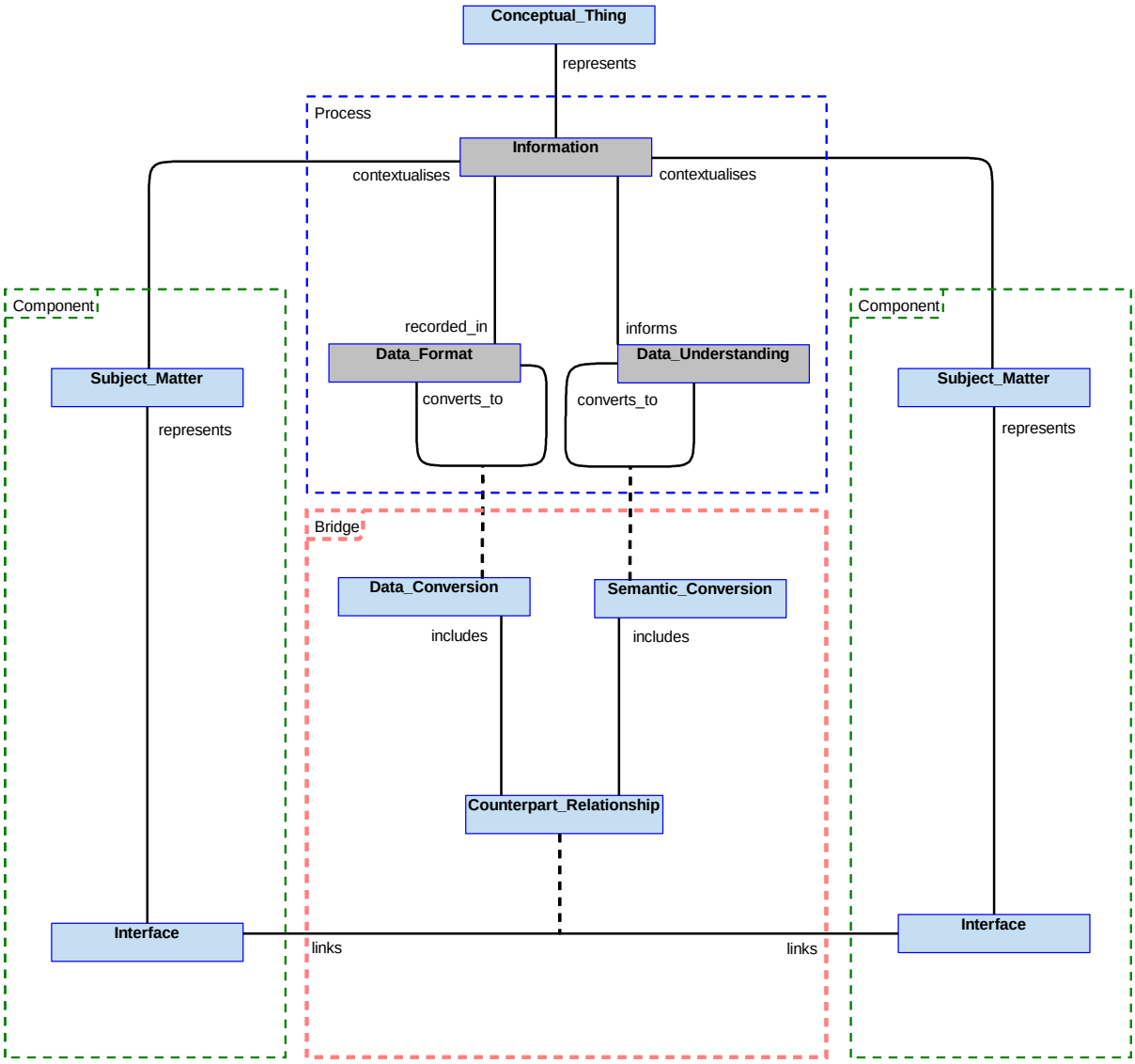

Figure 94: Data Modelling Relationships Diagram

A.3.1.6.4 Service Relationships

The Figure 95: Service Relationships Diagram illustrates the services relationship aspect of counterparts. It shows how a component's service connections may connect to another component's service.

In this view, components have the following elements (note that these are instances - a component is likely to have several of each):

- Service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. For more information, see Ref. [12].
- Activity is a trigger that invokes a functional operation or behaviour carried out by a component.
- Interface (in common with Figure 94: Data Modelling Relationships Diagram) is a point of interaction defined in data terms that allows a component to interact with other components or actors.
- Counterpart_Attribute is an element of Information (see Figure 94: Data Modelling Relationships Diagram) that forms part of an Interface.

Additional elements that are properties of the Bridge are:

- Event_Relationship is a mapping between the timing of behaviours performed by different components. This is a function of the Bridge not covered by the data relationship viewpoint above.
- Counterpart_Relationship is as described above for Figure 94: Data Modelling Relationships Diagram.
- Attribute_Relationship is a mapping between individual Counterpart_Attributes to be kept aligned.

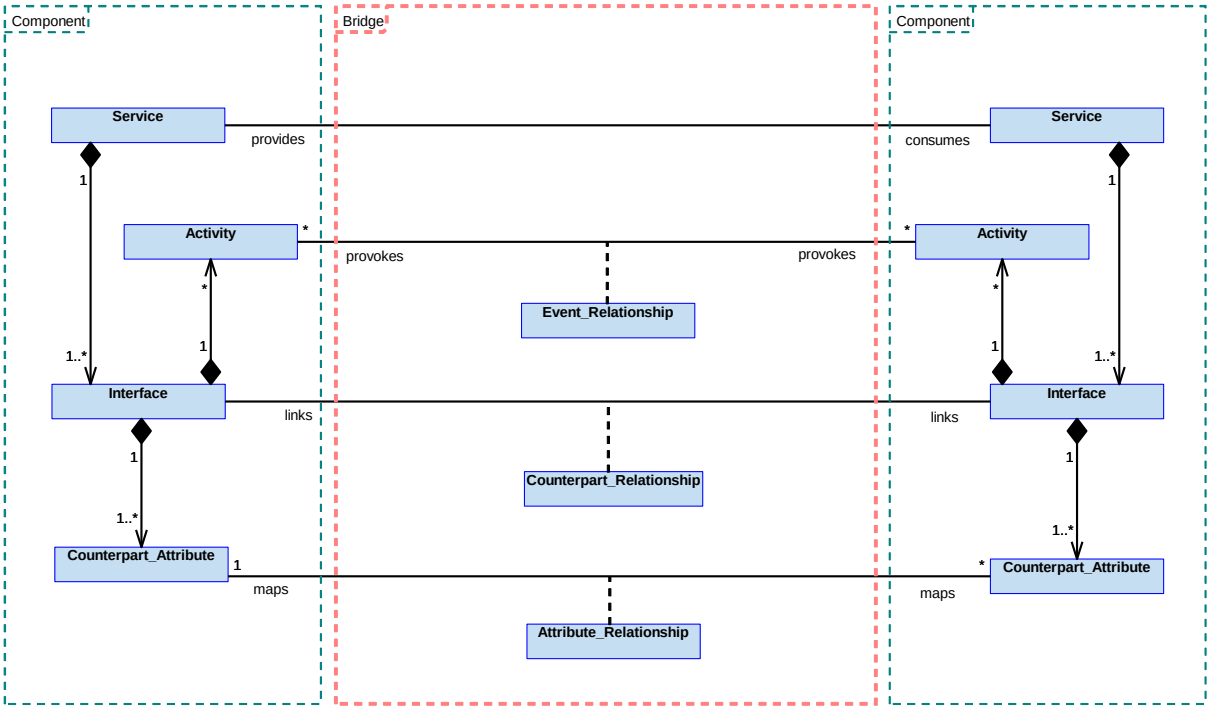

Figure 95: Service Relationships Diagram

A.3.1.6.5 Example of Counterparts

The Figure 96: Counterpart Attribute Mapping Diagram focuses on a simple counterpart example using PRA component services between the Routes component and the Resource Brokerage component. Here we are considering Routes requesting an allocation of fuel from Resource Brokerage. The counterpart relationship is between the Route_Dependency interface of Routes, which knows that it has a dependency on fuel, and the Resource_Request interface of Resource Brokerage, which determines if resource requests can be fulfilled. The response to the request is given by the Achievement interface (not shown in the diagram).

This example shows a simple one-to-one counterpart, although one-to-many counterparts can also be created (e.g. where a consumed service of one component is satisfied by more than one provided service from different components).

The diagram illustrates that it is instances of service classes which are counterparted, and it is the job of the bridge to map attributes appropriately to instances of service classes of other components.

For attribute mapping, this example brings out the following points:

- Not all attributes of the class need to be mapped (e.g. 'predicted_quality' has no relevance to the fuel transaction shown and so this is not mapped by the bridge).
- Attributes on one side of the bridge can be mapped to more than one attribute on the other side (e.g. 'cost' maps to three attributes on the Resource Brokerage).

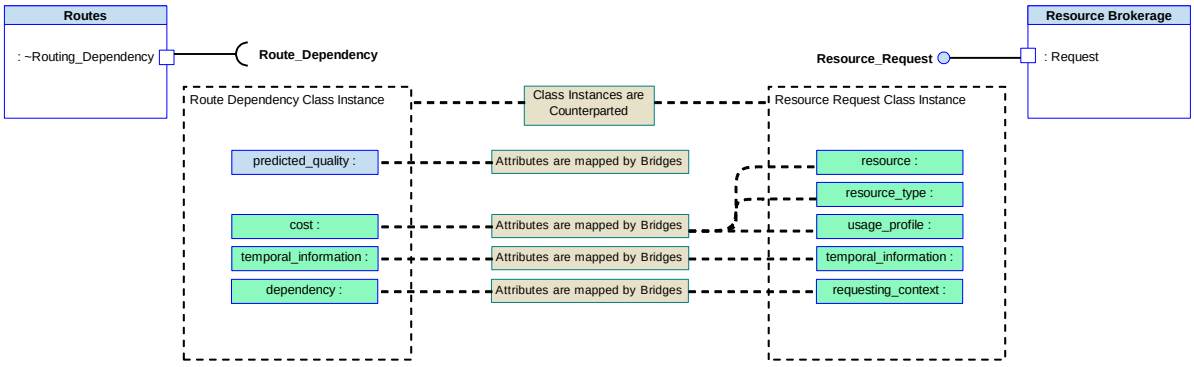

Figure 96: Counterpart Attribute Mapping Diagram

A.3.1.7 Service Contracts

A.3.1.7.1 What is a Service Contract?

Service contracts are used to define relationships between services in order to provide a capability. This allows the components that provide these services to work together in order to complete the task expressed in the service contract.

The full service contract shows both the provided services and consumed services that are used. This includes mandatory services that are required to complete the task along with any optional services. Optional services can be used to allow alternative workflows or to provide supplementary information relating to the activities being performed.

The relationships between the provided services and consumed services can be shown on a service contract diagram, see Figure 97: Service Contract Diagram.

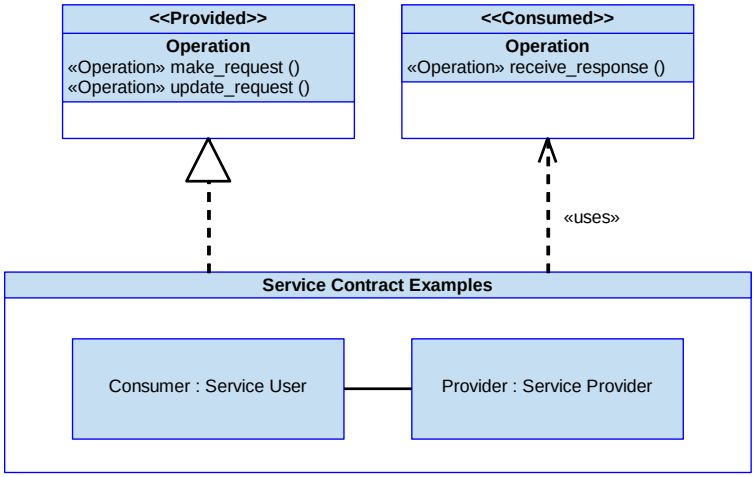

Figure 97: Service Contract Diagram

The provided services are shown as a realisation of what the provider provides, while the consumed services are shown as usage by the consumer.

When defining attributes and behaviour of the services, it is important to remember that the service object only exists for the life of the service. It will remove all information local to the service contract after it is complete. If necessary, additional services can be used to pass this data out to a different attribute or operation outside of the service contract.

The relationships between the provided services and consumed services can be shown on a choreography diagram, see Figure 98: Service Contract Sequence.

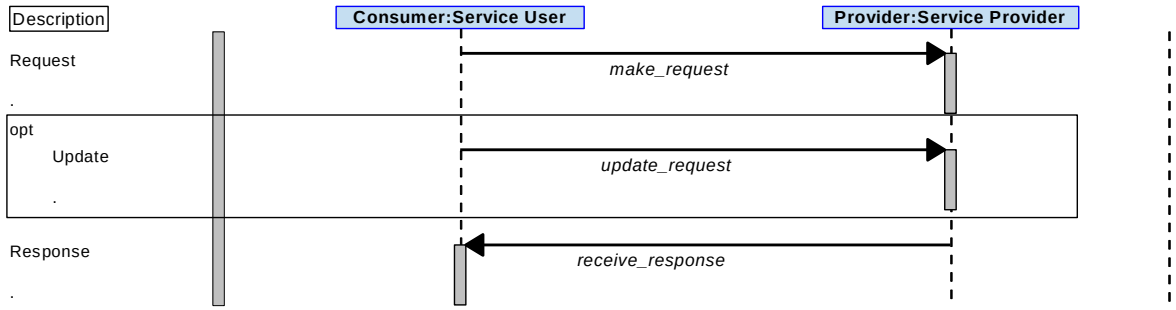

Figure 98: Service Contract Sequence

A service contract can also be defined from the perspective of only one of the components involved, providing a component centric view. This specifies the required pattern of use of either a consumed service or a provided service, where a sequence or order is essential to the correct operation of the service.

Figure 99: Component Centric Service Contract Sequence shows an example service contract sequence diagram for the Path Demands component service, Path_Arbitration. This example is to illustrate the use of a service contract; it should not be interpreted as prescriptive of the structure of a Path Demands service.

The Consumer in this case is shown as a generic component needing to place a demand for the movement of the air vehicle. The diagram captures the sequence of interactions that, in this example, must be undertaken in the defined order. Other examples might define unordered or partially ordered steps. In this respect, the service contract is specifying the flexibility that is required of the provided service.

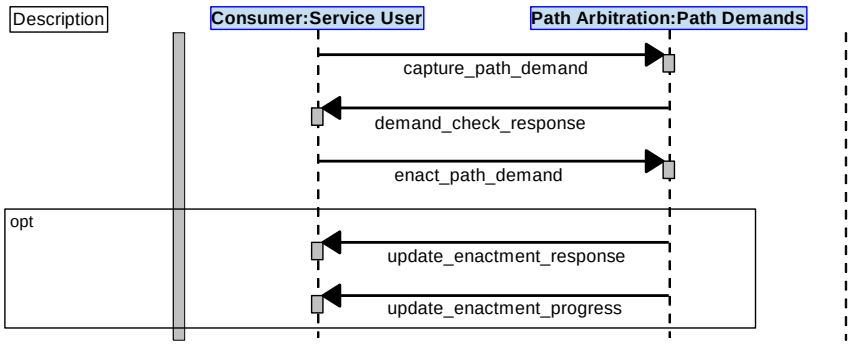

Figure 99: Component Centric Service Contract Sequence

In this example, the Consumer places a requirement to move the air vehicle. Path Demands provides an 'in principle' confirmation that the demand can be enacted as it is not currently inhibited by any constraints. At a subsequent time, the Consumer requests the enactment of the demand. Path Demands confirms enactment and provides subsequent progress updates (e.g. demand completed, or demand interrupted). The enactment response and enactment progress interactions are shown as optional in that a Consumer need not necessarily support these interactions. Not shown in the diagram are the checks that Path Demands performs or the application of arbitration rules, as these are not part of the service contract.

For more information on service contract diagrams and choreography diagrams, refer to the SoaML Standard, Ref. [25].

A.3.1.7.2 Whether to Use a Service Contract

Service contracts are useful in defining complex, multi-component service interactions. If a simple transaction can perform the required information exchange, a service contract is not needed.

When adding services to the service contract, it is important to include only the services relevant to the contract. There may be other actions, which while related, can be carried out independently of the service contract.

The duration or frequency at which a service is requested should also be considered. Even for a small number of interacting services, if the system needs to support concurrent processing of multiple service requests, having a well-defined service contract will allow the system to scale more easily to the requests that are made of it.

A.3.1.8 Component Interaction Patterns

The collaborative behaviour of the services can be further refined by applying a component interaction pattern.

The interaction pattern provides additional rules that govern the operation of the services; it is not a service pattern itself (these are found in the Services sub-section of Component Composition in Appendix B). The logic for controlling the sequence of activities should still be managed by the components. Depending on the type of pattern that is needed, this may mean one component is in control of the others, or it could be that a set of rules are provided indicating how components should communicate their progress.

Component interaction patterns can have the concept of activation. An Activation service detailed in the pattern can be called from an external source known as the Activator. This can be used to provide information into the service contract relating to the service request. Upon activation, the Participant components are selected to complete the service request.

A component interaction pattern may define a specific role for a controlling component (e.g. Coordinator or Orchestrator). The common terms Participant and Subordinate are used across the different patterns. Participants are components that provide or consume services that are required by the service contract. Subordinates are the components that are needed to facilitate the completion of the services required by the pattern.

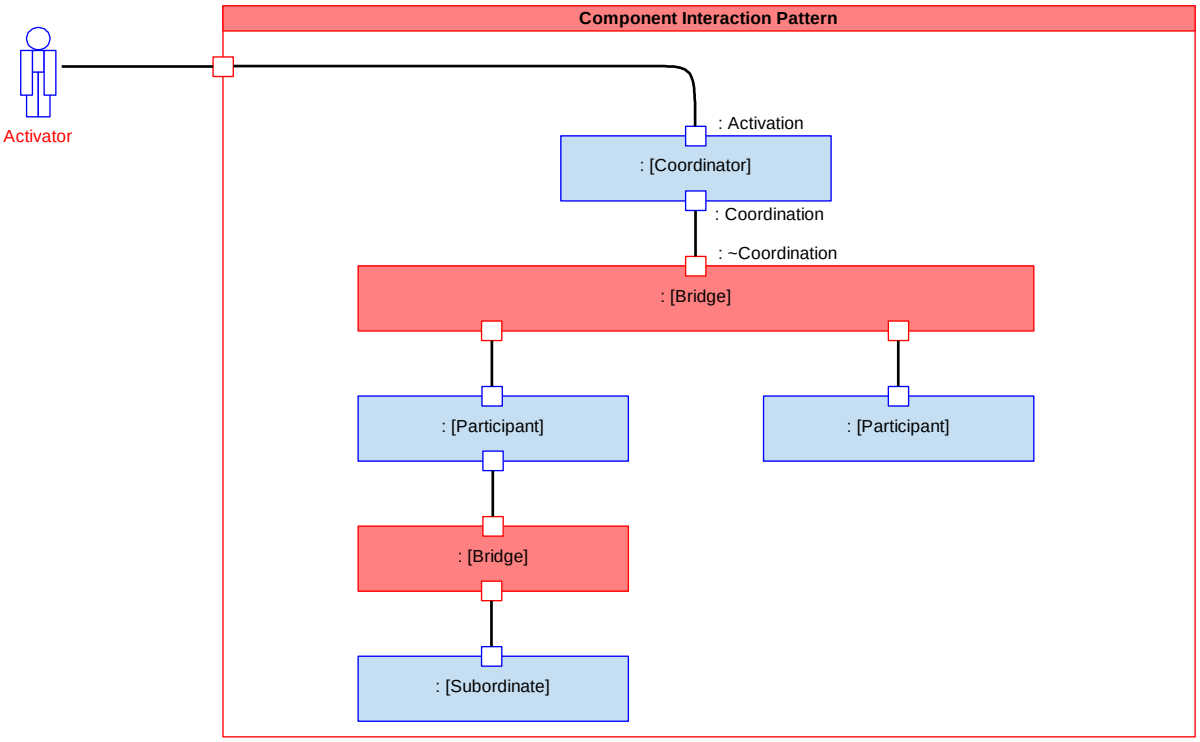

Figure 100: Component Interaction Pattern Roles

Where differing mechanisms of coordination are needed to complete a task, it may be necessary to link multiple patterns together. This is called a compound pattern. Each section of a compound pattern can be treated as if it was one of the individual component interaction patterns.

When defining the pattern, it should:

- Specify the mandatory services that are provided, or consumed.
- Identify the activation service (where applicable).

Additionally, it may:

- Identify optional services.
- Specify quality of service measurements.
- Place constraints on the usage of the services.

The job of a system integrator could become very complex if every component connection behaved in a unique way. The solution is to produce a standard a set of interaction behaviour types.

Simpler transactions are easier to develop and maintain. Using subject matter based components within the PRA reduces the number of connections between components, allowing for fewer component connections. However, one of the side effects of this approach is that there is more likely to be a need to coordinate behaviour between components.

The counterparts allow information to be linked between components, but to coordinate behaviour different patterns are needed. Four common patterns to manage this are singled out by Thomas Erl for special consideration in Ref. [11]. They are summarised below.

- **Atomic Transactions** are the most common coordination type. They are used when actions or data need to be synchronised between components.
- **Business Agreement** is used when the priorities of multiple parts of the system need to be coordinated when coming up with a solution. Requirements are derived by the coordinating component, which places requirements on participating components, often including criteria for measuring success. How a solution is determined and enacted is left up to the participating components.
- **Orchestration** is similar to a Business Agreement, except the orchestrating component owns the process, setting up other components to do different parts or steps of the process.
- **Choreography** is a different type of collaborative behaviour, where partner components contact a choreography service to discover the rules and process for working with other partner components. This is particularly useful in an extended system with numerous peer components across multiple platforms.

The choice to use a component interaction pattern, counterpart or simple transaction will depend on the Exploiting Programme. See PYRAMID Exploiter's Pack Annex B: Deployment Guide, Ref.[3], for examples of the above patterns.

A.3.2 Component Extensions

A.3.2.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

None

A.3.2.2 Introduction and Scope

This policy explains the concept of component extension, how it applies to the PRA and how the use of extensions supports the PYRAMID KURs. In the PRA, extensions can be developed to enhance the capabilities of PRA components. This policy defines what is meant by an extension component, considers their benefits and provides guidance on their appropriate usage.

A.3.2.2.1 What is Extension?

Extension is a design pattern and implementation mechanism used to open up a component in to different parts so that behaviour of one part can be easily extended or specialised, reducing the impact of change. This technique enables behaviour to be factored into a separate component called an extension.

A.3.2.2.2 Why is Extension Important?

The ability to extend components is important as it is a key enabler for multiple KURs and as such, it is important to consider extensibility as part of component development. Extension components allow the PRA to be extended in specific, well-defined ways that increase its adaptability, exploitability, future proofing and usability. This covers a variety of behaviour, including factoring out algorithms and catering for a range of special types. Factoring out behaviour in this way simplifies the component, allowing it to focus on the core behaviour without concern for special cases and allows changes to be restricted to extensions.

A.3.2.2.3 Aim of this Policy

This policy explains how extensions can be applied to the PRA in support of PYRAMID KURs.

The PYRAMID KURs which are applicable to this policy are:

- **Exploitable**: Extension components separate out deployment-specific concerns from general system behaviour, so that the PRA can be tailored for multiple Exploiting Programme needs without modification to core system behaviour.
- **Flight Certifiable and Security Accreditable:** Extension components allow functionality to be separated out. This means it would be easier to certify complex functionality.
- **Incorporate Future Growth:** Extension components allow implementations of the PRA to be adapted in response to evolutionary changes, in order to make use of more and newer devices and to keep pace with industry in adopting best practice, without modification to component definitions and component interactions. As modifications are confined to extension components, this limits the impact of change to the components being extended reducing overheads during development, testing and integration.
- **Scalable**: Extension components allow for the simplification and elimination of resourcing overheads by making it possible to support improved algorithms more suited to the needs and resources available to a given deployment.
- **Utility Across A Range Of Missions**: Extension components enable the PRA to be utilised for multiple operational scenarios by allowing different instances of extension components to support different operational scenarios.

Additional Considerations

The common theme is that by separating out ‘what the component does’ (defined in the parent component) from ‘how the component provides this behaviour’ (defined in the extension component) the cost of change is reduced; as the impact of change is confined to the extension components and has little to no impact on core system interactions. The fact that parent components can be tailored to meet specific needs without being modified greatly increases component reusability.

A.3.2.3 Overview

The policy for component extension can be summarised as follows:

- An extension component is a developed component that separates out or extends the functionality provided by a parent component whilst remaining within its subject matter.
- Any component can have extensions.
- Extensions are subject to the Rules for Extensions.
- The PRA identifies some possible extension components, but does not define any.

A.3.2.4 Extension Definitions

The following definitions apply to extensions:

Parent Component: a developed component that supports the use of extension components. A parent component makes itself extensible by defining one or more extension point(s).

Extension Component: a developed component that separates out or extends the functionality provided by a parent component whilst remaining within its subject matter. An extension component's provided services fulfil the needs of an extension point.

Extension Point: a consumed service that defines the parent's dependency upon an extension for a single purpose.

Extension Set: a set of one or more extension components that fulfil the same extension point.

See Figure 101: Extension Definitions for a visual representation of these definitions.

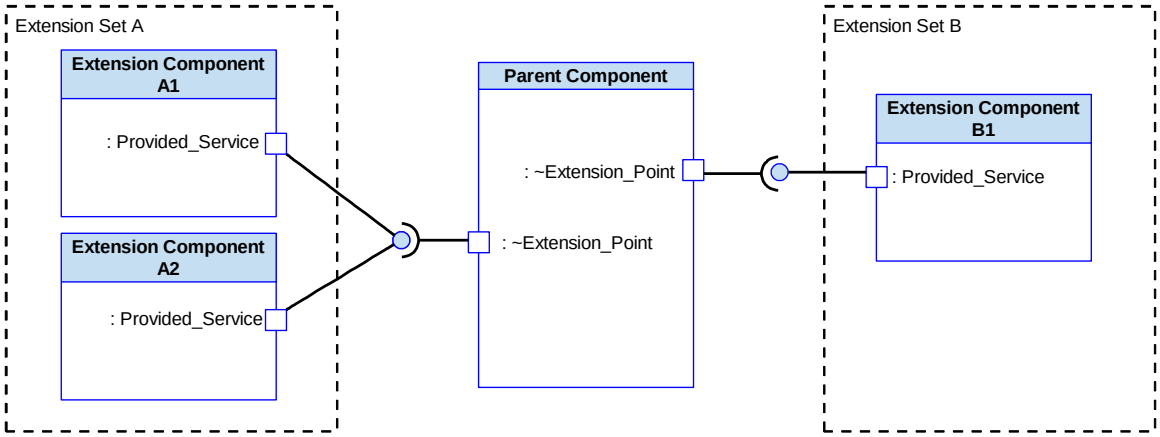

Figure 101: Extension Definitions

A.3.2.5 Rules for Extensions

The rules for component extensions are as follows:

- Any component can have extensions.
- An extension is defined within the subject matter of its parent, so:
  - Its responsibilities can only be a (possibly specialised) subset of the parent's.
  - Its services must conform to (but may specialise) the parent's services (if defined).
- Only the parent component has access to the provided services of an extension.
- Aside from these restrictions, an extension is defined and used in the same way as other PRA components. Therefore:
  - An extension component can consume services from elsewhere as long as they are compatible with the parent's services.
  - Extension components are integrated into the system through the use of bridges (including any interactions with its parent component).
  - Extension components can be data driven directly.
  - Extension components need to be designed, implemented and certified as part of the system.

Figure 102: Rules for Extensions shows how an extension component can consume services from another component, or from its parent. The parent may define additional services (beyond the Extension_Point) to support its extensions.

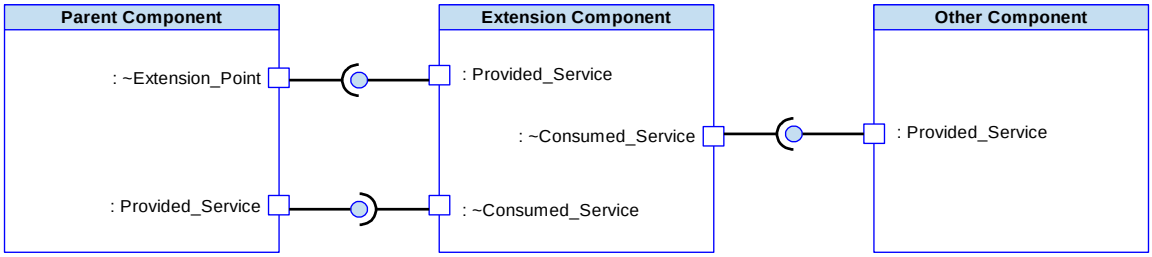

Figure 102: Rules for Extensions

A.3.2.6 When Should Extensions be Used?

The PRA identifies example extensions for the Tasks component, and suggests that others could have extensions. Exploiting Programmes should consider whether or not components would benefit from extensions when analysing component subject matters against their system requirements. It is highly likely that many components will benefit from using extensions.

The following might be reasons for using extensions:

- **Mission requirements:**

  Extensions can be used to support variations on mission requirements. Extensions can be used to capture the variations on the mission e.g. performing a sensing mission or performing an attack mission. In this scenario, the functionality for mission specific elements is performed by the extension component and any core functionality is performed by the parent component.

- **A component has a super-type specialised by sub-types:**

  A component that defines a super-type can delegate the definition of subtypes to extensions, factoring out specialisation and opening up the component to cater for an extensible range of subtypes. This simplifies the parent, minimises the impact of change and makes its capabilities future proof. When new subtypes with special behaviour are discovered, they can be added through extension without change to the parent. For example, a component dealing with hardware would not want to define concrete classes for all devices currently required by the system. It would likely define a super-type of device that captures the common behaviour. The specific devices would be subtypes of the super-type and the subtypes would be defined in one or more extensions.

- **The component's functionality is:**

  - Complex - it may benefit a component to factor out complexity in order to isolate it. This will simplify the component and provide easier maintenance and testing for both the now simplified core logic and the isolated complex logic.
  - Optional - any behaviour that is optional should be factored out so that this behaviour is only present in the system when required.
  - Open to alternative approaches - where alternative algorithms exist, use of extension can allow Exploiting Programmes to choose the algorithms that suit their needs. This enables a system to have a single algorithm where the choice is made at build time, or multiple options chosen at runtime by user selection or constraints. An example of multiple options would be multiple ways of calculating the same route, such as fastest, shortest or most fuel-efficient.

- **A component has a mix of requirements for:**

  - Safety integrity - it may be necessary to separate out behaviour based on safety integrity. For example, a parent component and some extensions may be required to be high integrity, but other extensions could be justified as lower integrity by the Exploiting Programme’s safety analysis.
  - Data security - at times data can be highly sensitive and on a strict need to know basis. A component’s core logic does not need to know the detailed data that is kept in an extension, meaning the definition and access to the data is kept discrete and knowledge of its detail can be removed from any system entirely by excluding the extension.

A.3.3 Data Driving

A.3.3.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

None

A.3.3.2 Introduction and Scope

This policy explains how configuration data could be utilised in a deployment of the PRA to modify component behaviour to cater for different conditions.

A.3.3.2.1 What is Data Driving?

Data Driving is one of the main design strategies for ensuring the system is able to incorporate future growth, is configurable, exploitable, resilient against obsolescence, and has utility across a range of missions. The strategy uses configuration files to drive either the specification or the internal structure of a component allowing components (and thus the system as a whole) to support different Exploiting Programmes (for example, differing vehicle types or hardware setups) and different end-user scenarios (for example, different role fit equipment or operational requirements) without the need for wholesale redesign.

A.3.3.2.2 Why is Data Driving Important?

Data driving enables system behaviour to be tailored for different variants, applications and in order to meet new user operational requirements through the specification of a configuration file rather than re-developing the component.

A.3.3.2.3 Aim of this Policy

This policy explains how the PRA has been designed to support data driving in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs which are applicable to this policy are:

- **Configurable:** This policy allows changes to be made to component behaviour at design or operational time to support vehicles with different capabilities, behaviour, equipment and HMI interfaces.
- **Exploitable:** This policy provides the ability to integrate and utilise the PRA on various platforms by allowing tailoring for vehicles with different capability (including performance characteristics), different (vehicle) equipment and for use in different physical environments.
- **Incorporate Future Growth:** This policy allows changes to the PRA design to be made without the need to update existing parts of the PRA to account for new components.
- **Resilient Against Obsolescence:** This policy allows portability of the system software on to different hardware and operating systems with minimal rework as the Recording and Logging policy includes data-driven elements allowing the PRA to be easily adapted between different software logging requirements.
- **Utility Across A Range Of Missions:** This policy provides the capability to create a system usable for various mission scenarios and organisational structures by allowing components to be used in different configurations to meet the needs of a specific implementation. This is achieved by providing the ability to modify properties associated with specific role fit equipment on a mission-by-mission basis with mission data loads.

A.3.3.3 Overview

This policy describes how the PRA satisfies a number of KURs through the support of data driving. It can be summarised as:

- Data driving can be used to tailor the system's behaviour in order to accommodate different conditions.
- A component with a data-driven specification can accommodate changes to its own behaviour and data.
- A component with a data-driven internal structure can accommodate changes to the data within the component.
- Configuration data can be developed and implemented independently from the component software.

A.3.3.4 When Should Data Driving be Used?

Data driving can be used in any components where significant changes are expected to occur to their area of subject matter as a result of variants or when variations in the way the system performs are needed. As an example, if a new electro-optical tactical sensor was developed and an Exploiting Programme wanted to integrate it into their system, this would necessitate changes to at least two components. The first component would be the component concerned with controlling tactical sensors, as this would require an understanding of new sensor characteristics. If this component had a data-driven internal structure, changes could be made to the data within that component to allow it to use the new sensor. The second component would be the

component concerned with how best to sense a target with available sensors, as this would need its specification configured to take account of a new 'optical' type of sensing capability. Both are required to work together for the system to find a target effectively, with both driven by different sets of data. Similarly, during the operational lifetime of an Exploiting Platform, there may be a need for optimising sensor capability to better support the mission needs or as a result of trials feedback. The use of data-driven components would allow this to occur without changes to any of the components. Another example would be for components that are concerned with weapons and their related effects. As new weapons are developed, their capabilities and control features will change; this will necessitate flexibility in the way this information is represented throughout the system.

It should be noted that use of data driving has implications for testing and certification. This may make its use inappropriate (or in need of careful management) when used in high integrity systems.

Any components for which data driving is an effective strategy for mitigating anticipated changes will have this rationale captured within the Assumptions and Design Considerations sections of their component definition. A Note that no services are defined for data driving (see the Services Not Defined in the PRA section).

A.3.3.5 Types of Data Driving

A.3.3.5.1 Data Driving the Component Specification

A component has services that describe its interfaces and behaviour and allow its effective use within the wider system. With part of this behaviour defined by the data that the component handles, the component design activity should identify the type of data that can be configured. An example would be the provision of data associated with two different weapon types (e.g. AIM-9L and ASRAAM). This method is unconcerned with the internal structure of the component, allowing any design methodology to be used. Services that relate to the loading of subject matter specific information within a component are not included among the services defined in the PRA as it is up to the component developer to decide if that information will be loaded via a data loading service or through other means.

A.3.3.5.2 Data Driving Internal Structure

This approach deals with specifically identifying how the population of the component internal structure will handle the configuration data allowing it to be data-driven. As an example, a class model would be an appropriate way to achieve this as the classes held within the model are treated as data. This data is acted upon by operations and methods, thus giving rise to the behaviour of the component. The class instances could be specified using a configuration file or, if extensions are used, it may be populated in the parent component by an extension component.

A.3.3.6 Configuration Data Development

The data can be specified for different variants in the component development (for example, execution platform, products and operational variants). There are two broad types: deployment time data and operational use data, with a particular component utilising either or both types of data as necessary.

- **Deployment Time Data:** This type of data is developed during the system design with the system qualified and certified within the context of that data set. It could include vehicle specific data (for example engines or vehicle control laws) and engineering authority data (for example system limits), and is defined any time before the component software has been produced.
- **Operational Use Data:** This type of data is developed after designing the system with the data verified separately from clearance of the system. It could include mission specific data (e.g. sensor optimised data) or non-mission specific data (e.g. maps) and is defined any time after the component software has been produced.

A.4 Safety and Security

This section contains the Safety Analysis and Security Approach policies. These have been considered during the development of the components.

A.4.1 Safety Analysis

A.4.1.1 Pre-Reading and Related Policies

Prerequisite

None

Read in conjunction with

None

A.4.1.2 Introduction and Scope

This policy explains what safety analysis means in the context of the development of the PRA. For the PRA there is no safety risk. This is because the Exploiting Programme will be entirely responsible for demonstrating the Exploiting Platform meets the safety targets applicable to the Exploiting Platform. It describes the process that has been followed in assigning an indicative design integrity to components and the context (i.e. the assumptions) under which that process has been carried out. The design integrity of a component determines the level of rigour required during software development in order to reduce the safety risk to an acceptable level. The indicative design integrities are not requirements on an Exploiting Programme.

A.4.1.2.1 What is Safety Analysis?

An artefact is safe if it poses an acceptable risk of harm (to people) and damage (to property). Safety analysis is the examination of an artefact to determine the level of (safety) risk it poses.

Safety analysis is dependent on a definition of the Exploiting Platform and context it operates within. It is not feasible to have a PRA safety analysis that covers all possible exploiting cases. Therefore, the PRA safety analysis has assumed a conservative context, but aims to avoid being driven by cases that are considered extremely unlikely (e.g. non-typical Exploiting Platform types or extreme operating scenarios). This is to improve the quality of the PRA. Therefore:

- The Exploiting Programme will be entirely responsible for demonstrating the Exploiting Platform meets the safety targets applicable to the Exploiting Platform.
- The safety considerations contained in the PRA are not requirements on an Exploiting Programme nor are they expected to contribute to the Exploiting Programme's safety case.

A.4.1.2.2 Why is Safety Analysis Important?

Safety analysis is important because new and modified air systems are required to be safe in order to be certified, meet legal obligations and meet moral expectations.

A.4.1.2.3 Aim of this Policy

This policy explains the process that has been followed in assigning an indicative design integrity to components and the context (i.e. the assumptions) under which that process has been carried out. The PYRAMID KURs are supported by performing the initial safety analysis which is included with each PRA component definition.

The PYRAMID KURs that are applicable to this policy are:

- **Flight Certifiable:** The primary purpose of performing safety analysis on the PRA was to improve the quality of the architecture. The need to separate high integrity functions from low integrity (potentially complex) functions was considered during the domain modelling that produced the PRA components, in the context of a typical military air platform.
- **Exploitable:** The secondary purpose is informing the Exploiting Programme (both system designers and safety analysts) where the PRA expects components to be relied upon from a safety point of view.

NOTE: The conclusions of the PRA safety analysis (e.g. that component X is expected to be high integrity) are guidelines only, based on conservative assumptions as to the context that may apply to a wide variety of Exploiting Platforms and operating scenarios. They do not place requirements on an Exploiting Programme.

A.4.1.3 Overview

The policy for safety analysis can be summarised as follows:

- All components have been assigned an indicative design integrity based on an analysis of (safety) risk within a defined context. These are not requirements on an Exploiting Programme.

A.4.1.4 Integrity Classifications

The indicative design integrities assigned to each component have been classified as Item Development Assurance Levels (IDALs) based on the civil aerospace standards ARP4754A Ref. [38] and ARP4761 Ref. [37]:

- DAL A
- DAL B
- DAL C

Notes:

- An indicative IDAL of C, covers C, D or E as the analysis has not attempted to differentiate between the lower DAL levels as it is judged it would not drive the PRA.
- IDALs are determined during the safety analysis process (e.g. using ARP4754A Ref. [38] and ARP4761 Ref. [37]). The level of rigour required during software development is defined in DO-178C Ref. [23]. DO-178C uses the term 'software level', which is synonymous with IDALs. For software development, IDALs A, B and C correspond to Software Levels A, B and C.
- ARP4754A Ref. [38] and ARP4761 Ref. [37] are acceptable means of compliance for safety analysis on UK MOD projects (Def Stan 00-56 Ref. [46]).
- DO-178C Ref. [23] is an acceptable means of compliance for software development on UK MOD projects (Def Stan 00-55 Ref. [47]).

A.4.1.5 Risk Acceptance Criteria

The risk acceptance criteria and severity classifications used are:

- Catastrophic: Failure conditions that result in the death of one or more people. Catastrophic outcomes are considered DAL A within the PRA safety considerations.
- 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.
- 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.

Notes:

- Where the failure of components can cause weapons to impact targets not intended by the crew, resulting in unintended harm to third parties, it is assumed this would drive an IDAL of B. This assumption was at the direction of UK MOD. The consideration of other accidents, such as impacting the host air vehicle during release, may drive a more onerous DAL.
- These risk acceptance criteria are assumptions on what would be appropriate for a UK military air system. An Exploiting Programme would agree the risk acceptance criteria for their project with the purchaser (e.g. UK MOD).

A.4.1.6 Safety Analysis Context

The context assumed for the safety analysis is:

- The air vehicle is large (>5700kg) (based on the Defence Standard 00-970 Part 9 Ref. [16] definition).
- The air vehicle may be manned.
- The air vehicle may be unmanned.
- The air vehicle may overfly populated areas.

The assessment of individual components has included additional component specific context in the rationale where necessary.

A.4.1.7 Determination of the Indicative IDAL

The components have been assigned an "indicative IDAL" of A, B or C based on the risk acceptance criteria and context defined above. The indicative IDAL and supporting rationale for each component is documented in the "Safety Considerations" section of each PRA component definition.

- The indicative IDALs reflect the likely criticality of a particular component and so take credit for external independent events or failures of other components.
- IDAL assignment has been largely based on engineering judgement and experience. The rationales are as concise as possible.
- IDAL assignment has been informed by the systematic safety analysis previously performed on the TIKAL programme. However, the safety consideration rationales are self-contained and so do not reference previous TIKAL safety analysis.
- IDAL assignment has focused on the most severe safety impact of the component (i.e. captures the most severe Functional DAL for the component). Other, non-driving, cases which may apply to some air vehicles or operational roles are generally not discussed.

A.4.2 Security Approach

A.4.2.1 Pre-Reading and Related Policies

Prerequisite

None.

Read in conjunction with

- Recording and Logging
- Cyber Defence
- Safety Analysis

A.4.2.2 Introduction and Scope

This policy explains what is meant by the PRA security approach.

PYRAMID is intended to be incorporated on a range of platforms, with different infrastructures and security environments, therefore the security approach is designed to support Exploiting Programmes with the accreditation process whilst remaining flexible and with no specific security architecture being mandated.

As the PRA is a framework that specifies components that will exist at the application layer, it cannot specify an accreditable solution without the support of security functionality offered by the infrastructure or externally to the system. Prior to its incorporation on an execution platform, without any understanding of the security environment, each component has no inherent security risk associated to it. Thus, the Exploiting Programme will be responsible for specifying the security environment and the security risks to be considered, and for demonstrating the Exploiting Platform meets the security targets applicable to its operational use.

A.4.2.2.1 What is the Security Approach?

The PRA architectural approach to security is that an assumed software and hardware infrastructure will provide many of the necessary security controls, atop of which will reside the components within the application layer. These components may contain additional security functions to support a multi-level integrated cyber defence.

Placing security functions within the components without the support of trust and assurance from the infrastructure would be difficult to accredit, thus the PRA expects lower-level security to be present, including security partitioning, VPNs, gateways and trust anchors back to the hardware (an established authoritative entity for which trust is assumed, not derived). However, these lower-level security functions may also require security-related metadata provided by the components in order to operate.

The approach taken for the component analysis does not provide detailed information on how these or any other security requirement may be satisfied, nor does it provide any information on how any non-technical security controls are addressed. These will be covered by the full security analysis to be undertaken for the Exploiting Platform.

The security approach taken for the PRA has been to look at the components and how they might support the accreditation goal, with each component including Security Considerations that:

- Identify security related and enforcing functions that may be required or provided by the components, see Security Functions for further details.
- Identify an indicative security classification for the components and therefore the security domain that the component may reside within, see Security Domains for further details.

As the Exploiting Programme will have its own security concerns and security architecture the indicative classification is meant only as a guide to support security domain segregation, and is not a requirement on the Exploiting Programme.

A.4.2.2.2 Why is the Security Approach Important?

To achieve security accreditation of any system, the integrator has to generate a security accreditation case based on a formally documented understanding of the security aspects of the system. As the security environment for each Exploiting Programme incorporating the PRA will be different, so each accreditation case will be different, but should cover topics including the operational context and the potential security risks the system could face. The risk assessment method can be as described in IAS 1&2 Ref. [51], ISO/IEC 27005 Ref. [52] or NIST Ref. [53] or as agreed with the Security Assurance Coordinator for the Exploiting Programme.

Security risk mitigation will be through a series of security controls. Generic sets of security controls can be found in ISO27002 Ref. [54], ISO/IEC 15408 Ref. [55] and NIST Ref. [53]; these will need to be tailored and are typically applied using a "defence in depth" approach, covering an array of personnel, physical, procedural and technical security controls to identify, protect, detect, respond and recover from a cyber attack. In practice, the controls identified will go beyond the scope of what can be achieved within the components, which can only address a subset of the technical controls. A PYRAMID deployment is therefore dependent on further external controls to provide this defence in depth.

For accreditation the Exploiting Programme will need to demonstrate:

- That there has been a formally documented approach to identifying security vulnerabilities.
- Any identified vulnerabilities have been mitigated as far as reasonably possible.
- Any classified information is suitably partitioned and protected from attacks on its confidentiality, integrity and availability (see Security Attributes).
- Safety and airworthiness elements of the system are protected from malicious attack.
- It is able to generate (and rapidly regenerate when required due to system reconfiguration/update) the information or metadata required to configure the system’s security functions.
- It is able to generate (and rapidly regenerate when required due to system reconfiguration/update) the evidence required for certification.

The Security Considerations provided for the components provide an indication of how they can support this process of accreditation through the development and documenting of security functions by:

- Identifying security functionality within components to act as risk mitigation controls.
- Supporting security partitioning by indicating the indicative classification of components.
- Identifying safety related functionality that should be protected.

Additionally, an indication of data flow is provided through the component service definitions.

Neither the PRA nor any standalone component can be considered accreditable in its own right, as this requires the context of the supporting infrastructure and security environment of the Exploiting Programme.

Security Guidance for PYRAMID Exploiters, Ref. [60], is available from Dstl on request. The document contains a list of "good practice" engineering and development standards and some contextual assumptions that are expected to apply to an Exploiting Programme, alongside additional considerations, e.g. for Software of an Unknown Pedigree (SOUP), types of storage, and a sample Protection Profile for a Common Criteria evaluation. Again, the Exploiting Programme will be expected to have its own specific considerations, security target, etc.

A.4.2.2.3 Aim of this Policy

This policy explains the approach taken in order to identify the security considerations for each PRA component that support Exploiting Programmes with the accreditation process in a manner consistent with the PYRAMID Key User Requirements (KURs).

The PYRAMID KURs that are applicable to this policy are:

- **Security Accreditable:** This primary PYRAMID KUR is supported at the PRA level through performing an initial security assessment against each component.
- **Flight Certifiable**: This policy supports identification of where safety-related security measures might need to be applied.

A.4.2.3 Overview

The policy for describing the security approach can be summarised as follows:

- Components are assigned an indicative security classification that corresponds to the appropriate levels of control that should be put in place to identify, protect, detect, respond to and recover from security threats. The assigned classification is based on an assessment of generic security risk within the context of a military security environment. These are not requirements on an Exploiting Programme.

The key points to be taken from the policy include:

- The security approach is flexible enough to be incorporated in a number of security environments.
- PYRAMID components cannot provide a complete security solution for the Exploiting Programme.
- Component developers and the integrators of a PYRAMID-based system will need to include configuration and metadata in their components in order that the underlying infrastructure can implement effective security controls and establish chains of trust, etc.
- The security approach is more than just protecting classified information but should also consider protecting the safety, operational advantage, freedom of operation and reputation of the entities that operate the Exploiting Platform and the intellectual property of its developers.
- The components contain a number of security functions (either SEFs or SRFs) from which a level of development rigour might be assumed, however the level of development rigour is ultimately dependant on the security environment and security requirements of the Exploiting Programme.

A.4.2.4 Determination

The determination of the component's security functions and indicative classifications has been based upon the assessment of a typical military platform, as well as assessment of the types of interactions defined in the IVs, the likely algorithms involved and the data that may reasonably be expected to be exchanged.

Any specific assumptions relevant to this assessment performed are included in the general Assumptions section of the PRA component definition's Design Rationale section. For example, this may include any instances whereby one component relies on another component for performing a required security-related task, such as encryption, or other applicable component behaviour and/or knowledge, etc.

The indicative classification has been stated at the lowest level that might be reasonably expected, often with a caveat that, in certain situations, this can be higher. It is important to note that reuse of developed components may adversely affect the security of Exploiting Platforms; the use of a developed component (or fragments thereof) on both secret and lesser classified platforms has the potential to compromise the secret one. This has not been seen as a reason to increase the indicative classification. The Security Guidance for PYRAMID Exploiters, Ref. [60], has further information on provenance and proliferation of developed components.

A.4.2.5 Context

Unless stated otherwise, the following context has been assumed:

- The component is primarily within a system belonging to an air vehicle, which may be manned or unmanned, and may be composed of different nodes in different locations, including ground-based elements.
- The air system is military and will be performing SNEO missions.

The Security Considerations for individual components have included additional component-specific context in the rationale where necessary.

The Security Guidance for PYRAMID Exploiters, Ref. [60], contains a list of contextual assumptions that are expected to apply to an Exploiting Programme utilising PYRAMID components.

A.4.2.6 Security Attributes

The traditional purpose of accreditation was to secure government classified information from breaches in confidentiality, integrity and availability. However, for military systems, accreditation needs to be cognisant of the effect on safety and operational effectiveness, together with operational advantage, freedom of operation and reputation as per the MOD's UK Prosperity Agenda, and not just the classification of its data and functionality. In support of this, the Exploiting Platform may also need to record and keep track of the flow of information or activities initiated by an authorised operator or the system itself; the PRA's security functions are intended to support the Exploiting Programme in the management of the following security attributes:

- Confidentiality - the property that information is not made available or disclosed to unauthorised individuals, entities, or processes. Ref. [41]
- Integrity - the property of accuracy and completeness. Ref. [41]
- Availability - the property of being accessible and usable upon demand by an authorised entity. Ref. [41]
- Accountability - the property that ensures actions performed by an entity may be traced uniquely to that entity. Ref. [62]
- Auditability - the ability to obtain audit evidence and evaluate it objectively to determine the extent to which the audit criteria are fulfilled.
- Authenticity - the property that an entity is what it claims to be. Ref. [41]
- Non-repudiation - the ability to prove the occurrence of a claimed event or action and its originating entities. Ref. [41]
- Safety related protection - security provided for safety related functions.

Protecting confidentiality, integrity and availability remains at the heart of ensuring the effectiveness of the product, which in turn maintains the operational advantage provided to the operator. Being able to reliably audit the actions carried out using the platform, knowing that the actions are authentic and non-repudiable will show where actions have been conducted in good faith and within the applicable rules of the air or engagement, thus protecting the operator from undue economic or reputational damage.

A.4.2.6.1 Safety Related Protection

The System Integrator of an Exploiting Platform has a responsibility to ensure that any safety related functionality is protected from cyber attack. Therefore, those components considered to be safety related may need specific security protection in accordance with DO-326A Ref. [21]/DO-356A Ref. [61] or another such acceptable means of compliance.

The approach specified in DO-326A is intended to cover airworthiness-related security, and the level of this security assurance should be commensurate with the risk associated with failure. Where other security analysis is required for the system there should be a coherent approach to reduce unnecessary overheads.

The Safety and Security Considerations of each component indicate where this will probably be needed, although this will depend on analysis performed by the Exploiting Programme. The Safety Considerations have been provided in line with the Safety Analysis policy.

A.4.2.7 Security Functions

The PRA is a framework for software within the application layer, however, most security functionality for the Exploiting Programme will be provided by the infrastructure; these infrastructure-based controls will be reliant on the provision of necessary configuration and metadata by the components, e.g. for establishing trust anchors, identification of data sources and maintaining separation. Additionally, some security functionality is specified within the PRA that can be tailored and configured depending on the requirements of the system. All components have been assessed as to how they might be expected to contribute to security; their Security Considerations will state whether there are:

- No anticipated security related functions.
- Security related functions (SRF) 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 an SRF will not directly lead to a security breach but may diminish the system's ability to detect or counter a threat (e.g. through security event logging).
- Security enforcing functions (SEF) are specific controls that provide protection to the system (e.g. providing cryptography). Failure of an SEF could lead to a security breach.

This is a broader classification than provided by the Common Criteria Evaluation Assurance Levels (EALs), see Ref. [55]. As with EALs, the level of rigour required for each component will increase with the security functions included, but will ultimately depend on the security requirements of the Exploiting Programme. However by documenting the functions and level of rigour using the same principles as Common Criteria, the functions could support the accreditation case.

These security functions support the concepts of a 'defence in depth' approach, with the components working with the infrastructure and any security controls external to the system to identify, protect, detect, respond, and recover from a cyber attack. The Cyber Defence policy provides a framework for how the PRA helps address cyber-related concerns.

A.4.2.7.1 Security Related Functions

The security related functions identified in the components include the following:

- **Back-up and Recovery**, supporting data archiving and the ability to recover data from that archive in the event of data loss.
- **Classification of Data**, supporting the handling of data, cross-domain separation and continued confidentiality.
- **Identifying Data Sources**, supporting trust in the system and authenticity.
- **Logging of Security Data**, supporting forensic investigations into possible security breaches and cyber attacks, etc. See the Recording and Logging policy for the types of data expected to be logged.
- **Maintaining Audit Records**, supporting audit, non-repudiation and accountability of mission data and events.
- **Supporting Safe Operation** and the continued airworthiness of the system.
- **Supporting Secure Remote Operation** of an unmanned platform.
- **System Status and Monitoring**, supporting identification of unexpected activity and therefore possible security breaches and cyber attacks.
- **Warnings and Notifications**, supporting awareness of unexpected activity and therefore possible security breaches and cyber attacks.

A.4.2.7.2 Security Enforcing Functions

The security enforcing functions identified in the components include the following:

- **Applying Emission Control Rules**, providing functions to restrict communications with outside entities.
- **Detecting Security Breaches**, providing functions to identify a security breach or cyber attack.
- **Encrypting Data**, providing functions to keep data confidential.
- **Ensuring Separation of Security Domains**, providing functions that prevent unintentional cross-domain communication.
- **HW Authentication**, providing a basis of trust in the underlying hardware.
- **Preventing Cyber Attacks and Malware**, providing protection from the effects of malicious cyber activity.
- **Protecting Integrity of Data**, providing functions to maintain data accuracy and completeness of data.
- **Rendering Sensitive Data Inaccessible**, providing functions to enable sanitisation of data.
- **Restricting Access to Data**, providing functions that ensure data access is restricted to allowed system elements or operators with appropriate clearance and a need to know.
- **Securing Communications**, providing functions to ensure confidentiality of data and voice communications.
- **User Login and Authentication**, providing functions to manage access to the system.
- **Verifying Integrity of Data**, providing functions that confirm data is accurate and complete.
- **Verifying Integrity of Software**, providing a basis of trust in the software within the system.

A.4.2.8 Security Domains

As the Exploiting Programme will have its own security concerns and architecture, the indicative classification provided is meant only as a guide to support security domain segregation, it is not a requirement on an Exploiting Programme.

To support the segregation of classified information and functionality within the Exploiting Programme, the following classifications are used:

- Official (O).
- Official Sensitive (O-S).
- Secret Coalition Eyes Only (SCEO).
- Secret National Eyes Only (SNEO).
- Top Secret (TS).

Note: Sensitive and Coalition or National Eyes Only are handling conditions and not classifications in their own right. The terms “Coalition” and “National” are generic placeholders that should be replaced by the appropriate terms when applied.

This classification within the component's Security Considerations is only a guide; the classification of the implemented component and other requirements for segregation will depend on the operational context of the Exploiting Programme and define its security architecture. The implementer is free to choose their own classifications and partition the components as they see fit, with some components being instantiated within several security classification domains, with communication between domains being limited to specifically defined and authorised communications channels with appropriate security enforcing functions to monitor and verify inter-domain traffic.

Within the PRA, as an indicative guide, each component has been defined as having one or more of these security classifications. The classification of the Platform Specific Model (PSM) component and other requirements for segregation will depend on the operational context of the Exploiting Programme and define its security architecture. The security domain that the component is in will depend on:

- The classification of the data the component is processing.
- The classification of algorithms within the component.
- The required integrity of the data processed by the component.
- Security protection requirements of safety-related functions.
- Any intellectual property concerns.

Due to intellectual property, the potential for invalidating certification, and international treaty considerations, there may be elements of the system that although accessible to the developer are not accessible to the operator or to potential adversaries. This level of protection needs to be included when considering the security domains.

B Appendix B: Components

This appendix defines the PRA components introduced in section 3, and also a generic component composition.

B.1 Component Composition

B.1.1 Overview

The Component Composition defines a generic component with model elements that enable it to support the PRA policies, especially:

- Control Architecture
- Dependency Management
- Capability Assessment
- Constraint Management
- Health Management

The Component Composition plays an important role in explaining the behaviour described in those policies as it applies to a generic component and how they relate to individual components.

The Component Composition has two additional goals:

- To help create a consistent framework for the use of components;
- To help component developers and implementers to understand the context in which certain aspects of their components are intended to work.

The Component Composition is particularly concerned with defining model elements that will allow components to cooperate to plan and execute system requirements, as described in the Control Architecture and related policies. It is therefore applicable to every component.

The model elements described in the Component Composition have been incorporated into the PRA components wherever they apply. Not all of the elements are applicable to every component (it depends on their role), and elements are tailored to make them specific to the subject matter of the component. Components include links to show where the Component Composition has been applied.

The Component Composition has a similar structure to an ordinary PRA component, except that it does not include a Role, Design Rationale, or Overview: there is no useful generic equivalent for these.

The Component Composition also has a set of Use Cases, which show how a component can be used.

B.1.2 Use Cases

The Use Cases show how a component is expected to interact with the wider system. They are shown from the viewpoint of the individual component. System-level views, showing how multiple components interact to achieve mission objectives, are given in the Control Architecture and other policies. Each individual use case has one or more sequence diagrams that give examples of typical scenarios for that use case.

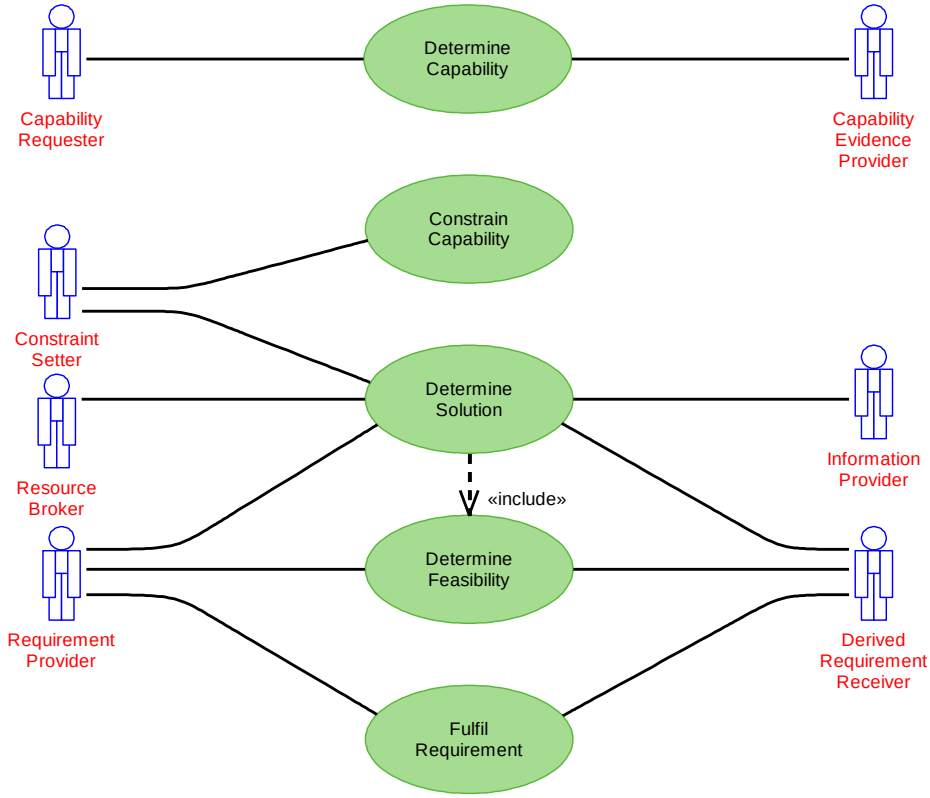

Figure 103: Use Case Summary Diagram

Figure 103: Use Case Summary Diagram summarises the use cases that the Component Composition supports.

B.1.2.1 Actors

Requirement Provider

An external actor or component that provides the requirement to be achieved, the measurement criterion by which the achievement of the requirement is assessed, and also tells the component to fulfil the requirement.

Information Provider

An external actor or component that provides information that the component needs to know in order to determine a solution.

Derived Requirement Receiver

An external actor or component that receives derived requirements (e.g. an action to be performed or a constraint on behaviour) and provides information against them.

Constraint Setter

An actor that represents a component or organisational entity that defines What_I'm_Not_Allowed_To_Do, which may include information about the operating conditions or environment in which the solution needs to be carried out.

Capability Evidence Provider

An external actor or component that provides evidence about a capability that this component relies on. This may be the provider of a capability, such as a physical resource or another component that this one relies on to fulfil its responsibilities. It could also be Anomaly Detection, Health Assessment or Cyber Defence, any of which could supply evidence for the capability of another component.

Capability Requester

An actor that requires an understanding of the capability of a component. This could be for a solution that requires the use of another capability, or for a need to assess its own capability which depends on another capability. It could be a HMI component or an actor external to the system.

Resource Broker

An external actor or component that manages the allocation of one or more resources needed by this component or by a Derived Requirement Receiver. It could be Resource Brokerage or Spectrum, or an external approver.

B.1.2.2 Use Cases

Determine Capability

In this use case, What_My_Capability_Is (current and predicted) is kept up-to-date as the evidence from Capability Evidence Providers changes. The three sequence diagrams show different styles of interaction between the component and a Capability Requester.

This use case shows how a component exchanges capability information with others to contribute to the Capability Assessment policy.

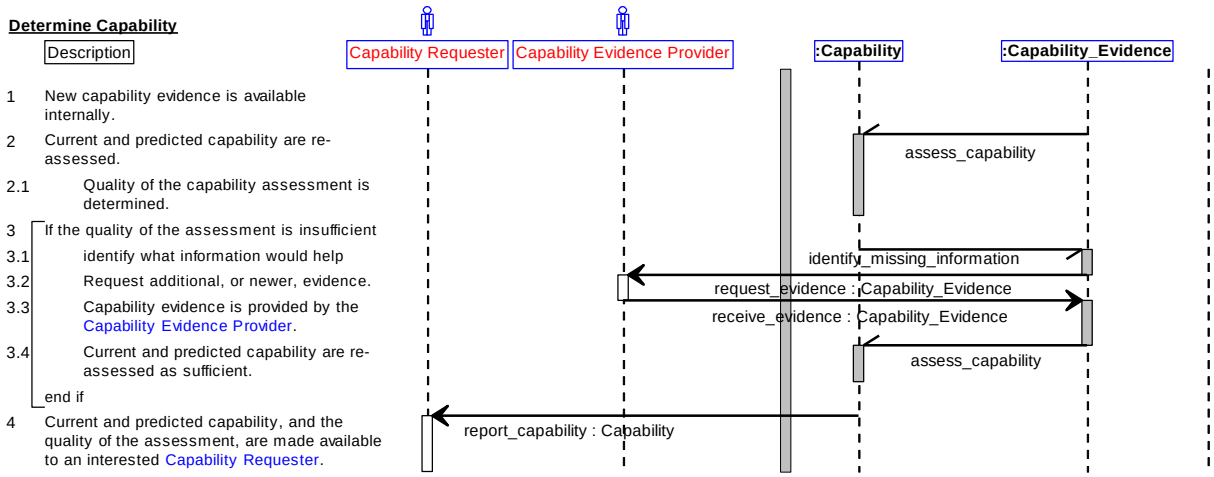

Figure 104: Capability Reassessment

In the Figure 104: Capability Reassessment sequence diagram no new capability evidence is provided, but capability assessment is triggered internally, for example by the passage of time. The component's capability is re-assessed and its quality re-determined. If the quality is determined to be insufficient then further capability evidence is obtained and the capability is reassessed. The revised assessment is made available to interested Capability Requesters.

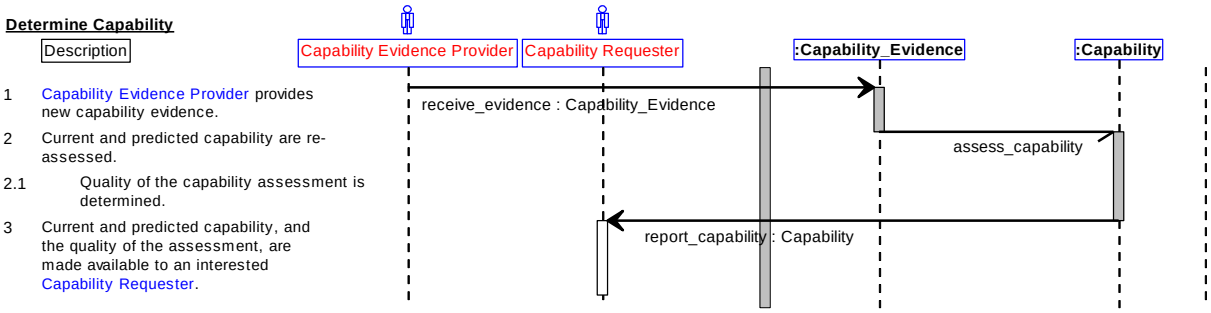

Figure 105: Fresh Capability Evidence

In the Figure 105: Fresh Capability Evidence sequence diagram, a Capability Evidence Provider provides new capability evidence. This could be routine information, test results (possibly as a consequence of a request for missing information that was sent out earlier) or an anomaly.

The component's capability is re-assessed and its quality re-determined. The revised assessment is made available to Capability Requesters.

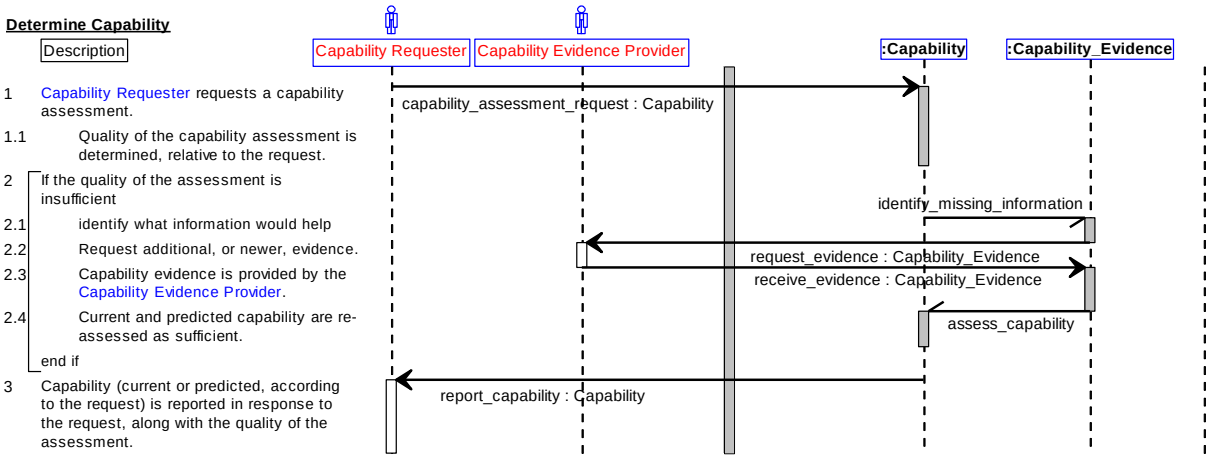

Figure 106: Requested Capability Assessment

In the Figure 106: Requested Capability Assessment sequence diagram, a Capability Requester makes an explicit request for capability information. The component's capability is assessed and if the quality is inadequate, the component requests additional information from a Capability Evidence Provider that could improve the quality, and the capability is reassessed. The assessment is then made available to interested Capability Requesters.

Constrain Capability

In this use case, the capability of the component to do something is constrained by an external source. This use case shows how a component contributes to the Constraint Management policy.

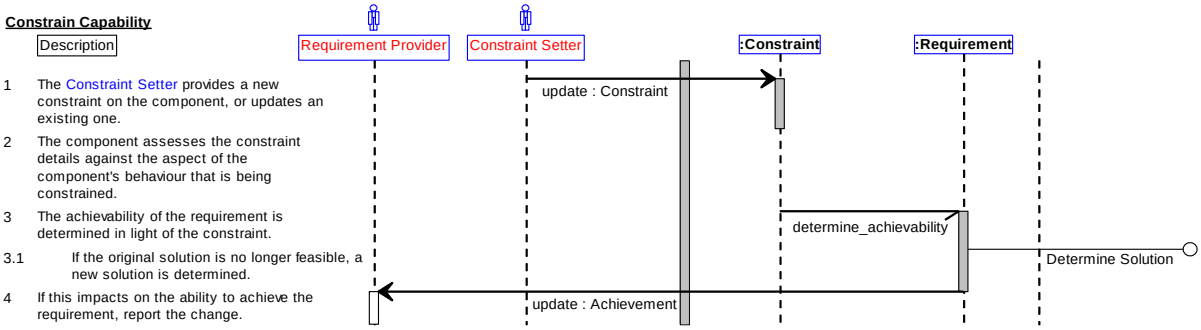

Figure 107: Apply New Constraint

In the Figure 107: Apply New Constraint sequence diagram, a Constraint Setter provides a new constraint on the component, or updates an existing one, which then constrains the component's behaviour. The feasibility of currently in progress solution is checked (see the Determine Feasibility use case), and if determined to be no longer achievable, a new solution is determined (see the Determine Solution use case). Any changes to the achievability of the requirement are also determined and reported to the Requirement Provider if necessary.

Determine Solution

In this use case a solution to achieve a requirement is planned, taking into account capability, constraints, resource availability and associated measurement criteria. An indication of whether the requirement is achievable is reported along with the cost, which is measured using metrics as defined by the measurement criteria. Issues (such as resource conflicts) which cannot be resolved within the scope of the component's subject matter are also reported.

This use case shows the part a single component plays in finding a solution to a mission objective, as described in the Control Architecture and Dependency Management policies.

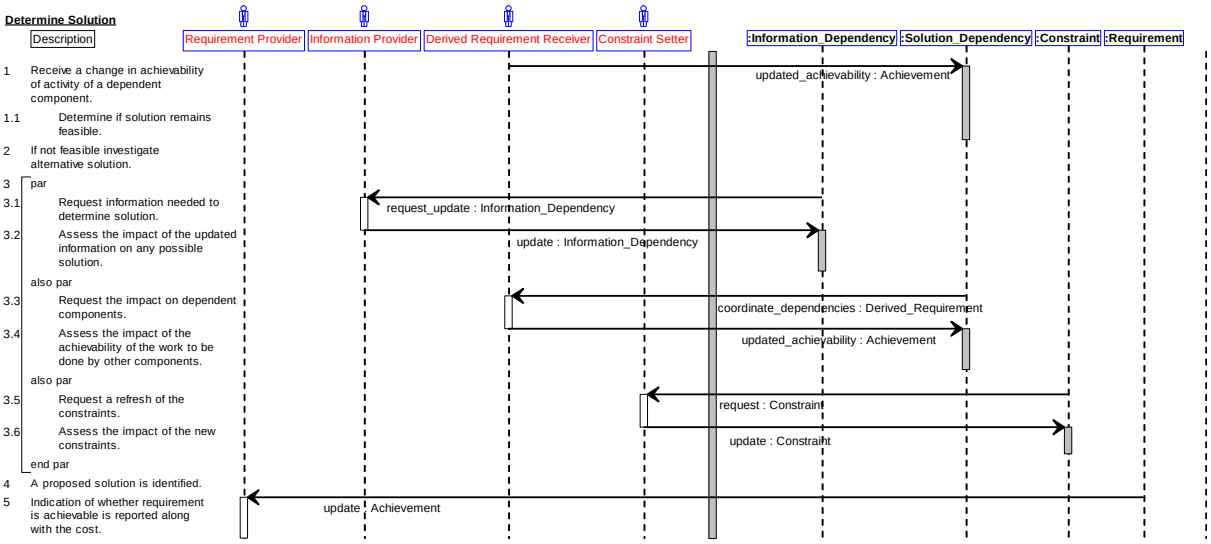

Figure 108: Change in the Achievability of Dependent Component Trigger

In the Figure 108: Change in the Achievability of Dependent Component Trigger sequence diagram, a Derived Requirement Receiver provides an achievability update on its activities. The solution feasibility is determined and a new solution is investigated if the original is no longer feasible. To determine the new solution, the component requests information from the Information Provider, solution dependencies to be achieved from the Derived Requirement Receiver and a refresh of constraints from the Constraint Setter, before identifying a preferred solution.

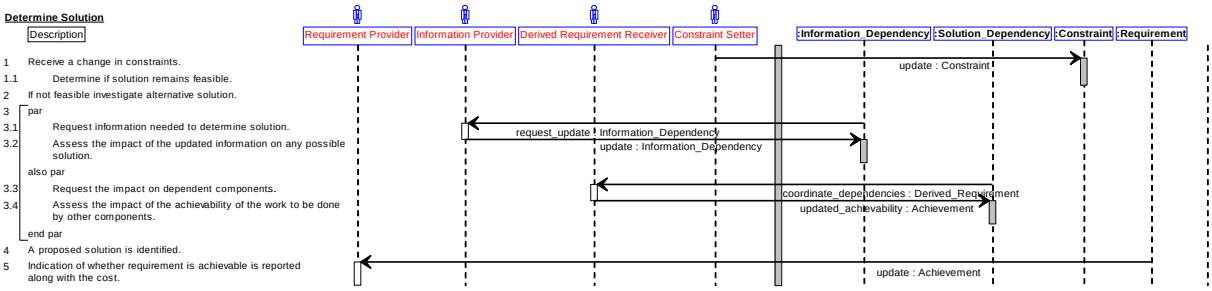

Figure 109: Change in the Constraints Trigger

In the Figure 109: Change in the Constraints Trigger sequence diagram, the Constraint Setter provides a change in constraints. The solution feasibility is determined and a new solution is investigated if the original is no longer feasible. To determine the new solution, the component requests information from the Information Provider, and solution dependencies to be achieved from the Derived Requirement Receiver, before identifying a preferred solution.

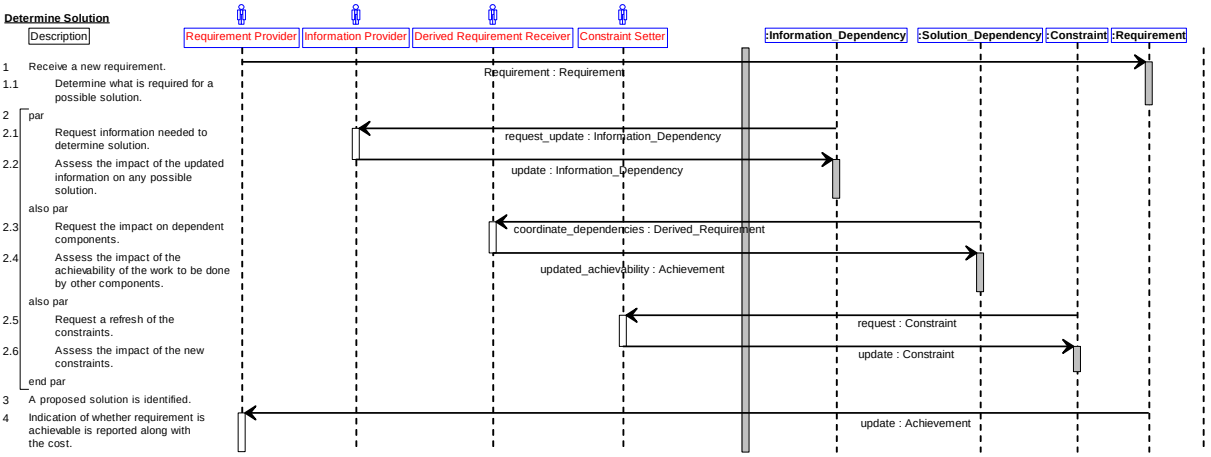

Figure 110: New Requirement Trigger

In the Figure 110: New Requirement Trigger sequence diagram, a Requirement Provider provides a new requirement. To determine the solution, the component requests information from the Information Provider, solution dependencies to be achieved from the Derived Requirement Receiver and a refresh of constraints from the Constraint Setter, before identifying a preferred solution. A change to an existing requirement would be handled in a similar way.

For a given component, the Requirement service is in its problem space, and achievability and achievement are expressed in terms of requirements. The Solution_Dependency and Information_Dependency services are in the component's solution space and deal with a specific solution and its feasibility.

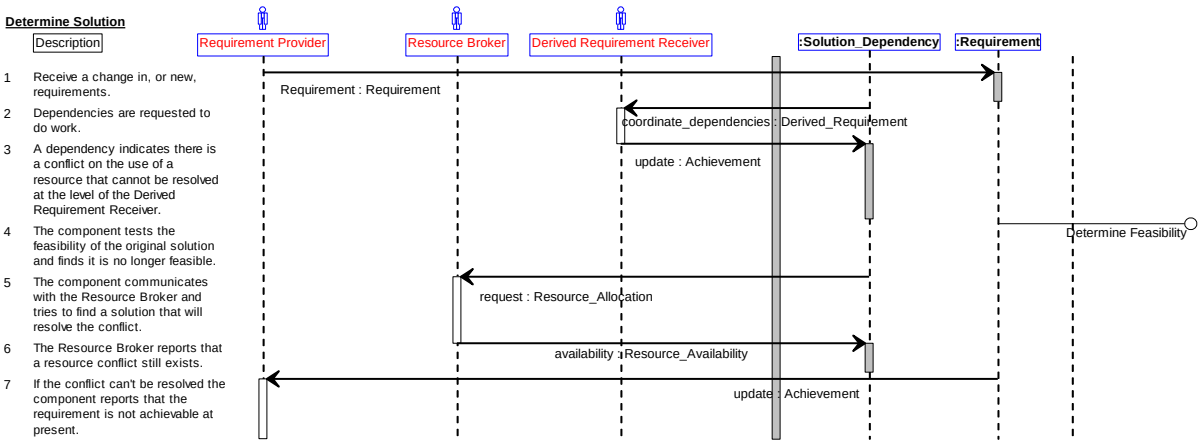

Figure 111: Report Resource Conflict

In the Figure 111: Report Resource Conflict sequence diagram, a change to an existing requirement, or a new requirement, is received. One of the dependencies requires the use of a resource that the Resource Broker is unable to allocate, which results in the solution no longer being feasible, as described in the Determine Feasibility use case. The component tries to resolve the conflict via a new solution, and if this is not achievable, the conflict is reported to the Requirement Provider.

Determine Feasibility

In this use case, the feasibility of carrying out the solution intended to achieve a requirement is determined. This includes solutions planned to be carried out in the future as well as those being carried out now.

This use case shows how each component monitors the feasibility of a planned solution when circumstances change, as discussed in the Dependency Management policy.

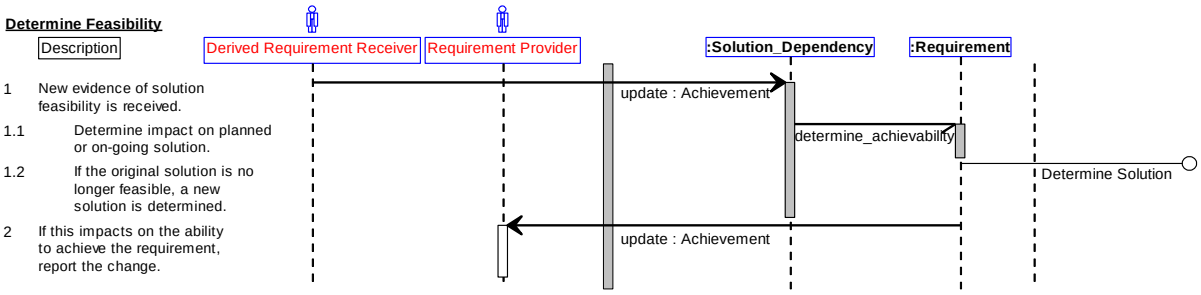

Figure 112: Determine Feasibility

In the Figure 112: Determine Feasibility sequence diagram, a Derived Requirement Receiver provides new evidence of the feasibility of a solution dependency. The impact of this new evidence on the planned or on-going solution is determined. If there are any changes to the achievability of the requirement this is reported to the Requirement Provider.

Fulfil Requirement

In this use case a requirement is fulfilled by executing the planned solution; achievement of the requirement is assessed and reported when requested by the Requirement Provider, at regular intervals, or when requirements have been met.

The use case assumes that the planned solution still meets the requirements. The Determine Feasibility use case shows how a solution replan is triggered if necessary.

This use case shows the part played by a single component in carrying out a mission objective, as described in the Control Architecture and Dependency Management policies.

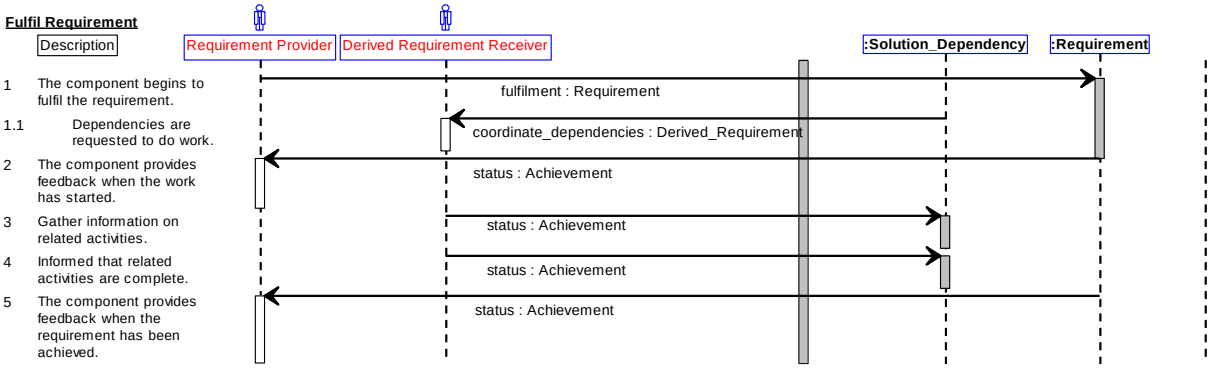

Figure 113: Report Achievement

In the Figure 113: Report Achievement sequence diagram, information is gathered on the progress of dependency activities and the component's own activities, and reported once the requirement has been achieved.

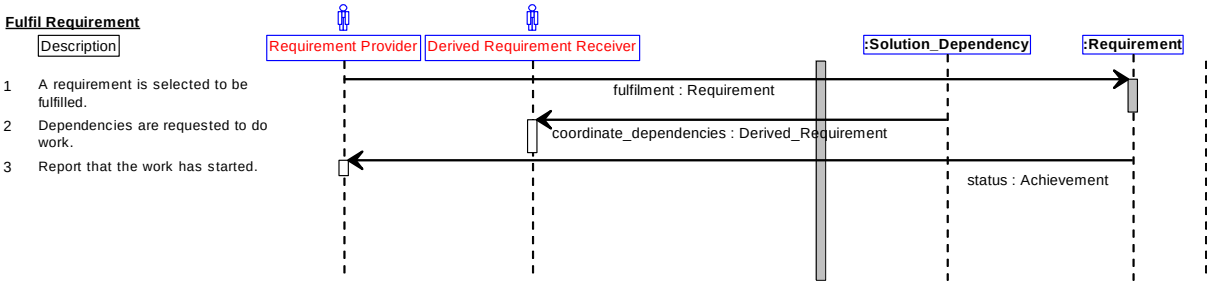

Figure 114: Select a Requirement to be Fulfilled

In the Figure 114: Select a Requirement to be Fulfilled sequence diagram, a Requirement Provider selects a previously defined requirement to be fulfilled. The component requests work from dependencies and then informs the Requirement Provider when the work has begun.

B.1.3 Responsibilities

This section lists generic responsibilities that apply to a component that satisfies all the use cases shown in Use Cases. Not all of these responsibilities apply to every component. Where they do apply, they are specialised to the subject matter of the component. A component may have other responsibilities specific to its subject matter.

capture_requirements

- To capture provided requirements.

capture_constraints

- To capture provided constraints.

capture_measurement_criteria

- To capture given measurement criteria.

determine_solution

- To determine a solution that meets the requirements within provided constraints.

determine_quality_of_deliverables

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

identify_progress

- To identify progress against a requirement.

determine_solution_dependencies

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

identify_missing_information

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

identify_whether_requirement_is_achievable

- To identify whether a requirement is still achievable given current or predicted capability and conditions.

coordinate_dependencies

- To coordinate the dependencies to execute a solution.

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).

predict_capability_progression

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

determine_if_solution_remains_feasible

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

B.1.4 Service Summary

Figure 115: Service Summary shows generic services that support the interactions in between components, and between components and actors, as shown in the Use Cases section. Not all of the services apply to every component. Where they do apply, they are specialised to the subject matter of the component. A component may have other services specific to its subject matter.

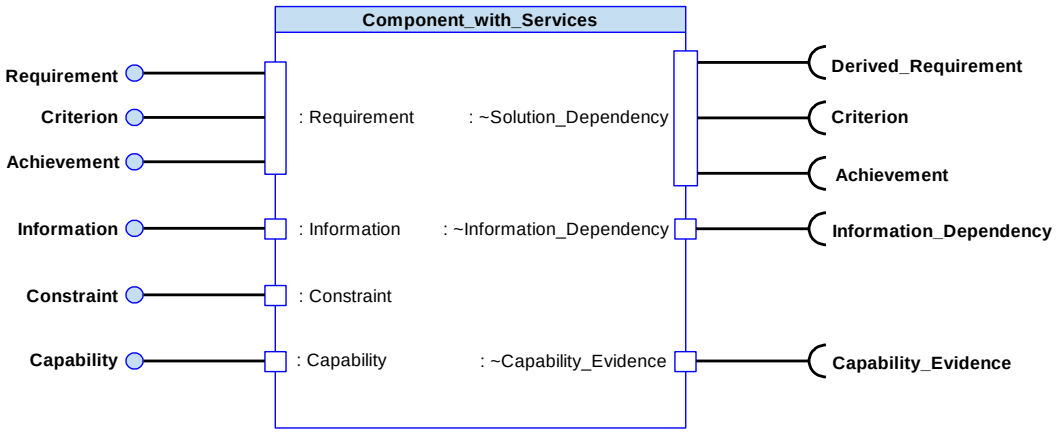

Figure 115: Service Summary

B.1.5 Subject Matter Semantics

The subject matter of the Component Composition is described in extremely generic terms. Components that satisfy the use cases shown in Use Cases will include entities that correspond to some or all of the entities shown here, but they will be expressed in terms suitable to the subject matter of the component.

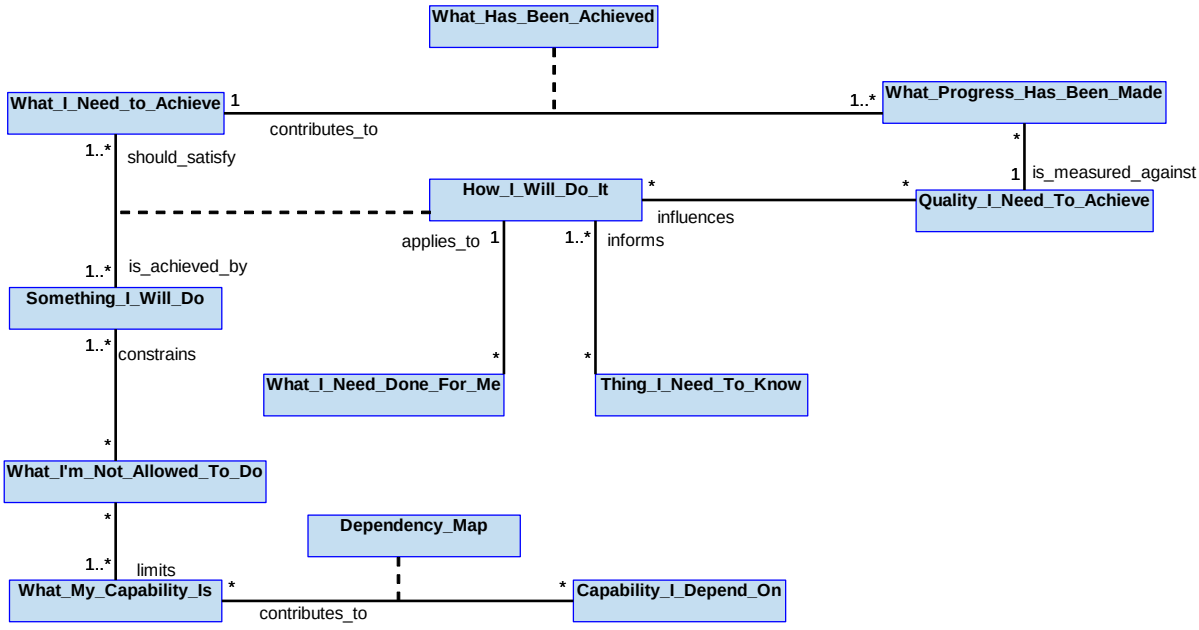

Figure 116: Semantics

B.1.5.1 Entities

What_I'm_Not_Allowed_To_Do

A limit or constraint on what the component is allowed to do.

Capability_I_Depend_On

A capability which this component relies on for its own capability.

Dependency_Map

The relationship between this component's capabilities, and the capabilities that it depends on.

How_I_Will_Do_It

How the component plans to meet the requirement(s) placed upon it.

What_My_Capability_Is

Something this component is able to do or provide.

Something_I_Will_Do

A specific activity the component will carry out.

What_I_Need_to_Achieve

A requirement that is placed on this component.

What_Progress_Has_Been_Made

Evidence of progress as part of an activity, e.g. jettison package has been sanitised prior to jettison.

What_Has_Been_Achieved

The degree to which the progress contributes towards achieving the requirement goal, e.g. waypoint A has been reached.

What_I_Need_Done_For_Me

Activities that the solution is dependent on, i.e. steps in the solution that are placed as derived requirements on other components. An example is that undercarriage must be deployed before landing can proceed.

Thing_I_Need_To_Know

Information the component needs to know to determine a solution, e.g. the environment in which the Exploiting Platform is currently, or will be, operating.

Quality_I_Need_To_Achieve

A measurement criterion used to determine progress or success.

B.1.6 Services

As with the Services Summary shown in Service Summary, not all of the services listed here apply to every component. Where they do apply, they are specialised to the subject matter of the component.

The services support the use cases shown in Use Cases, and examples of their use are given in the sequence diagrams in that section.

B.1.6.1 Service Definitions

B.1.6.1.1 Requirement

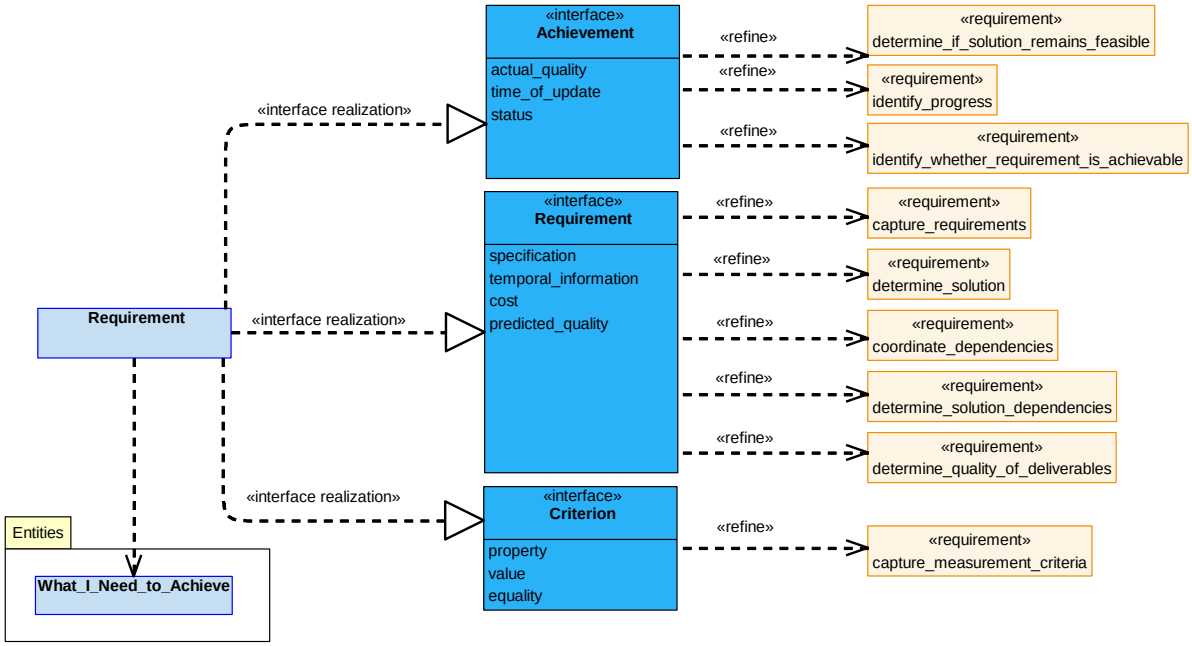

Figure 117: Requirement Service Definition

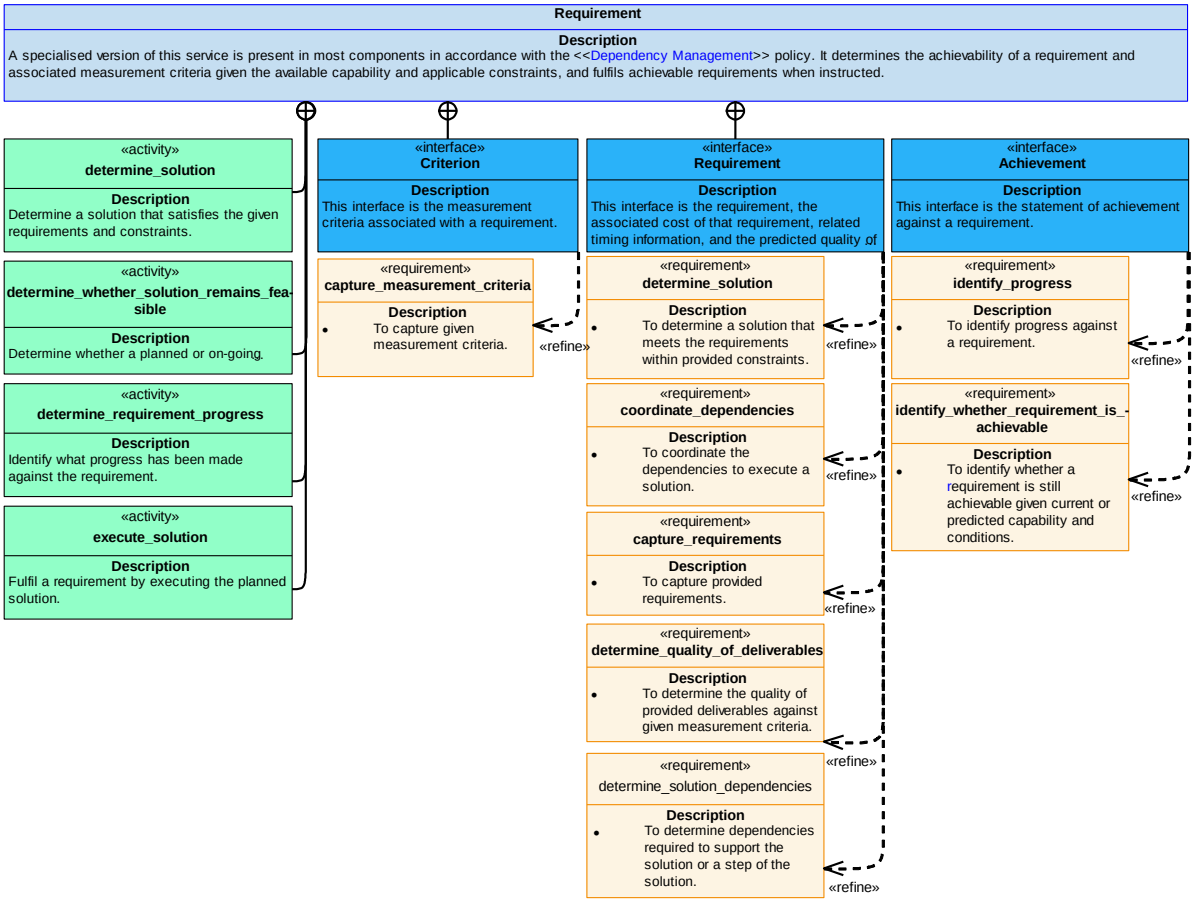

Figure 118: Requirement Service Policy

Requirement

A specialised version of this service is present in most components in accordance with the Dependency Management policy. It determines the achievability of a requirement and associated measurement criteria given the available capability and applicable constraints, and fulfils achievable requirements when instructed.

Interfaces

Requirement

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

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.

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.

Activities

determine_solution

Determine a solution that satisfies the given requirements and constraints.

determine_requirement_progress

Identify what progress has been made against the requirement.

execute_solution

Fulfil a requirement by executing the planned solution.

determine_whether_solution_remains_feasible

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

B.1.6.1.2 Solution_Dependency

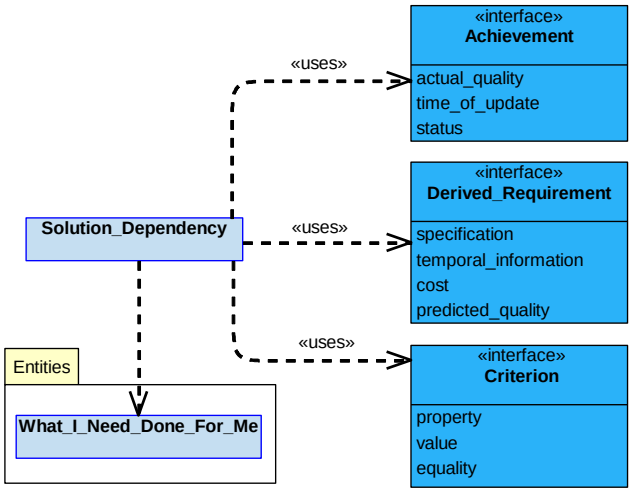

Figure 119: Solution_Dependency Service Definition

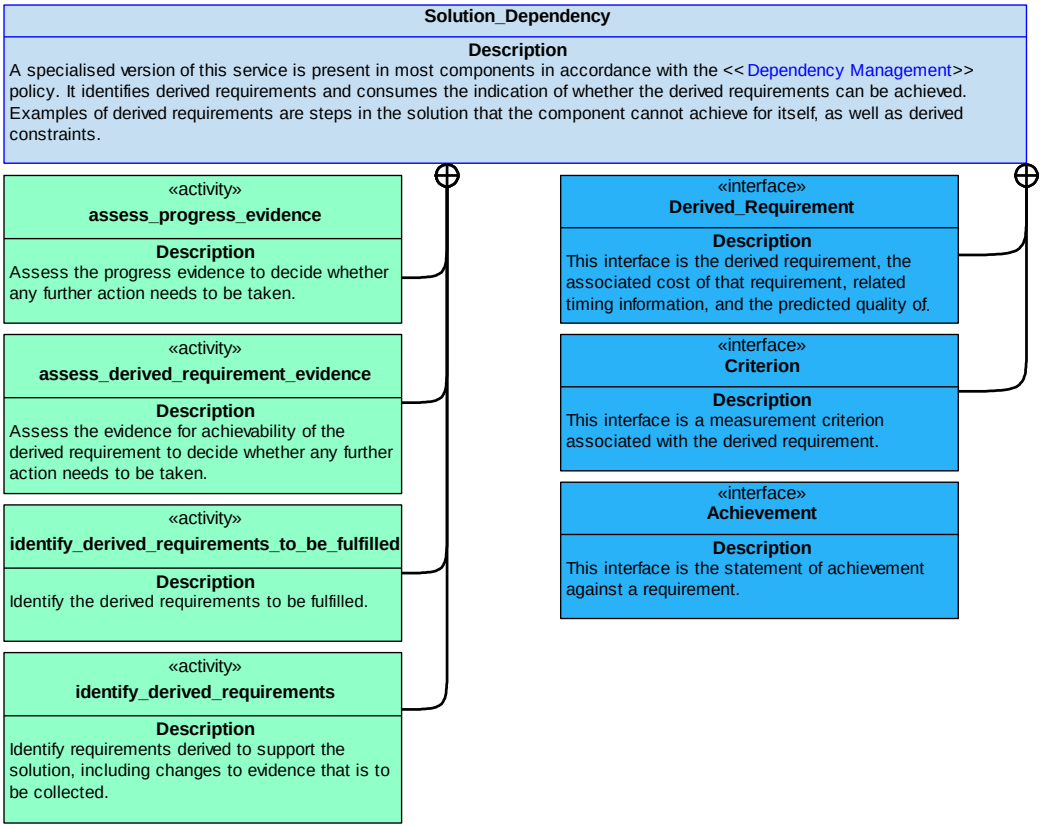

Figure 120: Solution_Dependency Service Policy

Solution_Dependency

A specialised version of this service is present in most components in accordance with the Dependency Management policy. It 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, as well as derived constraints.

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.

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.

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.

Activities

assess_progress_evidence

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_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.

B.1.6.1.3 Information

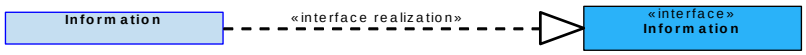

Figure 121: Information Service Definition

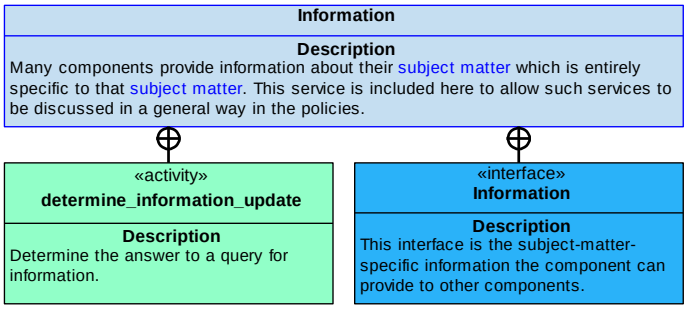

Figure 122: Information Service Policy

Information

Many components provide information about their subject matter which is entirely specific to that subject matter. This service is included here to allow such services to be discussed in a general way in the policies.

Interface

Information

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

Activity

determine_information_update

Determine the answer to a query for information.

B.1.6.1.4 Information_Dependency

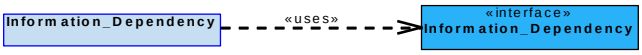

Figure 123: Information_Dependency Service Definition

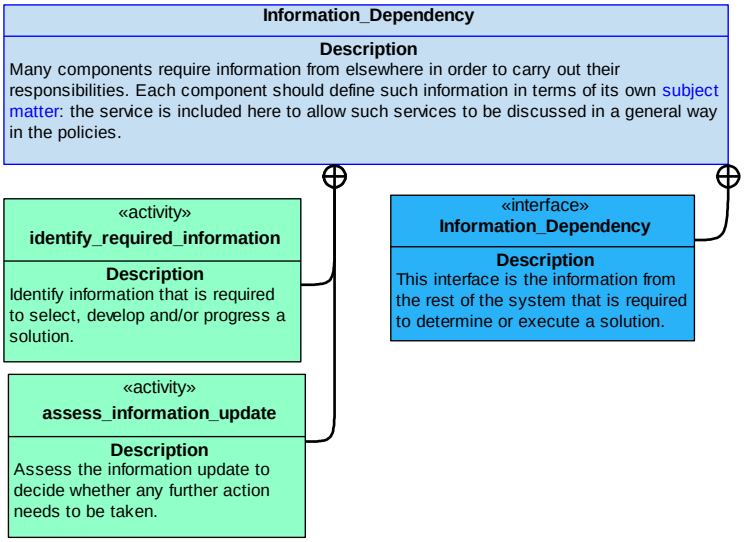

Figure 124: Information_Dependency Service Policy

Information_Dependency

Many components require information from elsewhere in order to carry out their responsibilities. Each component should define such information in terms of its own subject matter: the service is included here to allow such services to be discussed in a general way in the policies.

Interface

Information_Dependency

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

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 solution.

B.1.6.1.5 Constraint

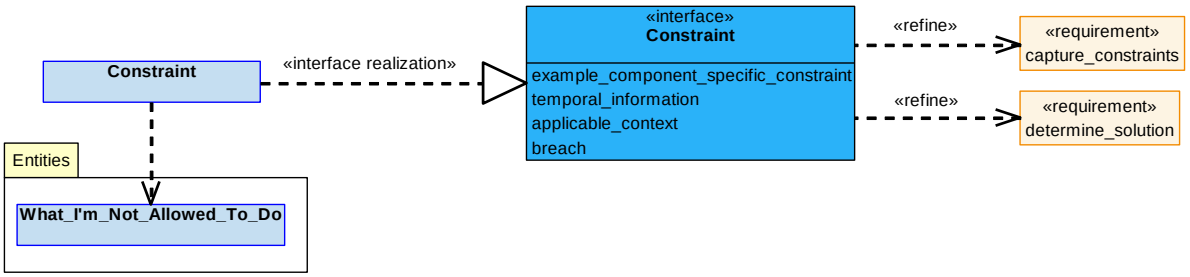

Figure 125: Constraint Service Definition

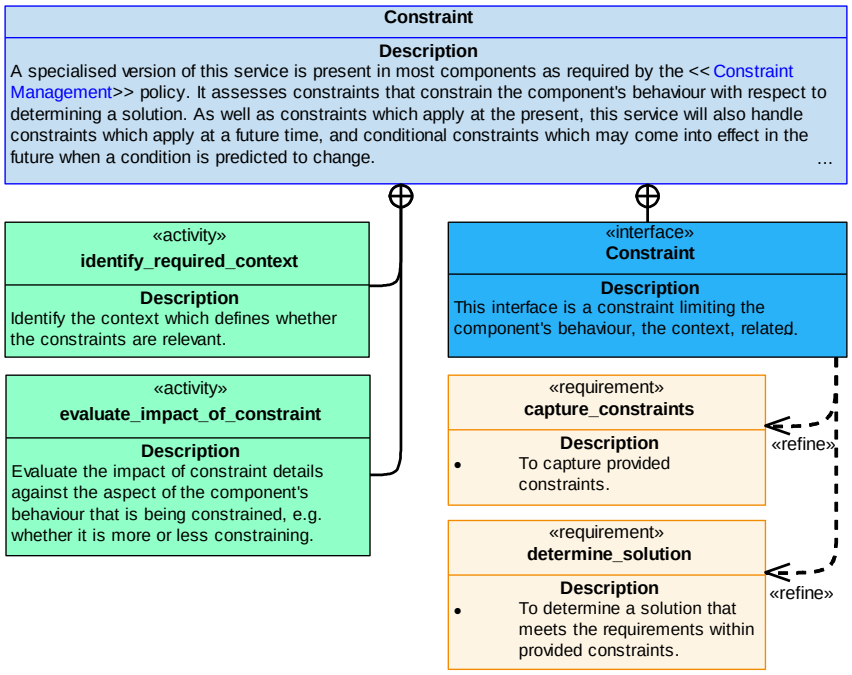

Figure 126: Constraint Service Policy

Constraint

A specialised version of this service is present in most components as required by the Constraint Management policy. It assesses constraints that constrain the component's behaviour with respect to determining a solution. As well as constraints which apply at the present, 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.

A component is not allowed to deliberately ignore a constraint, although it may vary its own actions to increase or decrease the likelihood of events outside of its control resulting in a breach when trading off against other factors. A breach is only applicable when events outside of the component’s control can result in the constraint being breached.

Interface

Constraint

This interface is a constraint limiting the component's behaviour, the context, related timing information, and a breach indication.

Attributes

example_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.

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.

B.1.6.1.6 Capability

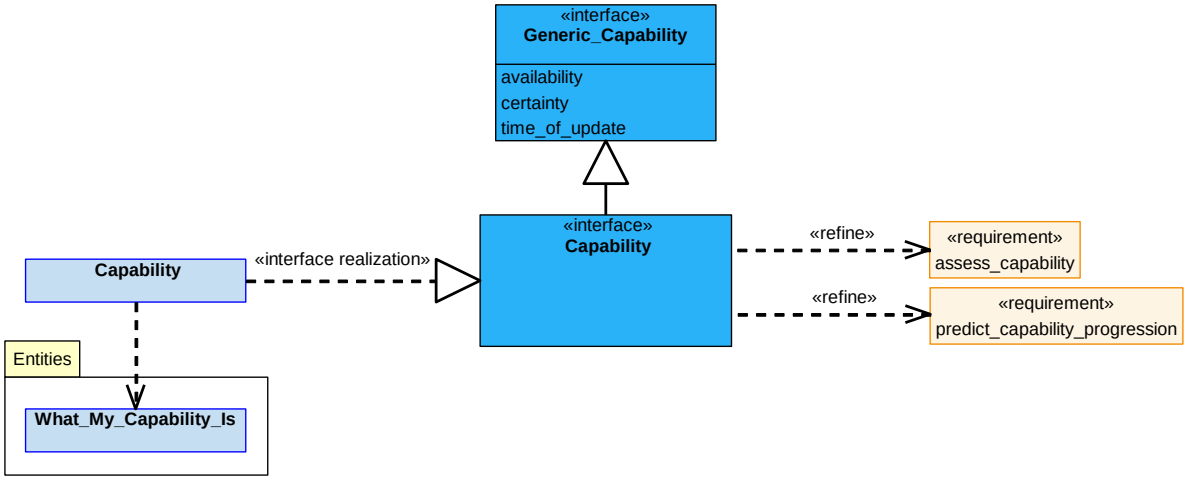

Figure 127: Capability Service Definition

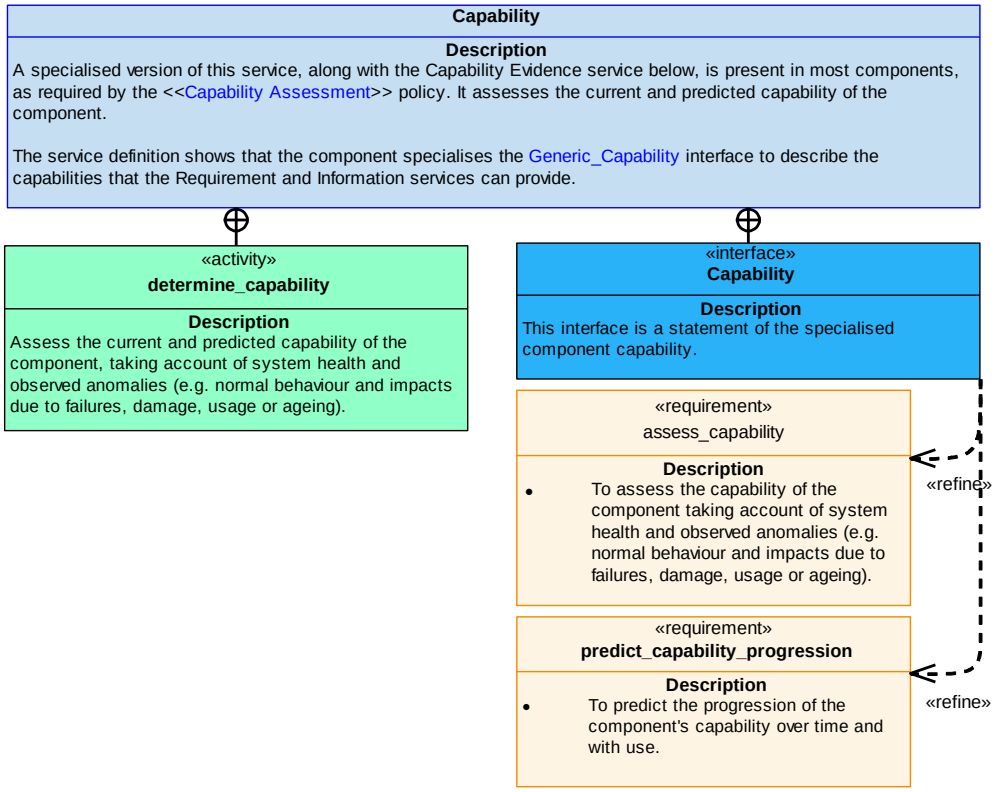

Figure 128: Capability Service Policy

Capability

A specialised version of this service, along with the Capability Evidence service below, is present in most components, as required by the Capability Assessment policy. It assesses the current and predicted capability of the component.

The service definition shows that the component specialises the Generic_Capability interface to describe the capabilities that the Requirement and Information services can provide.

Interface

Capability

This interface is a statement of the specialised component 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).

B.1.6.1.7 Capability_Evidence

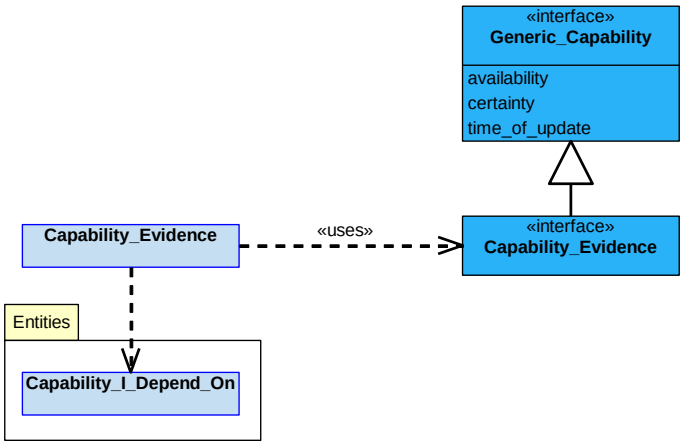

Figure 129: Capability_Evidence Service Definition

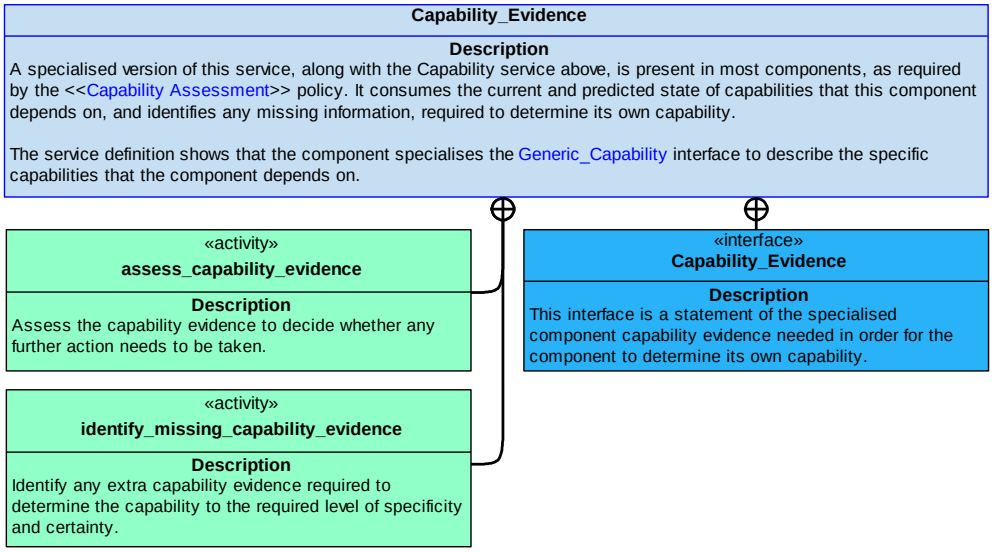

Figure 130: Capability_Evidence Service Policy

Capability_Evidence

A specialised version of this service, along with the Capability service above, is present in most components, as required by the Capability Assessment policy. It consumes the current and predicted state of capabilities that this component depends on, and identifies any missing information, required to determine its own capability.

The service definition shows that the component specialises the Generic_Capability interface to describe the specific capabilities that the component depends on.

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.

B.1.6.2 Service Dependencies

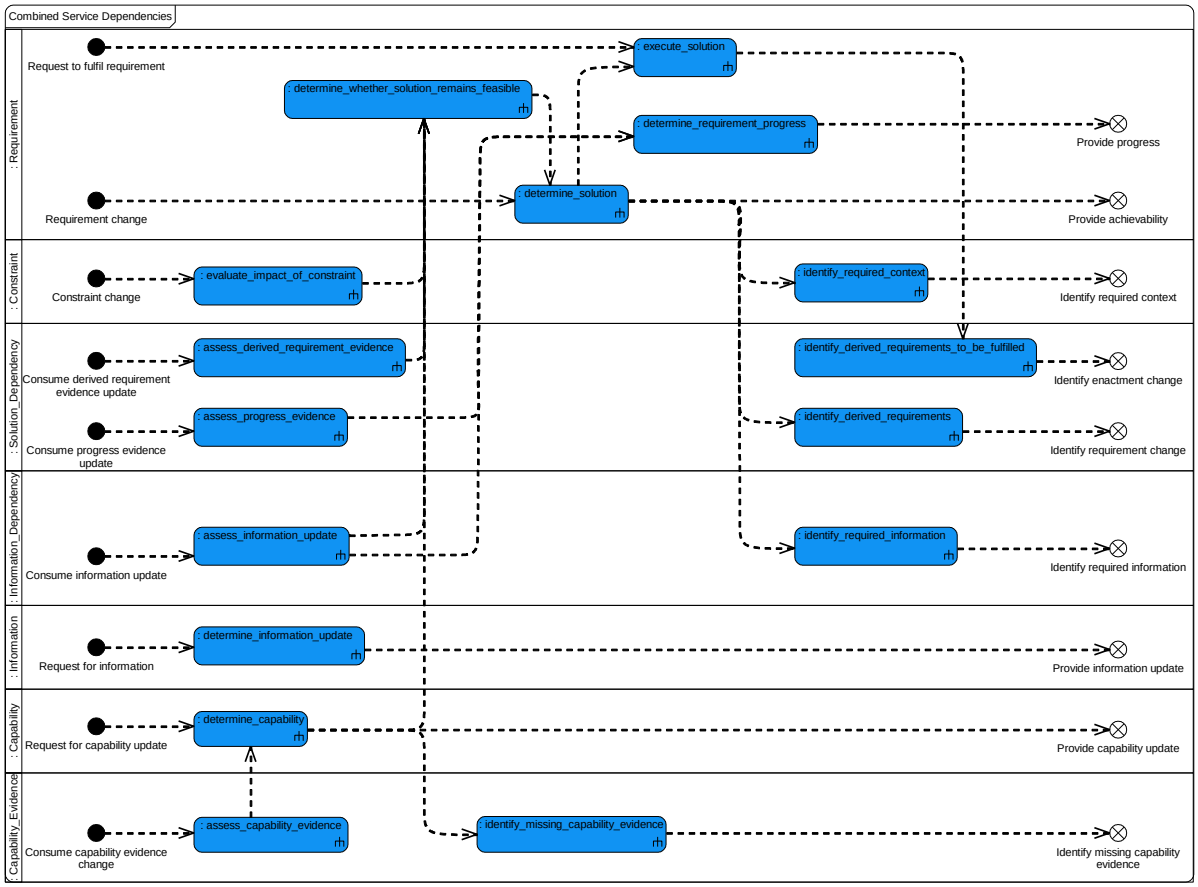

Figure 131: Combined Service Dependencies

This diagram has a swimlane for each service that is part of the Component Composition, and shows the dependencies between those services, as well as the various activities within each of the services.

B.1.7 Data Model

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 the Component Connections policy) to close the semantic gap and to map and translate between component services.

However, in some limited areas it is helpful to derive service data from a generic data model so that components can communicate some categories of data in a uniform way.

B.1.7.1 Capability

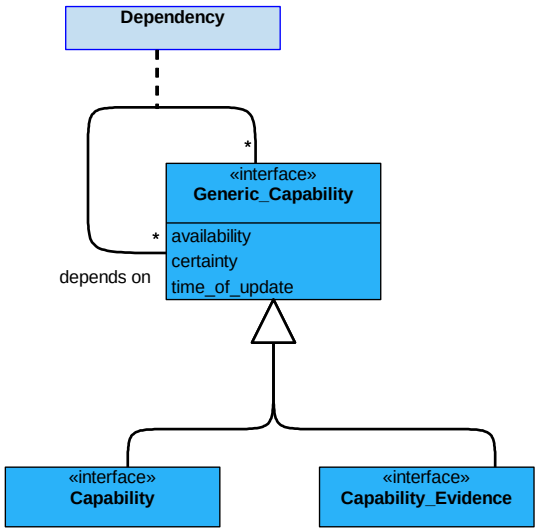

Figure 132: Capability Data Model

The capability data model shows the data structures that represent a capability assessment. Although it will need to be specialised to reflect the capabilities of a specific Exploiting Platform, it is a generic expression of capabilities that can form the basis of a mechanism to exchange capability information within that platform.

The diagram shows a self-referential Generic_Capability class, with an association class (Dependency) that defines how capabilities depend on each other. Individual components will specialise Generic_Capability by adding enumerations (or other data types) that reflect their own capabilities. An Exploiting Programme should provide further specialisation specific to the deployment.

The Dependency class is likely to be data-driven to capture the capability mapping for a particular Exploiting Programme.

Generic_Capability

This interface is the generic statement of capability that can be specialised for specific components.

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.

Dependency

This association class captures the way that Generic_Capabilities depend on each other.

B.1.7.2 Achievement

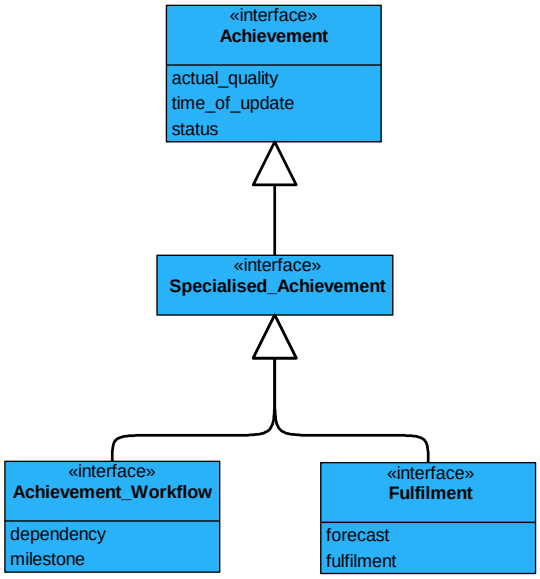

Figure 133: Achievement Data Model

The achievement data model is used to report all achievement of requirements in a uniform way.

The diagram shows a high-level Achievement class, which individual components will specialise if they are required to provide detailed achievement information, such as progress dependencies or achievement fulfilment.

Achievement

This interface is the statement of achievement against a requirement.

Attributes

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

Specialised_Achievement

This interface is a specialisation of the statement of achievement against a requirement.

Achievement_Workflow

A specialisation of the general case, used where there are complex dependencies, to make the requester aware of the dependency, the details of which will be on the related service. Similarly the milestone attribute provides information about the ordering of the dependencies.

Attributes

dependency The reliance of the parent requirement on the progress of other requirements.

milestone A significant point in progress towards the requirement.

Fulfilment

A specialisation that is used to predict the likely completion of a requirement in abstract terms, measured against a specific criterion.

Attributes

forecast A forecast of the remaining work required (e.g. time to complete or distance to destination).

fulfilment The comparison of achievement against the required work forecast (e.g. percentage complete).

B.2 Component Set

This section defines the PRA components introduced in section 3.

B.2.1 Anomaly Detection

B.2.1.1 Role

The role of Anomaly Detection is to detect states 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.

B.2.1.2 Overview

Control Architecture

Anomaly Detection is a service component as defined in the Control Architecture policy.

Standard Pattern of Use

Anomaly Detection determines the Actual_State of a System_Element using System_Data. For example the physical position of a resource such as an actuator, processing time, user privileges or position and vector of a detected aircraft. 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:

- The identification that a System_Element is in an unexpected State is necessary for health management or to identify a cyber attack.

B.2.1.3 Service Summary

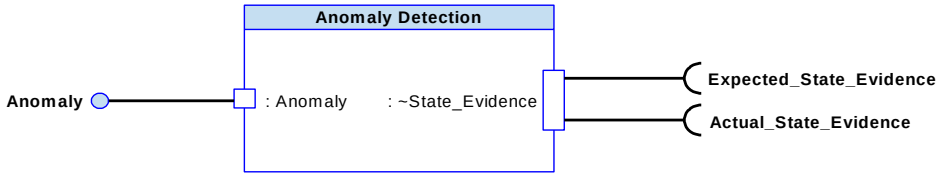

Figure 134: Anomaly Detection Service Summary

B.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.

B.2.1.5 Subject Matter Semantics

The subject matter of Anomaly Detection is the expected and actual States of System_Elements.

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.

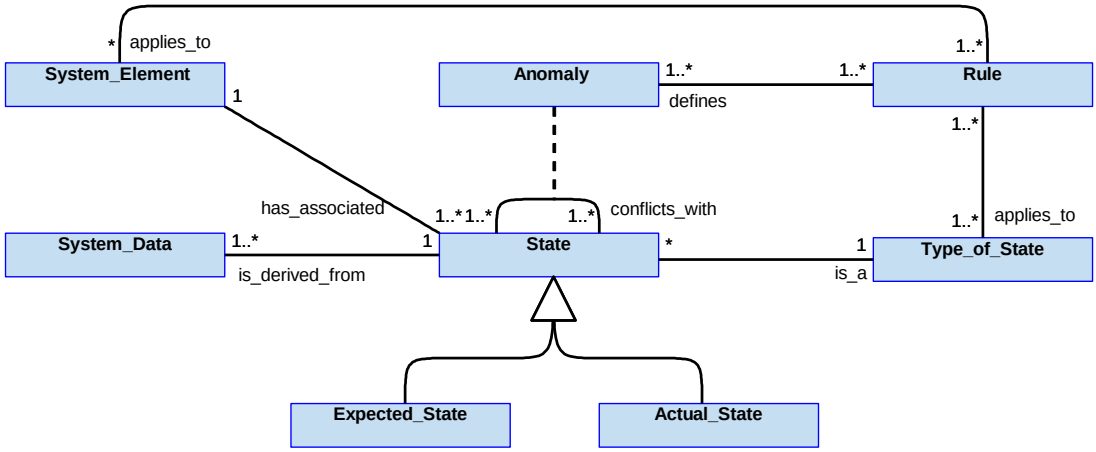

Figure 135: Anomaly Detection Semantics

B.2.1.5.1 Entities

Actual_State

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

Anomaly

An indication that part of a system is not in the expected State (or States) that may be a sign of a fault, damage or other degradation.

Expected_State

An expected State of an element of the system based on data generated by the system, including 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.

B.2.1.6 Design Rationale

B.2.1.6.1 Assumptions

- It is expected that instances of this component will be required to comply with ISO 13374 (Condition Monitoring and Diagnostics of Machines) Ref. [18]. See Health Management policy.
- 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.

B.2.1.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Anomaly Detection:

- Capability Assessment - Anomaly Detection supports implementation of the Capability Assessment policy, although Anomaly Detection itself does not provide an evolving view of its capabilities.
- Health Management - Anomaly Detection follows the Health Management policy. As such, it is likely, except for very simple deployments, that Anomaly Detection components will be deployed in multiple instances, with each instance being responsible for detecting anomalies in a particular component. 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

- It is not expected that this component will require extensions.

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 could be rule-based, with the specific rules data driven, 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.

B.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.

B.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.

Note: The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about the importance of detecting anomalous behaviour in ensuring the continued security of the Exploiting Platform.

B.2.1.7 Services

B.2.1.7.1 Service Definitions

B.2.1.7.1.1 Anomaly

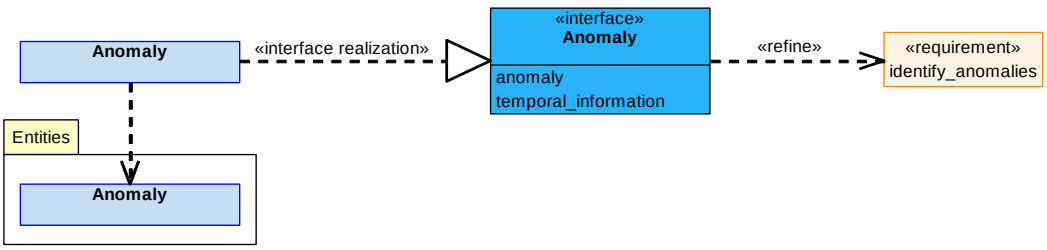

Figure 136: Anomaly Service Definition

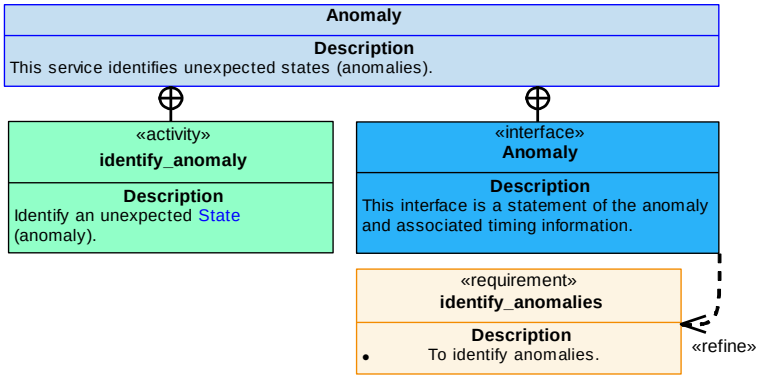

Figure 137: Anomaly Service Policy

Anomaly

This service identifies unexpected states (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 unexpected State (anomaly).

B.2.1.7.1.2 State_Evidence

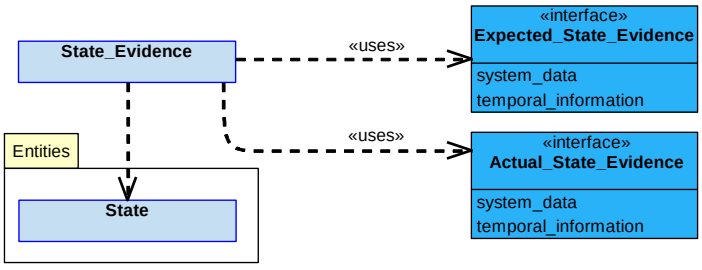

Figure 138: State_Evidence Service Definition

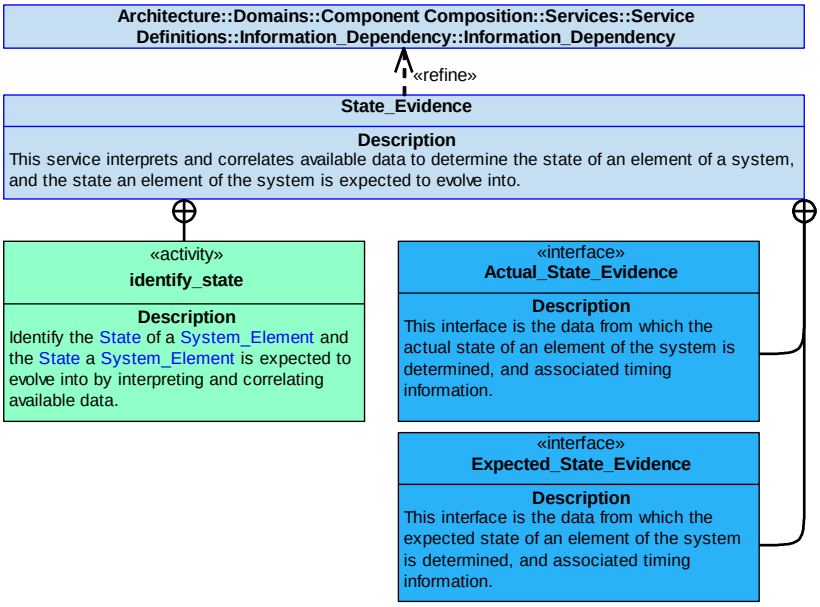

Figure 139: State_Evidence Service Policy

State_Evidence

This service interprets and correlates available data to determine the state of an element of a system, and the state an element of the system is expected to evolve into.

Interfaces

Expected_State_Evidence

This interface is the data from which the expected state of an element of the system is determined, and associated timing information.

Attributes

system_data	The data generated by the system, including commands, sensor data, etc. from which the expected state of an element of the system is determined.
temporal_information	The time at which the System_Data is expected to change.

Actual_State_Evidence

This interface is the data from which the actual state of an element of the system is determined, and associated timing information.

Attributes

system_data	The data generated by the system from which the actual state of an element of the system is determined.
temporal_information	The time at which the System_Data did change.

Activity

identify_state

Identify the State of a System_Element and the State a System_Element is expected to evolve into by interpreting and correlating available data.

B.2.1.7.2 Service Dependencies

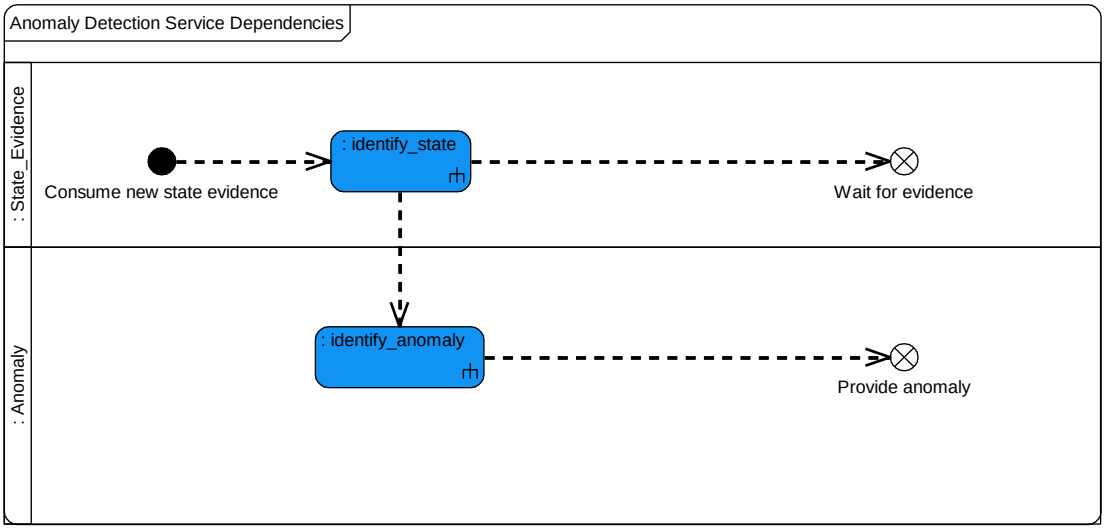

Figure 140: Anomaly Detection Service Dependencies

B.2.2 Asset Transitions

B.2.2.1 Role

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

B.2.2.2 Overview

Control Architecture

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

Standard Pattern of Use

When Asset Transitions obtains a Transition_Requirement, it may have many Transition_Solutions. Each Transition_Solution may contain one or more end State. The Asset transitions between Asset_States until the Transition_Solution has been achieved using only Allowable_Transitions. The Transition_Solution is measured against the Measurement_Criterion.

Examples of Use

Asset Transitions may be used when:

- Control of vehicle infrastructure State transitions is required, e.g. to prepare the vehicle for maintenance.
- Control of deployable asset State transitions is required, e.g. to prepare role fit equipment for release.
- Back-up systems are available to provide system redundancy in failure situations, e.g. reversionary capability on a processor to cover safety-critical functions.
- Control of power State transitions, e.g. to identify the need to apply power and ensure power is applied at the correct time relative to other activities that are required.

B.2.2.3 Service Summary

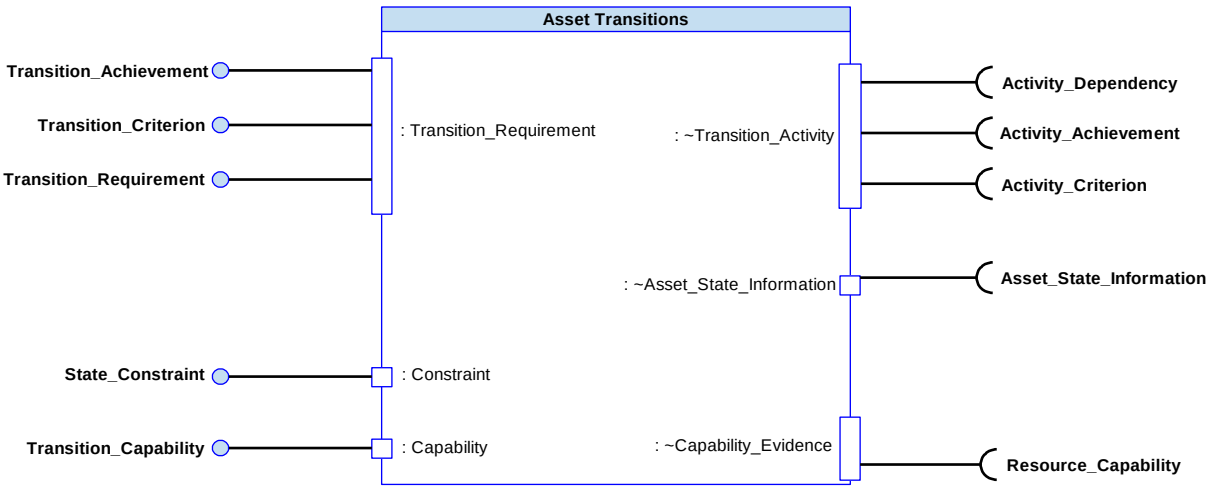

Figure 141: Asset Transitions Service Summary

B.2.2.4 Responsibilities

capture_transition_requirements

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

capture_state_constraints

- To capture given State_Constraints on a Transition_Solution.

assess_capability

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

determine_infrastructure_states

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

determine_transition_solution

- To determine a Transition_Solution to achieve desired States.

implement_transition_solution

- To implement a Transition_Solution.

identify_transition_solution_progress

- To identify the progress of an implementation of a Transition_Solution.

identify_states

- To maintain a view of the current States.

predict_capability_progression

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

identify_missing_information

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

capture_measurement_criteria

- To capture given Measurement_Criterion/criteria.

determine_quality_of_transition_execution

- To determine the quality of the execution of a Transition_Solution against the Measurement_Criterion/criteria.

determine_quality_of_transition_solution

- To determine the quality of a 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.

B.2.2.5 Subject Matter Semantics

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

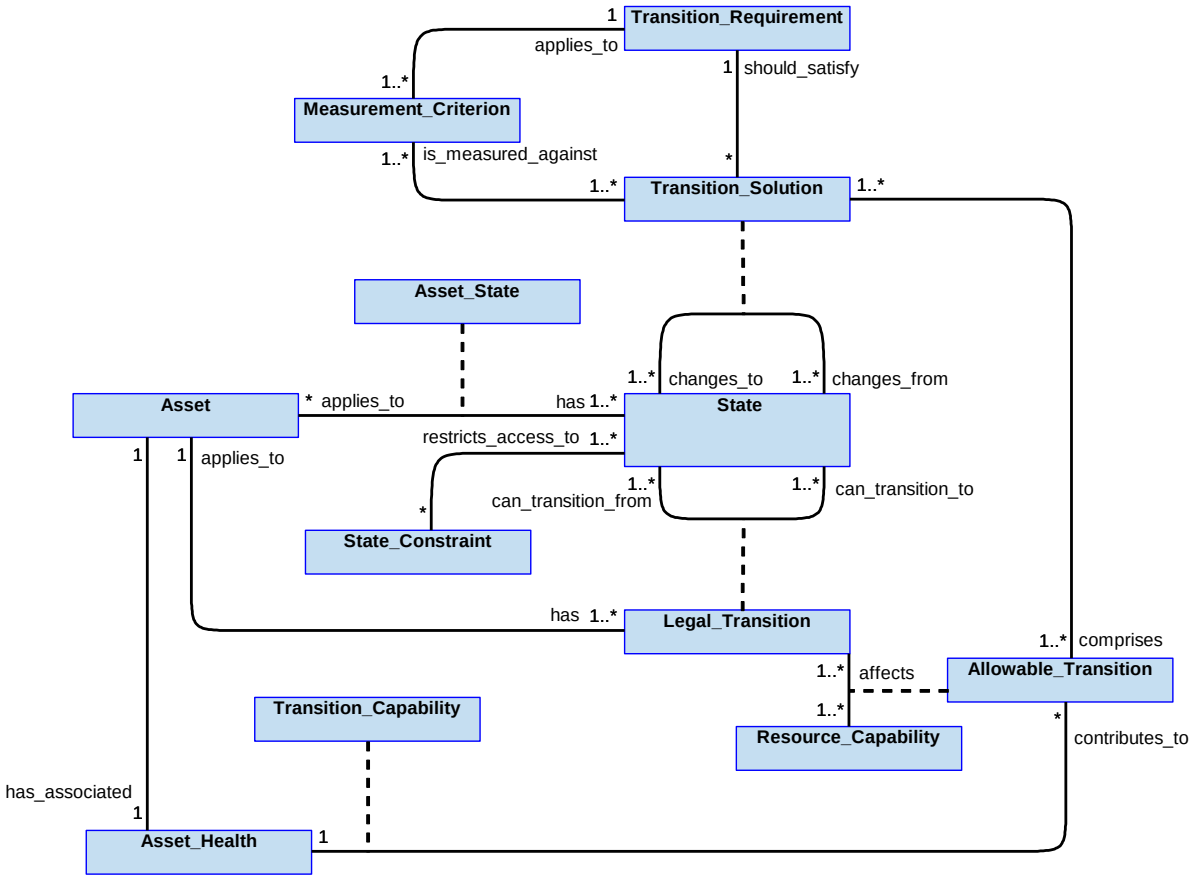

Figure 142: Asset Transitions Semantics

B.2.2.5.1 Entities

Asset_State

The Asset's current State.

Asset

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

Measurement_Criterion

A criterion by which the Allowable_Transition will be measured (e.g. the speed of a handover from a primary to a secondary process).

State

A condition of an Asset, e.g. on or off, primary or secondary.

Transition_Requirement

A requirement to achieve the desired State of an Asset.

Transition_Solution

A sequence of one or more Allowable_Transitions required to attain the desired State of an Asset.

Legal_Transition

A legal transition which an Asset may perform, e.g. for a particular Asset it may be legal to transition from "OFF" to "STANDBY" or from "STANDBY" to "ON", but not directly from "OFF to "ON".

Transition_Capability

The Allowable_Transitions that can be performed given the Asset_Health.

Asset_Health

The health of an Asset e.g. damage to a bay door.

Allowable_Transition

The Legal_Transitions that can be performed with the current Resource_Capability.

State_Constraint

A constraint that prevents an Asset from performing a Legal_Transition. For example, preventing the landing gear from being deployed if there is a need to reduce the observability of the Exploiting Platform.

Resource_Capability

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

B.2.2.6 Design Rationale

B.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.

B.2.2.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

Exploitation Considerations

- It is expected that there will be different instances of this component handling different Assets or groups of Assets, however some, if not all, of these instances will need to be coordinated.

B.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.

B.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 Asset_States 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.

B.2.2.7 Services

B.2.2.7.1 Service Definitions

B.2.2.7.1.1 Transition_Requirement

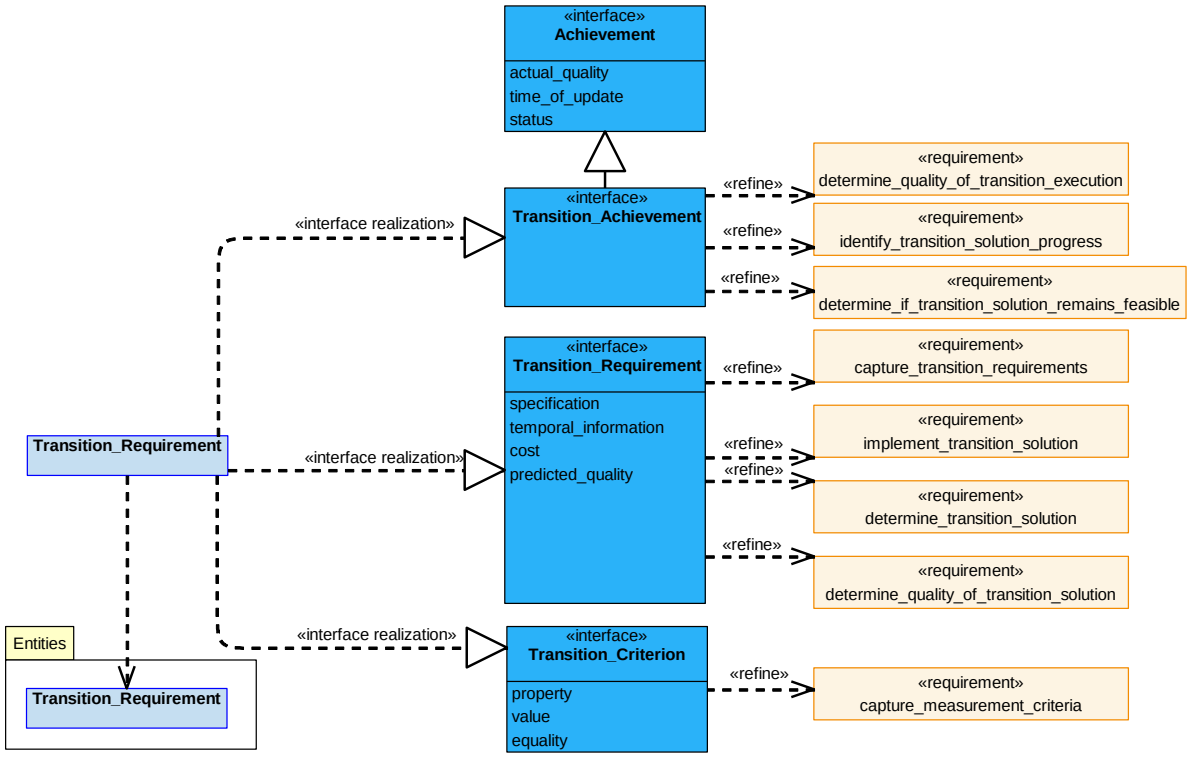

Figure 143: Transition_Requirement Service Definition

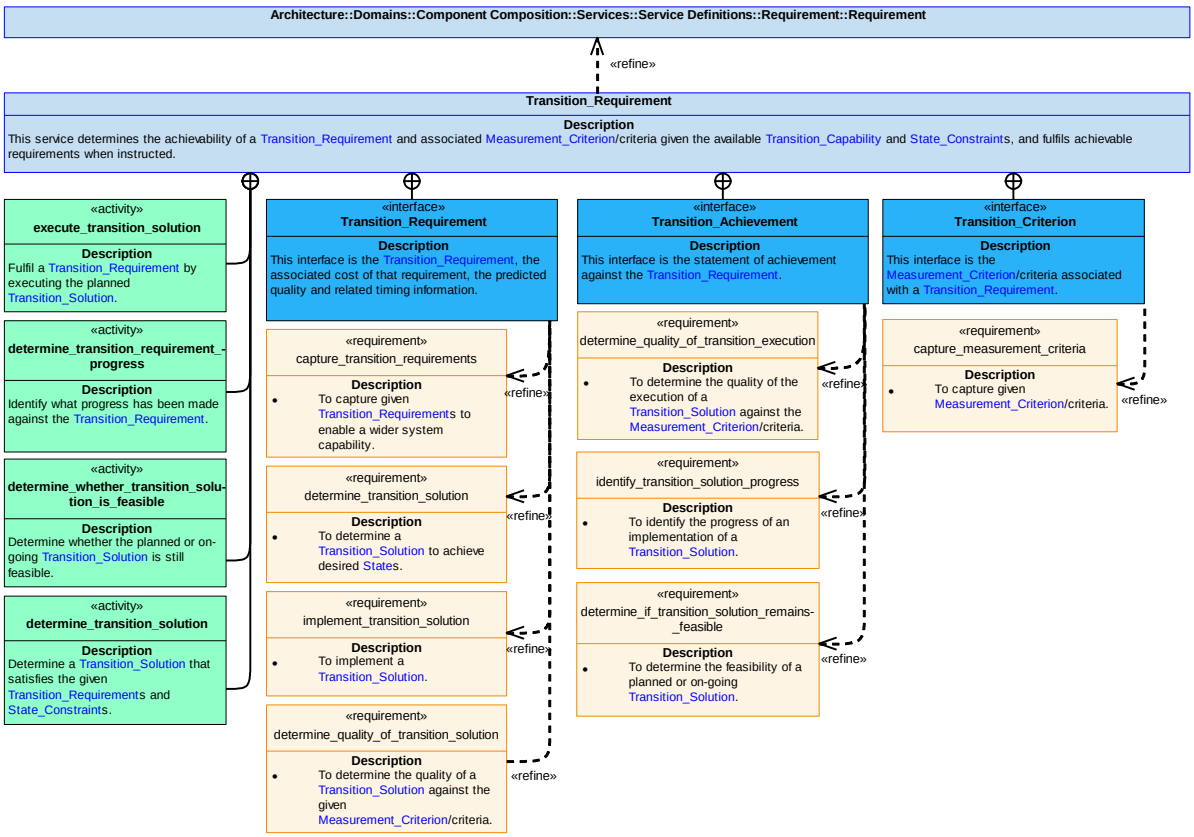

Figure 144: Transition_Requirement Service Policy

Transition_Requirement

This service determines the achievability of a Transition_Requirement and associated Measurement_Criterion/criteria given the available Transition_Capability and State_Constraints, 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, for example: 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 State_Constraints.

B.2.2.7.1.2 Transition_Activity

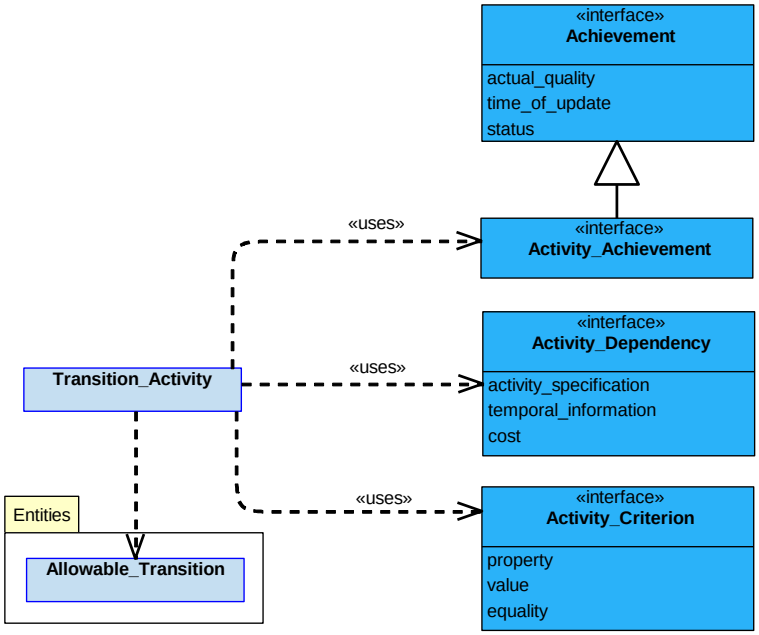

Figure 145: Transition_Activity Service Definition

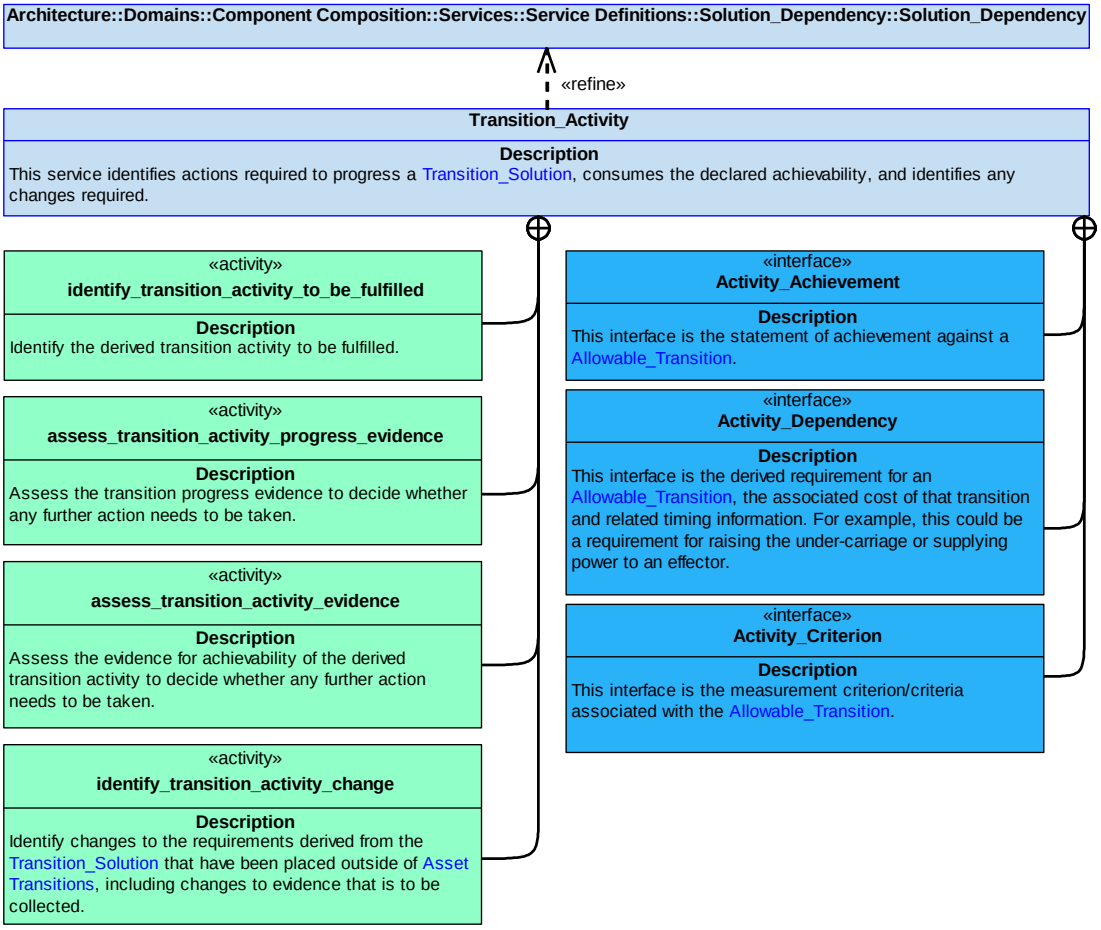

Figure 146: Transition_Activity Service Policy

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 an Allowable_Transition, 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 to be performed 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, for example: resources used, time taken.

Activity_Criterion

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

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 Allowable_Transition.

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.

B.2.2.7.1.3 Asset_State_Information

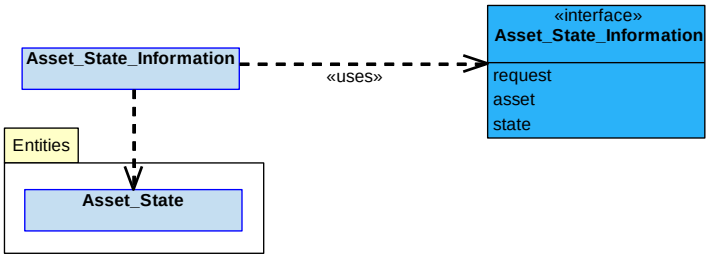

Figure 147: Asset_State_Information Service Definition

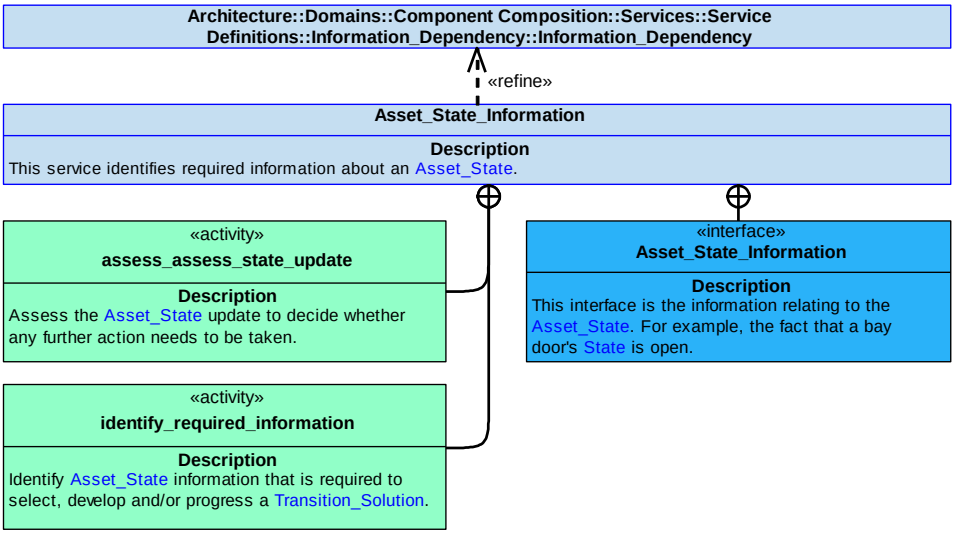

Figure 148: Asset_State_Information Service Policy

Asset_State_Information

This service identifies required information about an Asset_State.

Interface

Asset_State_Information

This interface is the information relating to the Asset_State. For example, the fact that a bay door's State is open.

Attributes

request	The request for information about an Asset_State.
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.

Activities

assess_assess_state_update

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

identify_required_information

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

B.2.2.7.1.4 Constraint

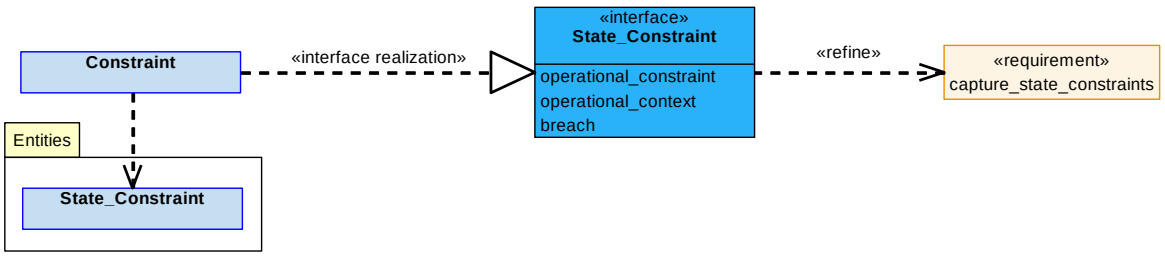

Figure 149: Constraint Service Definition

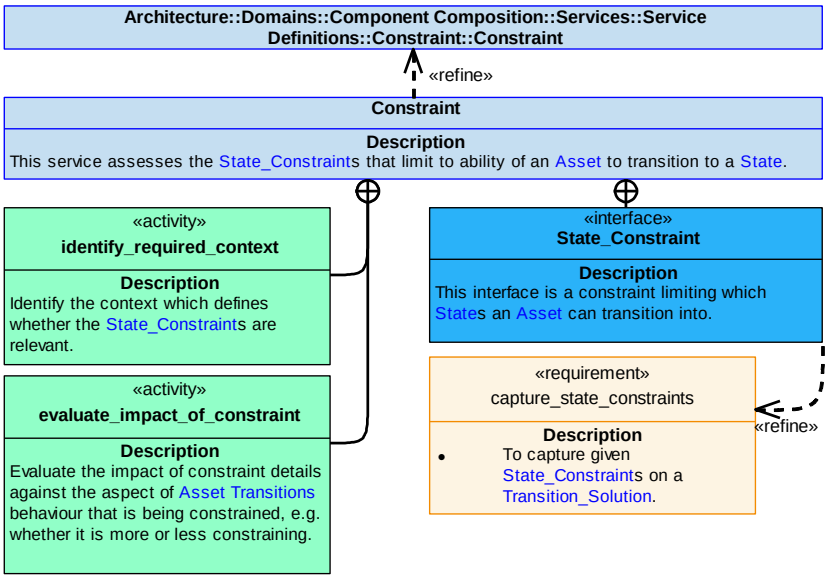

Figure 150: Constraint Service Policy

Constraint

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

Interface

State_Constraint

This interface is a constraint limiting which States an Asset can transition into.

Attributes

operational_constraint	A constraint which prevents a Legal_Transition. For example, 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 State_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 State_Constraints are relevant.

B.2.2.7.1.5 Capability

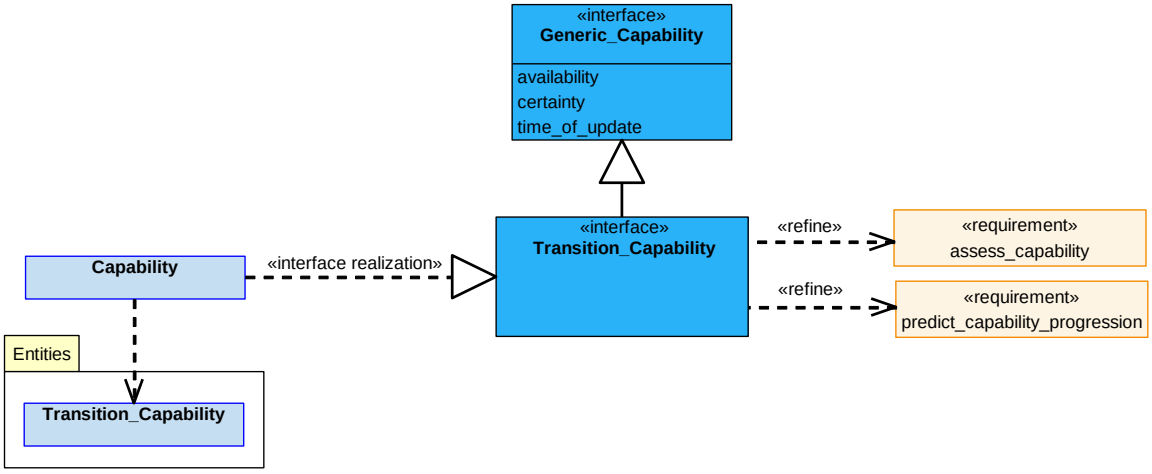

Figure 151: Capability Service Definition

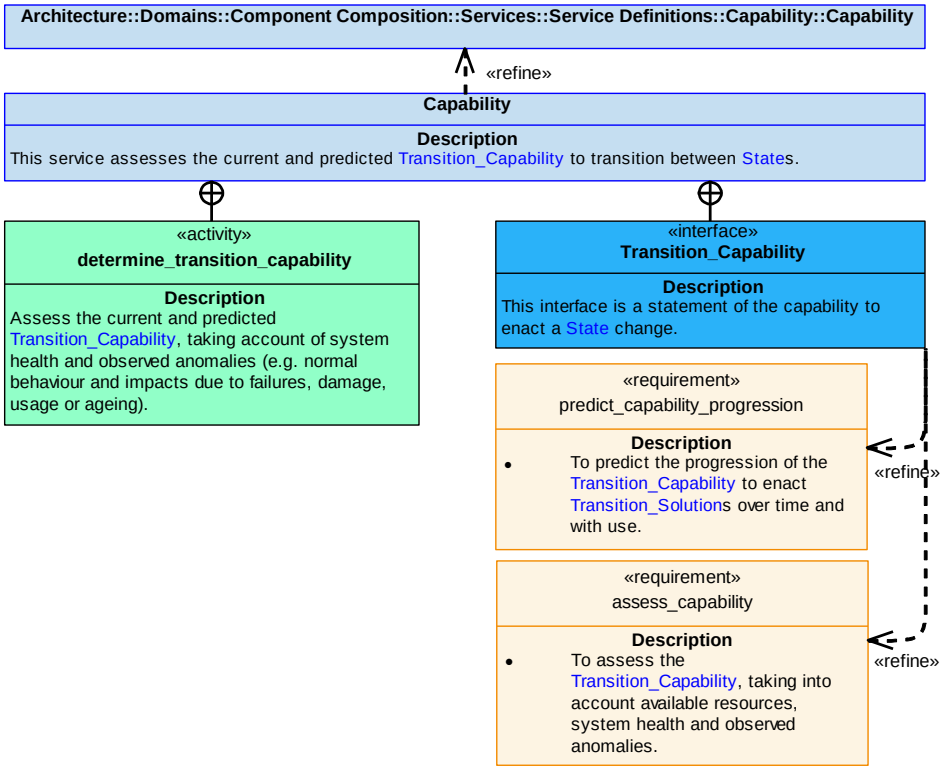

Figure 152: Capability Service Policy

Capability

This service assesses the current and predicted Transition_Capability to transition between States.

Interface

Transition_Capability

This interface is a statement of the capability to enact a State change.

Activity

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).

B.2.2.7.1.6 Capability_Evidence

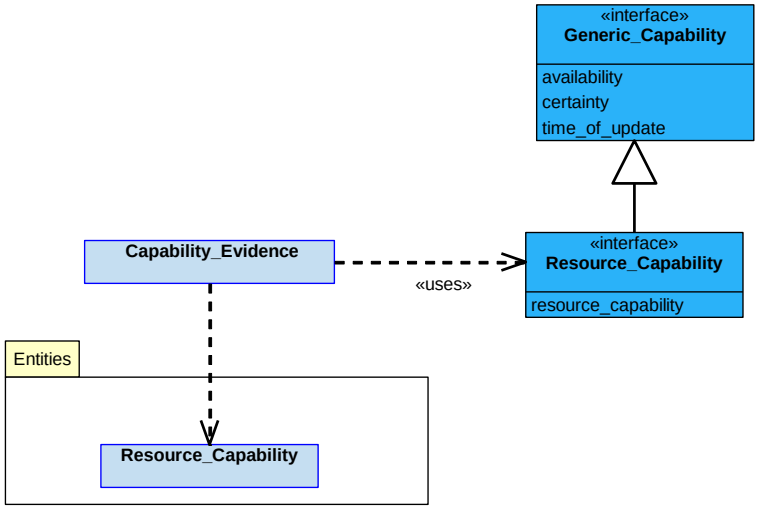

Figure 153: Capability_Evidence Service Definition

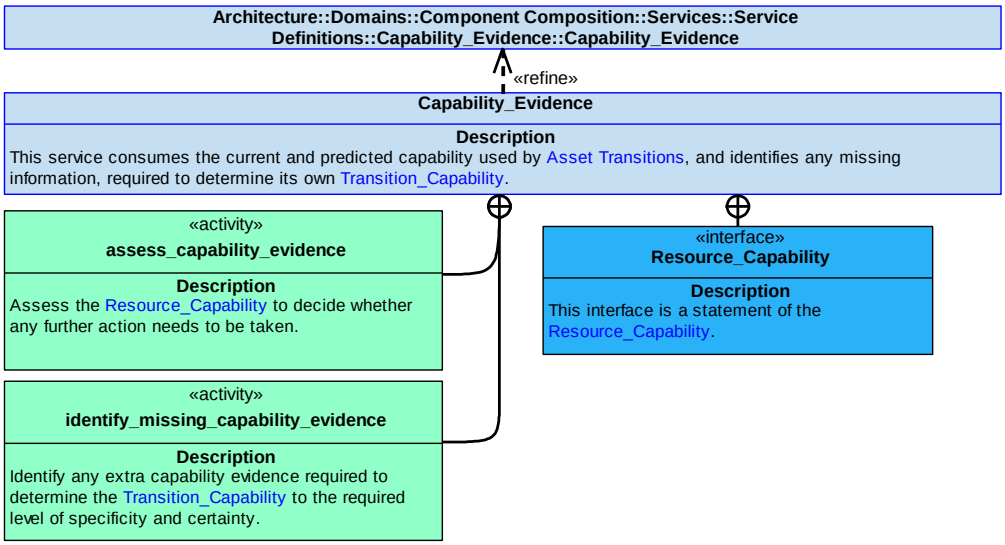

Figure 154: Capability_Evidence Service Policy

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.

Interface

Resource_Capability

This interface is a statement of the Resource_Capability.

Attribute

resource_capability	The capability of an Exploiting Platform to provide resources to enable a Legal_Transition. For example, to provide the required power.

Activities

assess_capability_evidence

Assess the Resource_Capability 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.

B.2.2.7.2 Service Dependencies

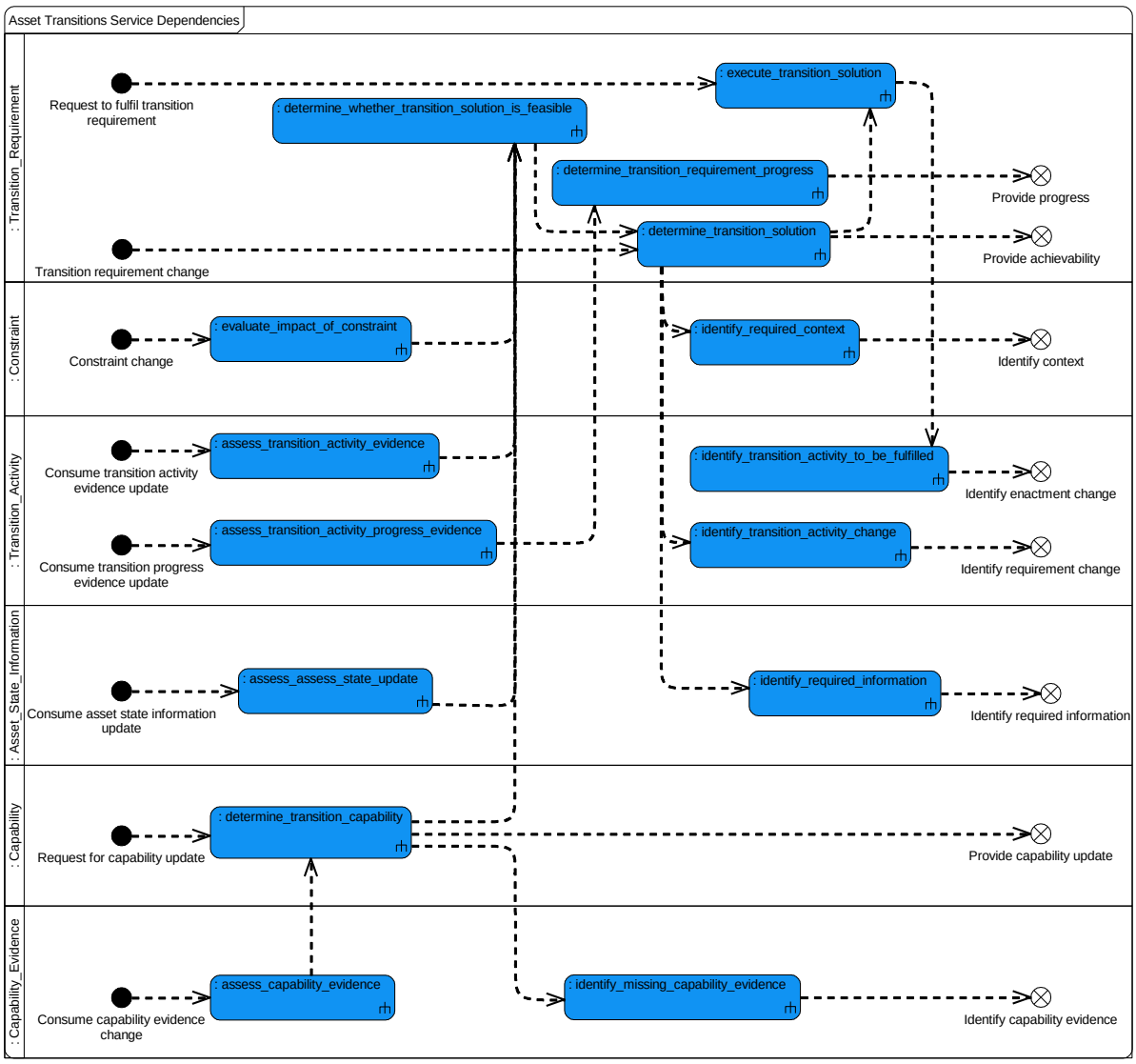

Figure 155: Asset Transitions Service Dependencies

B.2.3 Authorisation

B.2.3.1 Role

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

B.2.3.2 Overview

Control Architecture

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

Standard Pattern of Use

When an Authorisation requester plans an action 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 actions where a level of approval is required, e.g. weapon release, entering restricted zones.

B.2.3.3 Service Summary

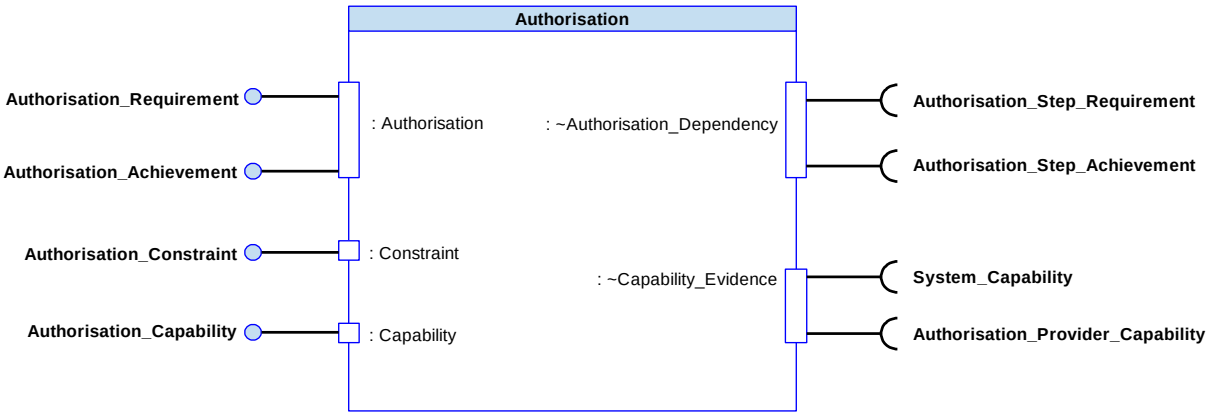

Figure 156: Authorisation Service Summary

B.2.3.4 Responsibilities

capture_authoriser_availability

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

determine_authorisation_policy

- To determine the active Authorisation_Policy.

capture_constraints_for_authorisations

- To capture Constraints related to obtaining Authorisations.

capture_requirements_for_authorisations

- To capture Authorisation_Requirements.

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).

predict_capability_progression

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

determine_permitted_authorisers

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

coordinate_authorisation_solution

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

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).

identify_pre-conditions

- To identify pre-conditions to support 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_if_authorisation_remains_feasible

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

B.2.3.5 Subject Matter Semantics

The subject matter of Authorisation is the conditions that govern explicit Authorisation of actions: 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 actions.

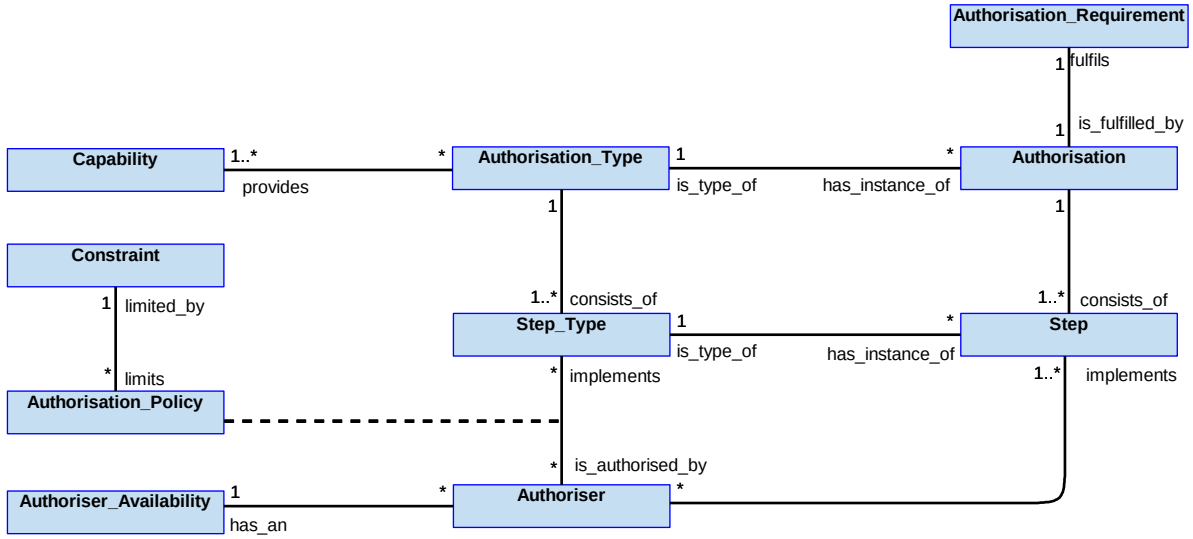

Figure 157: Authorisation Semantics

B.2.3.5.1 Entities

Authorisation

Approval to perform actions 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 action or group of actions.

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.

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)

B.2.3.6 Design Rationale

B.2.3.6.1 Assumptions

- A single act of Authorisation may be applied to a group of related actions, e.g. authorising an attack plan could authorise each action 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 actions 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.

B.2.3.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy) is critical for the Authorisation of actions by this component.

B.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 actions could be authorised that result in catastrophic accidents. Where an action could result in a catastrophic outcome it is expected that additional barriers (within the system) would also prevent the action 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.

B.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 actions 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.

B.2.3.7 Services

B.2.3.7.1 Service Definitions

B.2.3.7.1.1 Authorisation

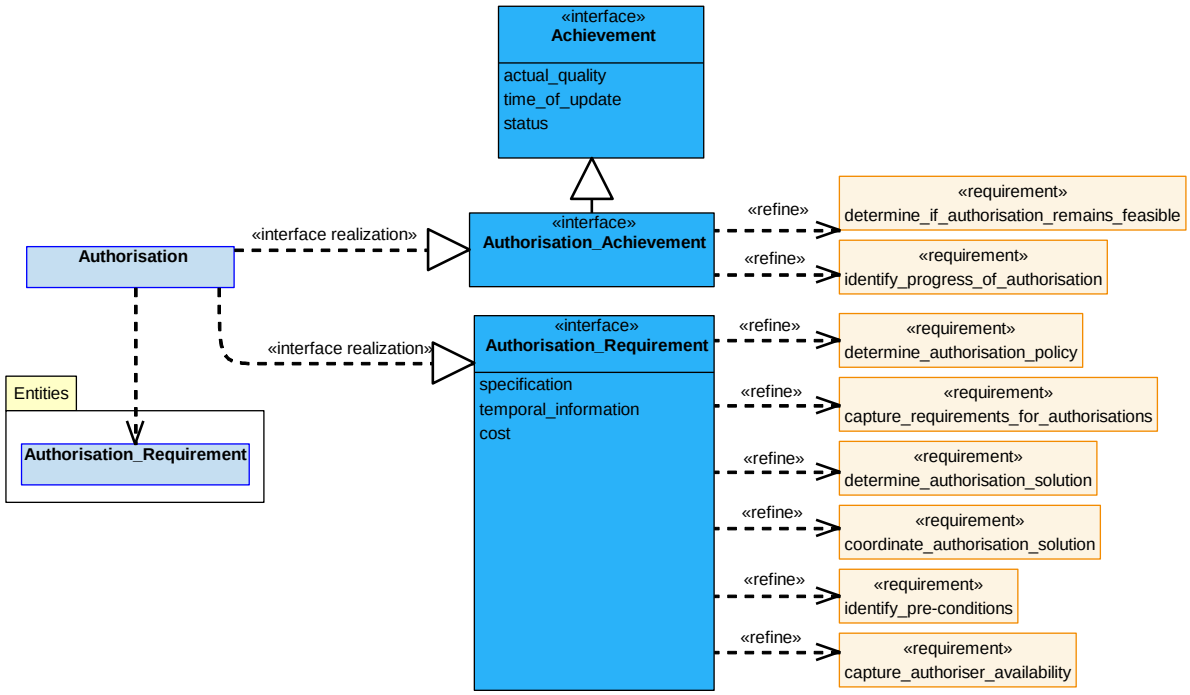

Figure 158: Authorisation Service Definition

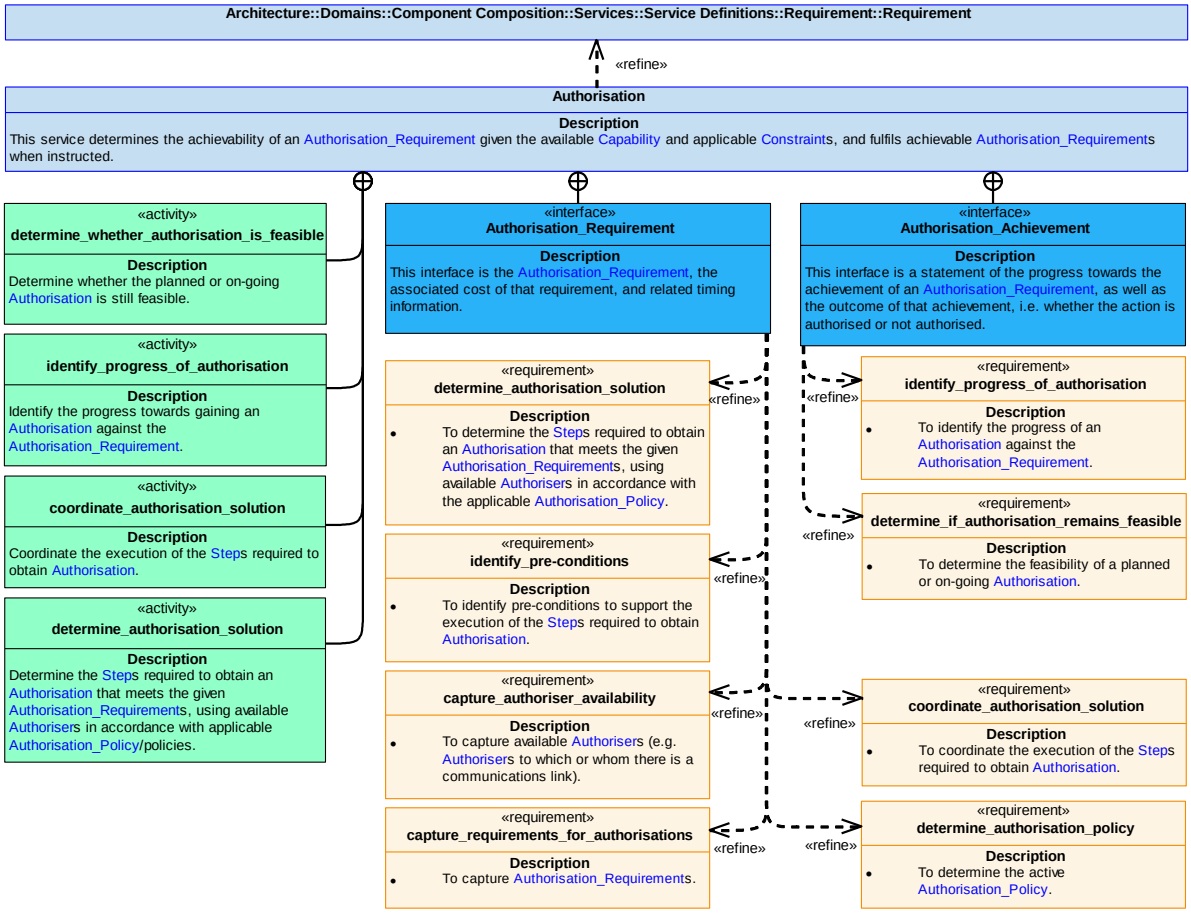

Figure 159: Authorisation Service Policy

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 action 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.

B.2.3.7.1.2 Authorisation_Dependency

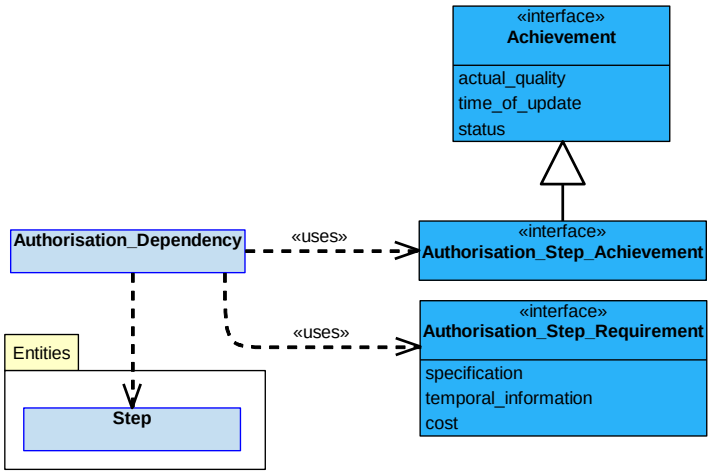

Figure 160: Authorisation_Dependency Service Definition

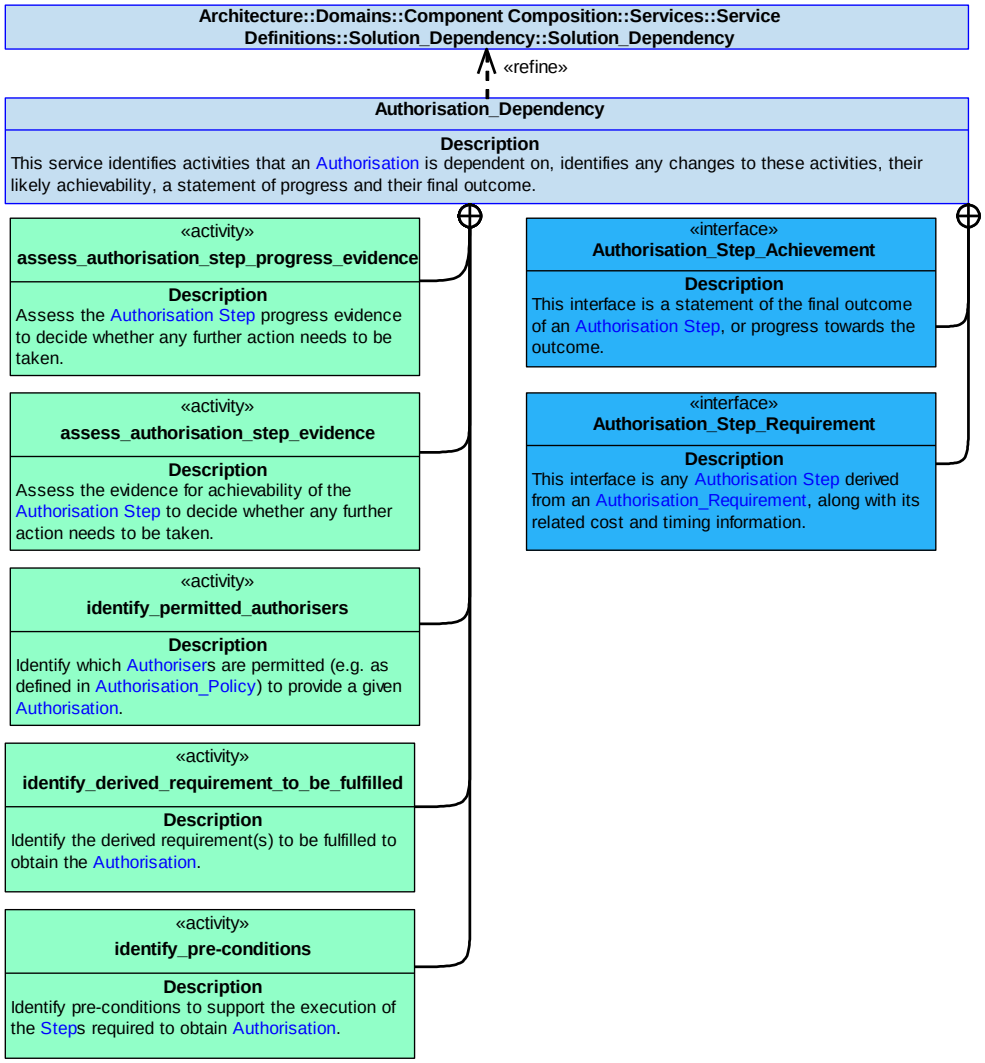

Figure 161: Authorisation_Dependency Service Policy

Authorisation_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.

Interfaces

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.
cost	The cost of executing the solution, for example: resources used and time taken.

Authorisation_Step_Achievement

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

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.

identify_pre-conditions

Identify pre-conditions 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.

B.2.3.7.1.3 Constraint

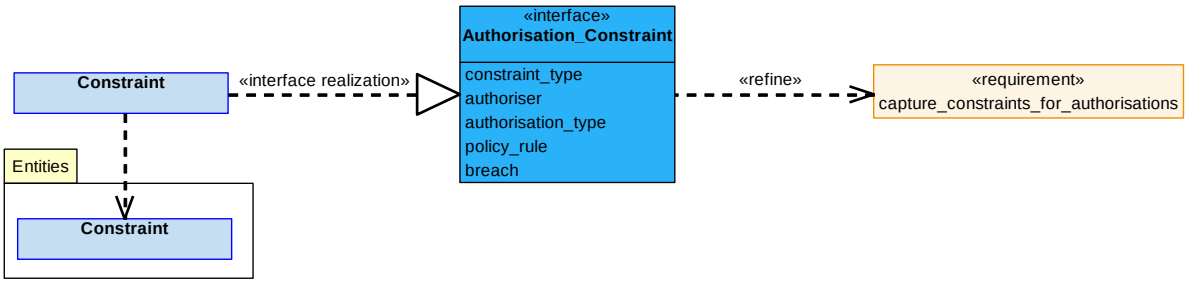

Figure 162: Constraint Service Definition

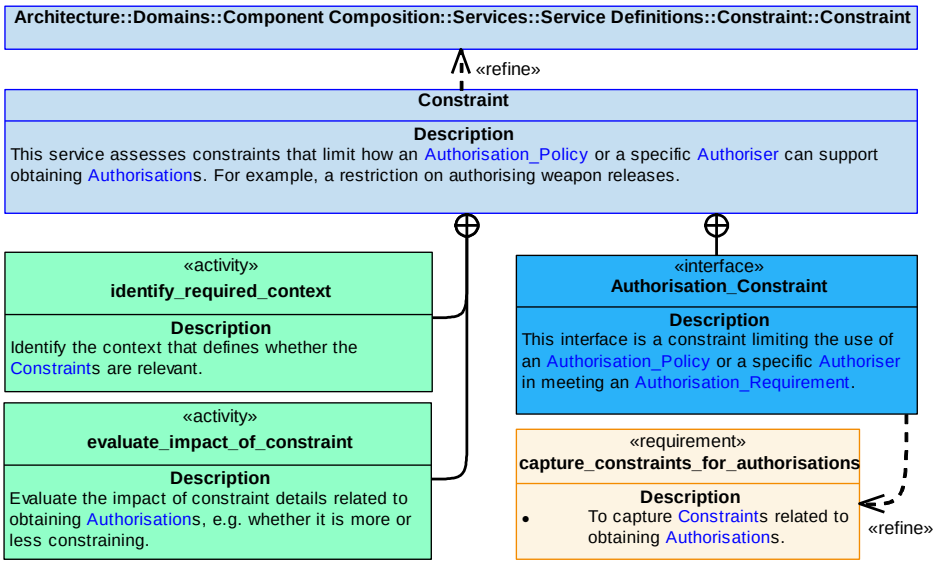

Figure 163: Constraint Service Policy

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

constraint_type	Type of restrictions to be applied to the use of the Authorisation_Policy, e.g. restrict certain Authorisers, restrict certain Authorisation_Types or restrict certain Authorisation_Policy rules.
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.

B.2.3.7.1.4 Capability

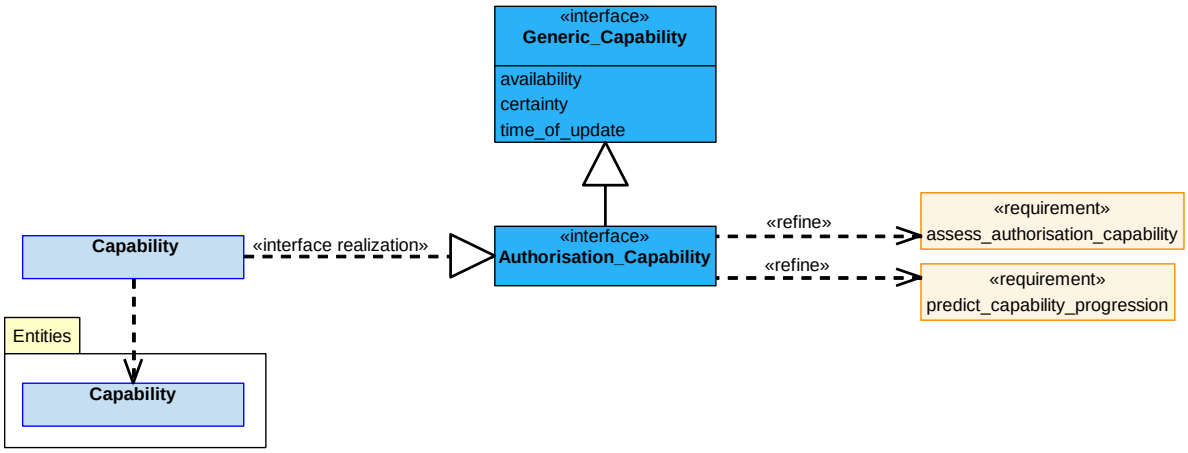

Figure 164: Capability Service Definition

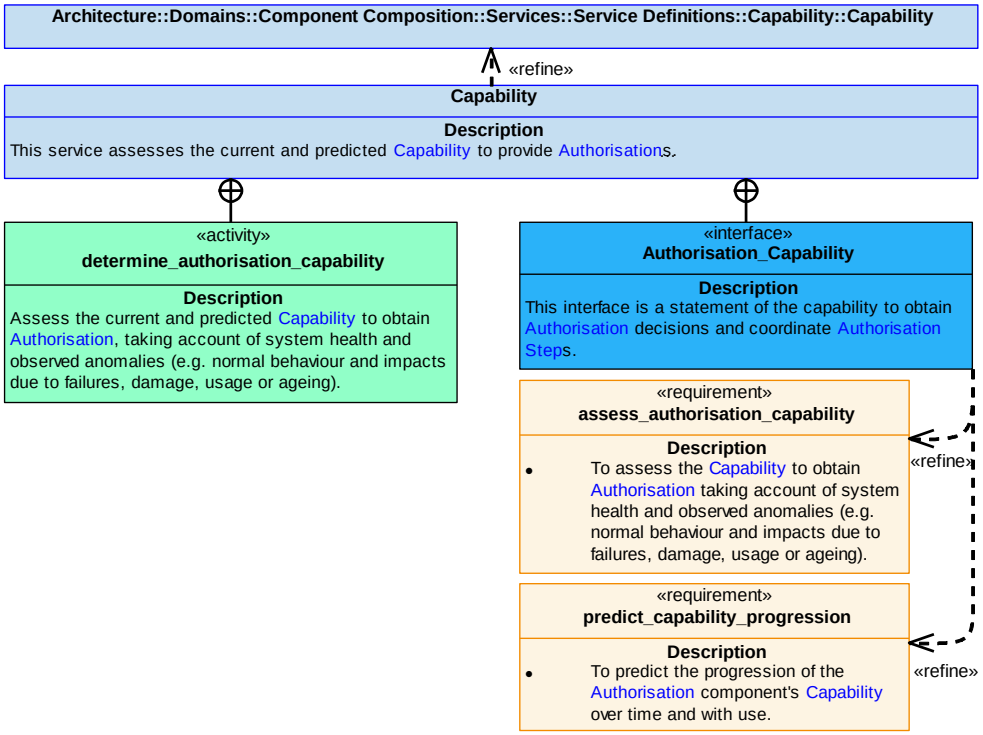

Figure 165: Capability Service Policy

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 account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

B.2.3.7.1.5 Capability_Evidence

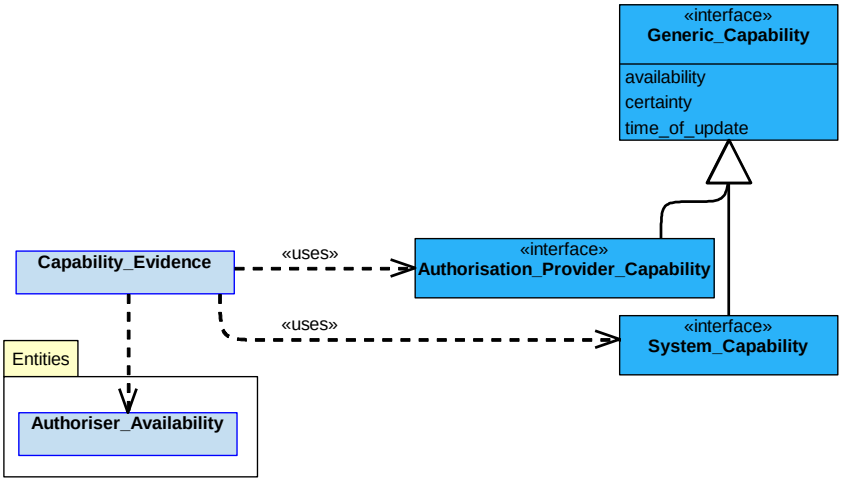

Figure 166: Capability_Evidence Service Definition

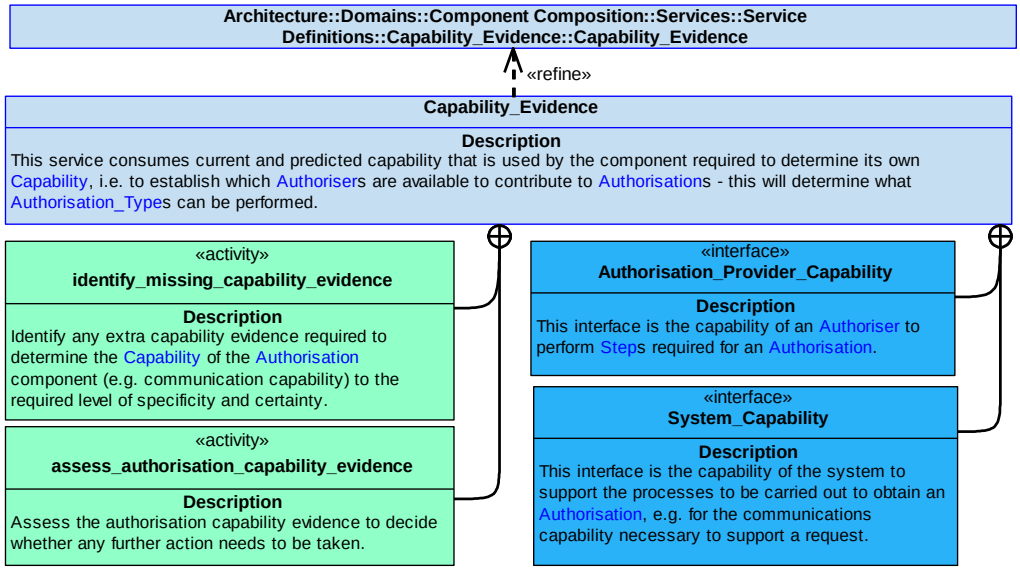

Figure 167: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability that is used by the component required to determine its own Capability, i.e. 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.

B.2.3.7.2 Service Dependencies

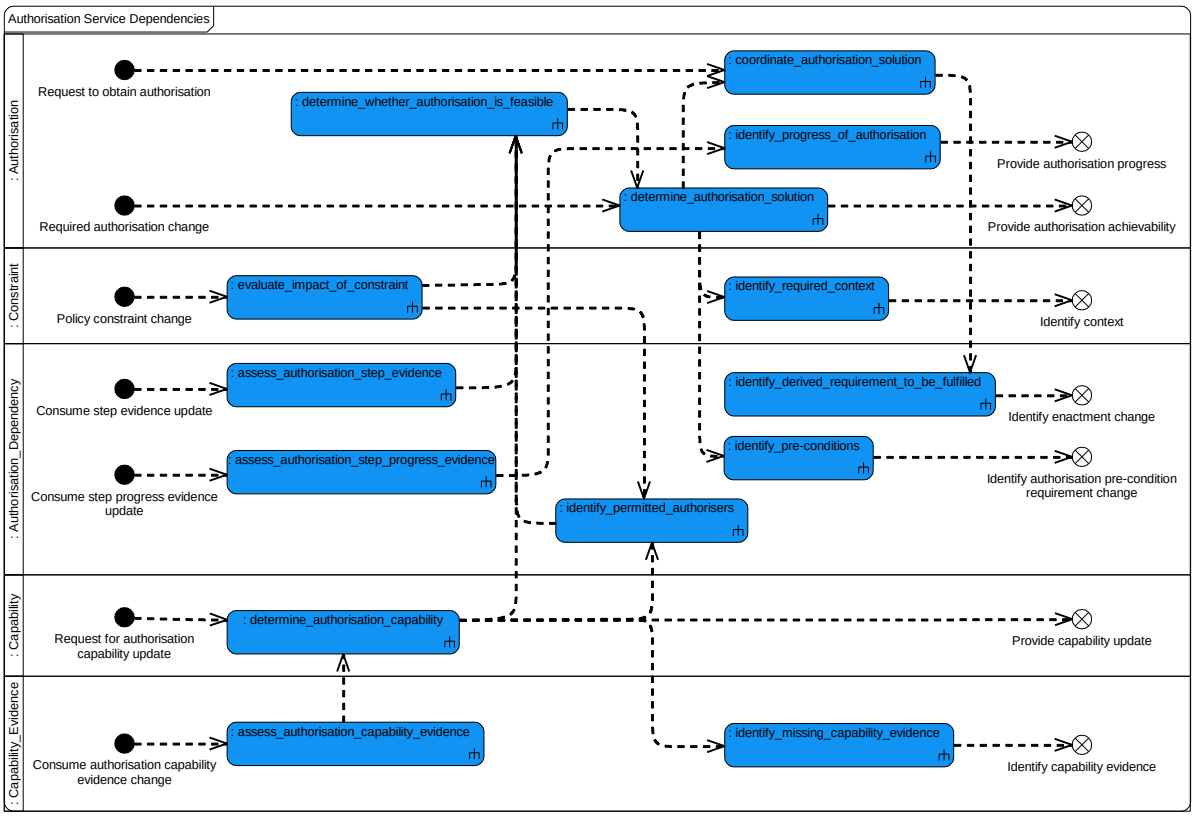

Figure 168: Authorisation Service Dependencies

B.2.4 Collision Avoidance

B.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.

B.2.4.2 Overview

Control Architecture

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

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).

B.2.4.3 Service Summary

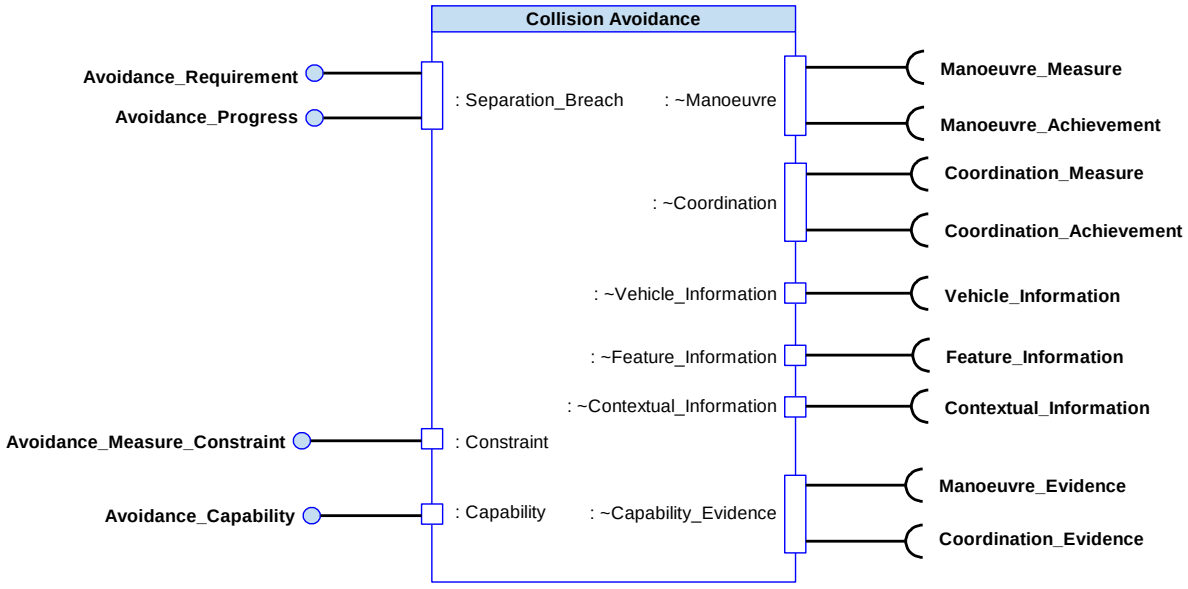

Figure 169: Collision Avoidance Service Summary

B.2.4.4 Responsibilities

determine_avoidance_measure

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

capture_separation_breach

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

identify_progress_of_avoidance_measure

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

identify_avoidance_measure_in_progress_remains_feasible

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

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).

predict_progression_of_avoidance_capability

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

identify_missing_information

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

capture_avoidance_constraints

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

B.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.

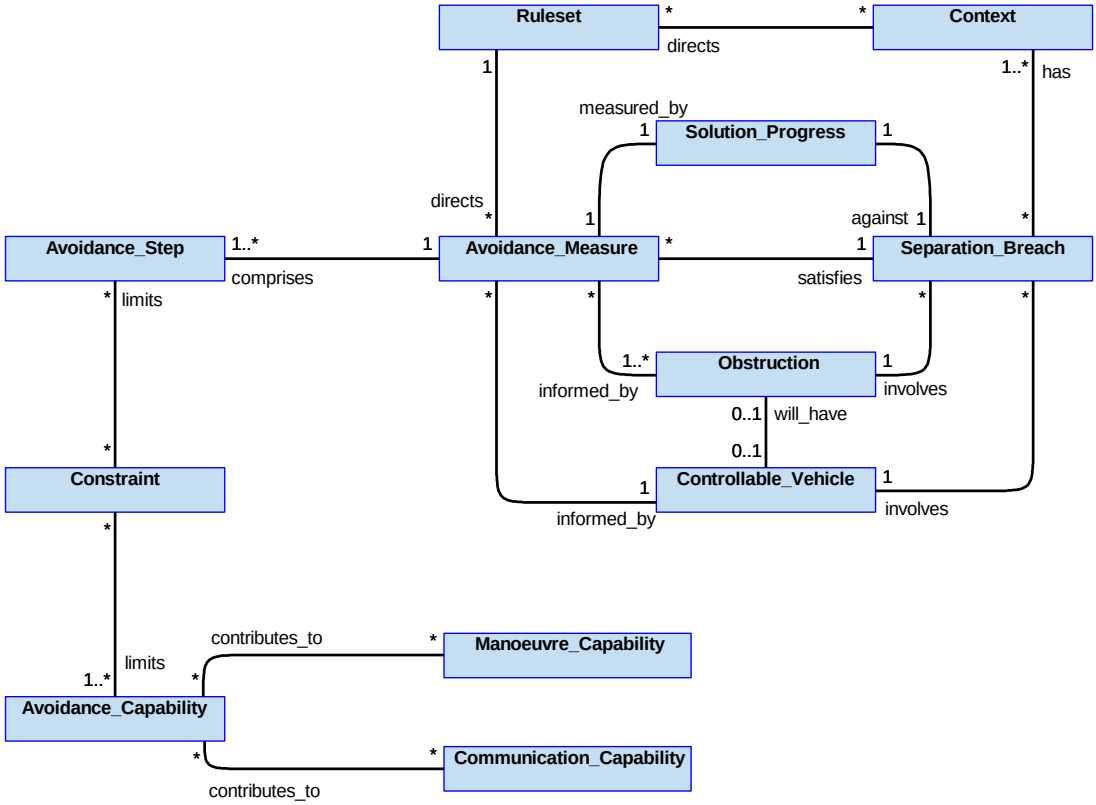

Figure 170: Collision Avoidance Semantics

B.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, an unmanned adjunct 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.

B.2.4.6 Design Rationale

B.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.

B.2.4.6.2 Design Considerations

Directly Applicable Policies

These policies 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 unmanned adjunct. In such a case, cooperative avoidance may include direct control of both ownship and the adjunct.
- 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. The rules will also dictate which avoidance manoeuvre to pass to the path demand arbitrator, should more than one potential or actual breach be determined.
- Once Collision Avoidance has identified the manoeuvre, it passes it to the path demand arbitrator, should a deployment be architected that way. This allows the path demand arbitrator to take into account mission and task objectives when determining if the collision avoidance manoeuvre is appropriate. For example, Collision Avoidance may determine an avoidance manoeuvre to prevent collision between an air vehicle and a building, but the path demand arbitrator will reject this avoidance manoeuvre if the mission objective 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).

B.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.

B.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.

B.2.4.7 Services

B.2.4.7.1 Service Definitions

B.2.4.7.1.1 Separation_Breach

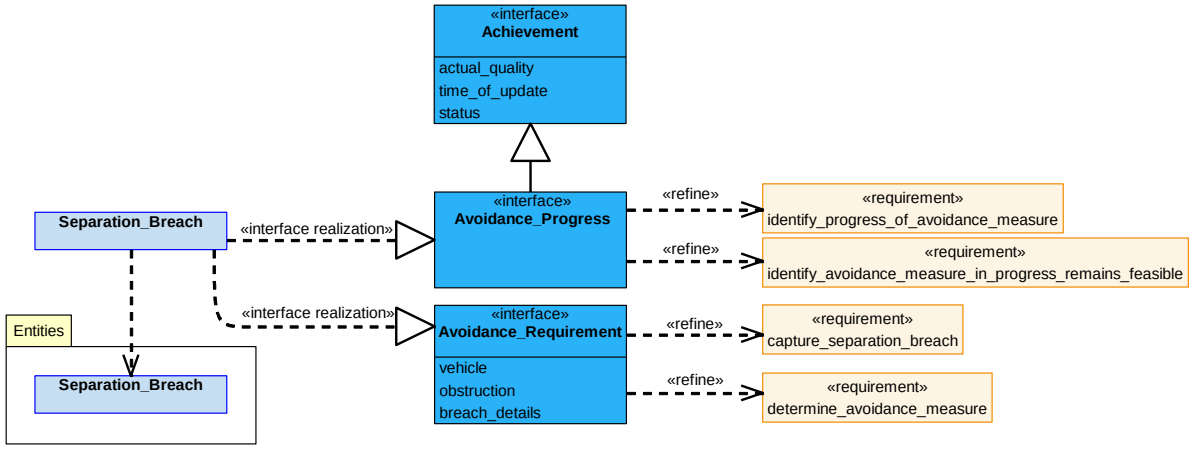

Figure 171: Separation_Breach Service Definition

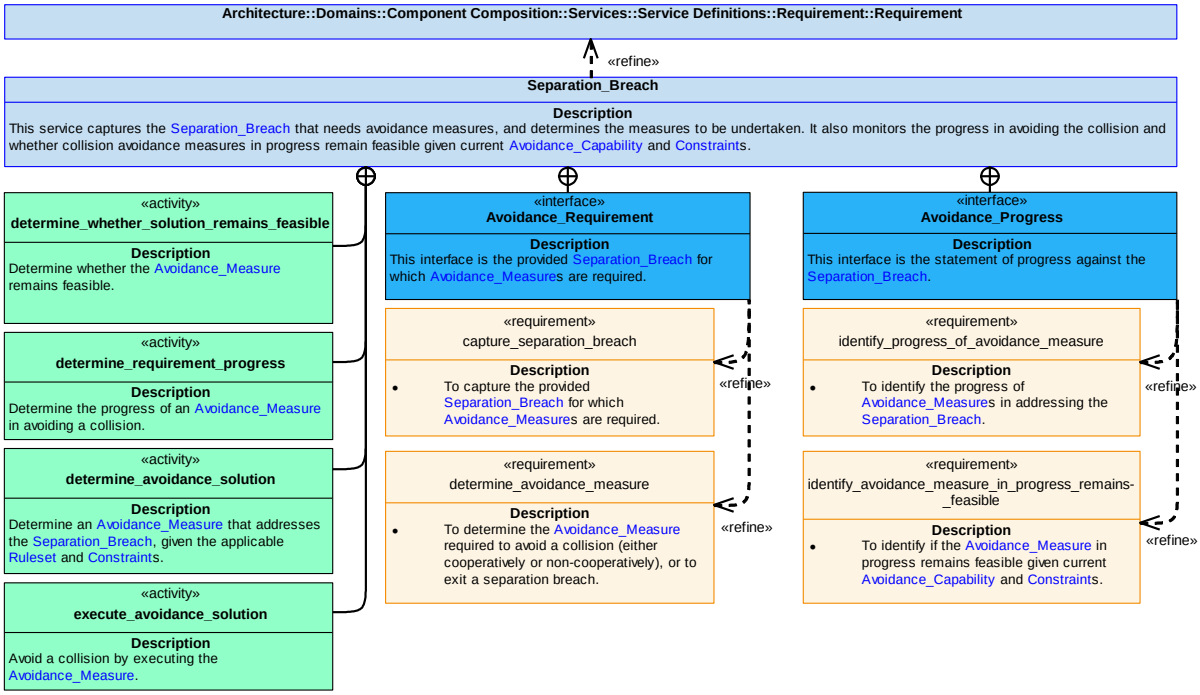

Figure 172: Separation_Breach Service Policy

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 an adjunct.
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.

B.2.4.7.1.2 Manoeuvre

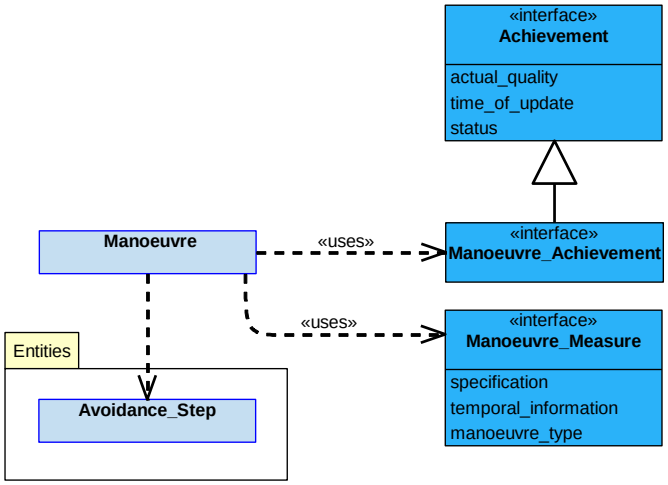

Figure 173: Manoeuvre Service Definition

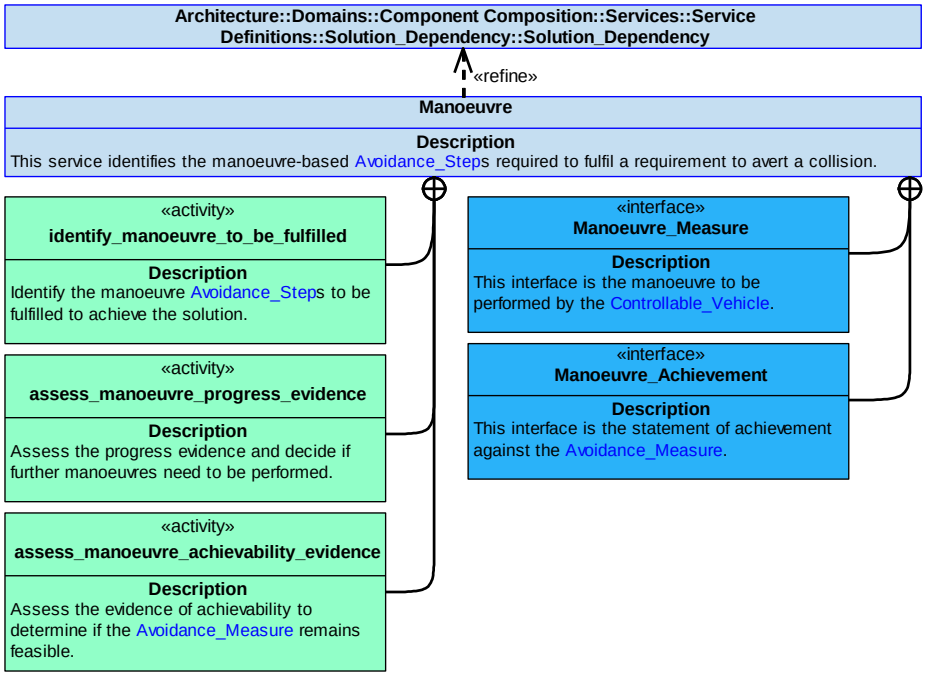

Figure 174: Manoeuvre Service Policy

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.

B.2.4.7.1.3 Coordination

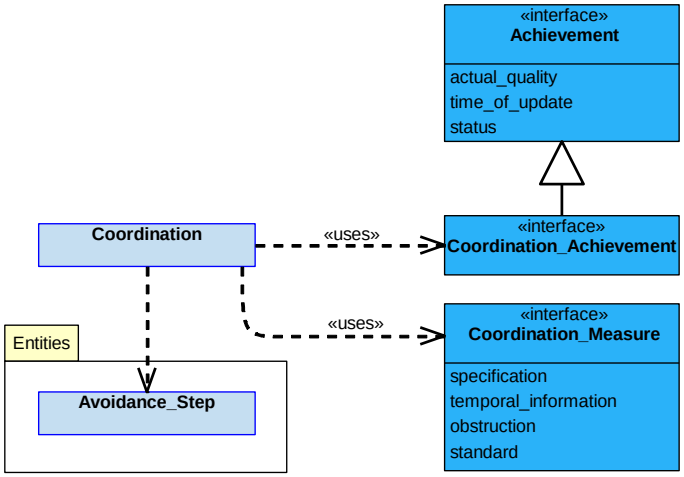

Figure 175: Coordination Service Definition

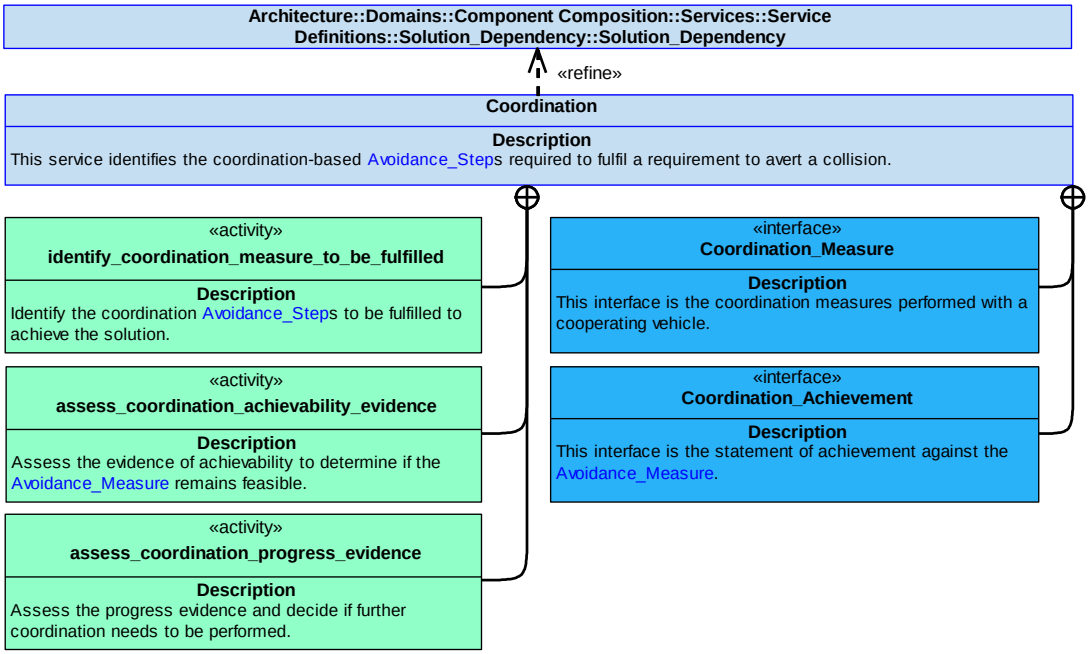

Figure 176: Coordination Service Policy

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.

B.2.4.7.1.4 Vehicle_Information

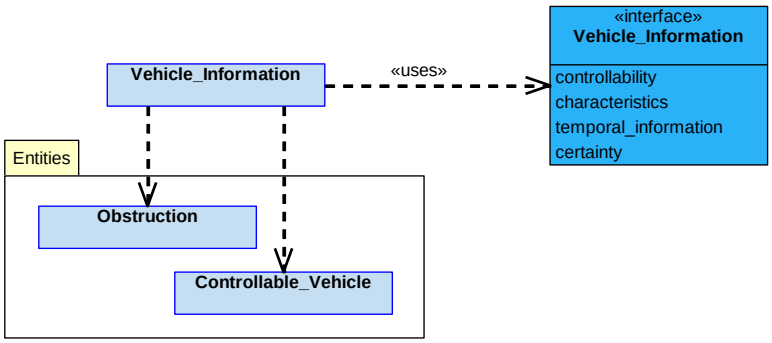

Figure 177: Vehicle_Information Service Definition

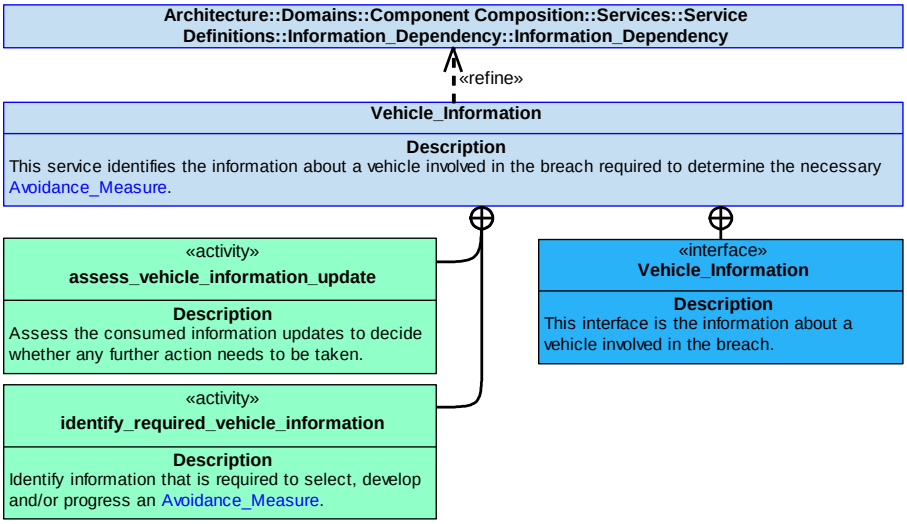

Figure 178: Vehicle_Information Service Policy

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.

B.2.4.7.1.5 Feature_Information

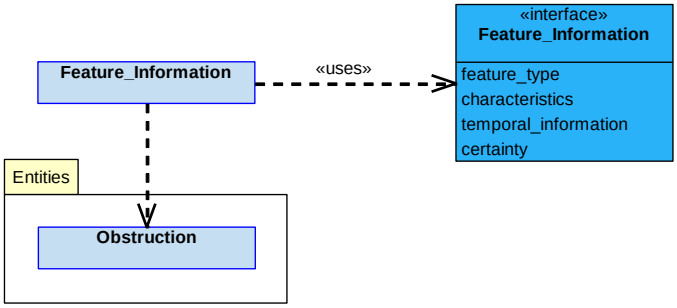

Figure 179: Feature_Information Service Definition

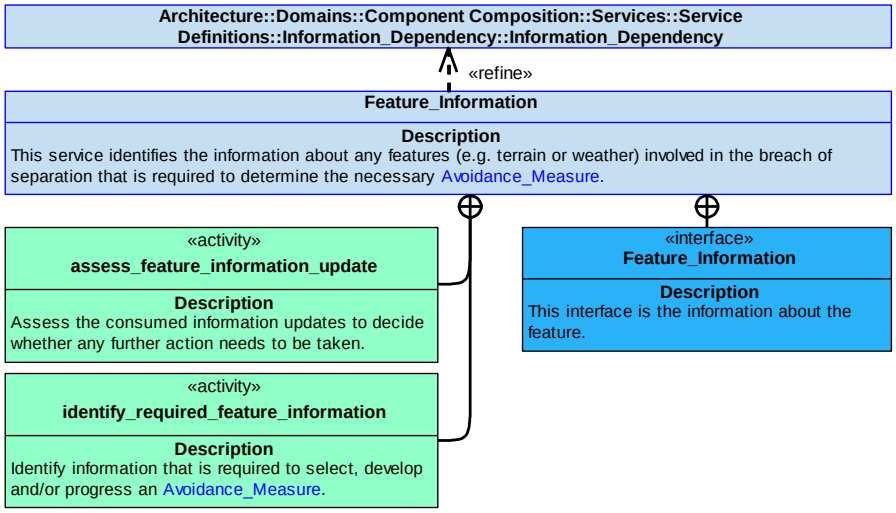

Figure 180: Feature Information Service Policy

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.

Interface

Feature_Information

This interface is the information about the feature.

Attributes

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.

Activities

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.

B.2.4.7.1.6 Contextual_Information

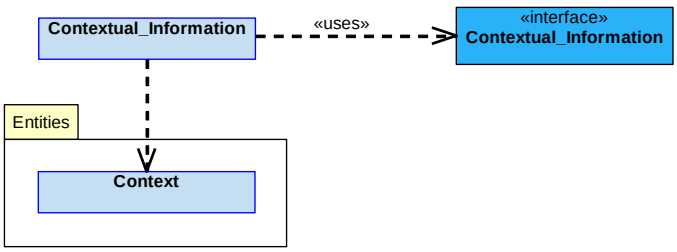

Figure 181: Contextual_Information Service Definition

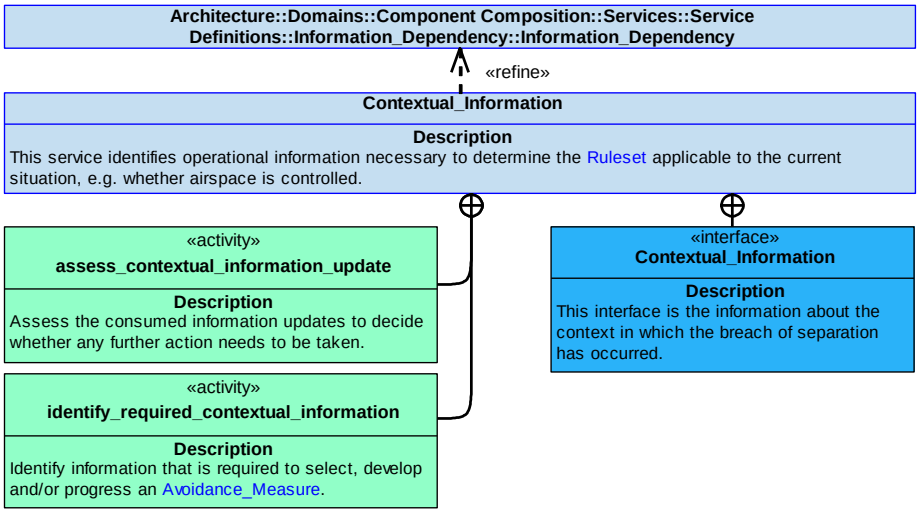

Figure 182: Contextual_Information Service Policy

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.

B.2.4.7.1.7 Constraint

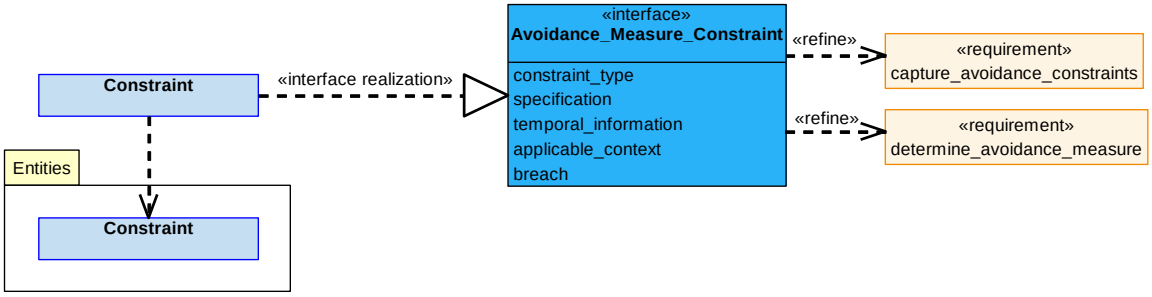

Figure 183: Constraint Service Definition

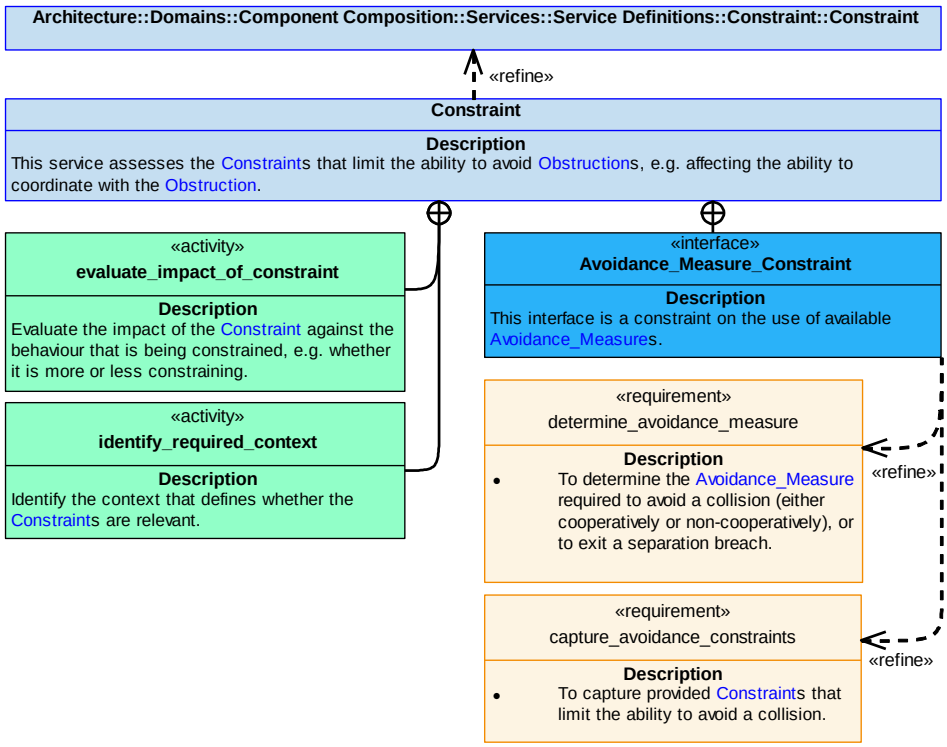

Figure 184: Constraint Service Policy

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, 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.

B.2.4.7.1.8 Capability

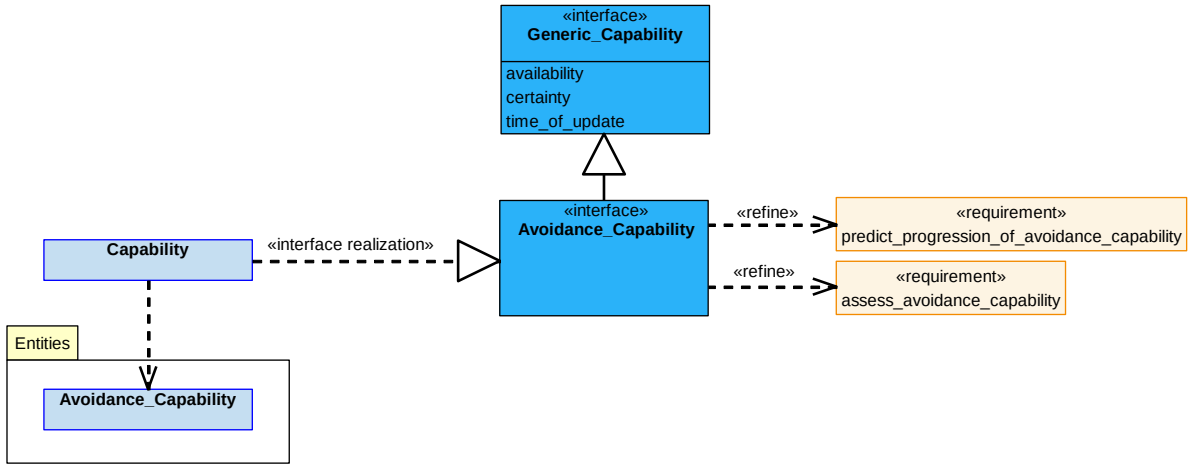

Figure 185: Capability Service Definition

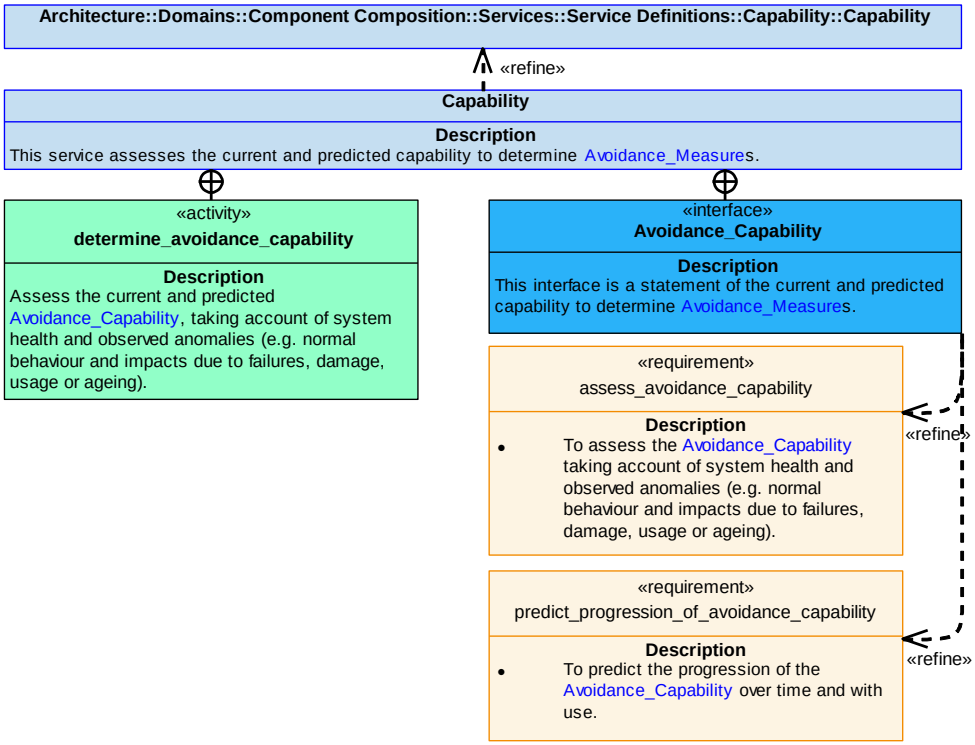

Figure 186: Capability Service Policy

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).

B.2.4.7.1.9 Capability_Evidence

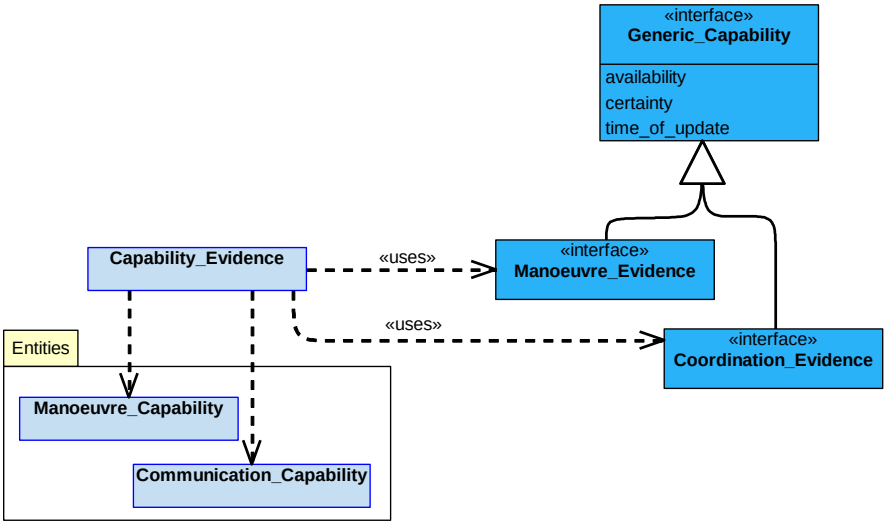

Figure 187: Capability_Evidence Service Definition

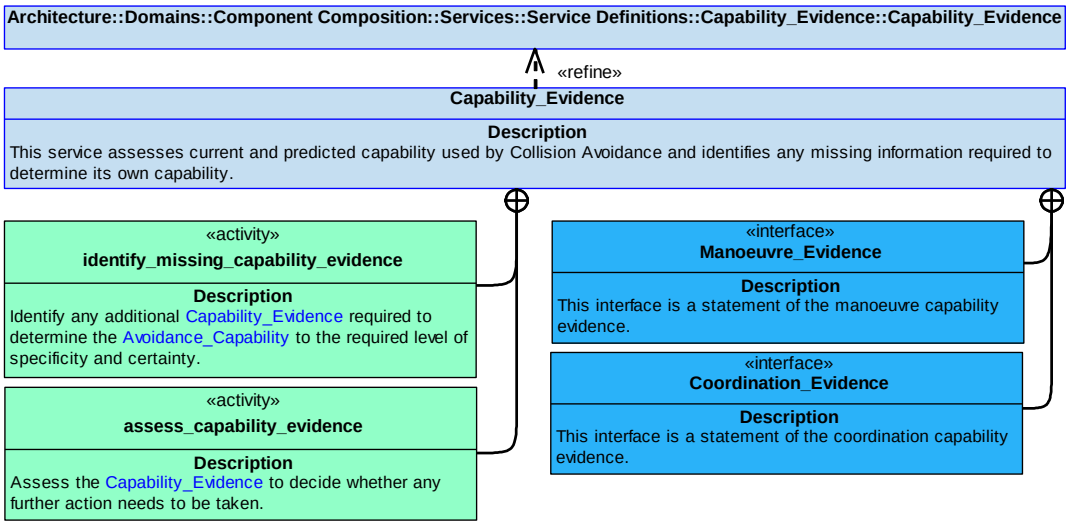

Figure 188: Capability_Evidence Service Policy

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.

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.

B.2.4.7.2 Service Dependencies

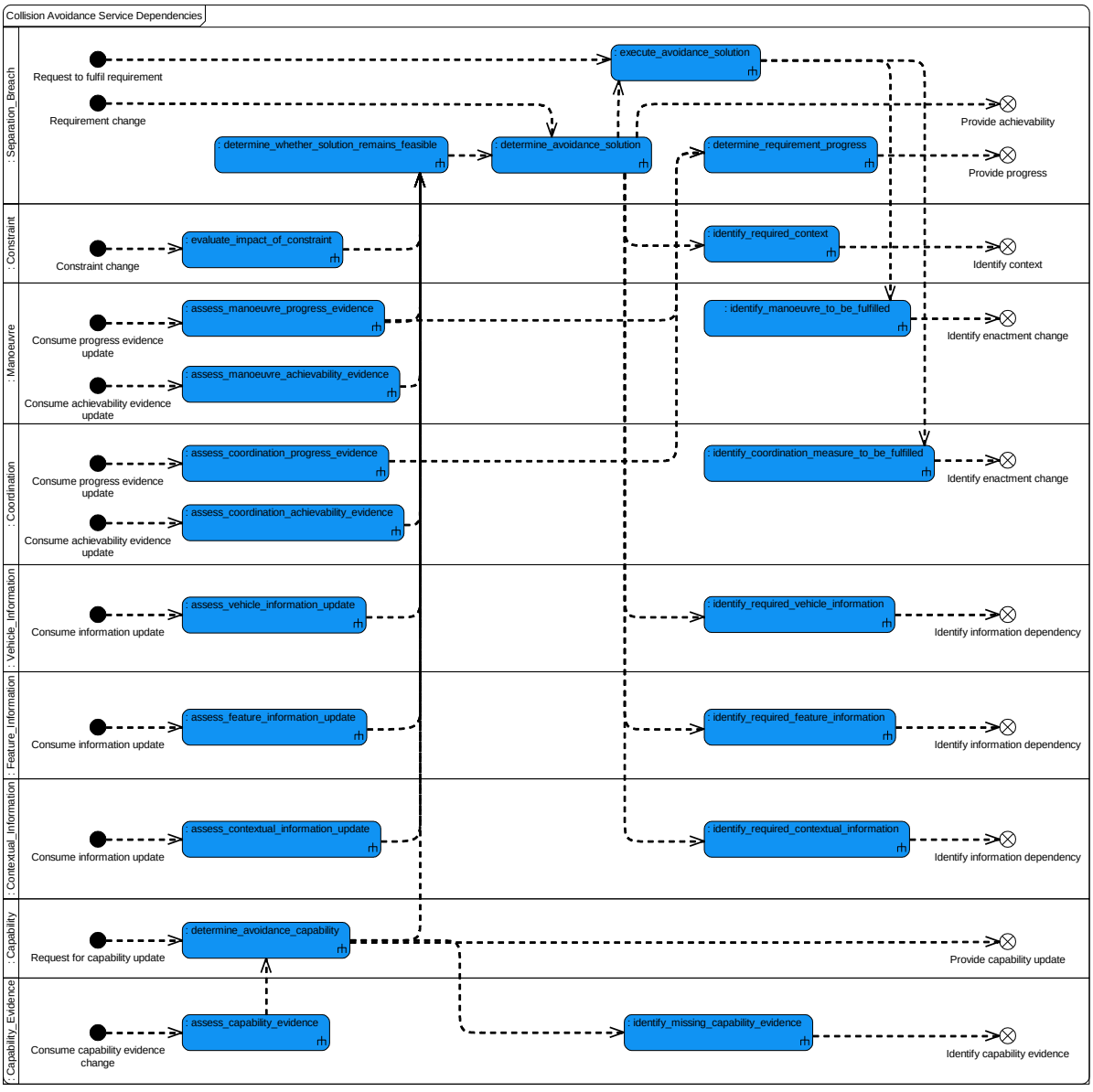

Figure 189: Collision Avoidance Service Dependencies

B.2.5 Collision Prediction

B.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.

B.2.5.2 Overview

Control Architecture

Collision Prediction is a service component as defined in the Control Architecture policy.

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).

B.2.5.3 Service Summary

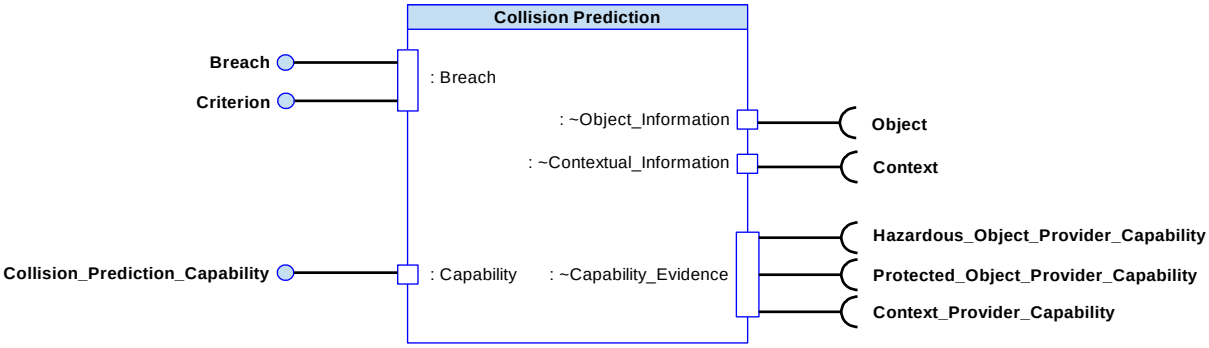

Figure 190: Collision Prediction Service Summary

B.2.5.4 Responsibilities

capture_prediction_requirements

- To capture Requirements for collision prediction (e.g. during taxi, transit or air-to-air refuelling).

identify_rules

- To identify the rules that apply when determining if a Breach has occurred or is predicted.

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.

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.

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).

predict_capability_progression

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

identify_missing_information

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

B.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.

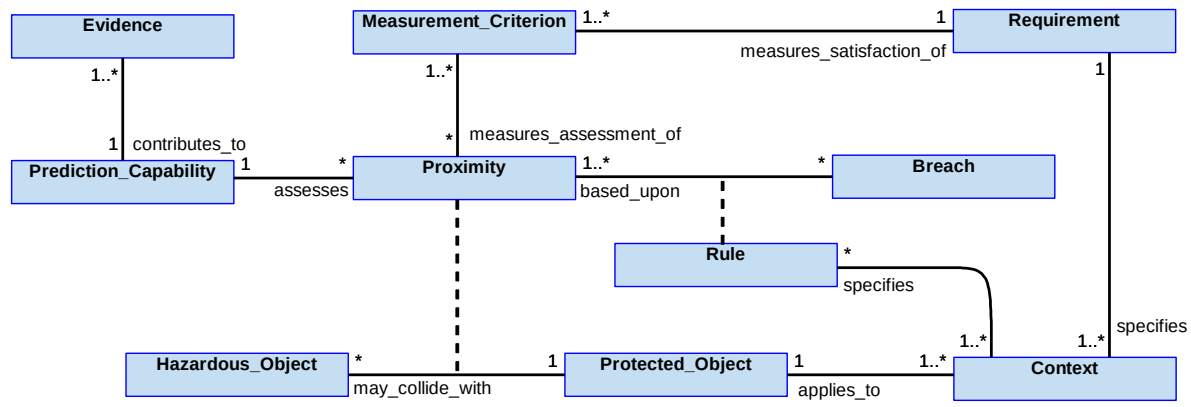

Figure 191: Collision Prediction Semantics

B.2.5.5.1 Entities

Breach

A violation of the acceptable Proximity. E.g. 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.

B.2.5.6 Design Rationale

B.2.5.6.1 Assumptions

- A group of vehicles, including less-capable unmanned adjuncts (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 unmanned adjunct.

B.2.5.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Collision Prediction:

- Data Driving - Prediction algorithms and Breach rules could be data-driven at build time (i.e. development 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.

B.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.

B.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.

B.2.5.7 Services

B.2.5.7.1 Service Definitions

B.2.5.7.1.1 Breach

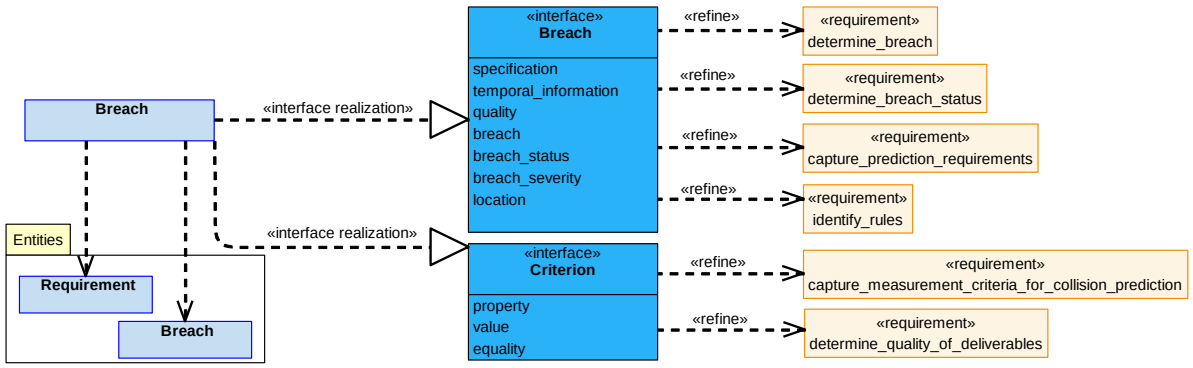

Figure 192: Breach Service Definition

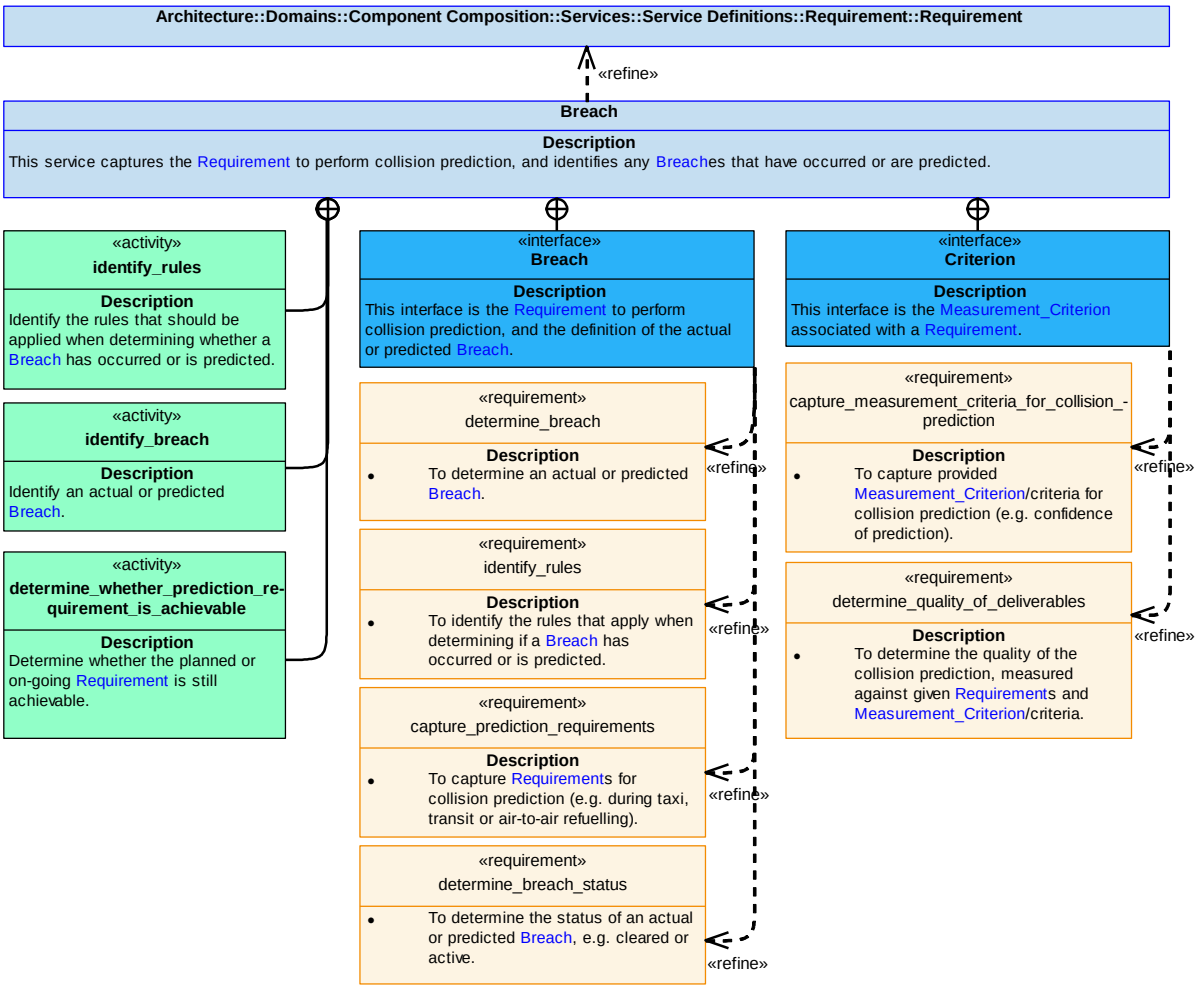

Figure 193: Breach Service Policy

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.

B.2.5.7.1.2 Object_Information

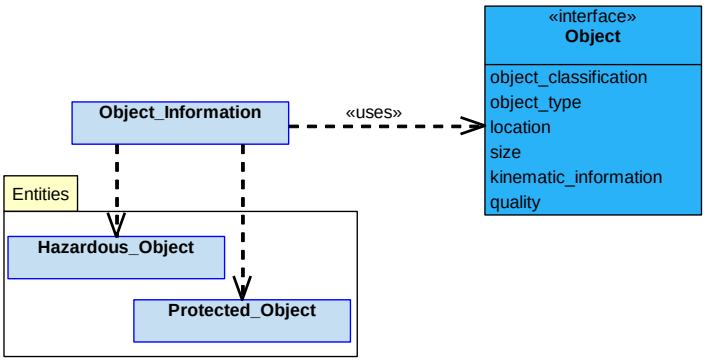

Figure 194: Object_Information Service Definition

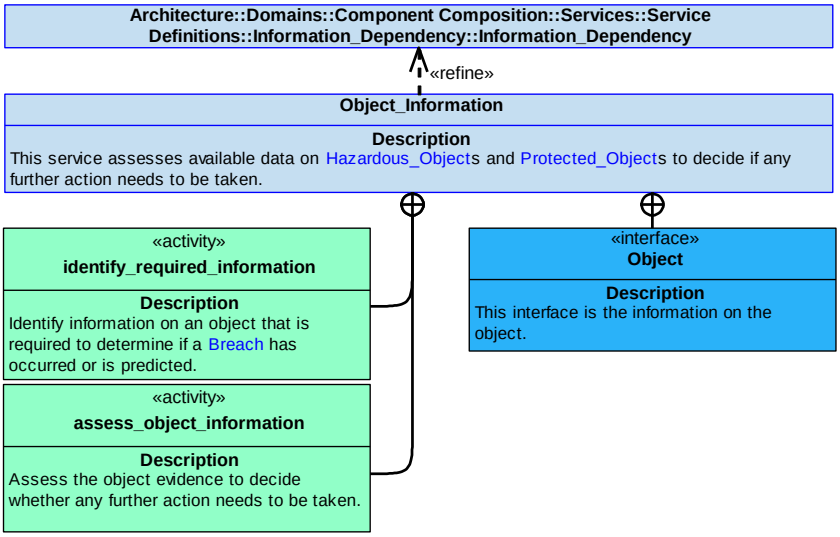

Figure 195: Object_Information Service Policy

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, unmanned adjunct, 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.

B.2.5.7.1.3 Contextual_Information

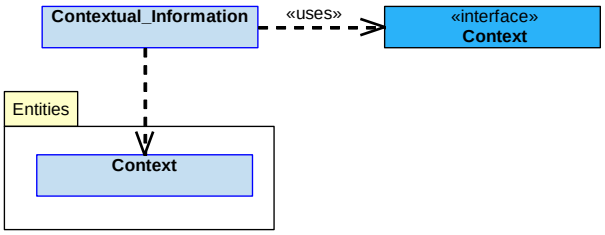

Figure 196: Contextual_Information Service Definition

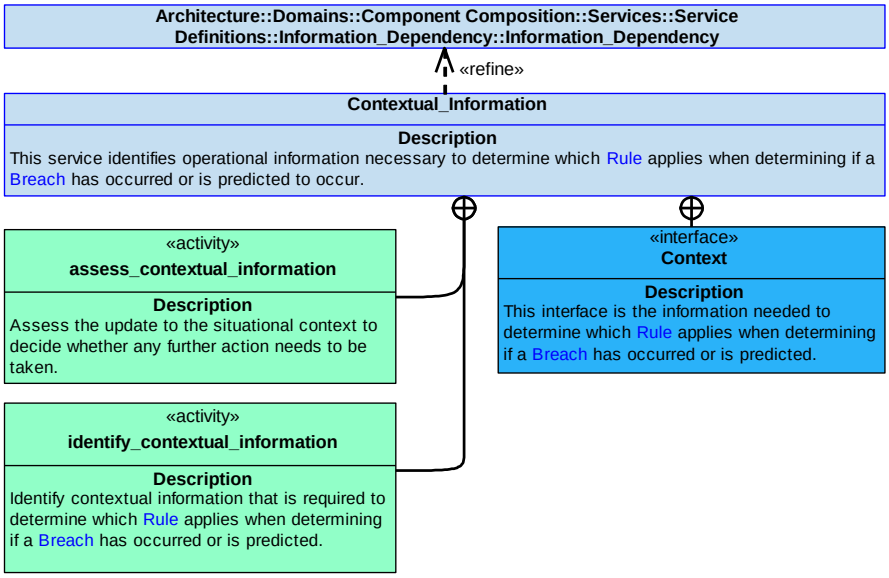

Figure 197: Contextual_Information Service Policy

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.

B.2.5.7.1.4 Capability

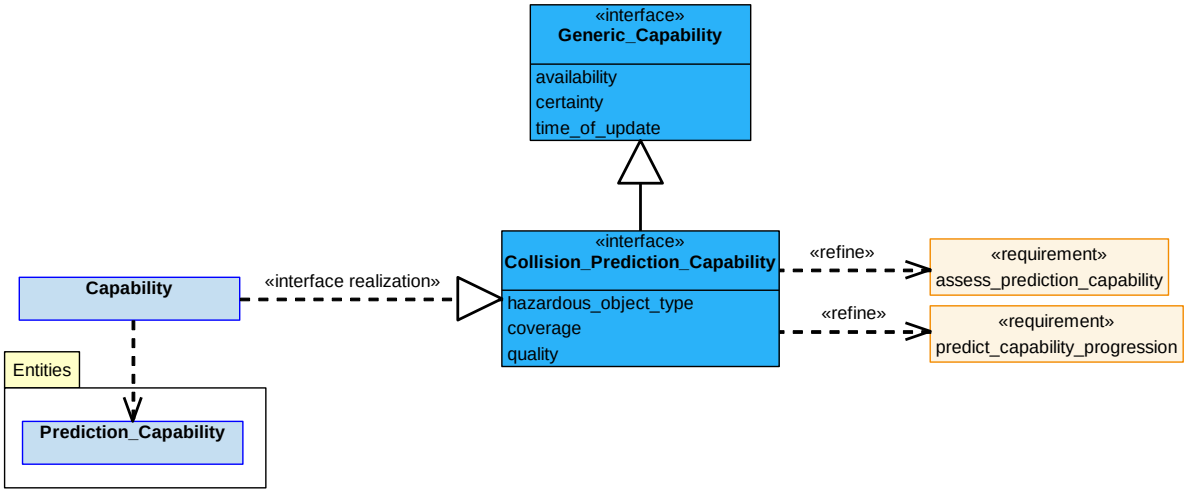

Figure 198: Capability Service Definition

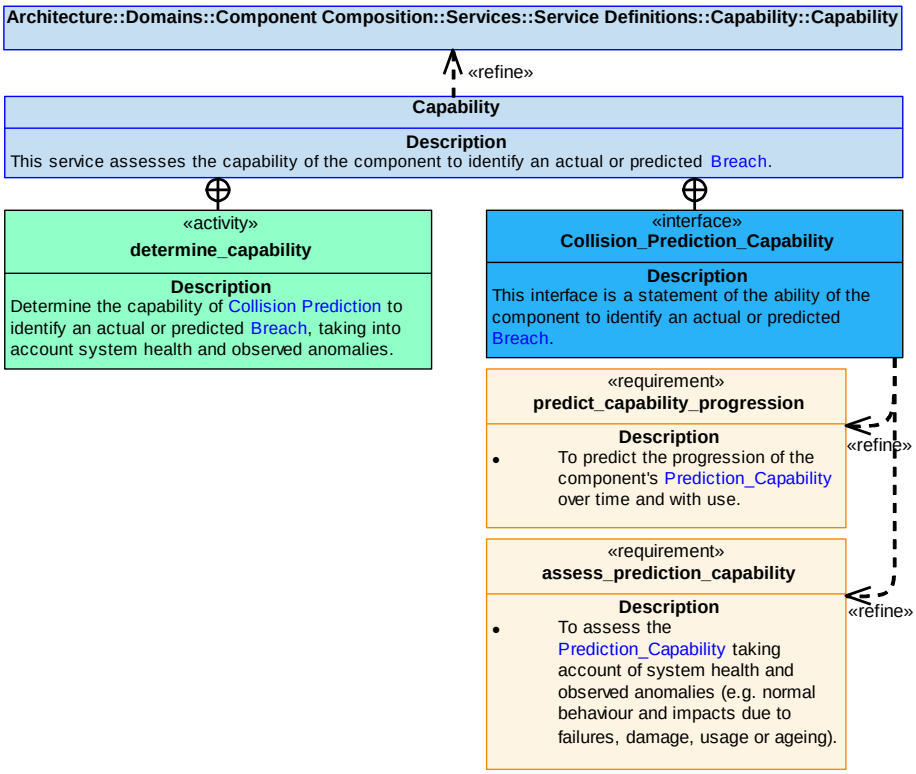

Figure 199: Capability Service Policy

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.

B.2.5.7.1.5 Capability_Evidence

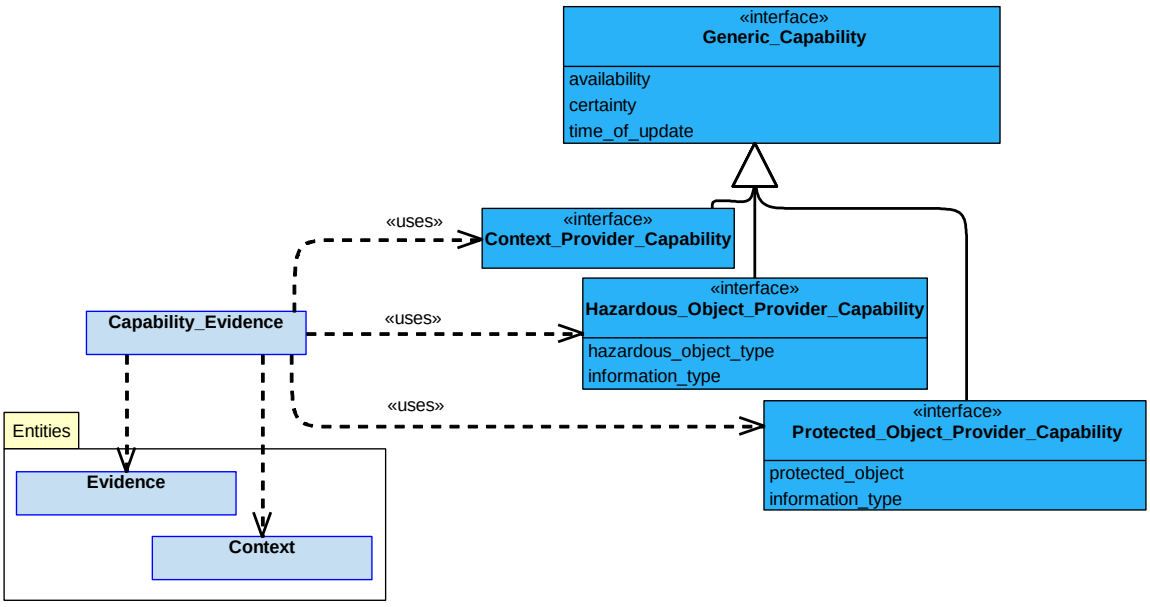

Figure 200: Capability_Evidence Service Definition

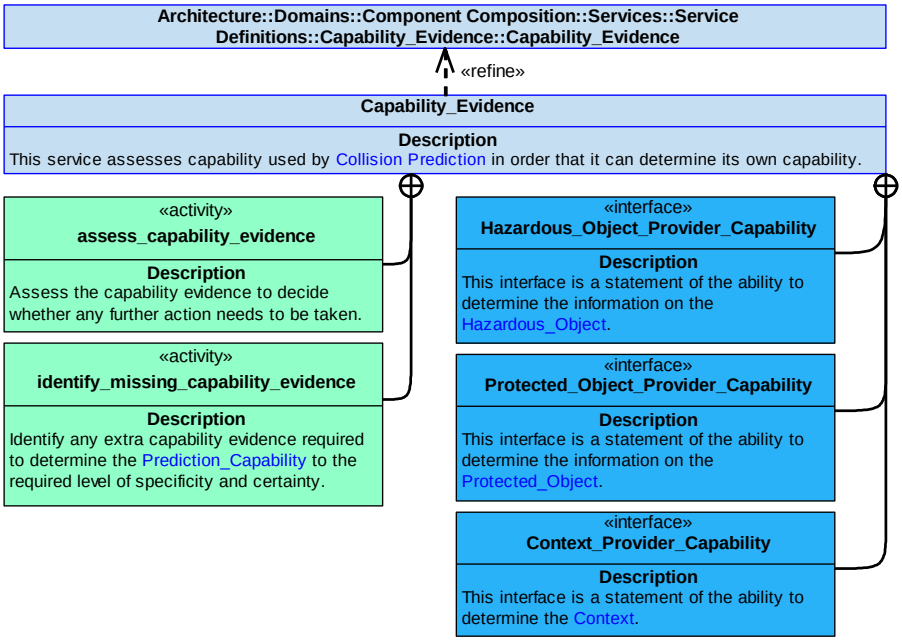

Figure 201: Capability_Evidence Service Policy

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.

B.2.5.7.2 Service Dependencies

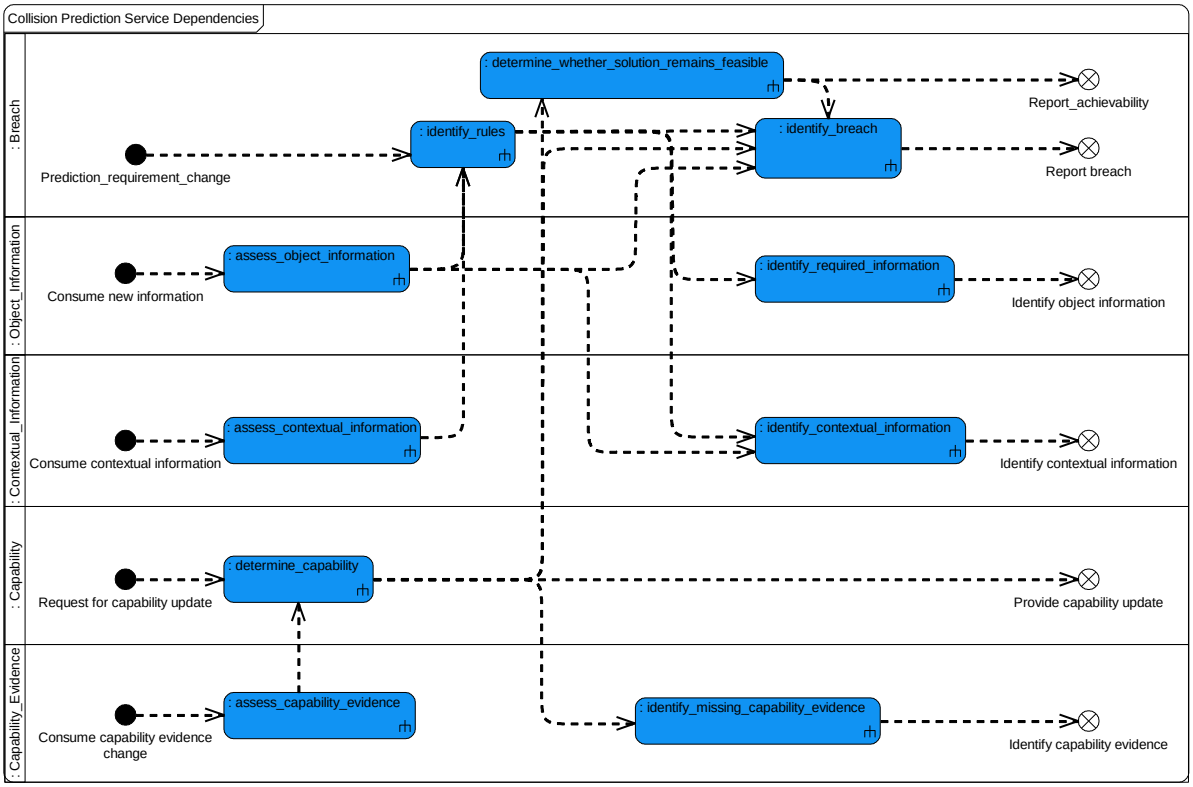

Figure 202: Collision Prediction Service Dependencies

B.2.6 Communication Links

B.2.6.1 Role

The role of Communication Links is to establish, maintain and optimise direct communication links between nodes.

B.2.6.2 Overview

Control Architecture

Communication Links is an action component as defined in the Control Architecture policy.

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.

B.2.6.3 Service Summary

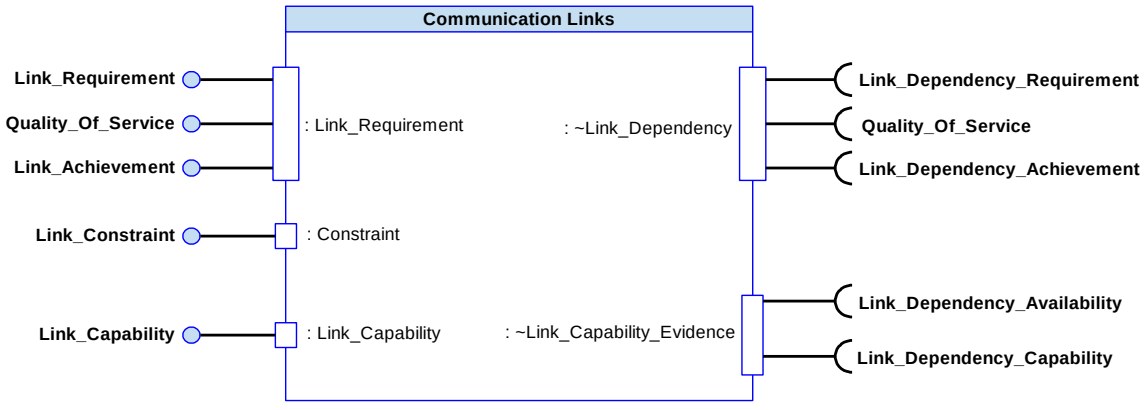

Figure 203: Communication Links Service Summary

B.2.6.4 Responsibilities

capture_link_requirements

- To capture given communication Link_Requirements (e.g. endpoint, throughput, reliability and latency).

capture_link_constraints

- To capture given communication link Constraints (e.g. maximum power level and spectrum usage).

capture_link_measurement_criteria

- To capture the criteria by which Link_Quality will be measured (e.g. reliability, throughput, and latency) for Link_Solutions.

assess_link_capability

- To assess the system Capability to establish and maintain links using available Link_Resources within given Constraints.

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_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).

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).

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_missing_information

- To identify missing information that could improve the certainty or specificity of the link management Capability assessment (e.g. observability/LoS assessment).

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).

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).

determine_if_link_solution_remains_feasible

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

B.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.
- Communication Links will also determine how Link requirements are met, not the determination of the need for a Link, which is the subject of another component (e.g. Networks or an appropriate Tasks extension).

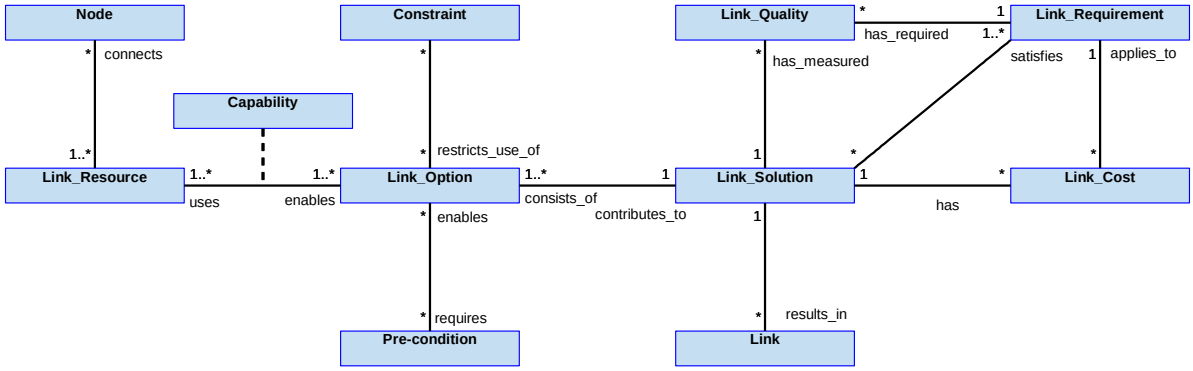

Figure 204: Communication Links Semantics

B.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, 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).

B.2.6.6 Design Rationale

B.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 (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.

B.2.6.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy shows how tasks can be coordinated across different vehicles. This is useful in coordinating changes to communication links.
- Data Driving - This policy 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.

B.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.

B.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.

B.2.6.7 Services

B.2.6.7.1 Service Definitions

B.2.6.7.1.1 Link_Requirement

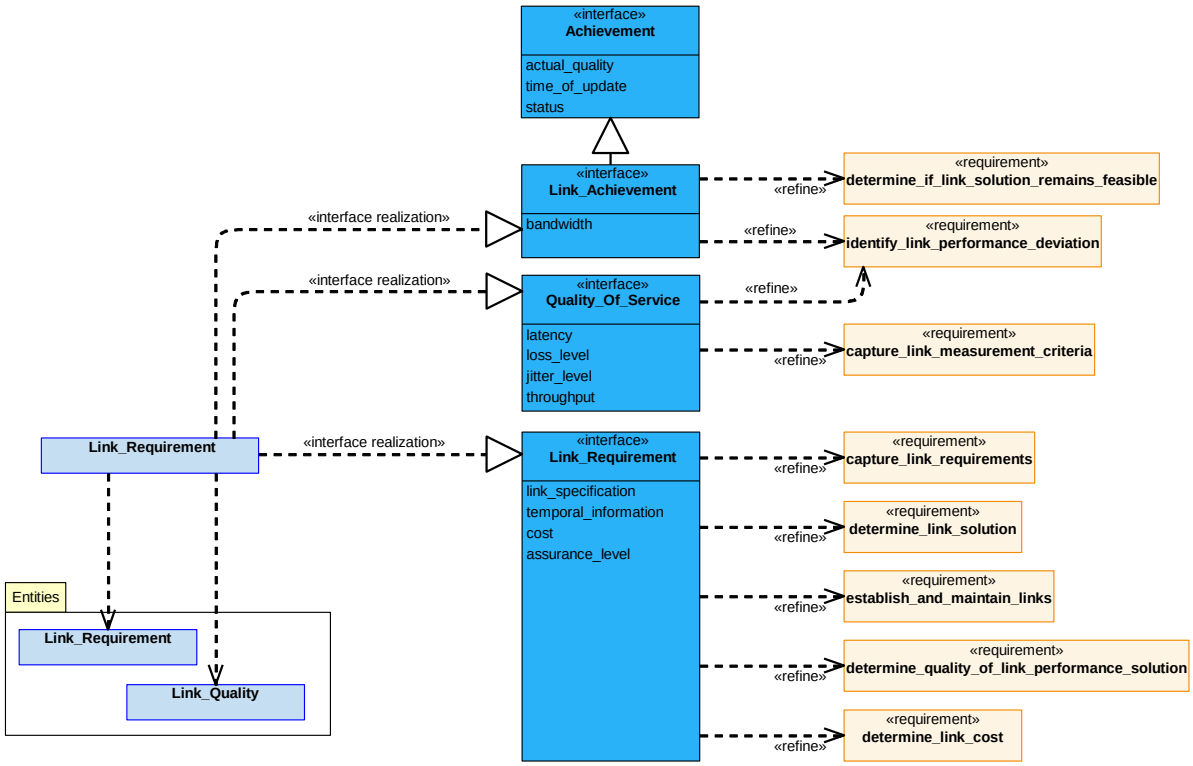

Figure 205: Link_Requirement Service Definition

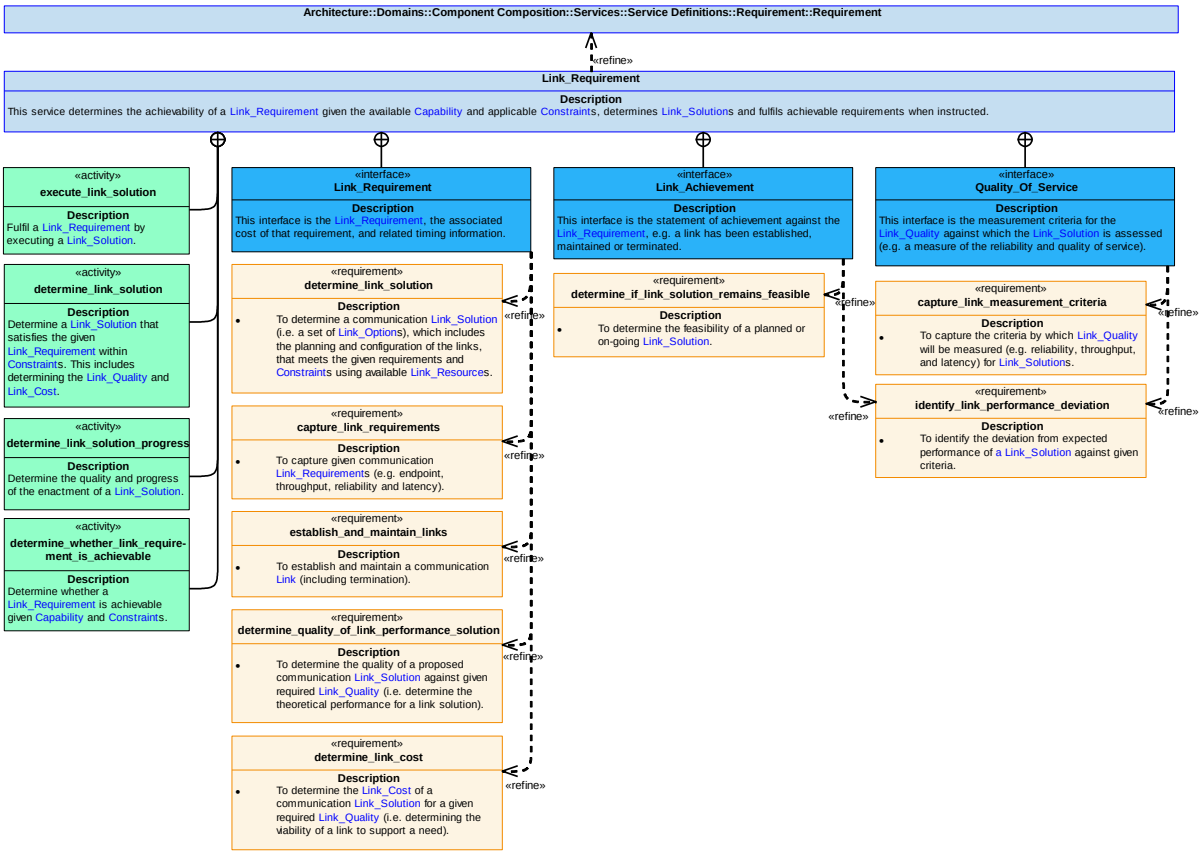

Figure 206: Link_Requirement Service Policy

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

execute_link_solution

Fulfil a Link_Requirement by executing a Link_Solution.

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_link_solution_progress

Determine the quality and progress of the enactment of a Link_Solution.

determine_whether_link_requirement_is_achievable

Determine whether a Link_Requirement is achievable given Capability and Constraints.

B.2.6.7.1.2 Link_Dependency

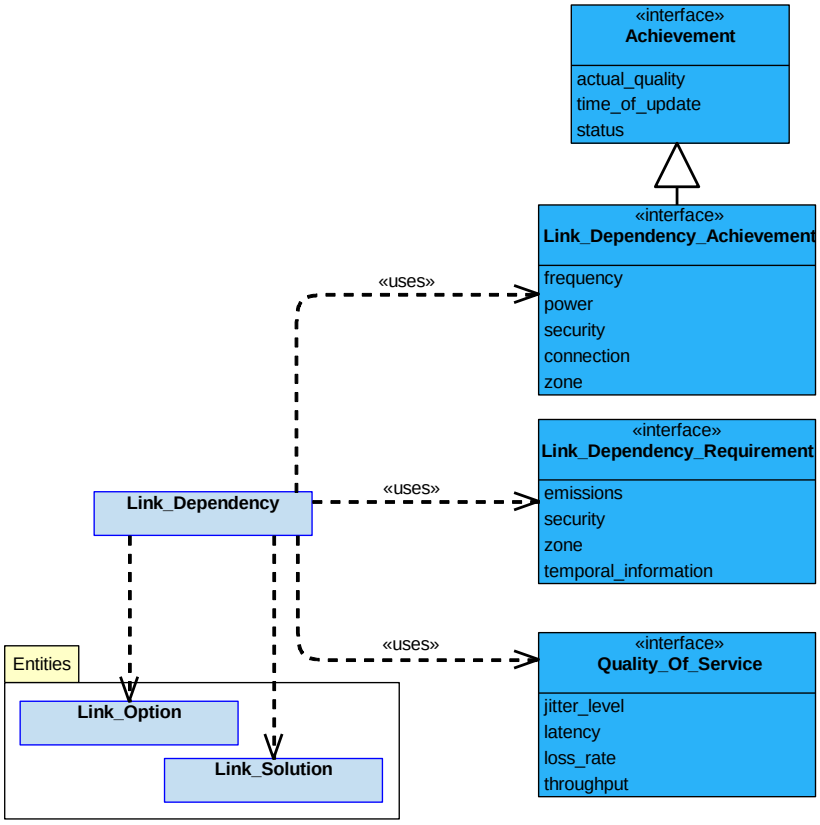

Figure 207: Link_Dependency Service Definition

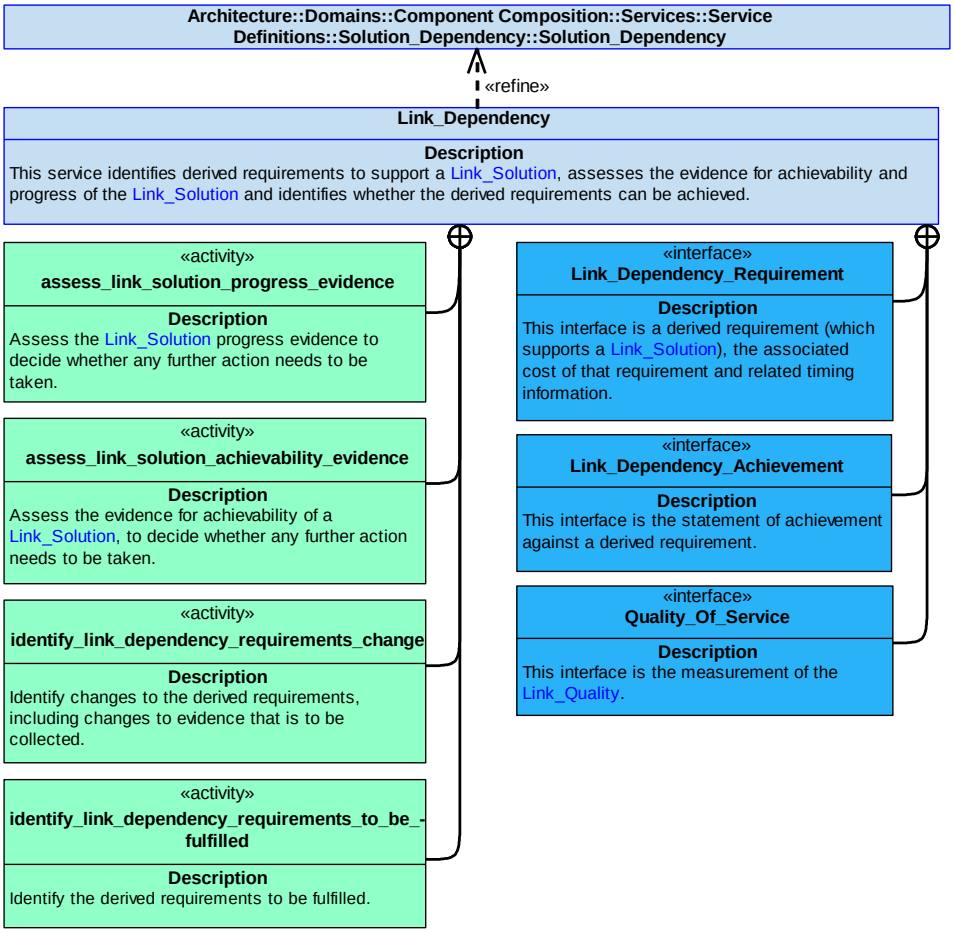

Figure 208: Link_Dependency Service Policy

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.

B.2.6.7.1.3 Constraint

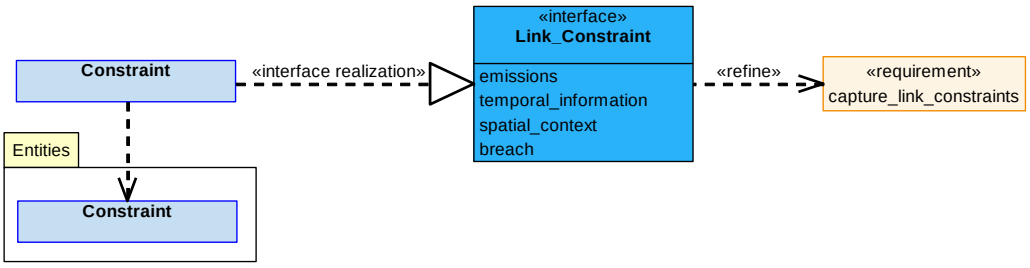

Figure 209: Constraint Service Definition

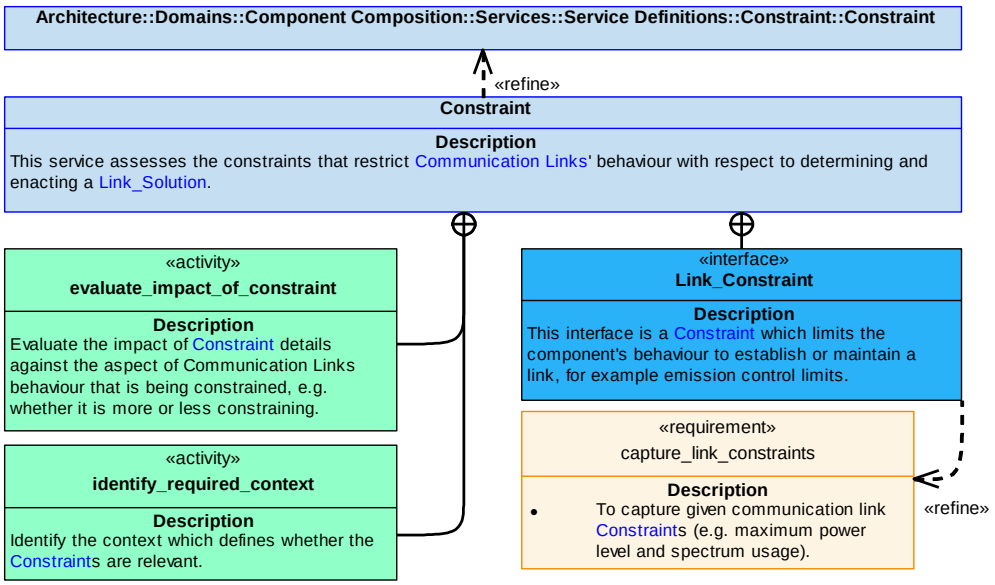

Figure 210: Constraint Service Policy

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.

B.2.6.7.1.4 Link_Capability

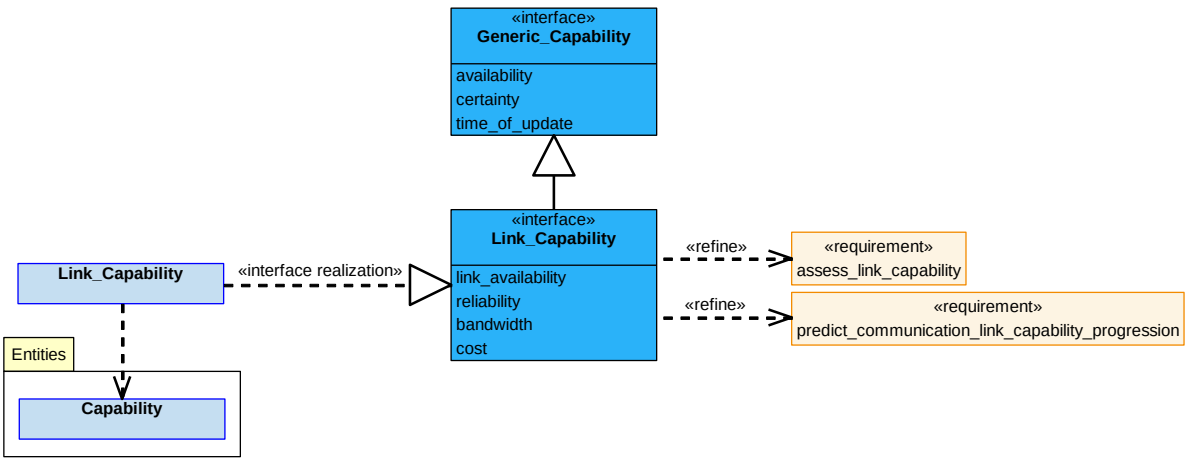

Figure 211: Link_Capability Service Definition

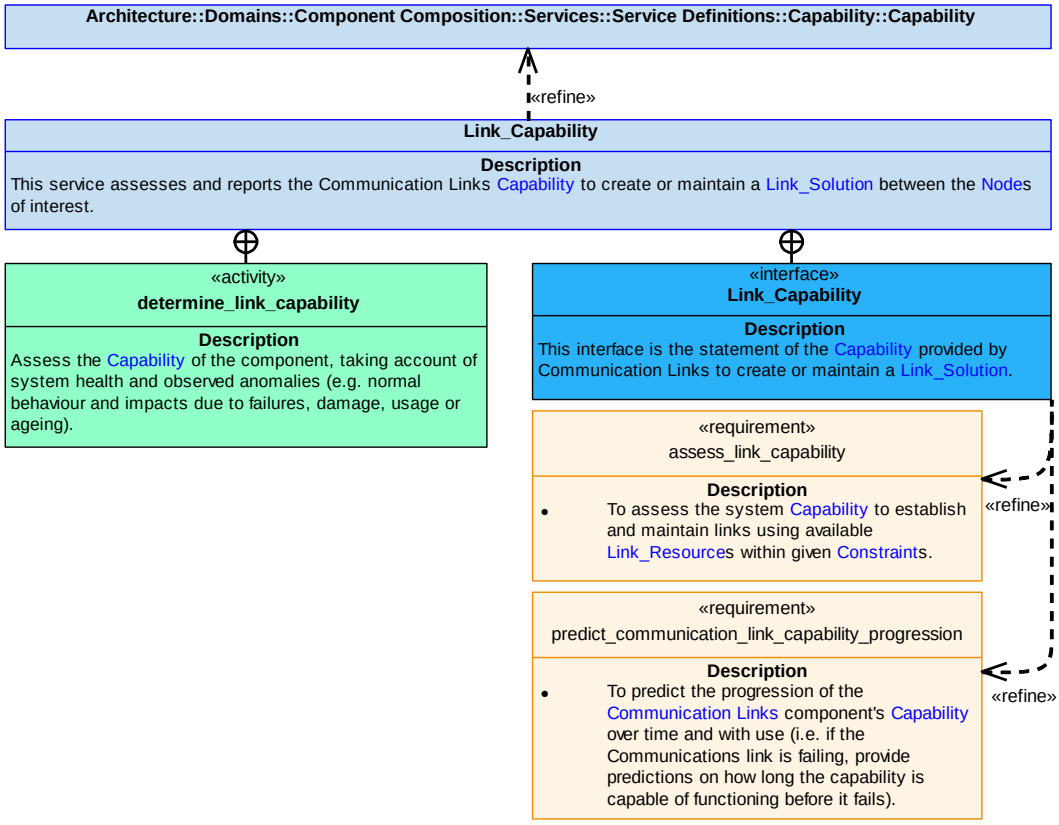

Figure 212: Link_Capability Service Policy

Link_Capability

This service assesses and reports the Communication Links Capability to create or maintain a Link_Solution between the Nodes of interest.

Interface

Link_Capability

This interface is the statement of the Capability provided by Communication Links to create or maintain a Link_Solution.

Attributes

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).

B.2.6.7.1.5 Link_Capability_Evidence

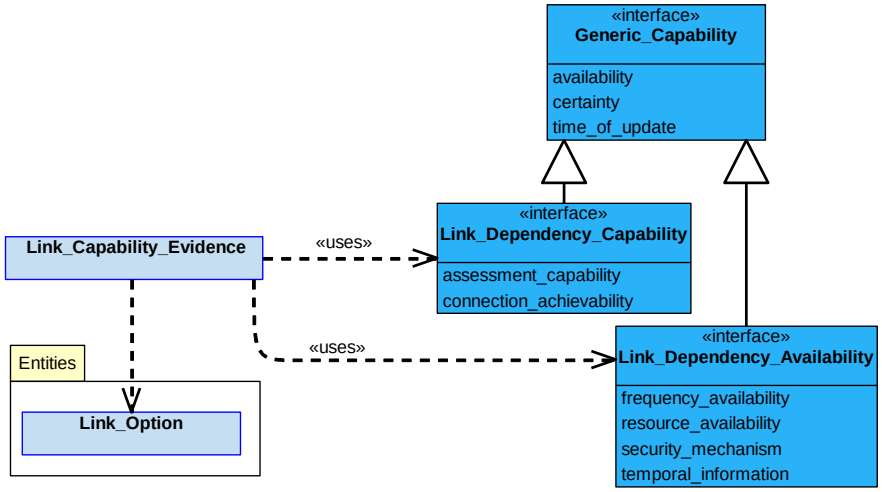

Figure 213: Link_Capability Evidence Service Definition

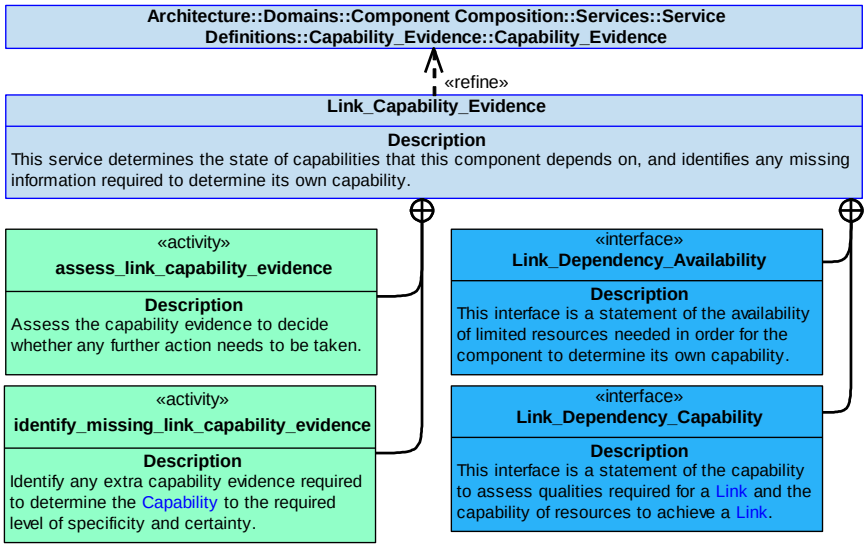

Figure 214: Link_Capability Evidence Service Policy

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.

B.2.6.7.2 Service Dependencies

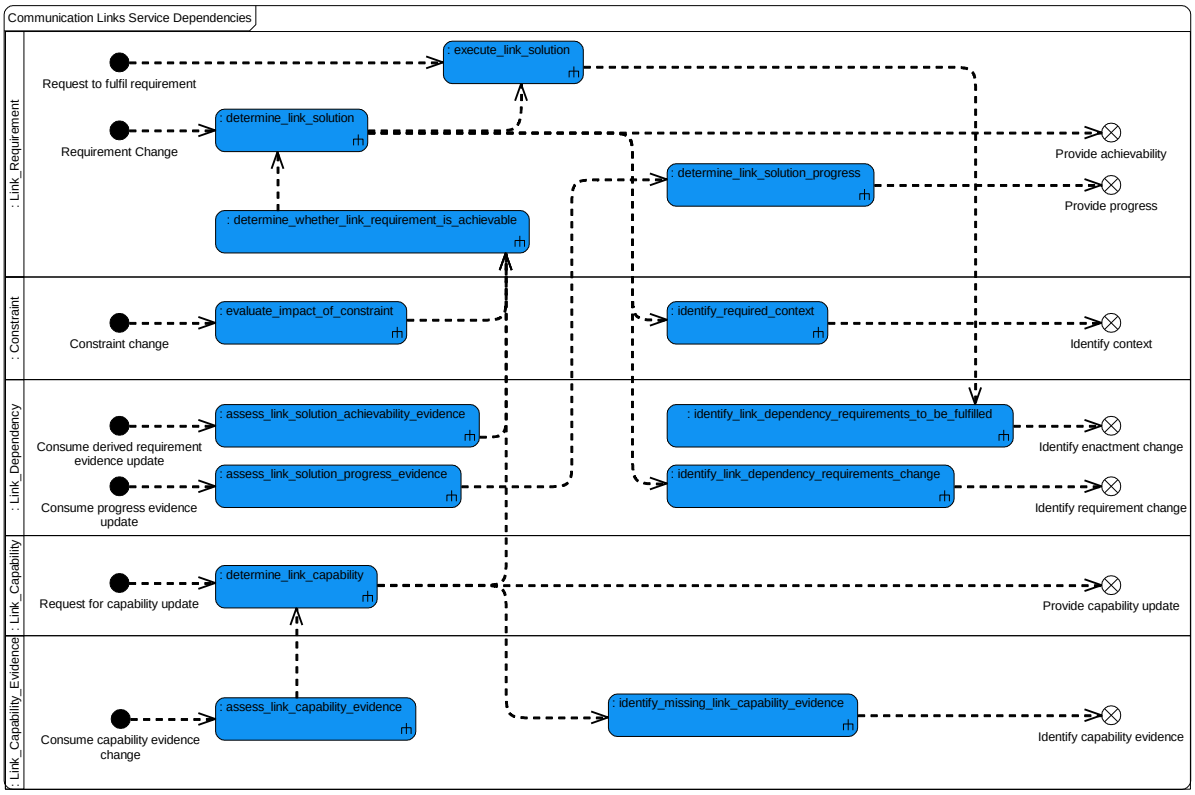

Figure 215: Communication Links Service Dependencies

B.2.7 Communicator

B.2.7.1 Role

The role of Communicator is to provide an interface to manage communication resources.

B.2.7.2 Overview

Control Architecture

Communicator is a resource component as defined in the Control Architecture policy.

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).

B.2.7.3 Service Summary

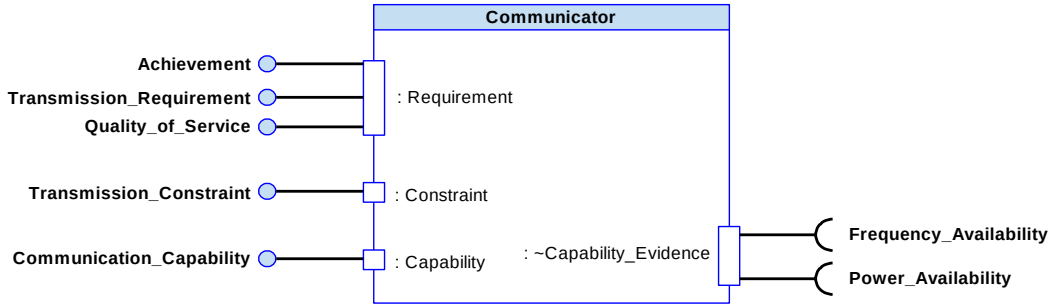

Figure 216: Communicator Service Summary

B.2.7.4 Responsibilities

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).

capture_constraints_for_communicator_resources

- To capture provided Constraints for use of Communicator_Resources (e.g. maximum power or frequency).

capture_requirements_for_communicator_resources

- To capture provided Requirements (e.g. power, latency, loss rate, direction and throughput) for the use of Communicator_Resources.

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.

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.

coordinate_use_of_resources

- To coordinate the use of Communicator_Resources (e.g. determining signal direction including signal lock).

identify_progress_of_transmission_sequence

- To identify the progress of a Communicator_Resource Transmission_Sequence against the Requirements.

predict_capability_progression

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

identify_missing_information

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

identify_transmission_sequence_remains_feasible

- To identify if a Transmission_Sequence in progress remains feasible given current Communicator_Resources.

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).

B.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.

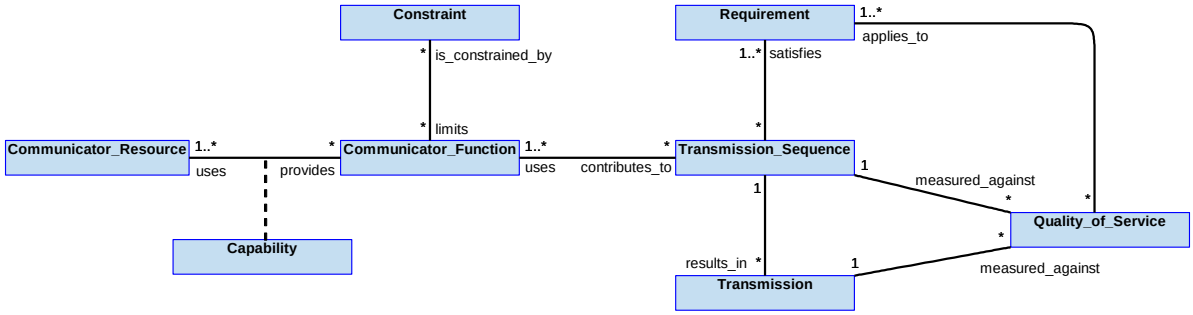

Figure 217: Communicator Semantics

B.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.

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).

B.2.7.6 Design Rationale

B.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.

B.2.7.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Communicator:

- Interaction with Equipment - This policy specifies how this component supports new and different pieces of equipment to interact with the system.
- Use of Communications - This policy 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.

B.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.

B.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.

B.2.7.7 Services

B.2.7.7.1 Service Definitions

B.2.7.7.1.1 Requirement

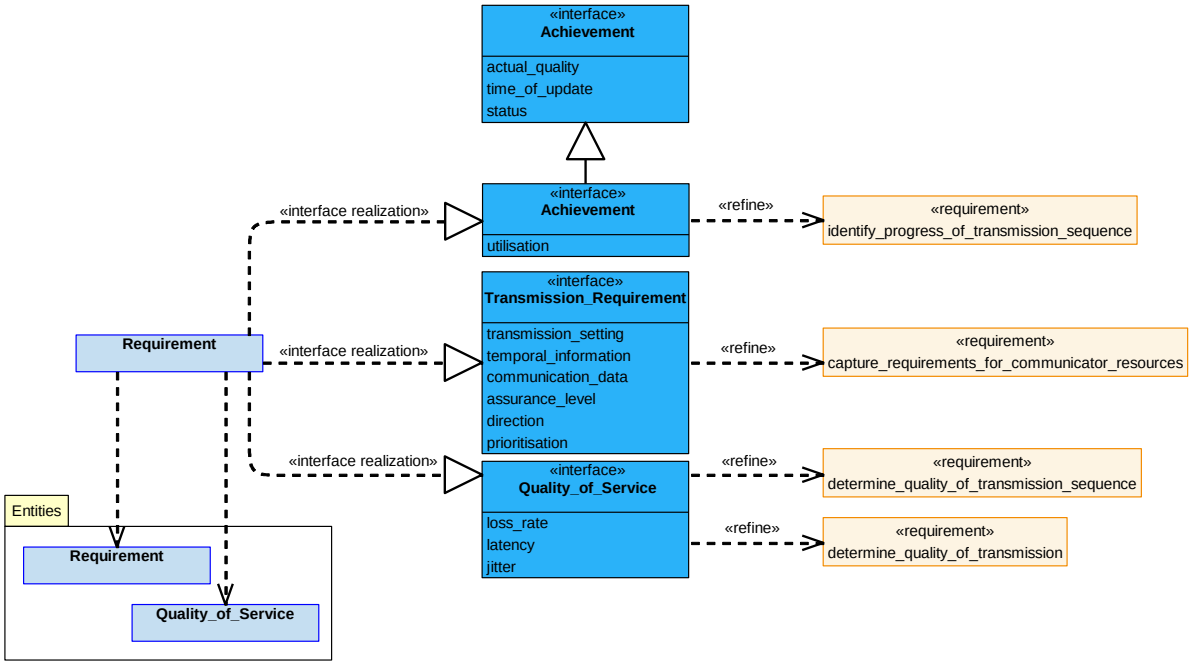

Figure 218: Requirement Service Definition

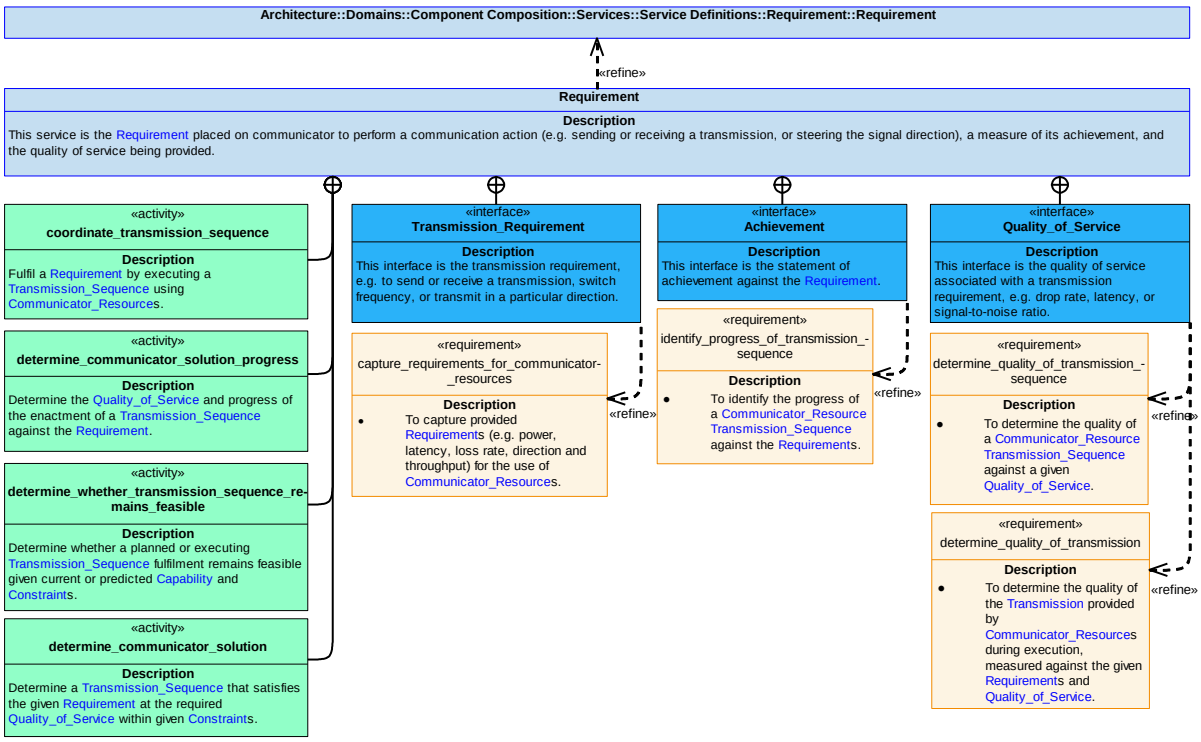

Figure 219: Requirement Service Policy

Requirement

This service is the Requirement placed on communicator to perform a communication action (e.g. sending or receiving a transmission, or steering the signal direction), a measure of its achievement, and 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.

B.2.7.7.1.2 Constraint

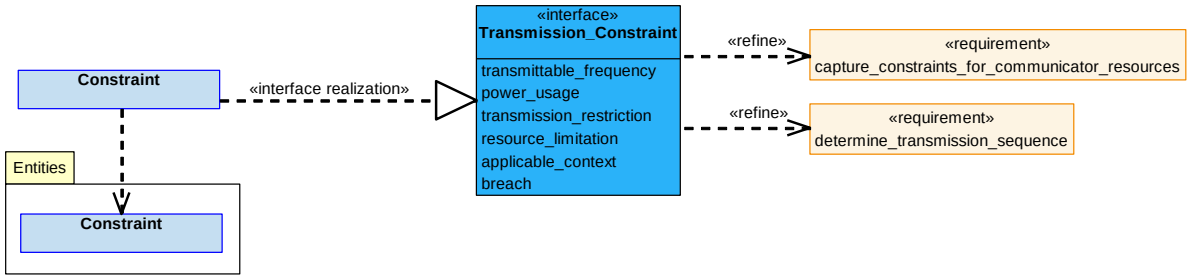

Figure 220: Constraint Service Definition

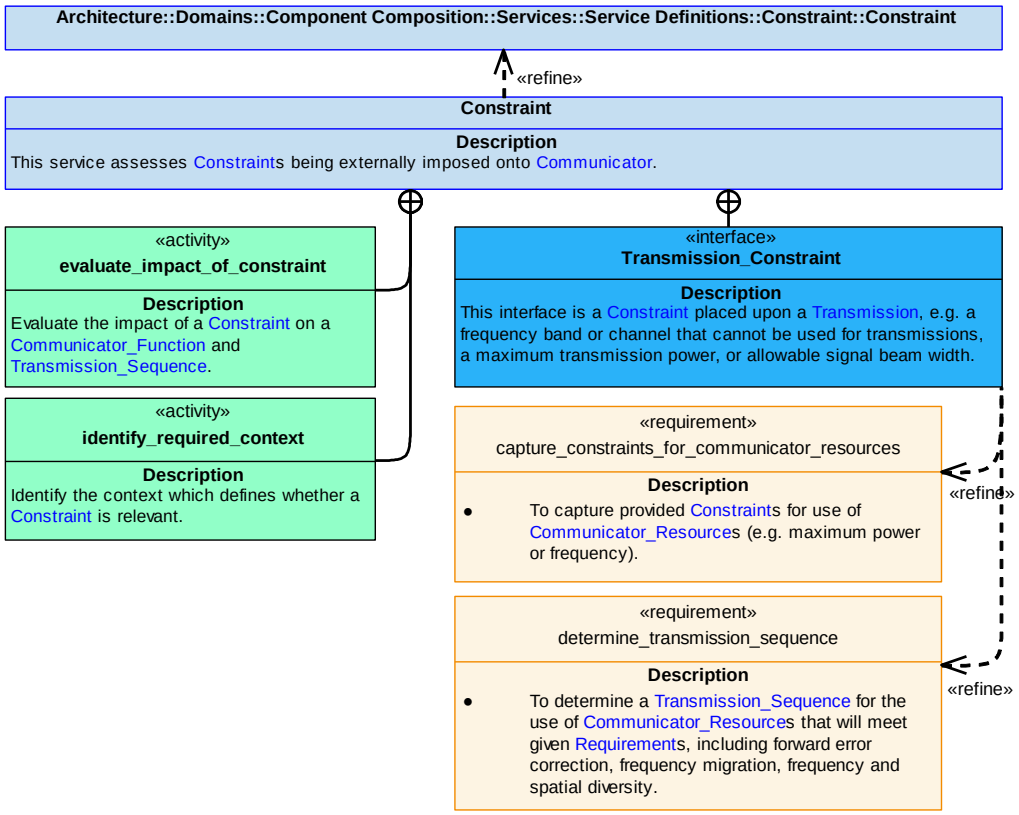

Figure 221: Constraint Service Policy

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 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.

B.2.7.7.1.3 Capability

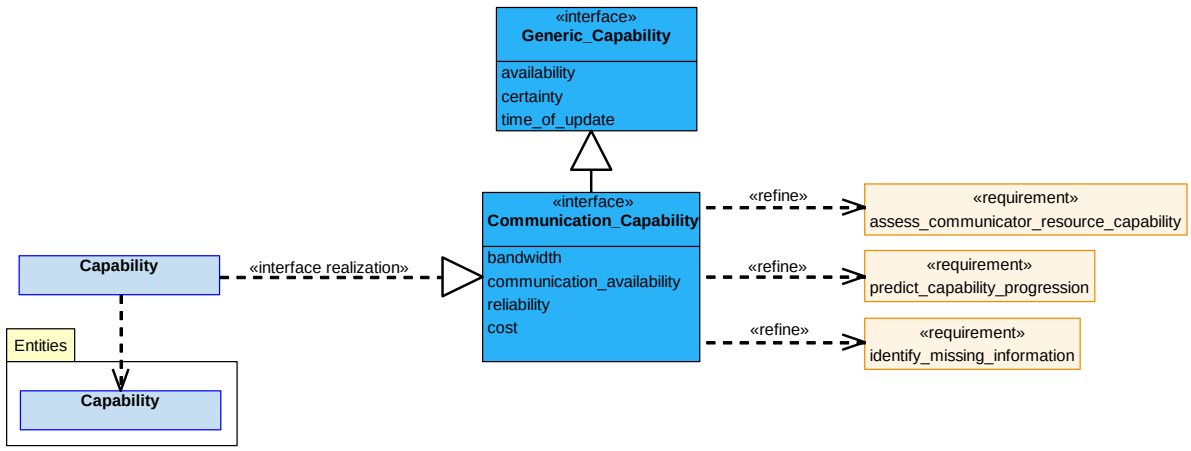

Figure 222: Capability Service Definition

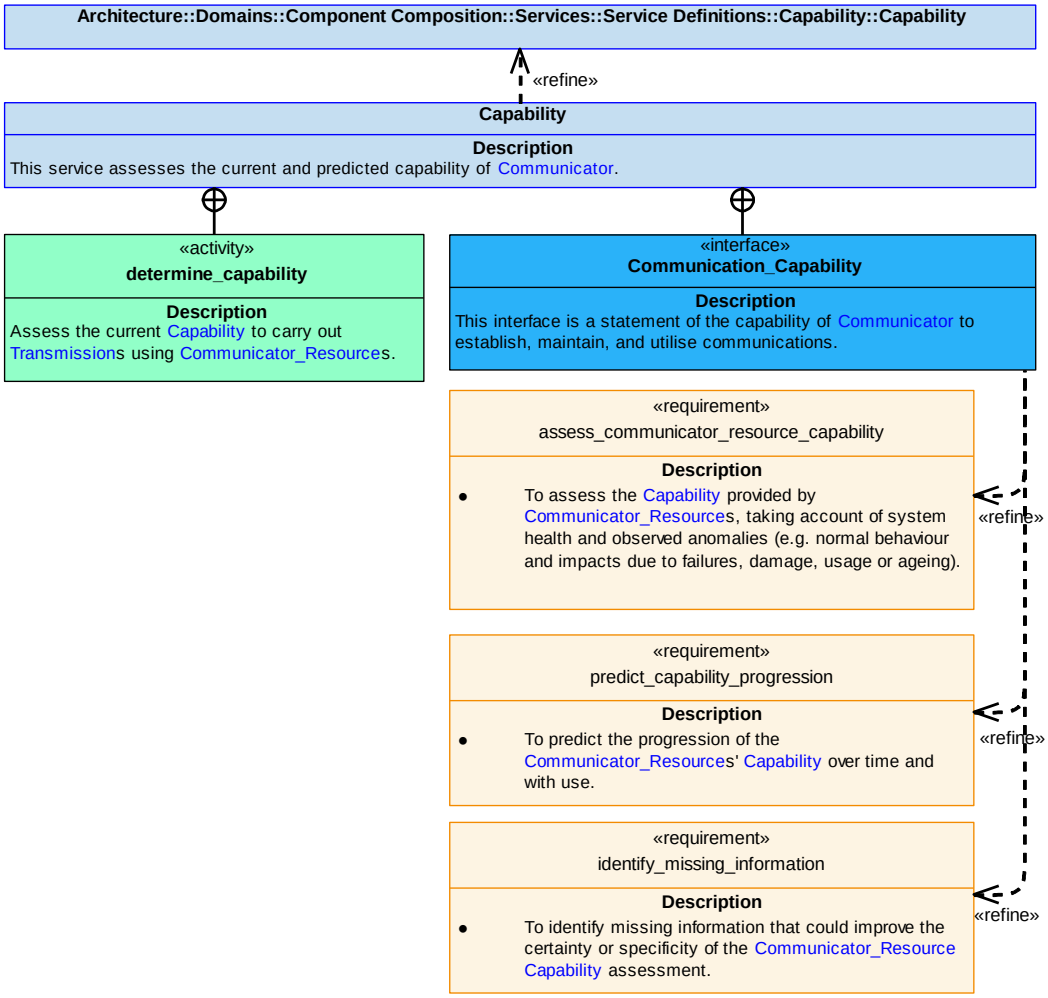

Figure 223: Capability Service Policy

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.

B.2.7.7.1.4 Capability Evidence

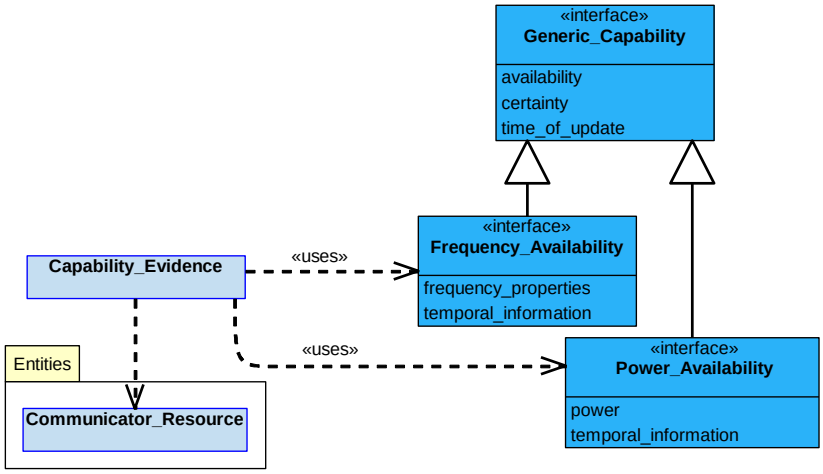

Figure 224: Capability_Evidence Service Definition

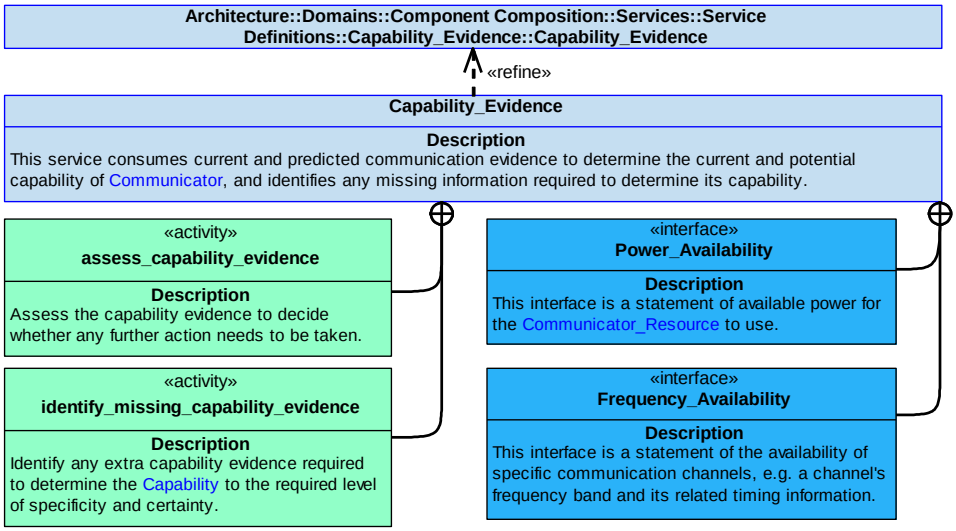

Figure 225: Capability_Evidence Service Policy

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.

Interfaces

Frequency_Availability

This interface is a statement of the availability of specific communication channels, e.g. a channel's frequency band and its related timing information.

Attributes

frequency_properties	The properties of the specific portion of the spectrum (e.g. a nominated frequency band) that is available to communicate on.
temporal_information	Timing information related to the availability of a communications channel, e.g. how long that specific channel will be available for.

Power_Availability

This interface is a statement of available power for the Communicator_Resource to use.

Attributes

power	The amount of power that is available for powering Communicator_Resources.
temporal_information	Timing information related to the availability of power, e.g. how long the power 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.

B.2.7.7.2 Service Dependencies

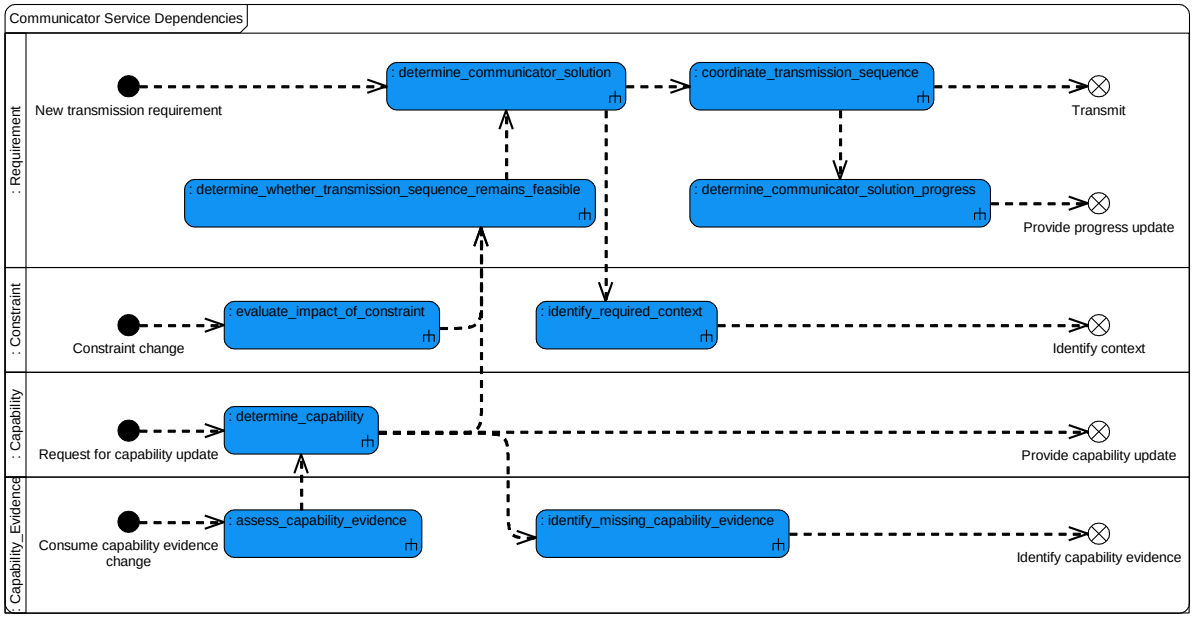

Figure 226: Communicator Service Dependencies

B.2.8 Countermeasures

B.2.8.1 Role

The role of Countermeasures is to counteract immediate threats to an air vehicle or multiple air vehicles.

B.2.8.2 Overview

Control Architecture

Countermeasures is an action component as defined in the Control Architecture policy.

Standard Pattern of Use

Countermeasures has been pre-authorised to enact a Countermeasure_Strategy in response to particular threats. Countermeasures receives a threat notification that a threat has exceeded the threat level threshold. Countermeasures receives a Countermeasure_Requirement to lower this threat to an acceptable level. Countermeasures determines a Countermeasure_Strategy to lower the threat to an acceptable level, taking into account Capability and Constraints. Countermeasures enacts the Countermeasure_Strategy utilising Countermeasure_Resources, and monitors the Countermeasure_Strategy to determine the effectiveness until the Deliverable has been met.

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.

B.2.8.3 Service Summary

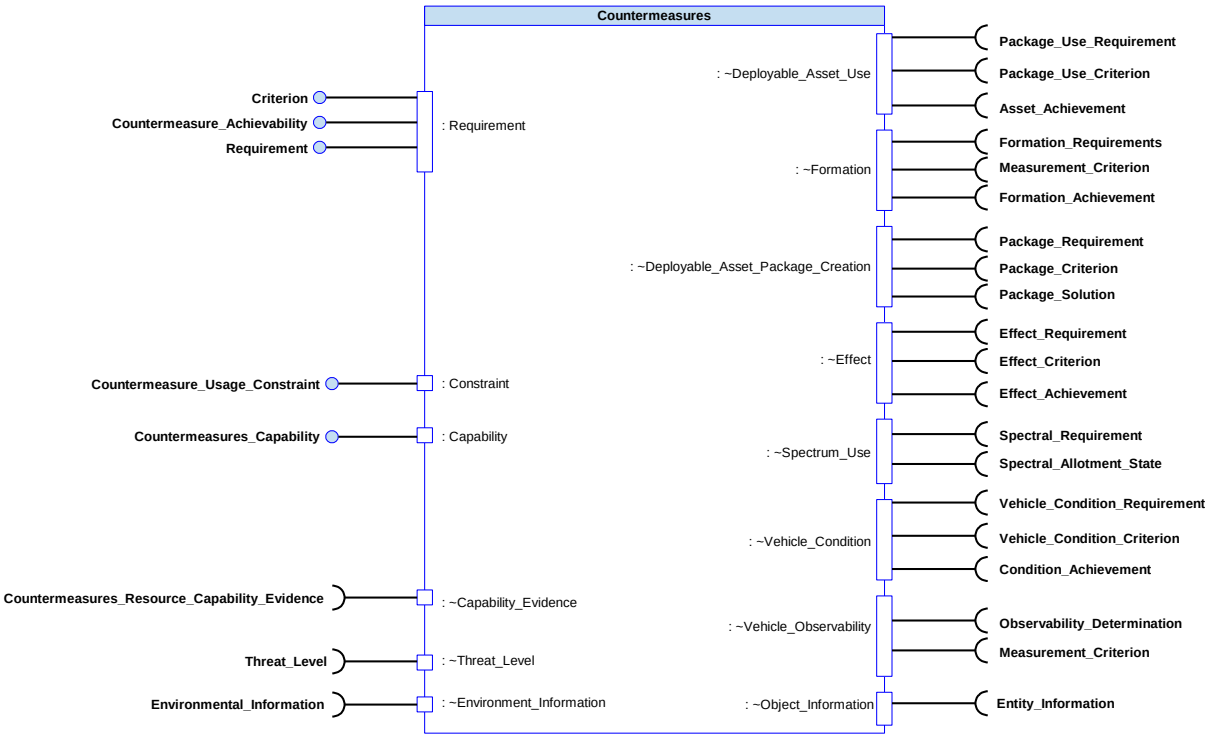

Figure 227: Countermeasures Service Summary

B.2.8.4 Responsibilities

capture_countermeasure_requirements

- To capture given Countermeasure_Requirements (e.g. threatening element or acceptable risk reduction level).

capture_countermeasure_constraints

- To capture given countermeasure Constraints (e.g. transmission restrictions or allowable level of response).

capture_measurement_criteria

- To capture given Measurement_Criterion/criteria (e.g. reduction in threat level) for countermeasure solutions.

assess_countermeasure_capability

- To assess the Capability to carry out Countermeasure_Strategy(ies) using available Countermeasure_Resources, taking into account observed anomalies.

predict_capability_progression

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

determine_countermeasure_strategy

- To determine a Countermeasure_Strategy that meets the Countermeasure_Requirements with available Countermeasure_Resources, and within given Constraints.

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_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).

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.

identify_missing_information

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

identify_if_countermeasure_requirement_remains_achievable

- To identify whether a Countermeasure_Requirement is still achievable given current or predicted capability and conditions.

B.2.8.5 Subject Matter Semantics

The subject matter of Countermeasures is the defensive Countermeasure_Strategy(ies) 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.

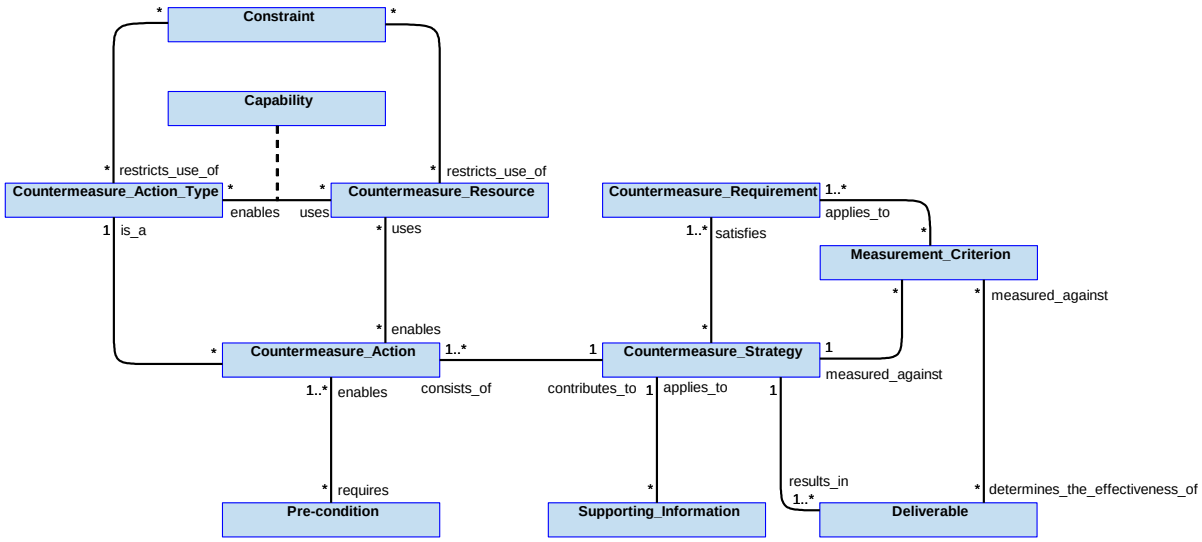

Figure 228: Countermeasures Semantics

B.2.8.5.1 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.

B.2.8.6 Design Rationale

B.2.8.6.1 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.

B.2.8.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Countermeasures:

- Multi-Vehicle Coordination - This policy is applicable in scenarios where defensive actions would need co-ordination between vehicles and/or platforms.
- Data Driving - This policy is applicable to cope with the change in countermeasure types and techniques used against particular threatening elements, with the countermeasures defined as deployment time data. The data would also need to be defined for operational use. This allows the component to be reusable between multiple Exploiting Programmes and maintainable as behaviours change and resources are replaced.
- Recording and Logging - This policy 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 Countermeasure_Strategy(ies) 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.

B.2.8.6.3 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, Authorisation and Path Demands) 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.

B.2.8.6.4 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.

B.2.8.7 Services

B.2.8.7.1 Service Definitions

B.2.8.7.1.1 Requirement

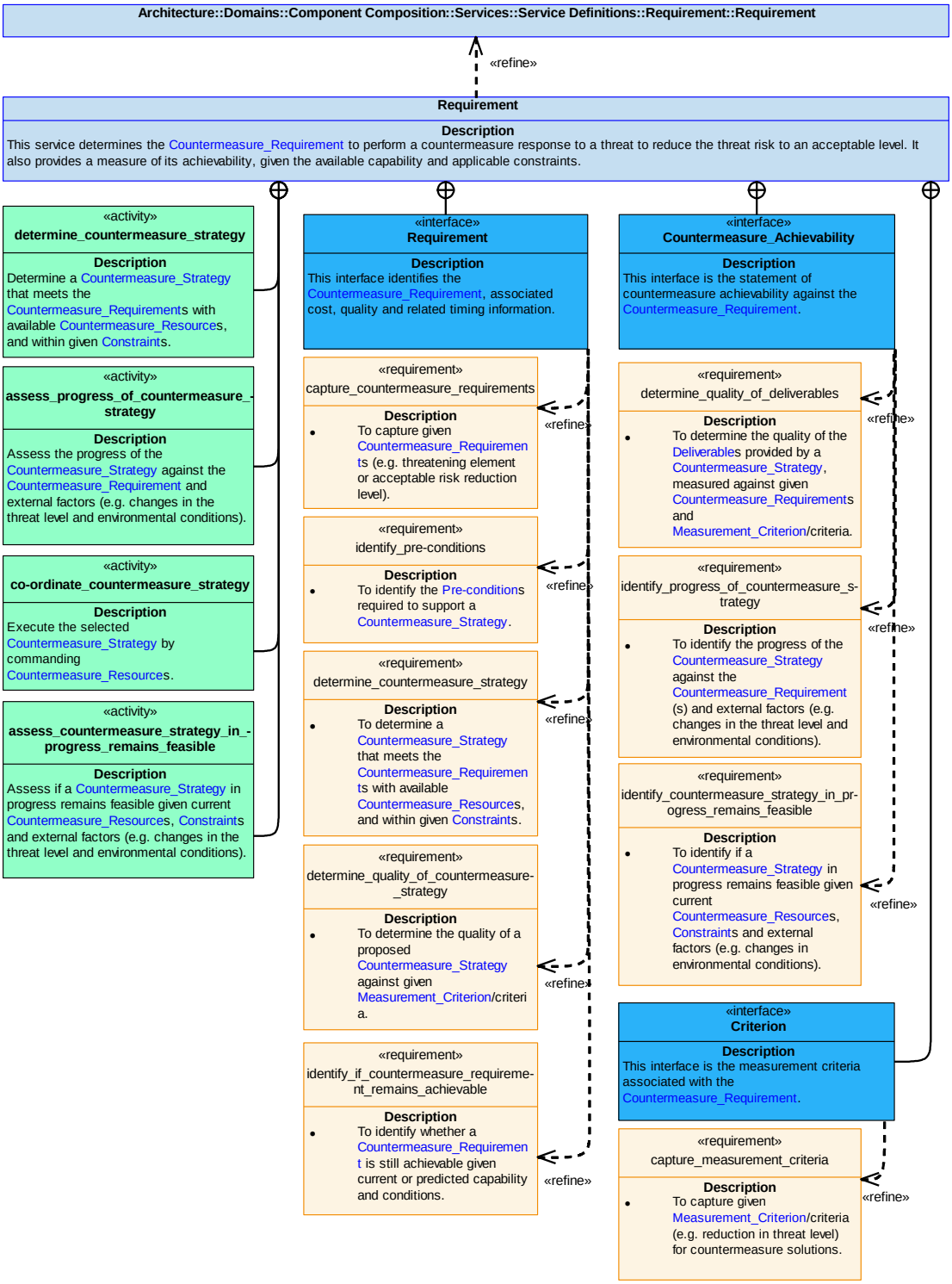

Figure 229: Requirement Service Definition

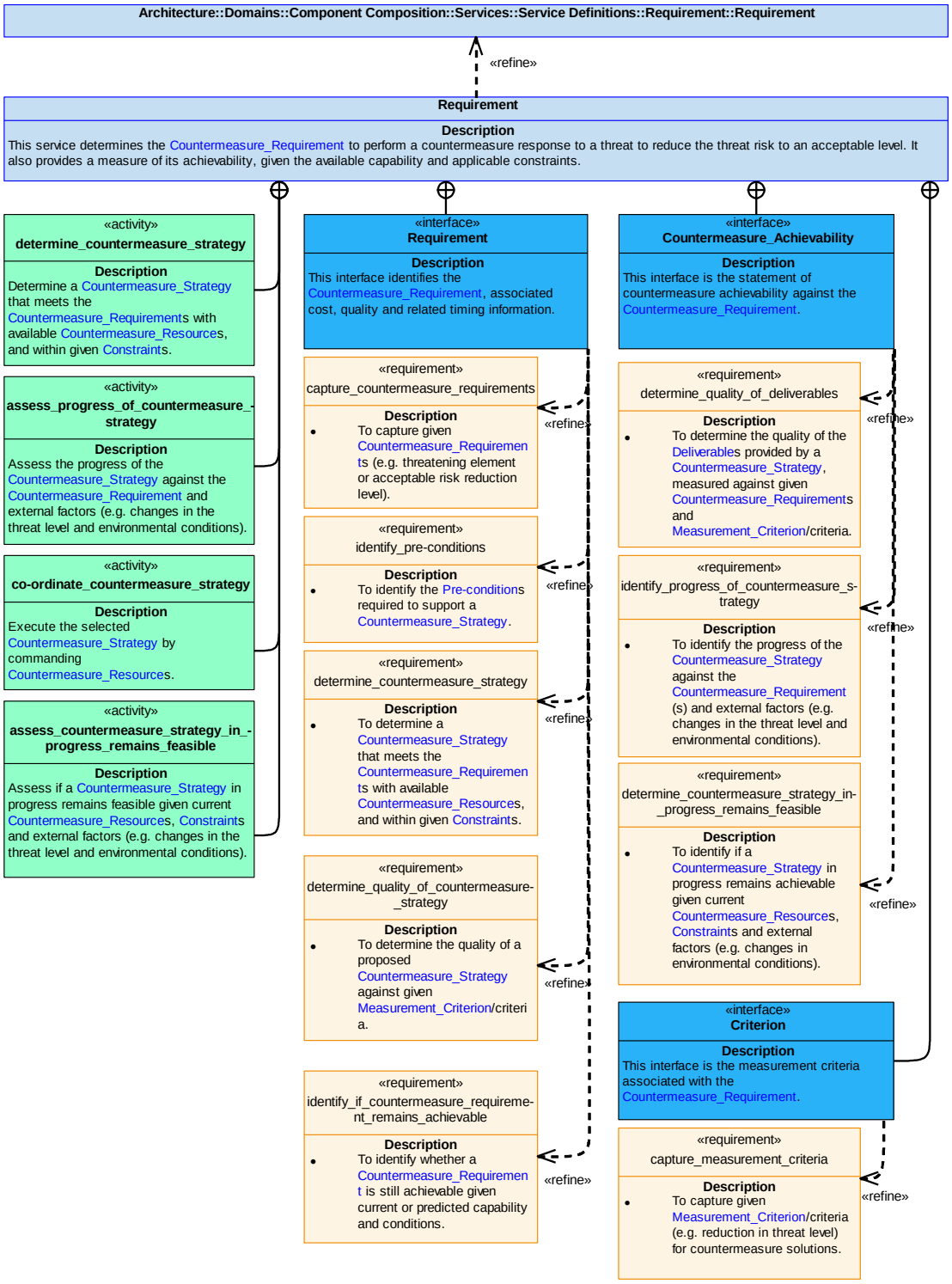

Figure 230: Requirement Service Policy

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).

B.2.8.7.1.2 Deployable_Asset_Use

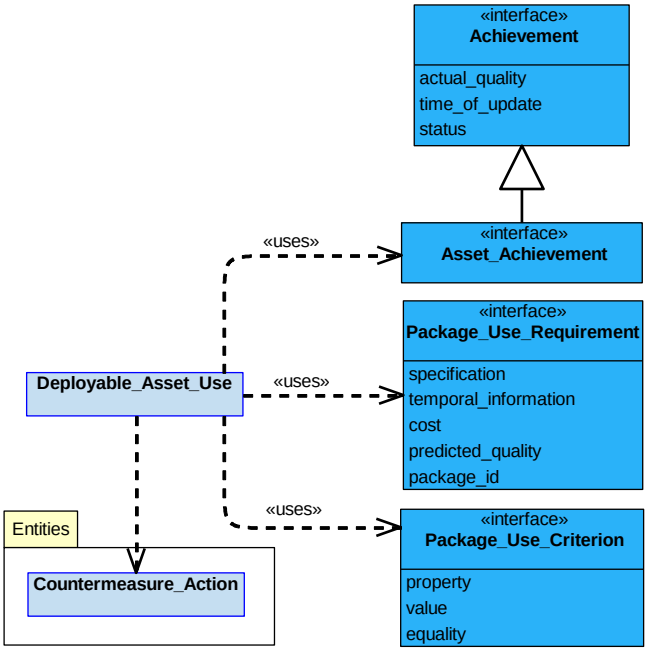

Figure 231: Deployable_Asset_Use Service Definition

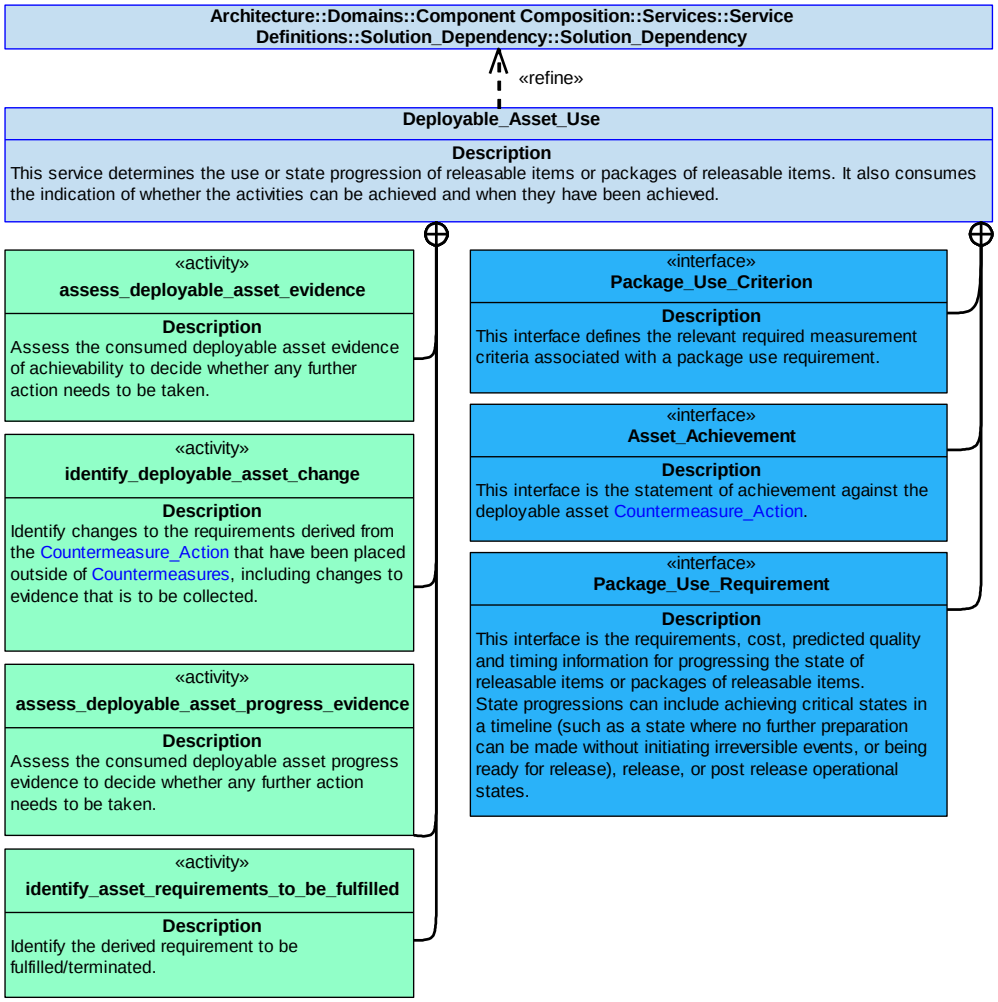

Figure 232: Deployable_Asset_Use Service Policy

Deployable_Asset_Use

This service determines the use or state progression of releasable items or packages of releasable items. It also consumes the indication of whether the activities can be achieved and when they have been achieved.

Interfaces

Package_Use_Criterion

This interface defines the relevant required measurement criteria associated with a package use requirement.

Attributes

property	The property to be measured, e.g. threat mitigation.
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.

Package_Use_Requirement

This interface is the requirements, cost, predicted quality and timing information for progressing the state of releasable items or packages of releasable items.

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.

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.
predicted_quality	How well the proposed deployable asset solution is predicted to satisfy the requirement.
package_id	A package identifier, as created by the Deployable_Asset_Package_Creation service.

Asset_Achievement

This interface is the statement of achievement against the deployable asset Countermeasure_Action.

Activities

assess_deployable_asset_progress_evidence

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

identify_deployable_asset_change

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

identify_asset_requirements_to_be_fulfilled

Identify the derived requirement to be fulfilled/terminated.

assess_deployable_asset_evidence

Assess the consumed deployable asset evidence of achievability to decide whether any further action needs to be taken.

B.2.8.7.1.3 Formation

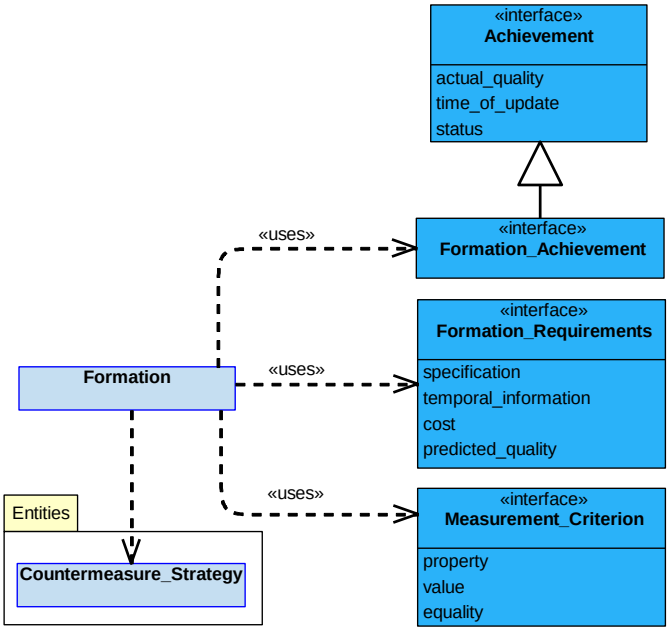

Figure 233: Formation Service Definition

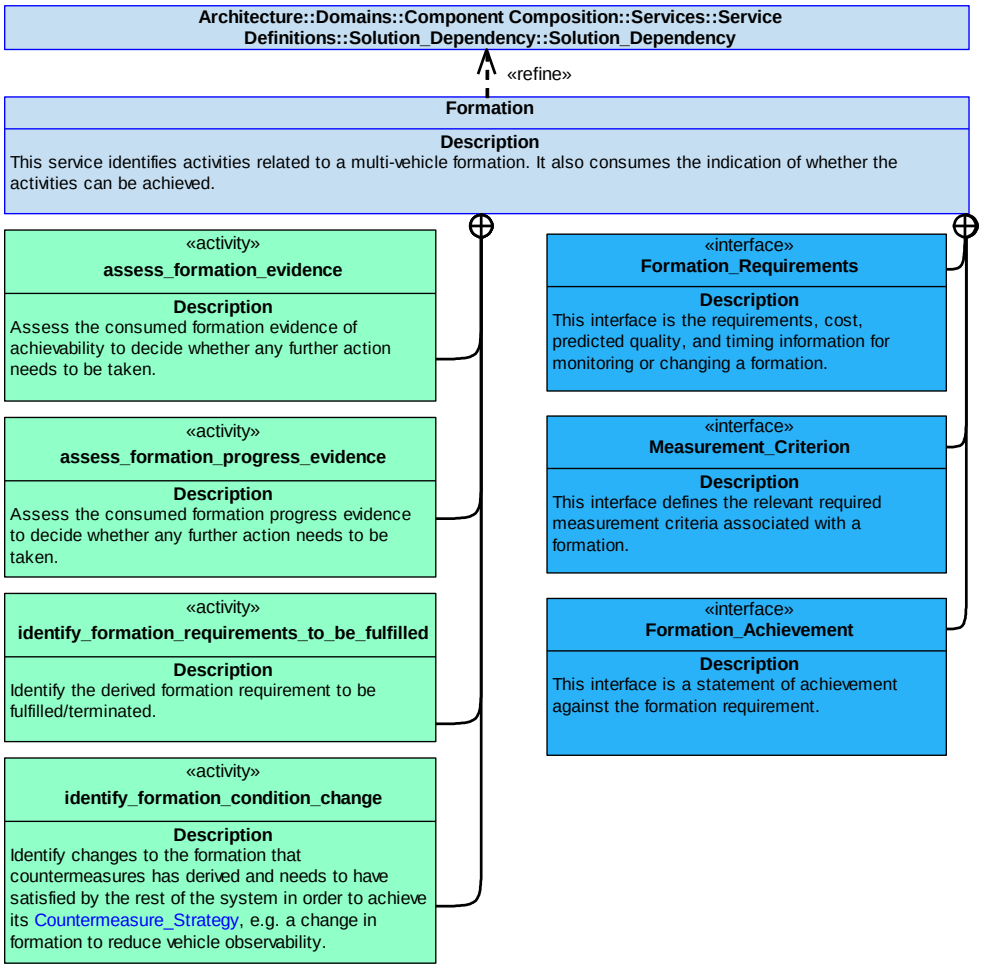

Figure 234: Formation Service Policy

Formation

This service identifies activities related to a multi-vehicle formation. It also consumes the indication of whether the activities can be achieved.

Interfaces

Formation_Requirements

This interface is the requirements, cost, predicted quality, and timing information for monitoring or changing a formation.

Attributes

specification	The definition of the derived requirement, e.g. change formation to pattern 'nnnn'
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 formation solution is predicted to satisfy the requirement.

Measurement_Criterion

This interface defines the relevant required measurement criteria associated with a formation.

Attributes

property	The property to be measured, e.g. the new formation arrangement.
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.

Formation_Achievement

This interface is a statement of achievement against the formation requirement.

Activities

assess_formation_evidence

Assess the consumed formation evidence of achievability to decide whether any further action needs to be taken.

assess_formation_progress_evidence

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

identify_formation_condition_change

Identify changes to the formation 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 formation to reduce vehicle observability.

identify_formation_requirements_to_be_fulfilled

Identify the derived formation requirement to be fulfilled/terminated.

B.2.8.7.1.4 Deployable_Asset_Package_Creation

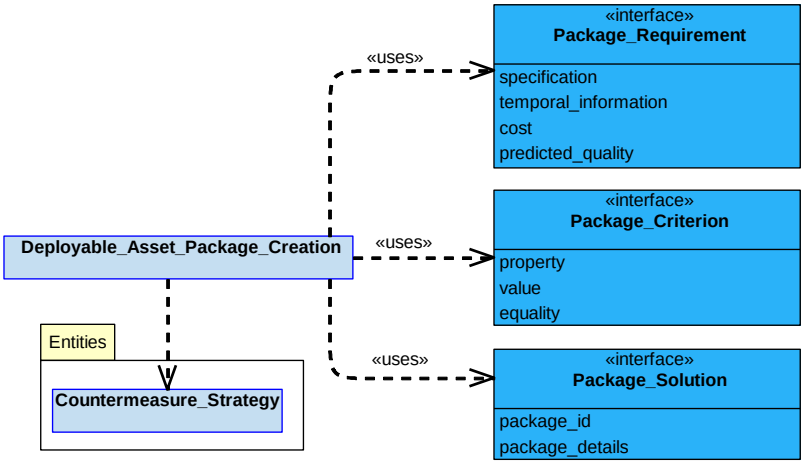

Figure 235: Deployable_Asset_Package_Creation Service Definition

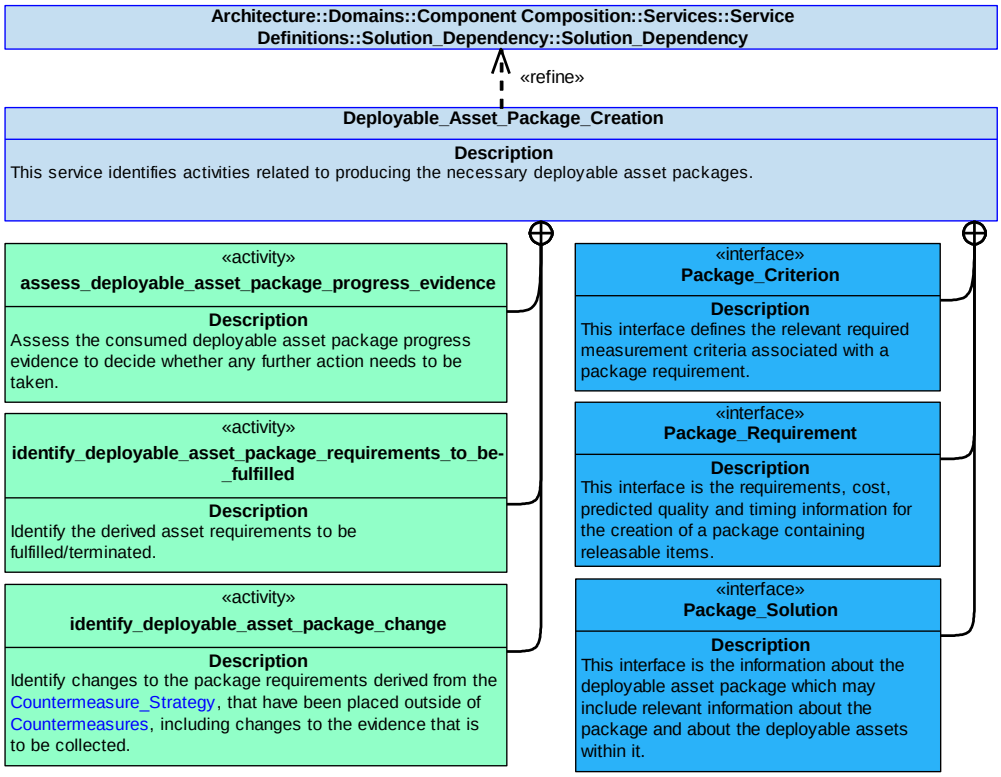

Figure 236: Deployable_Asset_Package_Creation Service Policy

Deployable_Asset_Package_Creation

This service identifies activities related to producing the necessary deployable asset packages.

Interfaces

Package_Criterion

This interface defines the relevant required measurement criteria associated with a package requirement.

Attributes

property	The property to be measured, e.g. minimum probability of success against a specified target.
value	The amount related to the property to be measured, e.g. 80% probable.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Package_Requirement

This interface is the requirements, cost, predicted quality and timing information for the creation of a package containing releasable items.

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.
predicted_quality	How well the proposed asset package solution is predicted to satisfy the requirement.

Package_Solution

This interface is the information about the deployable asset package which may include relevant information about the package and about the deployable assets within it.

Attributes

package_id	A package identifier, allowing the package to be referenced by other services.
package_details	Information about the package and about the deployable assets within it. This may include unique references to each item within the package allowing them to be handled separately.

Activities

assess_deployable_asset_package_progress_evidence

Assess the consumed deployable asset package progress evidence to decide whether any further action needs to be taken.

identify_deployable_asset_package_change

Identify changes to the package requirements derived from the Countermeasure_Strategy, that have been placed outside of Countermeasures, including changes to the evidence that is to be collected.

identify_deployable_asset_package_requirements_to_be_fulfilled

Identify the derived asset requirements to be fulfilled/terminated.

B.2.8.7.1.5 Effect

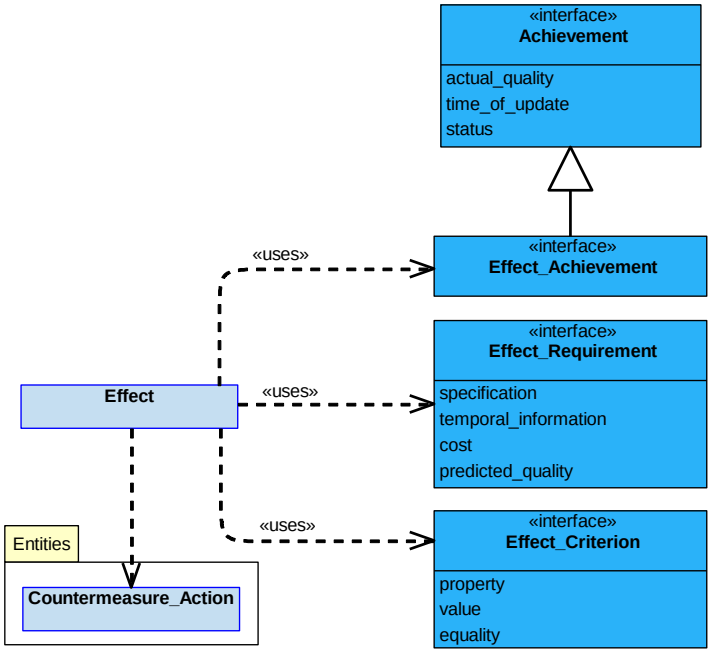

Figure 237: Effect Service Definition

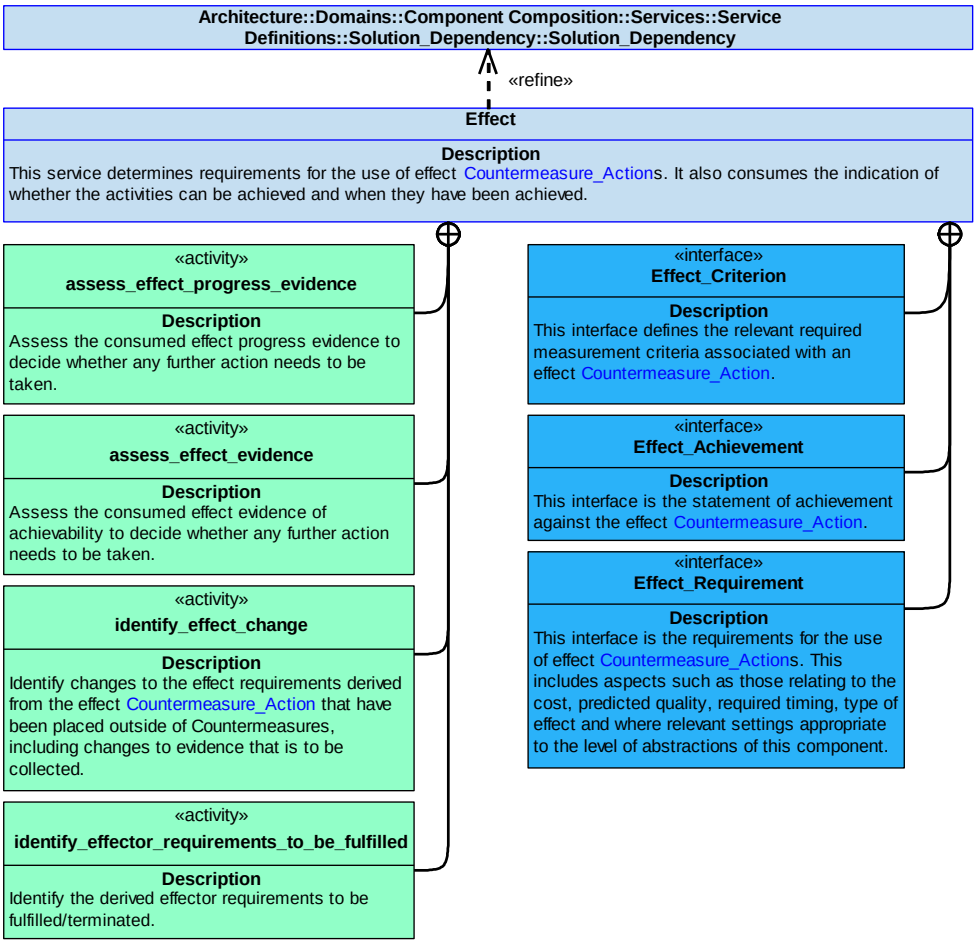

Figure 238: Effect Service Policy

Effect

This service determines requirements for the use of effect Countermeasure_Actions. It also consumes the indication of whether the activities can be achieved and when they have been achieved.

Interfaces

Effect_Criterion

This interface defines the relevant required measurement criteria associated with an effect Countermeasure_Action.

Attributes

property	The property to be measured, e.g. minimum level of reduction in threat risk.
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.

Effect_Requirement

This interface is the requirements for the use of effect Countermeasure_Actions. This includes aspects such as those relating to the cost, predicted quality, required timing, type of effect and where relevant settings appropriate to the level of abstractions of this component.

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.
predicted_quality	How well the proposed effect is predicted to satisfy the requirement.

Effect_Achievement

This interface is the statement of achievement against the effect Countermeasure_Action.

Activities

assess_effect_progress_evidence

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

identify_effect_change

Identify changes to the effect requirements derived from the effect Countermeasure_Action that have been placed outside of Countermeasures, including changes to evidence that is to be collected.

identify_effector_requirements_to_be_fulfilled

Identify the derived effector requirements to be fulfilled/terminated.

assess_effect_evidence

Assess the consumed effect evidence of achievability to decide whether any further action needs to be taken.

B.2.8.7.1.6 Spectrum_Use

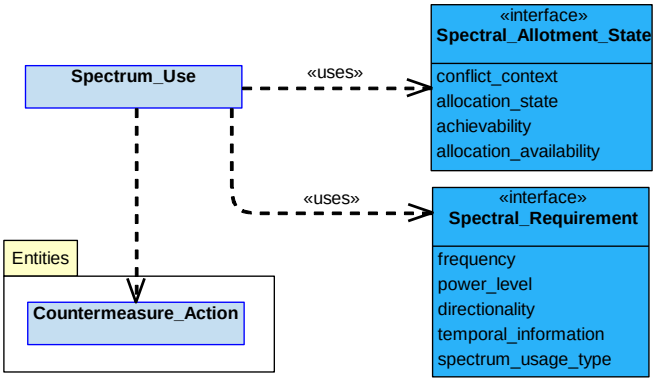

Figure 239: Spectrum_Use Service Definition

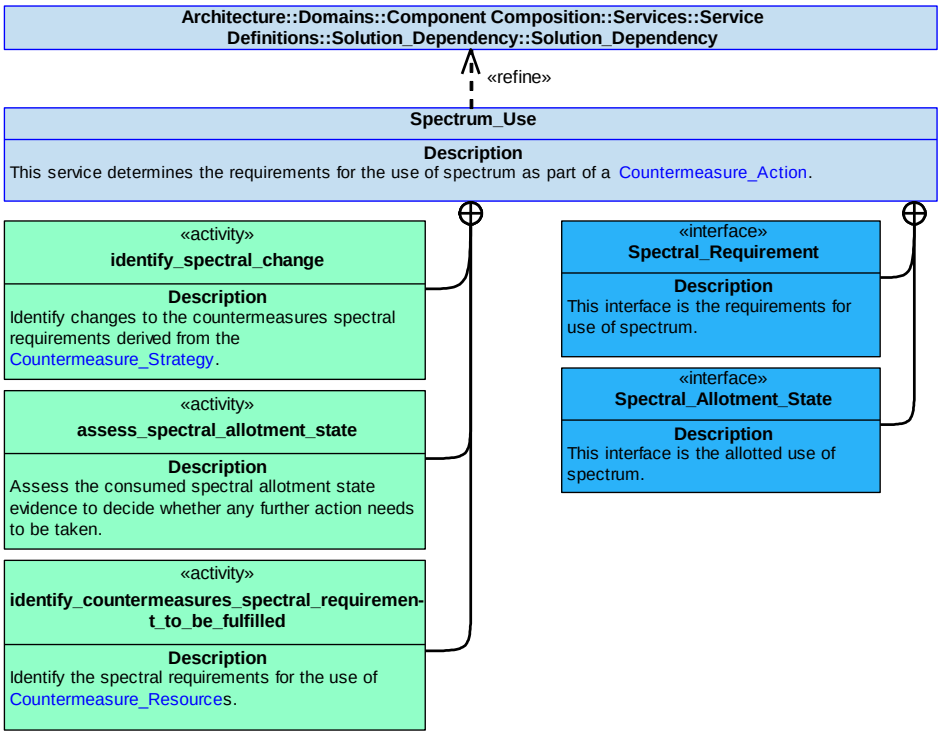

Figure 240: Spectrum_Use Service Policy

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

conflict_context	Identifies the source or reason for an allocation conflict.
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). The response can include conditions/constraints that are non-binary allowing for operationally acceptable consequences for the requester.
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.

B.2.8.7.1.7 Vehicle_Condition

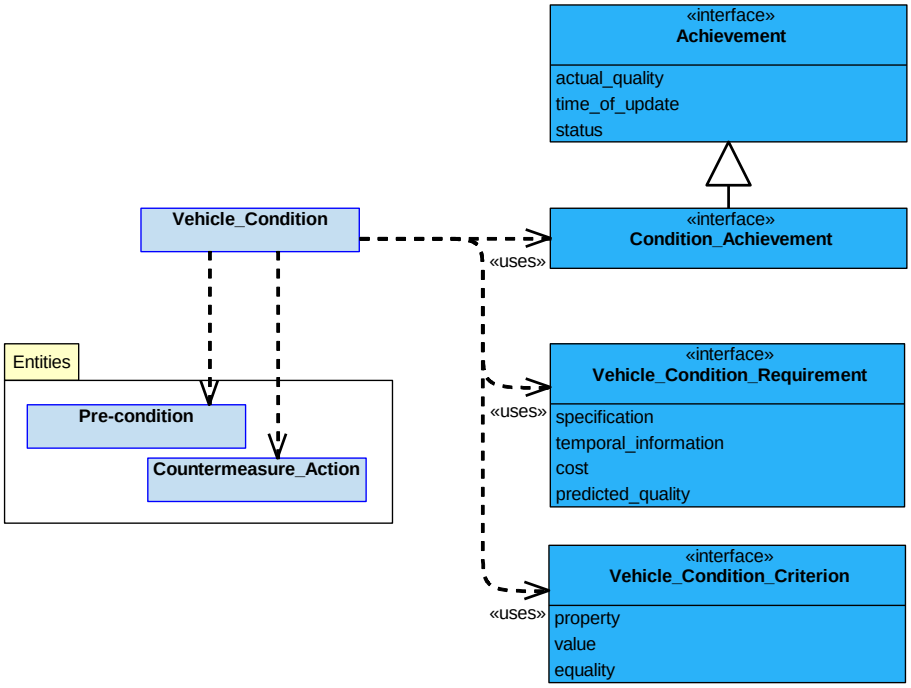

Figure 241: Vehicle_Condition Service Definition

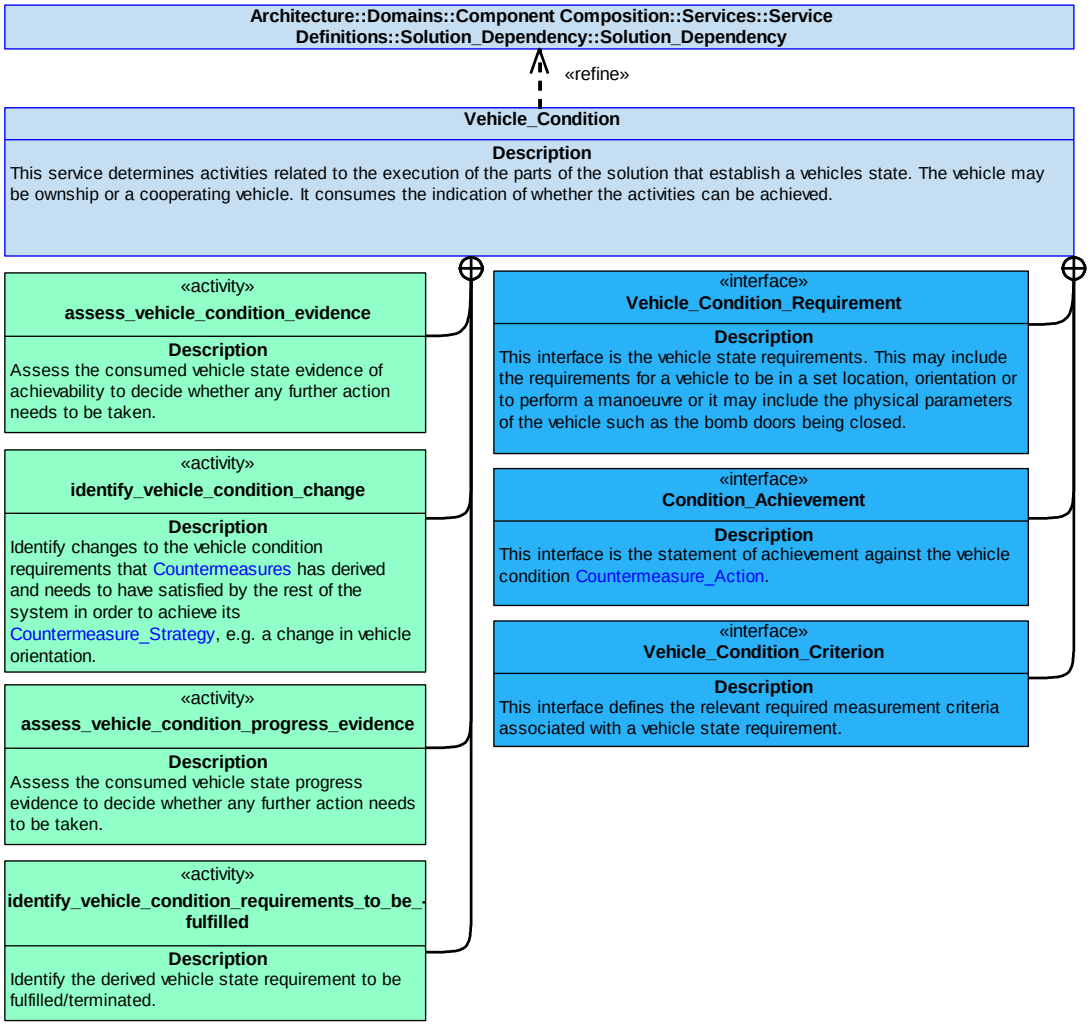

Figure 242: Vehicle_Condition Service Policy

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 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 closed.

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.
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 Countermeasure_Action.

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.

B.2.8.7.1.8 Vehicle_Observability

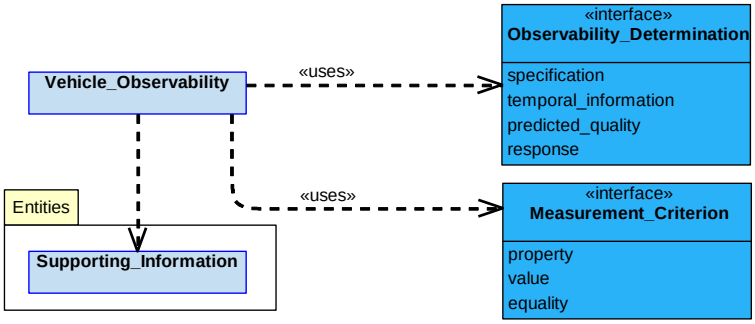

Figure 243: Vehicle_Observability Service Definition

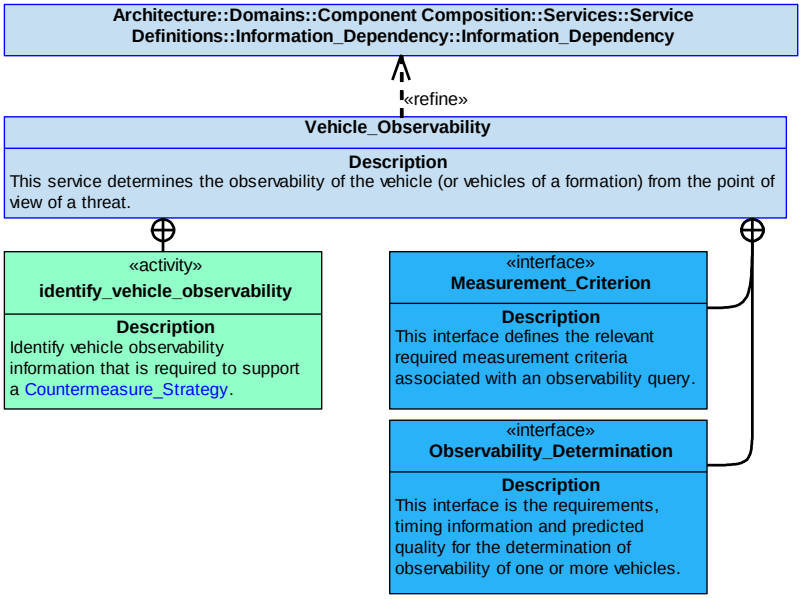

Figure 244: Vehicle_Observability Service Policy

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.

Activity

identify_vehicle_observability

Identify vehicle observability information that is required to support a Countermeasure_Strategy.

B.2.8.7.1.9 Object_Information

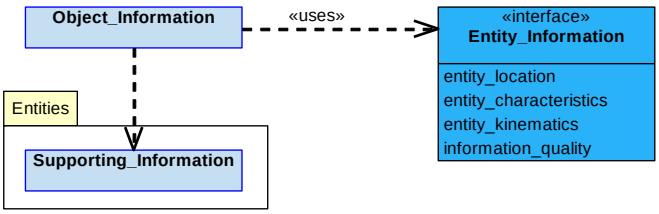

Figure 245: Object_Information Service Definition

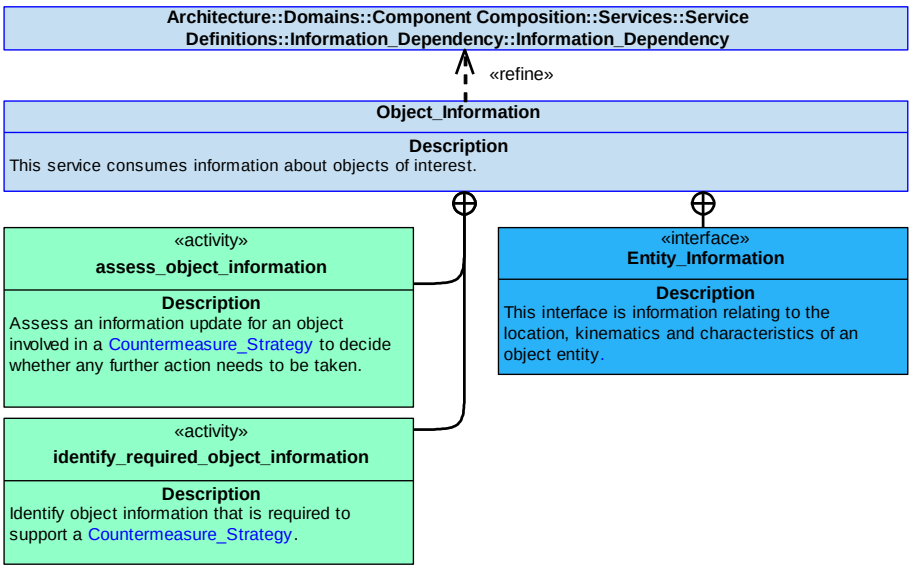

Figure 246: Object_Information Service Policy

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.

B.2.8.7.1.10 Environment_Information

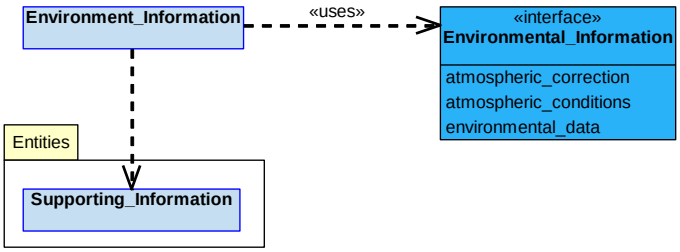

Figure 247: Environment_Information Service Definition

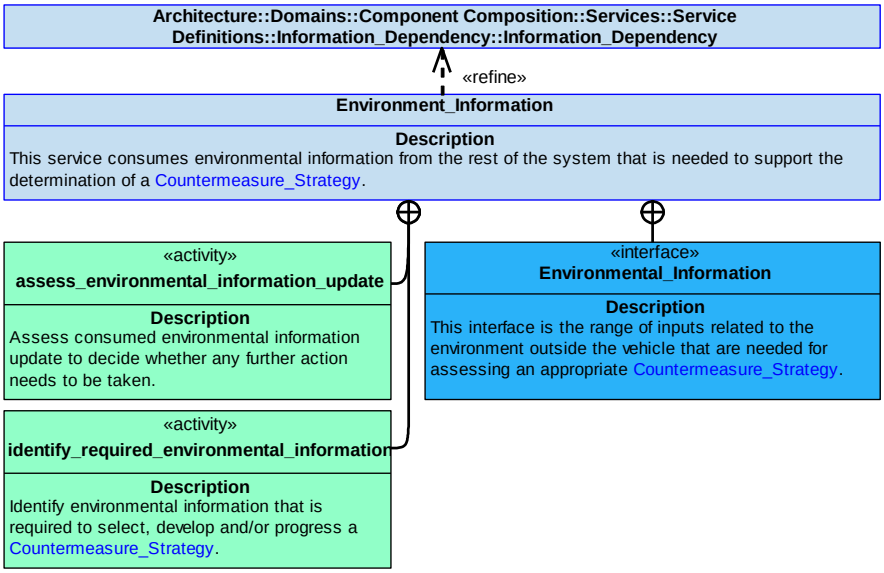

Figure 248: Environment_Information Service Policy

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.

B.2.8.7.1.11 Threat_Level

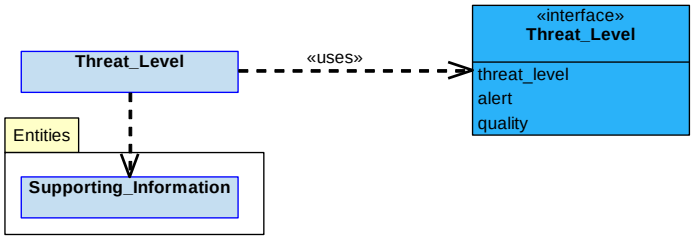

Figure 249: Threat_Level Service Definition

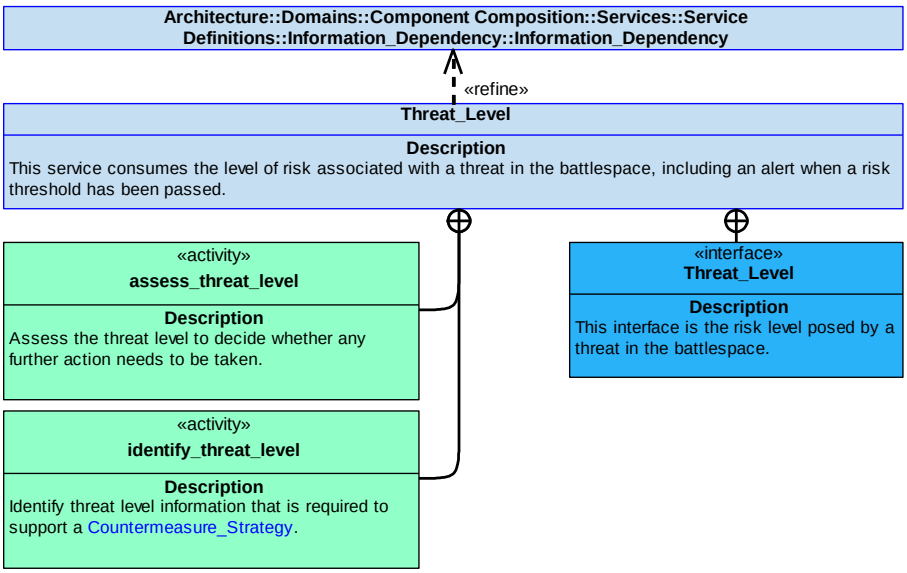

Figure 250: Threat_Level Service Policy

Threat_Level

This service consumes the level of risk associated with a threat in the battlespace, including an alert when a risk threshold has been passed.

Interface

Threat_Level

This interface is the risk level posed by a threat in the battlespace.

Attributes

threat_level	The level of risk posed by a threat in the battlespace.
alert	An indication that the risk threshold of a threat has been passed.
quality	The quality of the level of risk information.

Activities

assess_threat_level

Assess the threat level to decide whether any further action needs to be taken.

identify_threat_level

Identify threat level information that is required to support a Countermeasure_Strategy.

B.2.8.7.1.12 Constraint

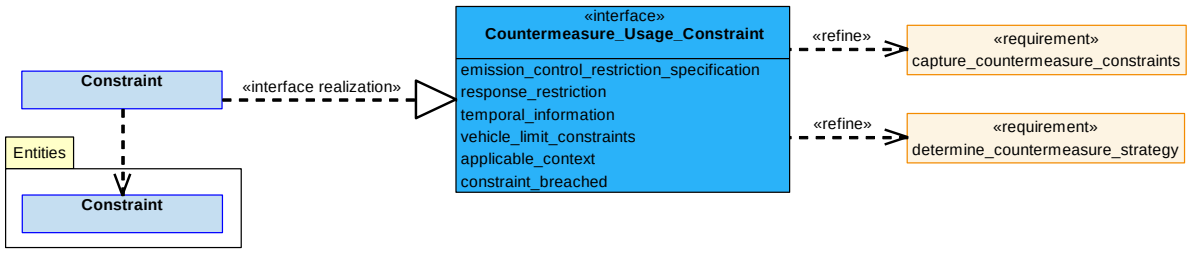

Figure 251: Constraint Service Definition

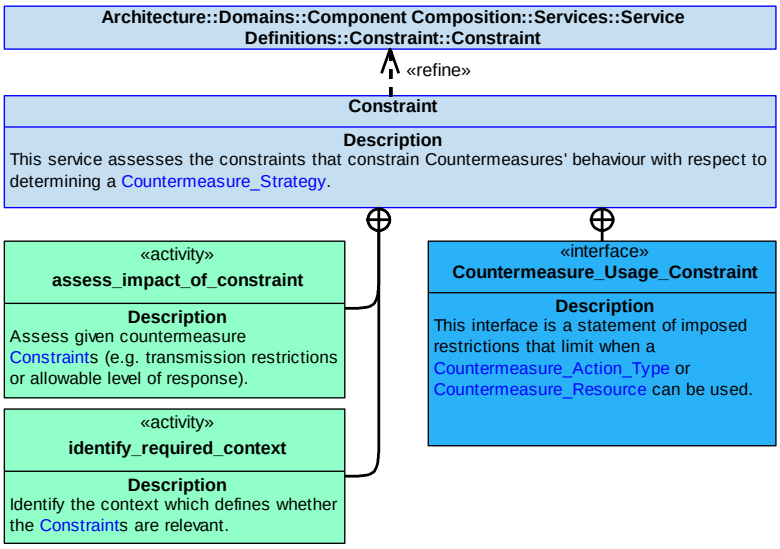

Figure 252: Constraint Service Policy

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.

B.2.8.7.1.13 Capability

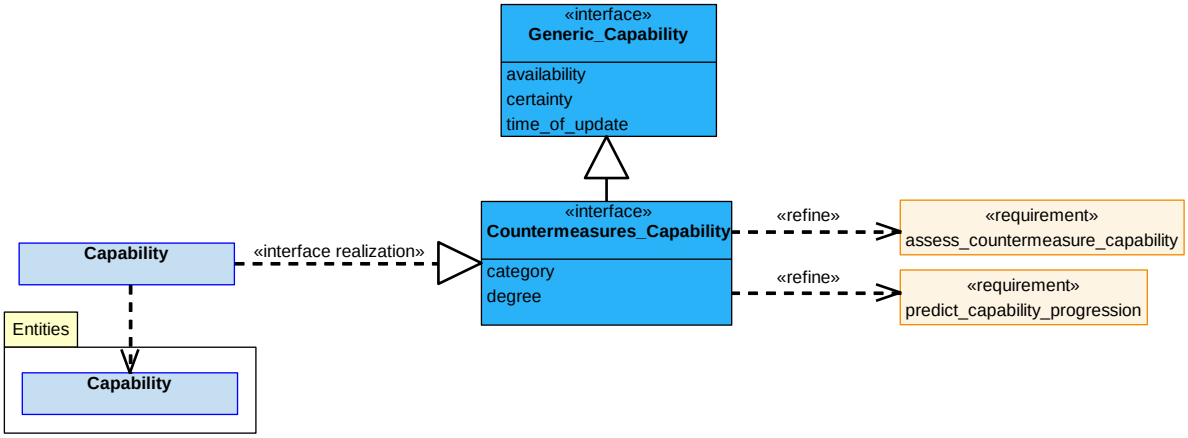

Figure 253: Capability Service Definition

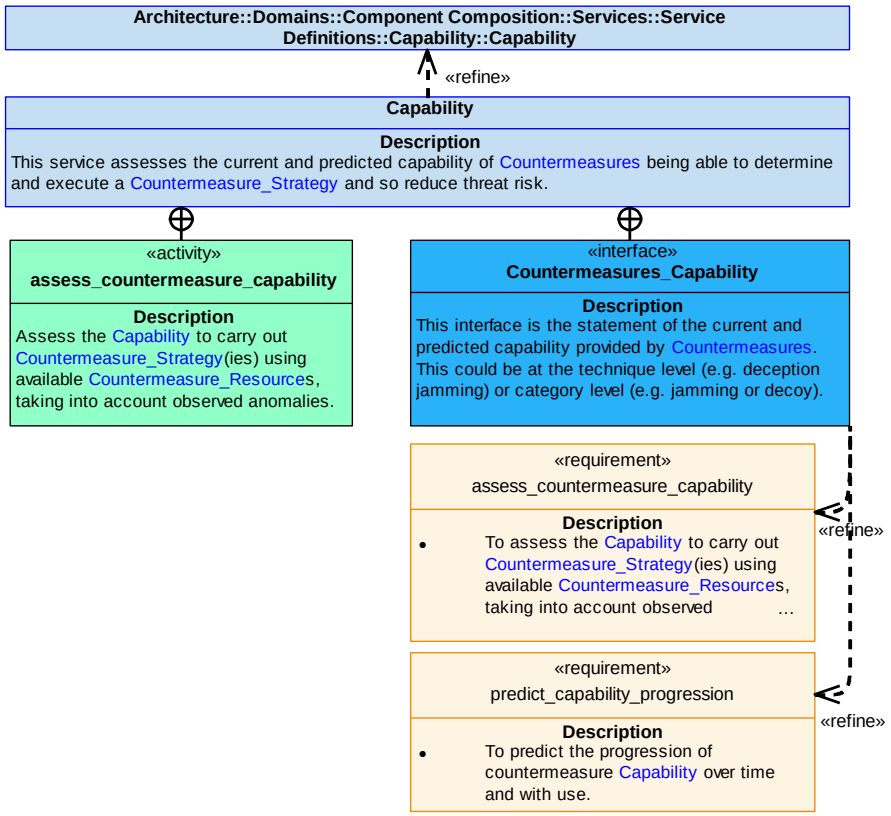

Figure 254: Capability Service Policy

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 Countermeasure_Strategy(ies) using available Countermeasure_Resources, taking into account observed anomalies.

B.2.8.7.1.14 Capability_Evidence

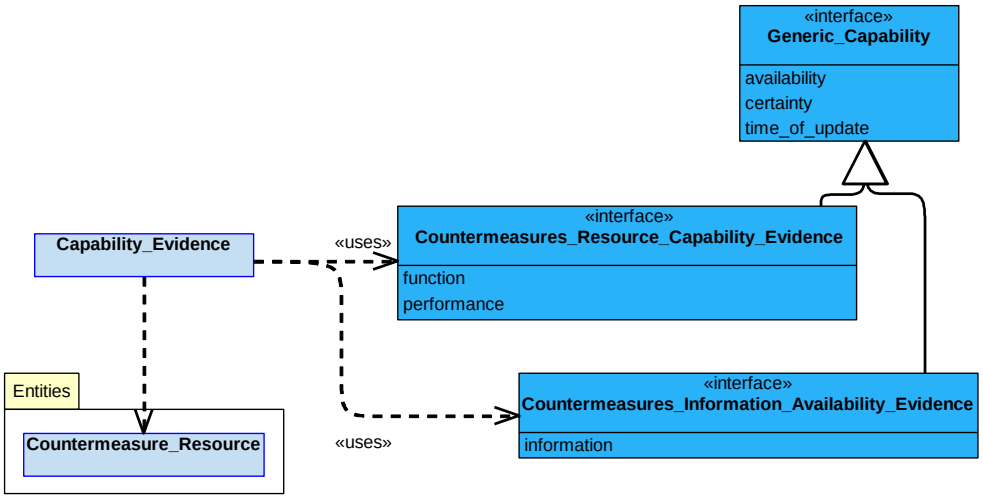

Figure 255: Capability_Evidence Service Definition

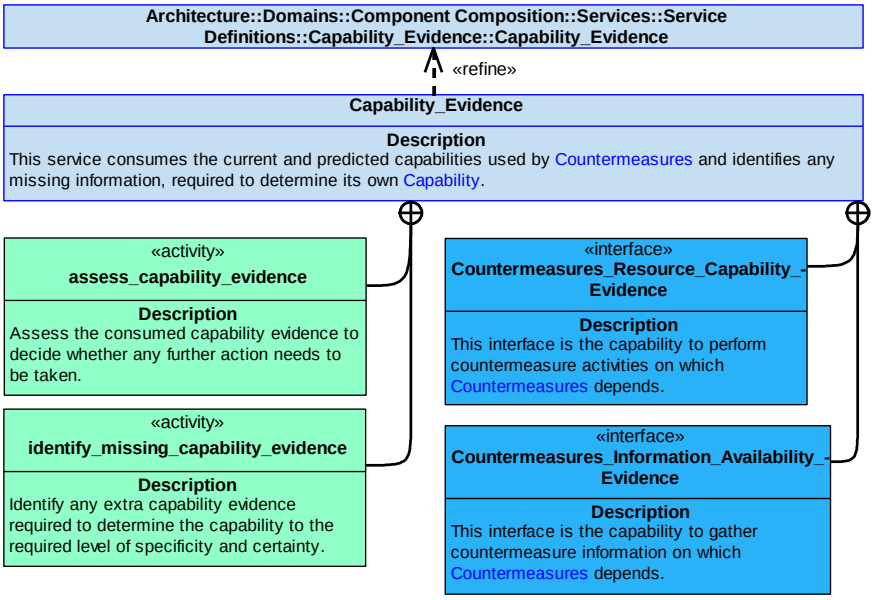

Figure 256: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capabilities used by Countermeasures and identifies any missing information, required to determine its own Capability.

Interfaces

Countermeasures_Resource_Capability_Evidence

This interface is the capability to perform countermeasure activities on which Countermeasures depends.

Attributes

function	The specific function or technique defined by this Countermeasure_Resource, including the control options for its use (e.g. spot jamming or barrage jamming).
performance	The level or degree of capability available for this particular function, taking into account the type, location and fit of the Countermeasures equipment on the vehicle. For example, field of regard, jamming to signal ratio or burnthrough range.

Countermeasures_Information_Availability_Evidence

This interface is the capability to gather countermeasure information on which Countermeasures depends.

Attribute

information	The definition of the information required to allow creation of a Countermeasure_Strategy.

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.

B.2.8.7.2 Service Dependencies

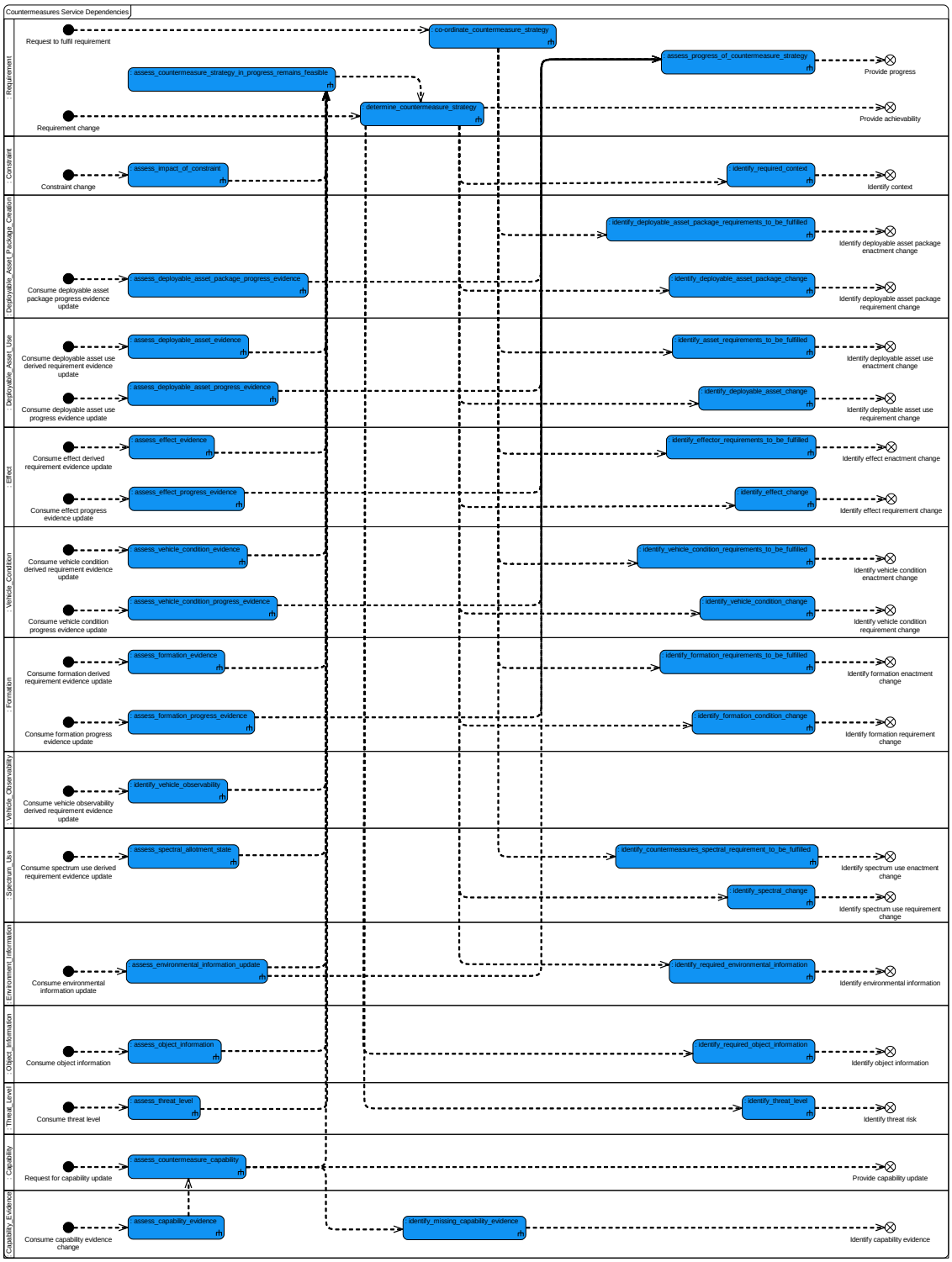

Figure 257: Countermeasures Service Dependencies

B.2.9 Cryptographic Materials

B.2.9.1 Role

The role of Cryptographic Materials is to manage and distribute cryptographic keys, algorithms and certificates.

B.2.9.2 Overview

Control Architecture

Cryptographic Materials is a service component as defined in the Control Architecture policy.

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.

B.2.9.3 Service Summary

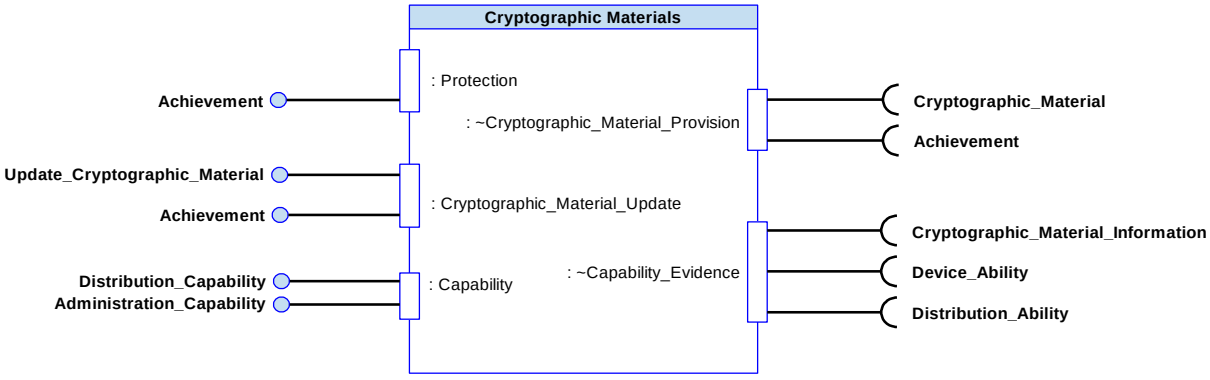

Figure 258: Cryptographic Materials Service Summary

B.2.9.4 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).

distribute_crypto_material

- To distribute the required Cryptographic_Material.

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.

determine_crypto_material_usage

- To determine when and where particular Cryptographic_Material needs to be used.

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.

identify_material_solution_progress

- To identify what progress has been made against the Requirement.

B.2.9.5 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.

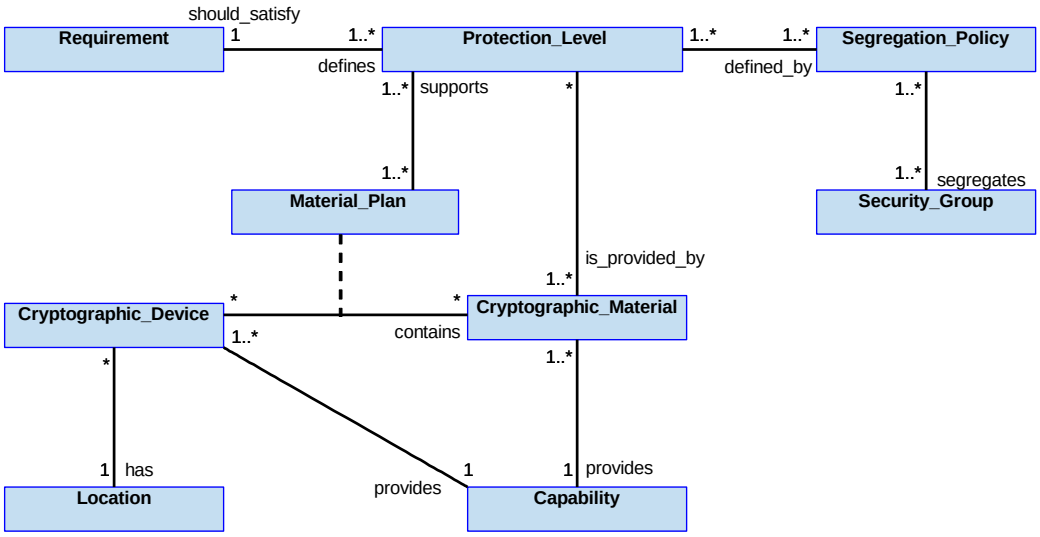

Figure 259: Cryptographic Materials Semantics

B.2.9.5.1 Entities

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.

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.

Cryptographic_Material

An item used in the process of encryption or decryption, e.g. a key, algorithm (including for hash functions) or certificate.

Segregation_Policy

The definition of the specific security domains, their segregation and use.

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.

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.

Capability

The capability of the component to update and make cryptographic material available for use.

Security_Group

A group of items that have similar segregation requirements, e.g. security domains, types and classification of data.

B.2.9.6 Design Rationale

B.2.9.6.1 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.

B.2.9.6.2 Design Considerations

Directly Applicable Policies

These policies 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. Cryptographic Materials will however report any store/device that fails to confirm that a request to sanitise has 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.

B.2.9.6.3 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.

B.2.9.6.4 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.

B.2.9.7 Services

B.2.9.7.1 Service Definitions

B.2.9.7.1.1 Protection

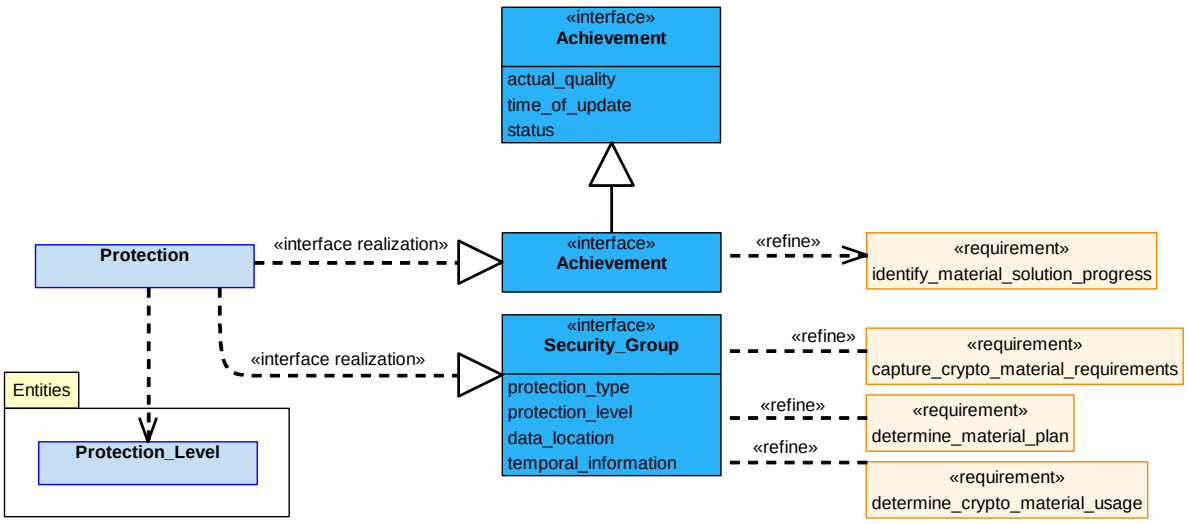

Figure 260: Protection Service Definition

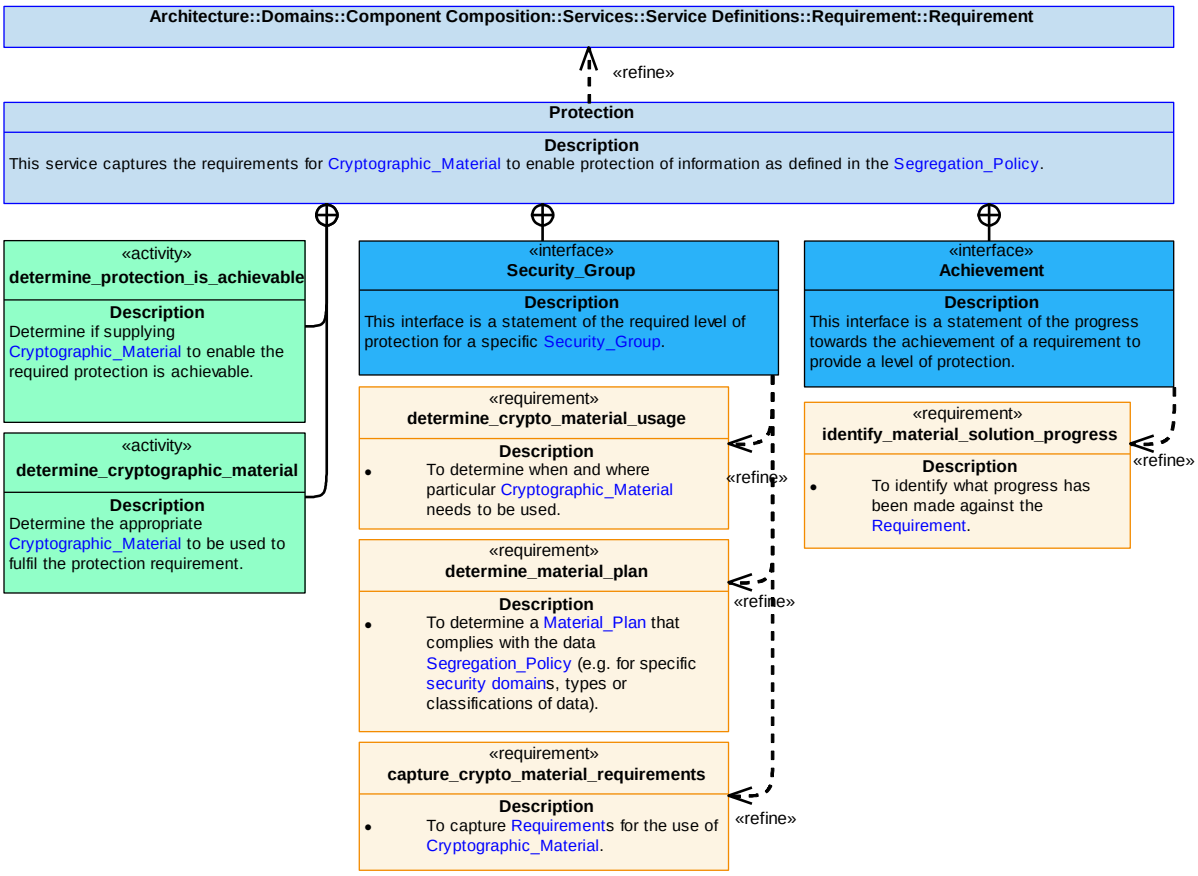

Figure 261: Protection Service Policy

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.

B.2.9.7.1.2 Cryptographic_Material_Provision

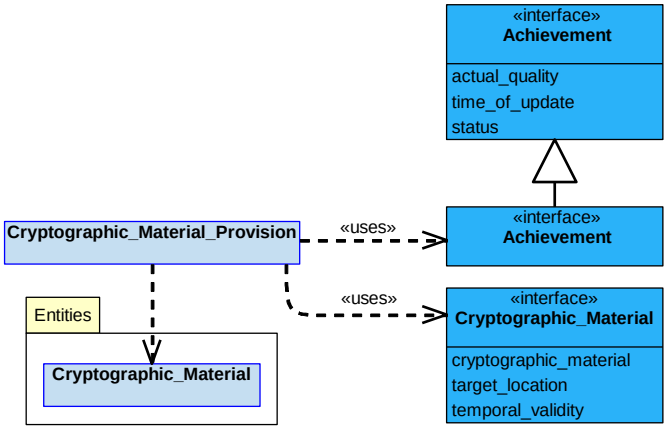

Figure 262: Cryptographic_Material_Provision Service Definition

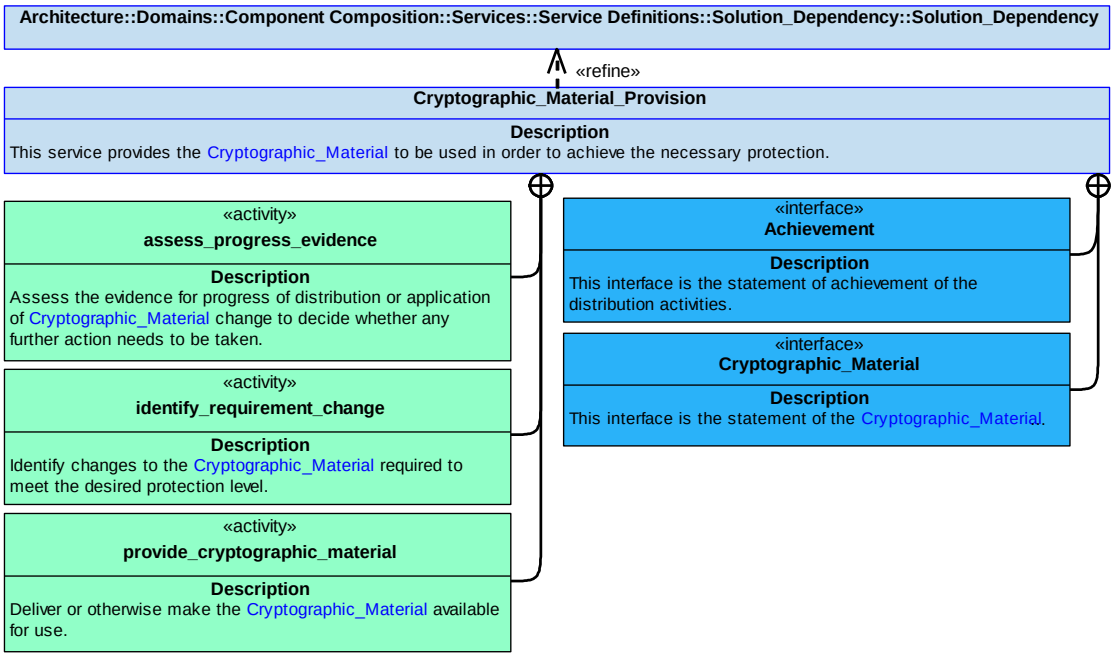

Figure 263: Cryptographic_Material_Provision Service Policy

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.

B.2.9.7.1.3 Cryptographic_Material_Update

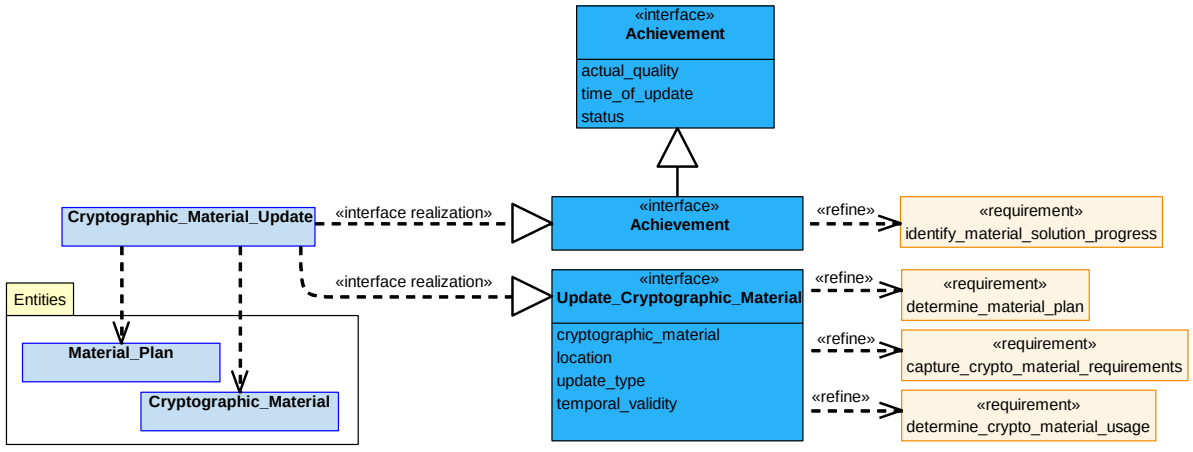

Figure 264: Cryptographic_Material_Update Service Definition

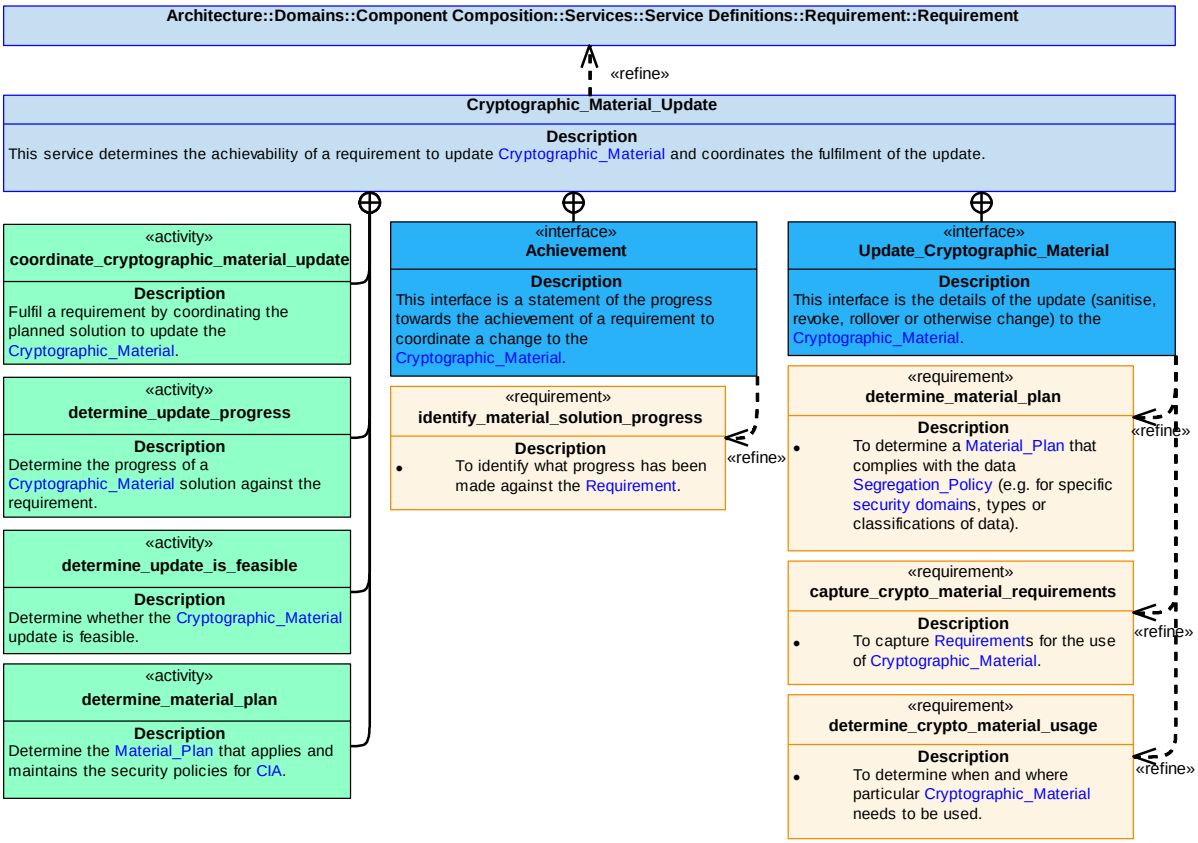

Figure 265: Cryptographic_Material_Update Service Policy

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.

B.2.9.7.1.4 Capability

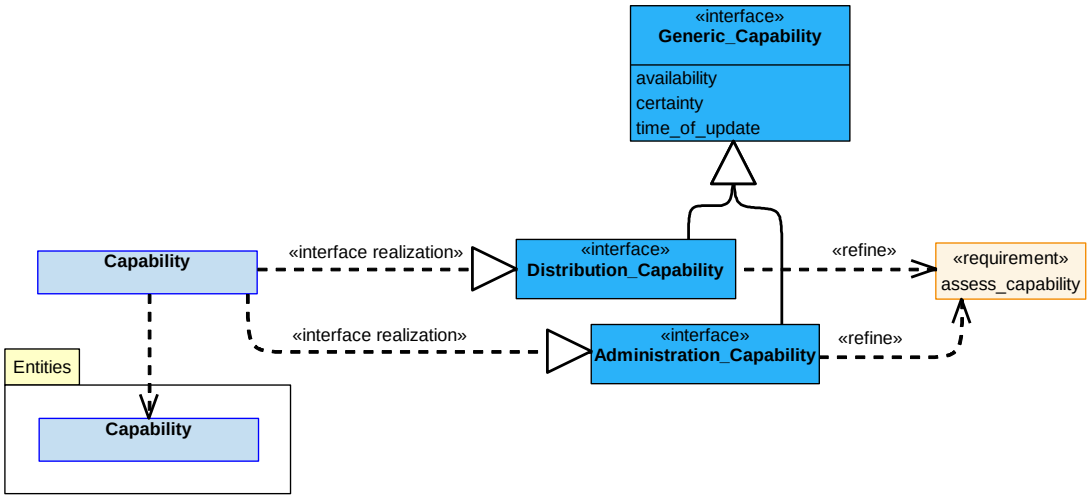

Figure 266: Capability Service Definition

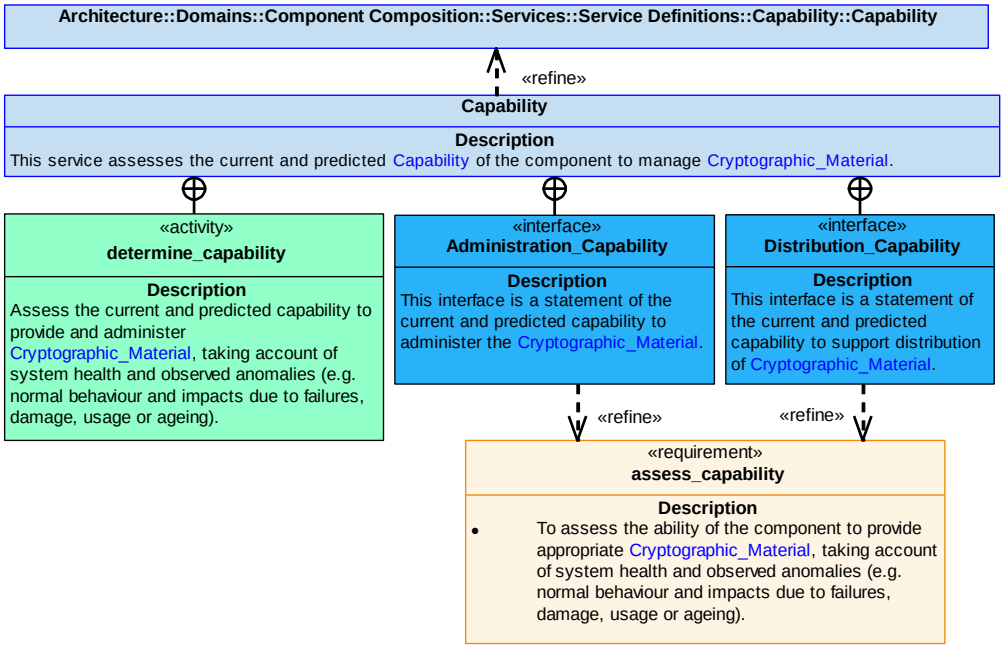

Figure 267: Capability Service Policy

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).

B.2.9.7.1.5 Capability_Evidence

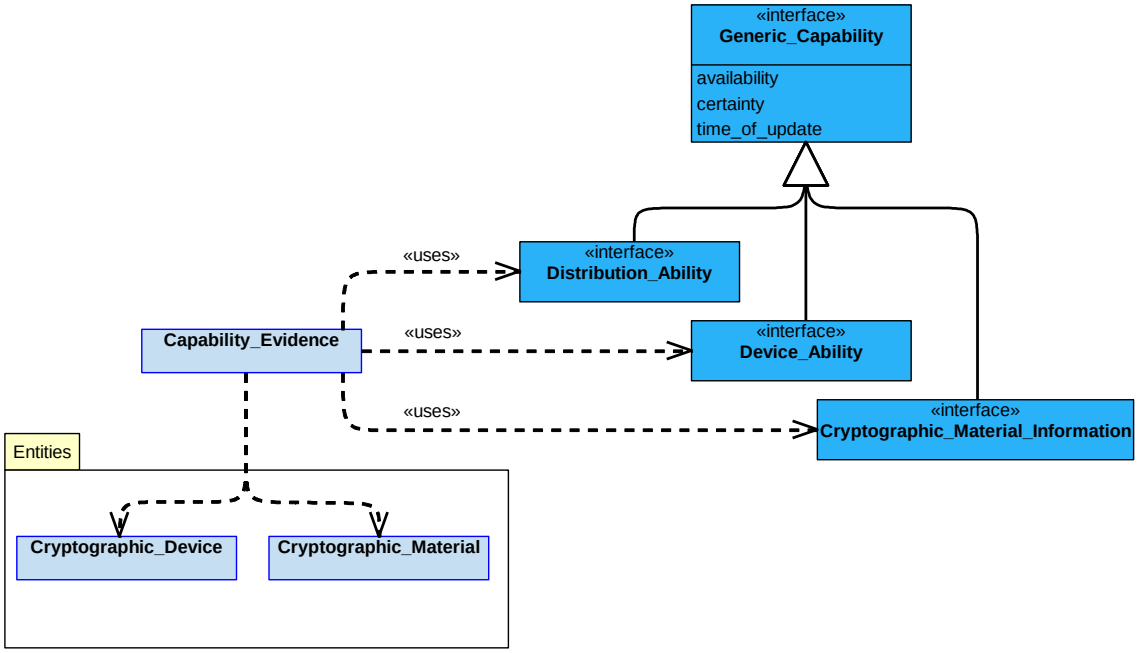

Figure 268: Capability_Evidence Service Definition

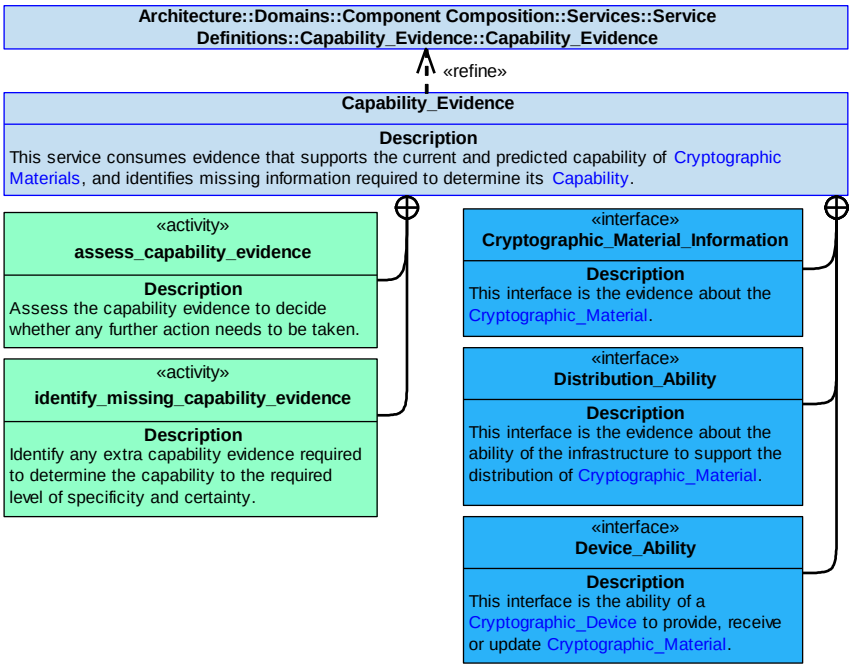

Figure 269: Capability_Evidence Service Policy

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.

B.2.9.7.2 Service Dependencies

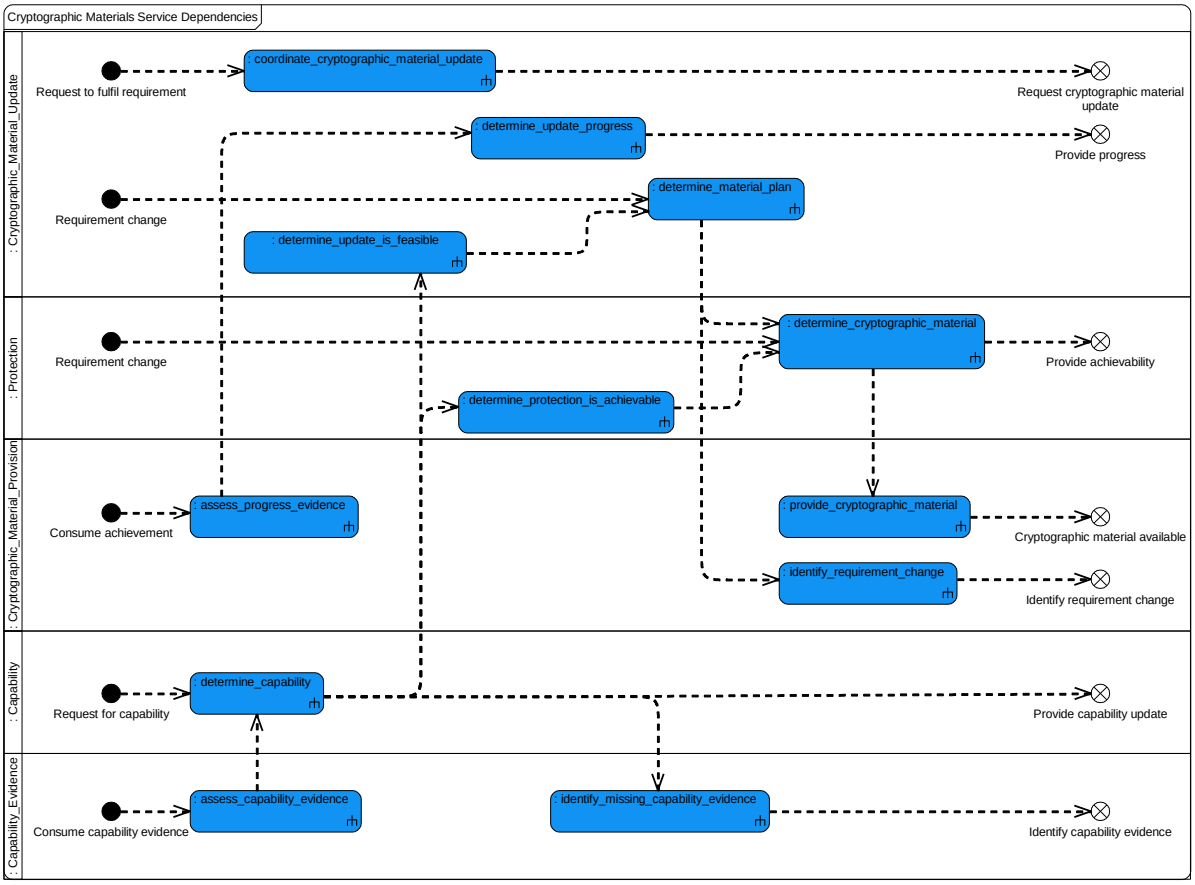

Figure 270: Cryptographic Materials Service Dependencies

B.2.10 Cryptographic Methods

B.2.10.1 Role

The role of Cryptographic Methods is to perform cryptographic transformations.

B.2.10.2 Overview

Control Architecture

Cryptographic Methods is a service component as defined in the Control Architecture policy.

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).

B.2.10.3 Service Summary

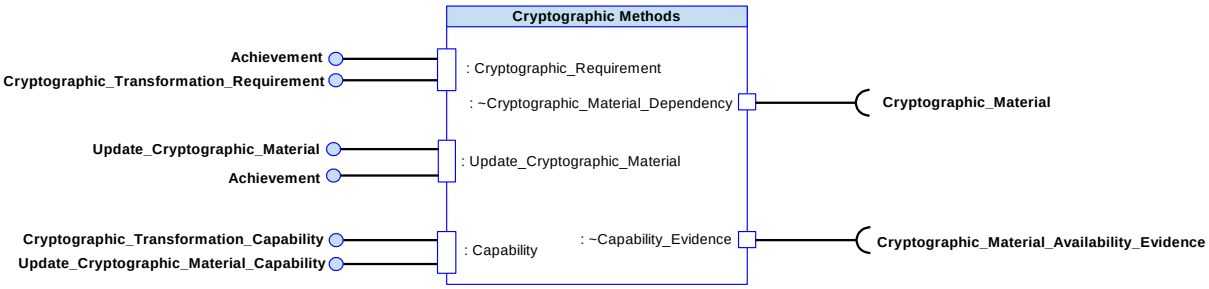

Figure 271: Cryptographic Methods Service Summary

B.2.10.4 Responsibilities

capture_cryptographic_requirement

- To capture Requirements for a Cryptographic_Function (e.g. encryption, decryption, or hashing).

capture_cryptographic_material

- To capture provided Cryptographic_Material.

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.

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.

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.

B.2.10.5 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.

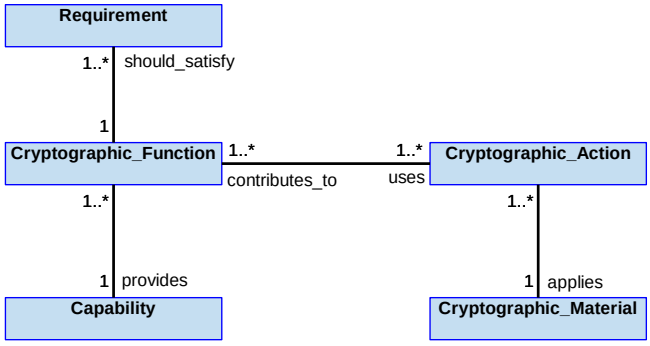

Figure 272: Cryptographic Methods Semantics

B.2.10.5.1 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.

B.2.10.6 Design Rationale

B.2.10.6.1 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.

B.2.10.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy will carry out logging of cryptography events.

Extensions

- Different cryptographic methods can be accommodated by extensions.

B.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. 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 action 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 policy.

B.2.10.6.4 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.

B.2.10.7 Services

B.2.10.7.1 Service Definitions

B.2.10.7.1.1 Cryptographic_Requirement

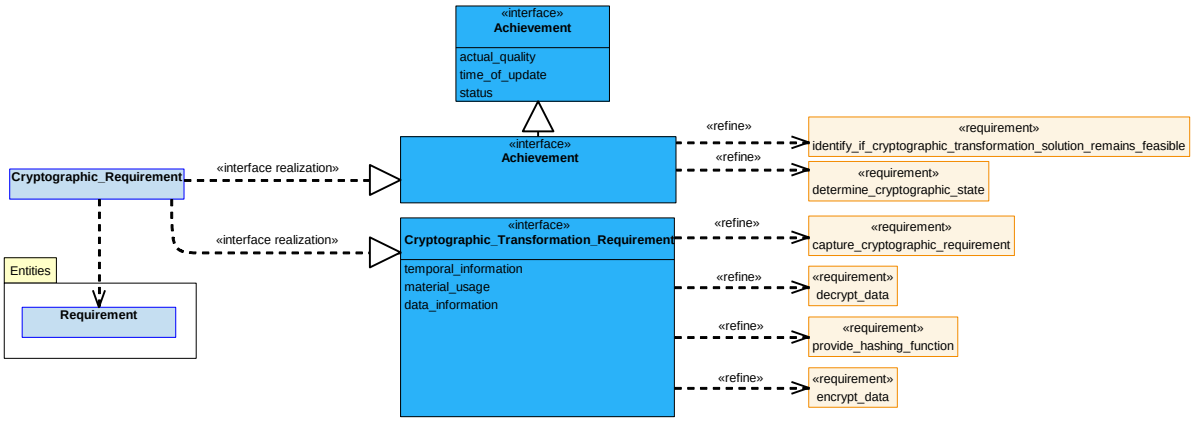

Figure 273: Cryptographic_Requirement Service Definition

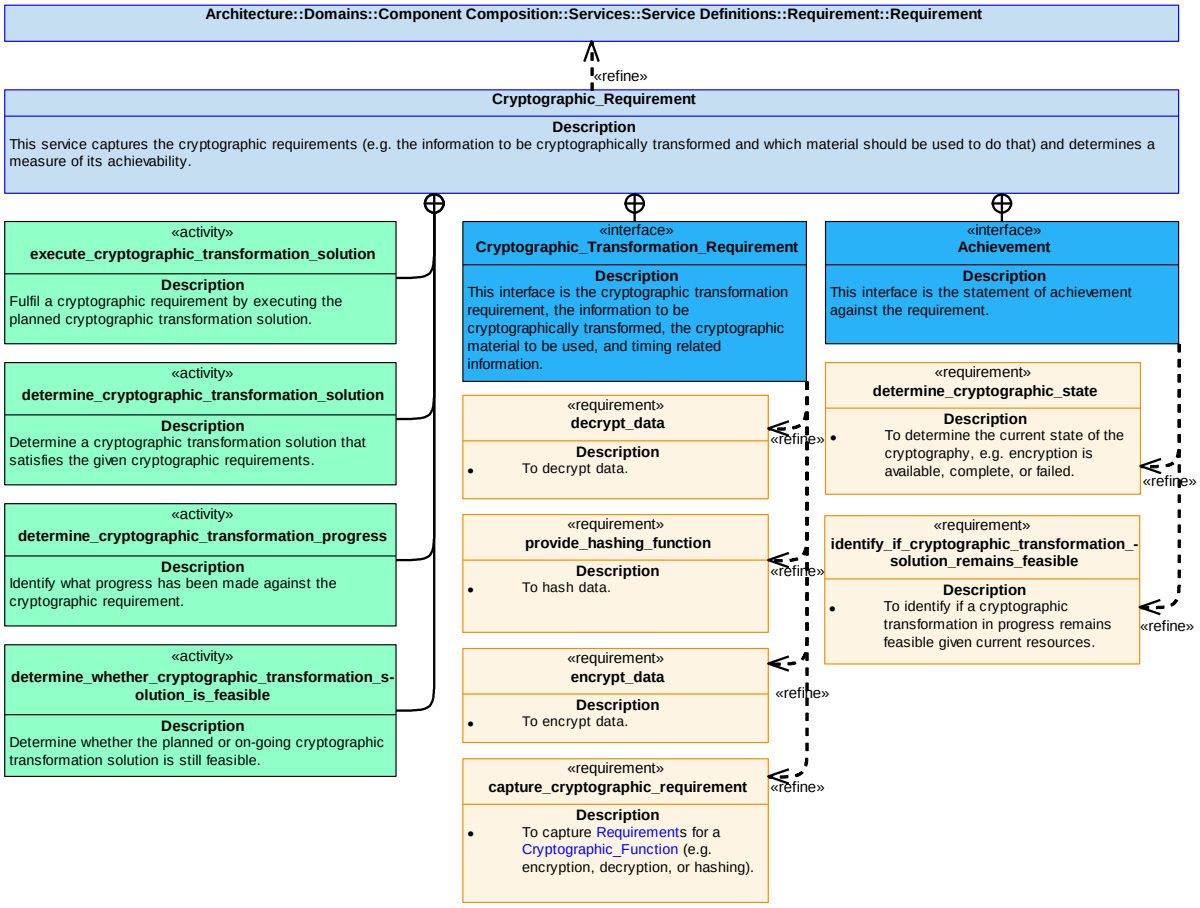

Figure 274: Cryptographic_Requirement Service Policy

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.

B.2.10.7.1.2 Cryptographic_Material_Dependency

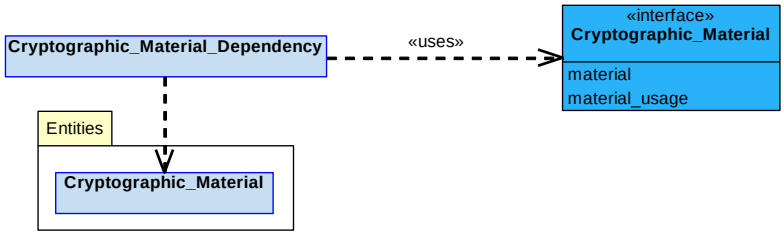

Figure 275: Cryptographic_Material_Dependency Service Definition

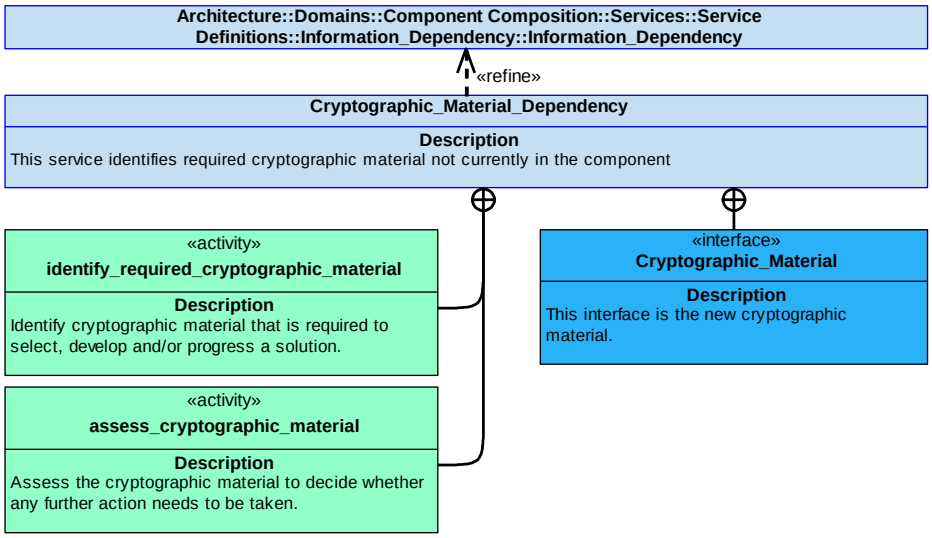

Figure 276: Cryptographic_Material_Dependency Service Policy

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.

B.2.10.7.1.3 Update_Cryptographic_Material

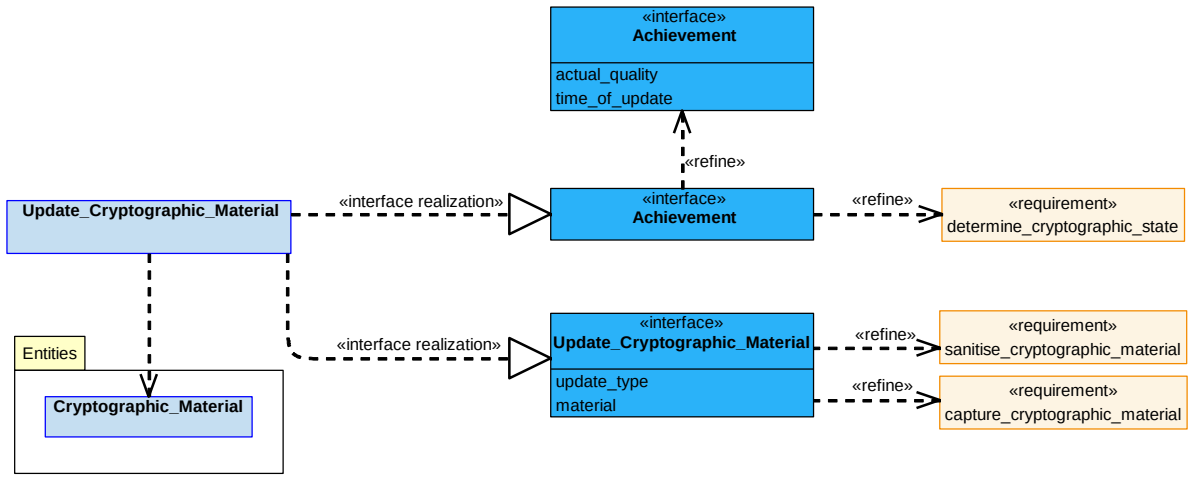

Figure 277: Update_Cryptographic_Material Service Definition

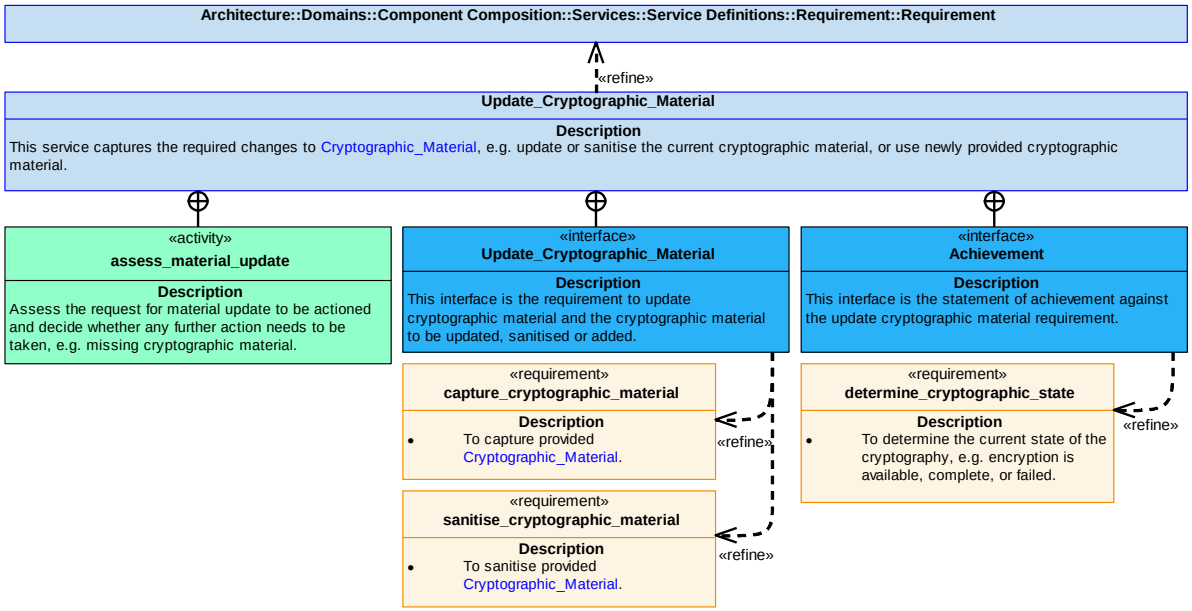

Figure 278: Update_Cryptographic_Material Service Policy

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.

B.2.10.7.1.4 Capability

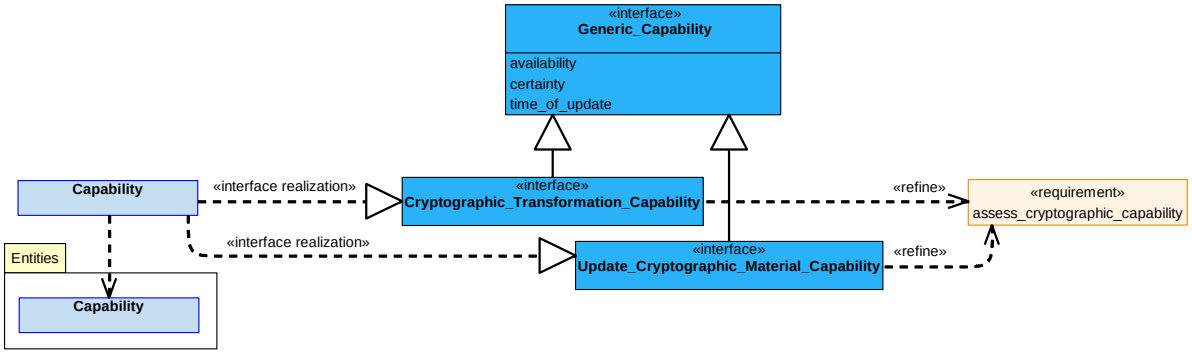

Figure 279: Capability Service Definition

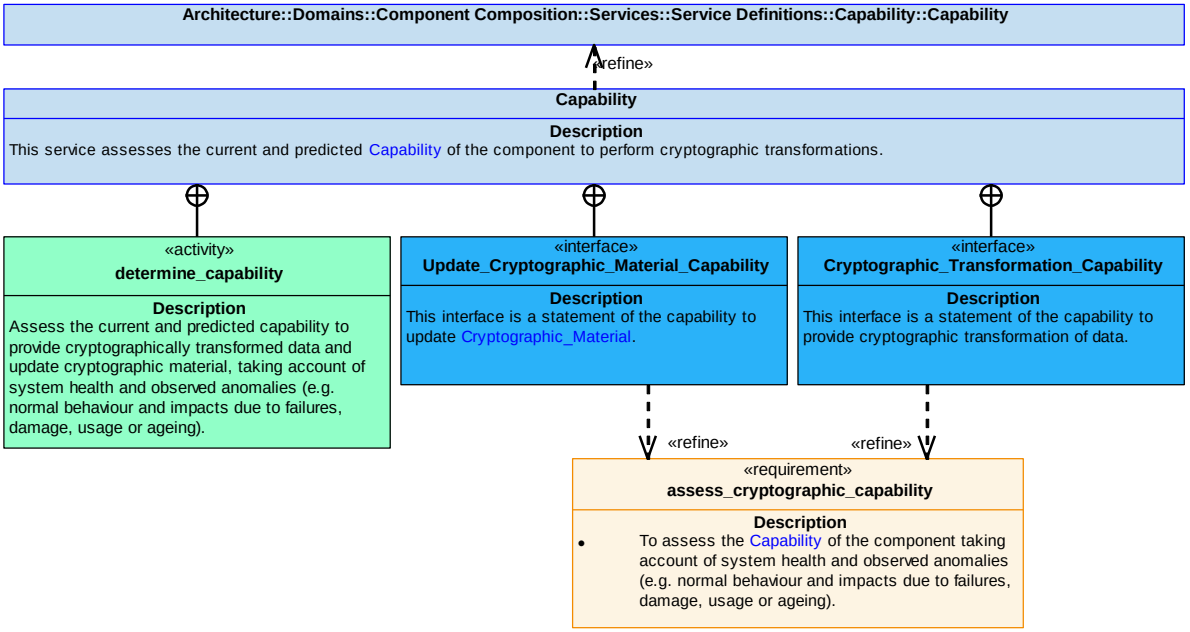

Figure 280: Capability Service Policy

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).

B.2.10.7.1.5 Capability_Evidence

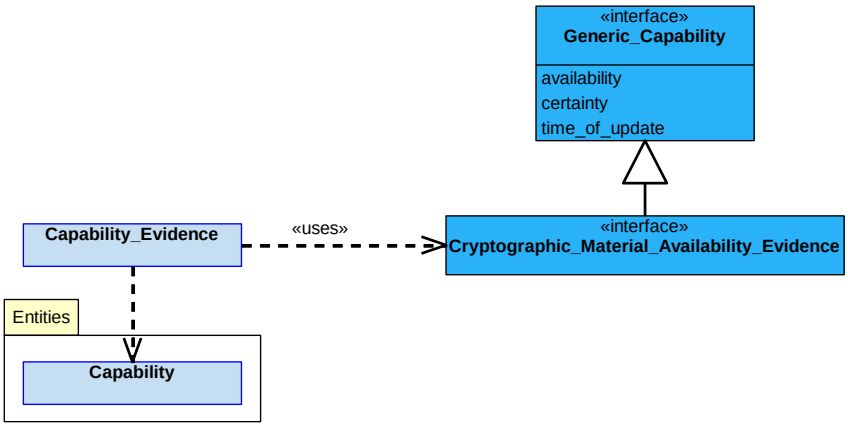

Figure 281: Capability_Evidence Service Definition

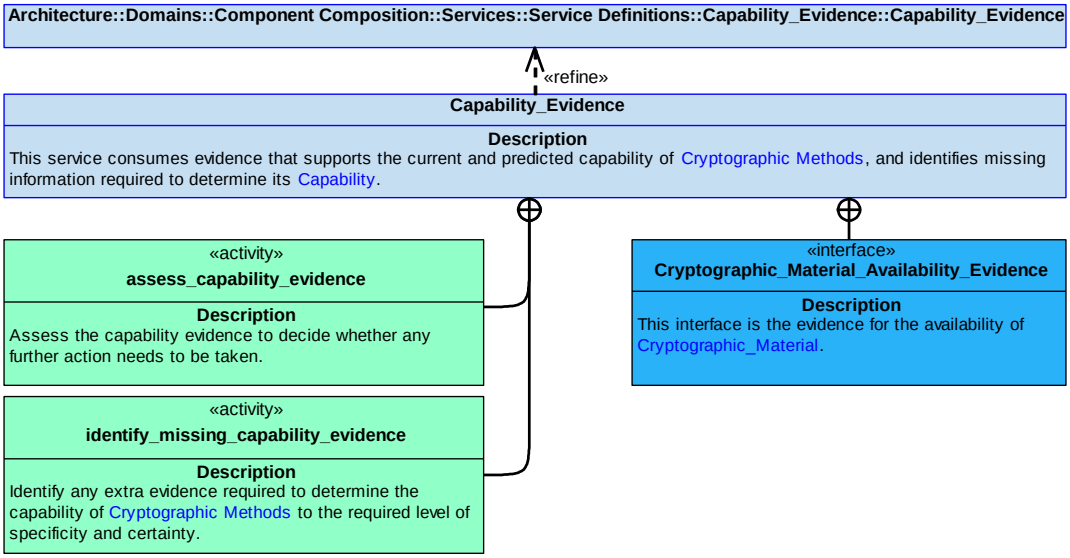

Figure 282: Capability_Evidence Service Policy

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.

Interface

Cryptographic_Material_Availability_Evidence

This interface is the evidence for the availability of 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 evidence required to determine the capability of Cryptographic Methods to the required level of specificity and certainty.

B.2.10.7.2 Service Dependencies

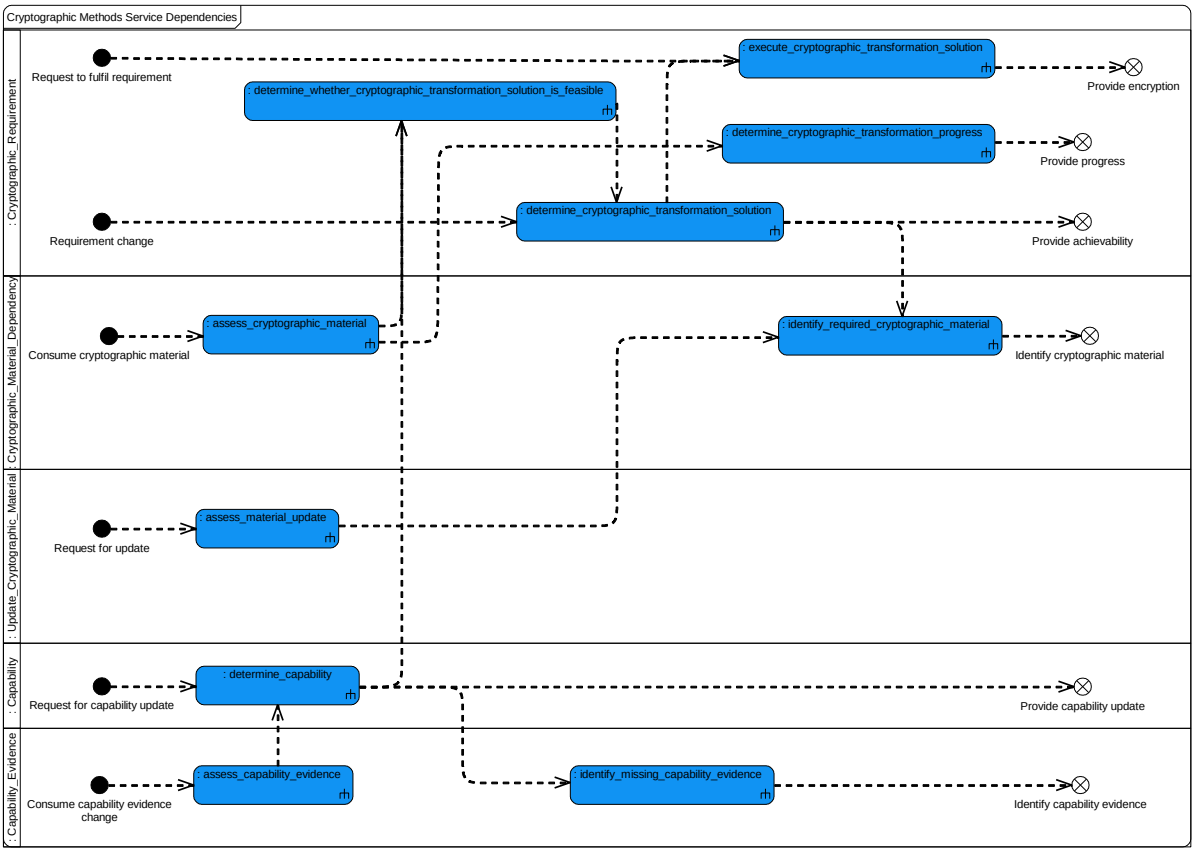

Figure 283: Cryptographic Methods Service Dependencies

B.2.11 Cyber Defence

B.2.11.1 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.

B.2.11.2 Overview

Control Architecture

Cyber Defence is a service component as defined in the Control Architecture policy.

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.

B.2.11.3 Service Summary

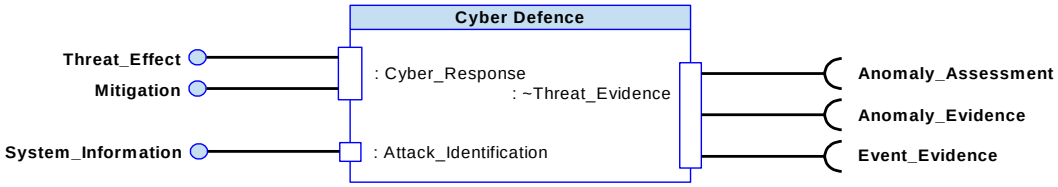

Figure 284: Cyber Defence Service Summary

B.2.11.4 Responsibilities

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_possible_actions

- To identify possible actions to counteract a suspected cyber attack.

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.

B.2.11.5 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.

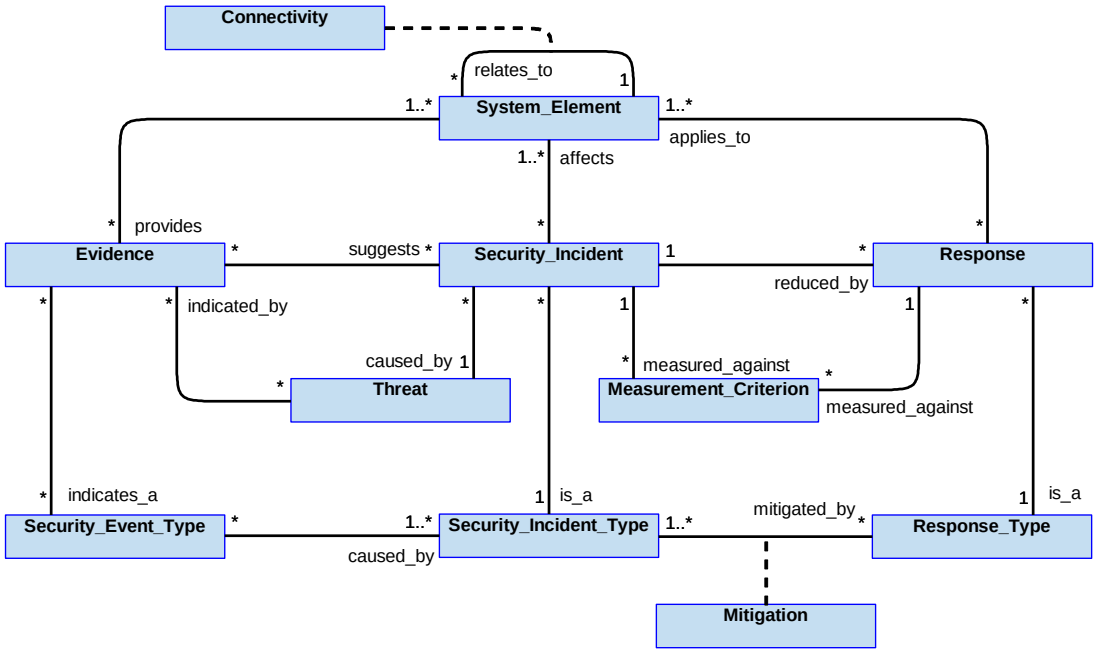

Figure 285: Cyber Defence Semantics

B.2.11.5.1 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_Incident

A specific security incident for assessment of whether the system is the subject of a cyber attack.

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.

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_Type

The kind of security incident that a system may be subject to. For example, loss of confidentiality or denial of service.

Threat

A known threat (vulnerability, attack vector, exploit, etc.) that might affect the system.

B.2.11.6 Design Rationale

B.2.11.6.1 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).

B.2.11.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Cyber Defence:

- Data Driving - This policy 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 Assessment - 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.

B.2.11.6.3 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 policy.

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

B.2.11.6.4 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, but may require to communicate across protection boundaries. 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.

The Security Guidance for PYRAMID Exploiters, Ref. [60] contains a section on cyber defence and resilience.

B.2.11.7 Services

B.2.11.7.1 Service Definitions

B.2.11.7.1.1 Cyber_Response

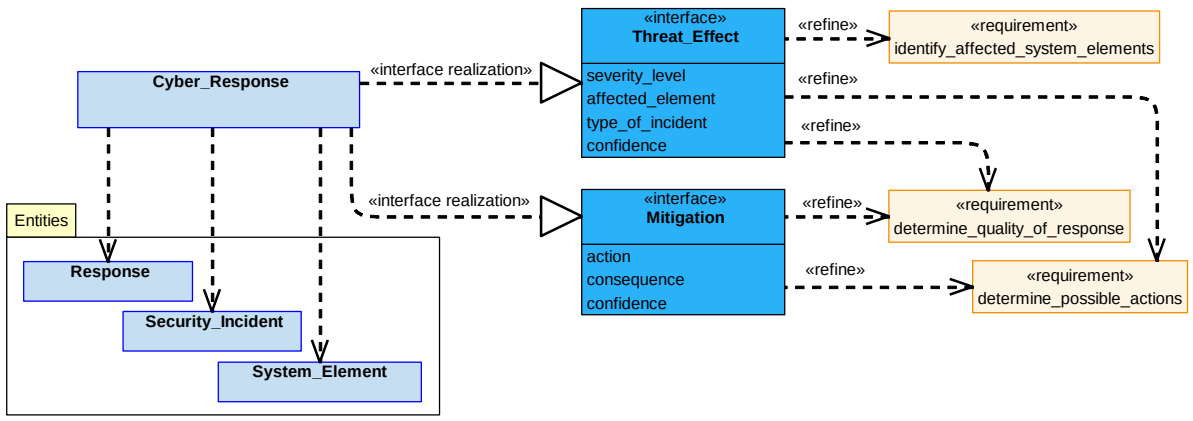

Figure 286: Cyber_Response Service Definition

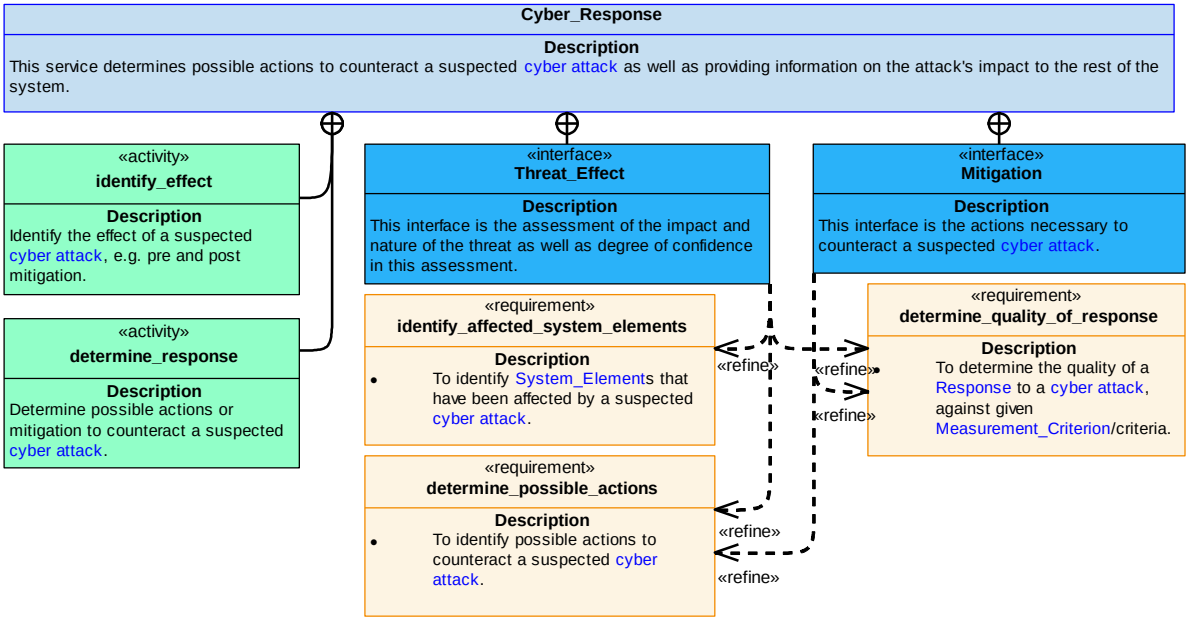

Figure 287: Cyber Response Service Policy

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.

Interfaces

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.

B.2.11.7.1.2 Attack_Identification

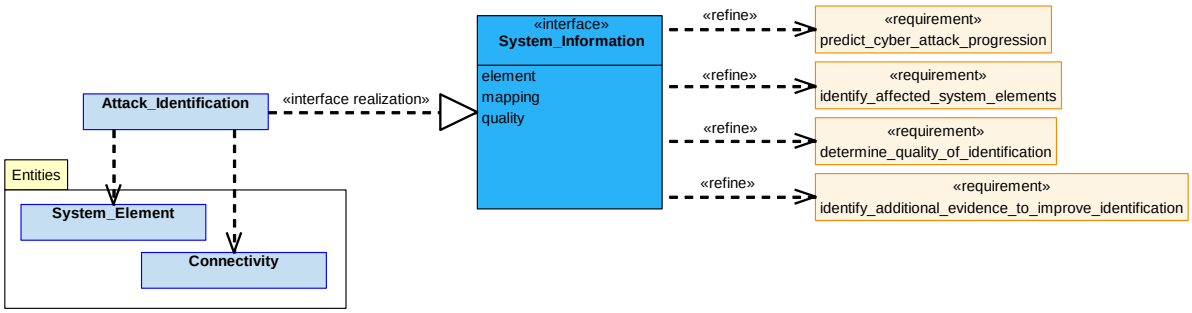

Figure 288: Attack_Identification Service Definition

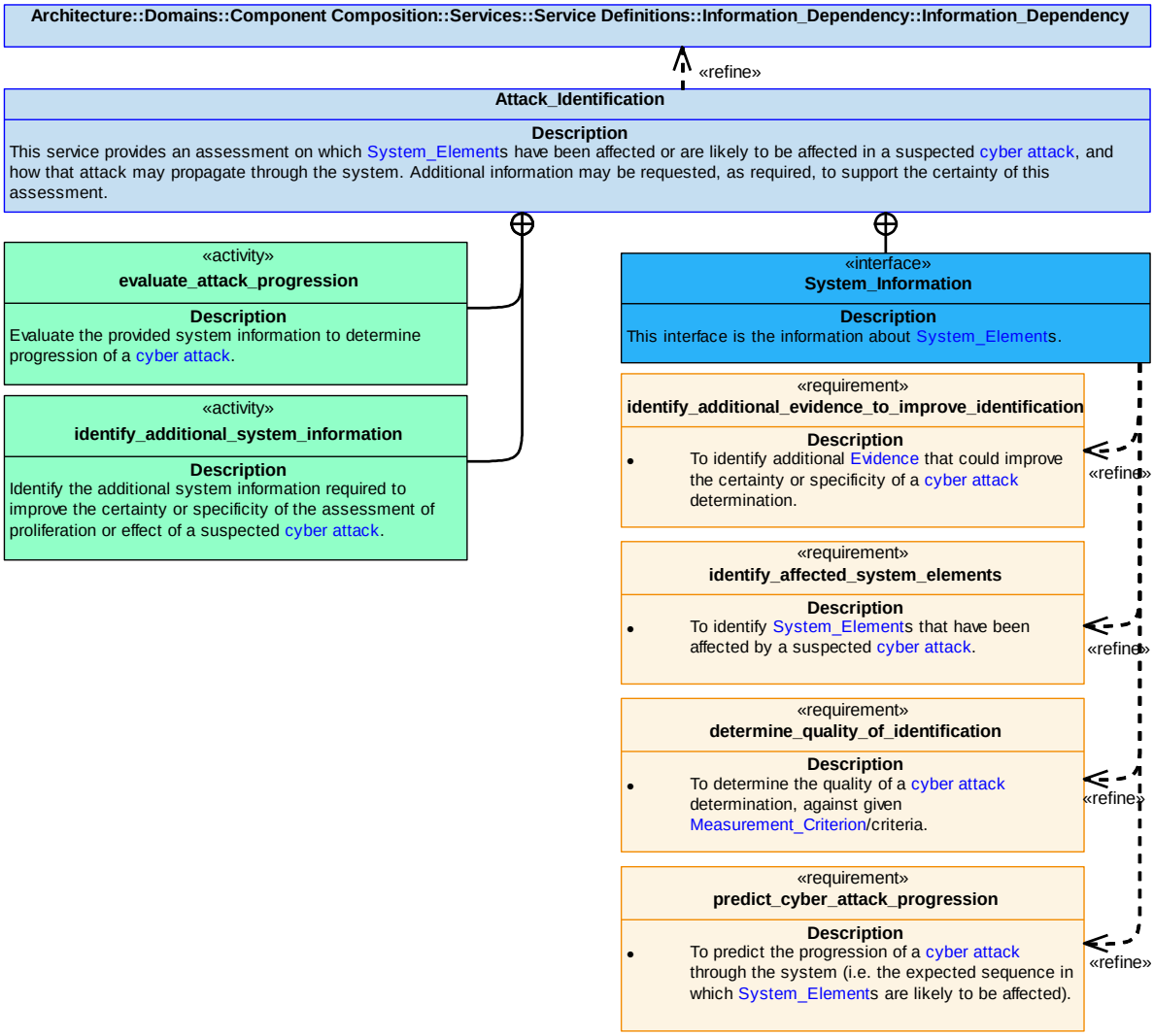

Figure 289: Attack Identification Service Policy

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.

B.2.11.7.1.3 Threat_Evidence

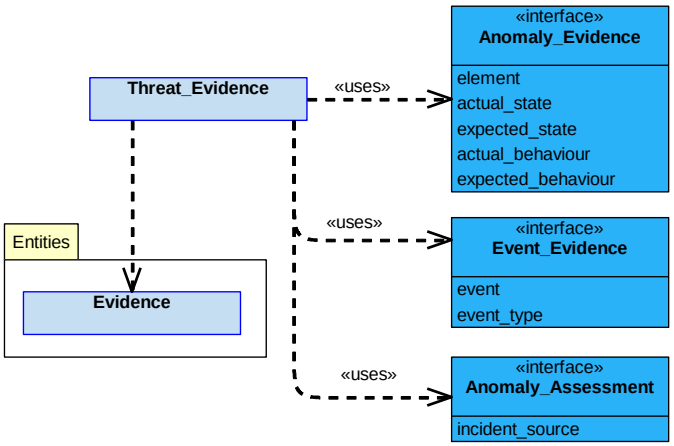

Figure 290: Threat_Evidence Service Definition

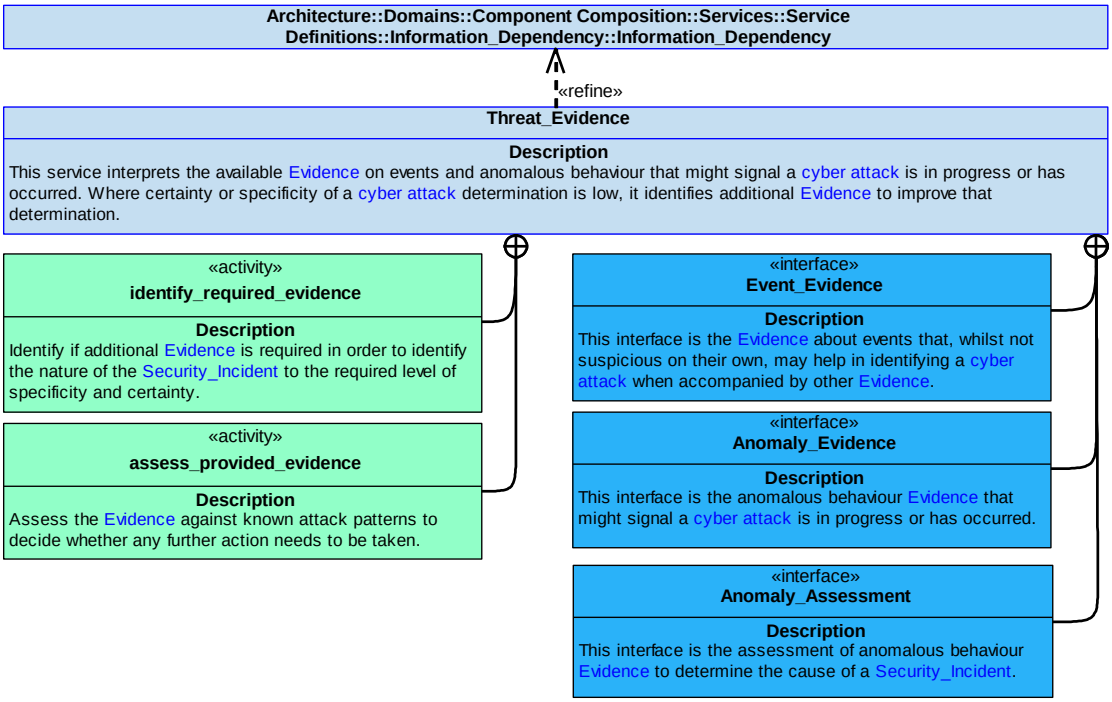

Figure 291: Threat Evidence Service Policy

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.

B.2.11.7.2 Service Dependencies

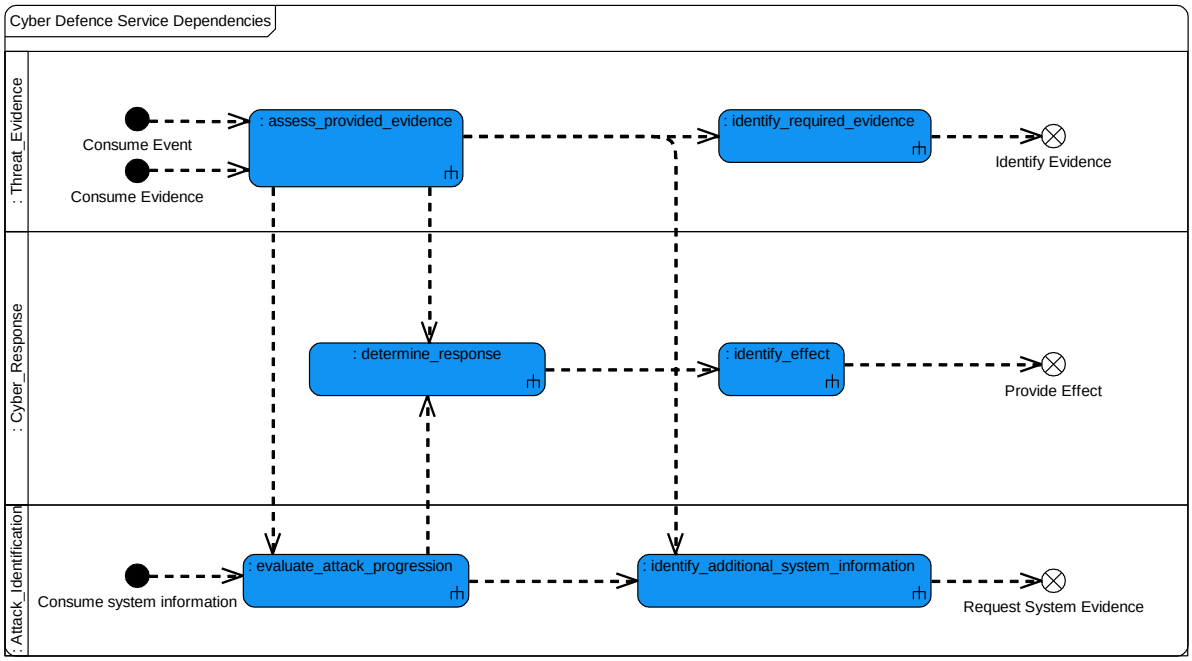

Figure 292: Cyber Defence Service Dependencies

B.2.12 Data Distribution

B.2.12.1 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.

B.2.12.2 Overview

Control Architecture

Data Distribution is a resource component as defined in the Control Architecture policy.

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. Data Distribution gathers Data_Items, formats, protects and delivers the Delivery_Item(s) in accordance with an agreed data distribution solution.

Examples of Use

Data Distribution will be used:

- When data needs to be provided in a specified format to a Participant.
- To automatically produce reports.

B.2.12.3 Service Summary

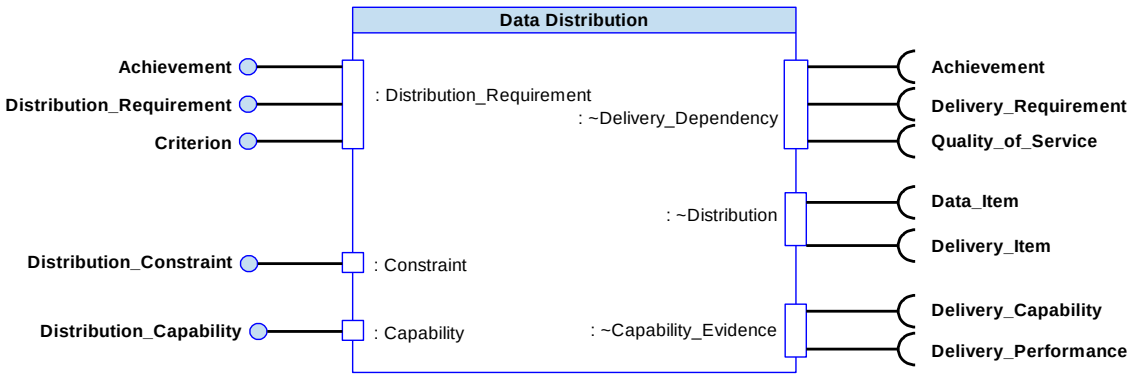

Figure 293: Data Distribution Service Summary

B.2.12.4 Responsibilities

capture_requirements_for_data_distribution

- To capture Distribution_Requirements for distribution of Delivery_Items.

capture_constraints_for_data_distribution

- To capture Distribution_Constraints for the distribution of Delivery_Items.

capture_measurement_criteria_for_data_distribution

- To capture Measurement_Criteria for distribution of a Delivery_Item.

assess_data_distribution_capability

- To assess the Distribution_Capability to distribute Delivery_Items taking account of system health and observed anomalies.

predict_capability_progression

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

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.

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).

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.

identify_missing_information

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

identify_data_distribution_solution_in_progress_remains_feasible

- To identify whether a data distribution solution currently in progress remains feasible.

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.

B.2.12.5 Subject Matter Semantics

The subject matter of Data Distribution is the distribution, and reception of data, including collation and formatting for delivery and decomposition on receipt.

Exclusions

The subject matter of Data Distribution does not include:

- Formatting of data items between different systems.
- Manipulating the format of data items.

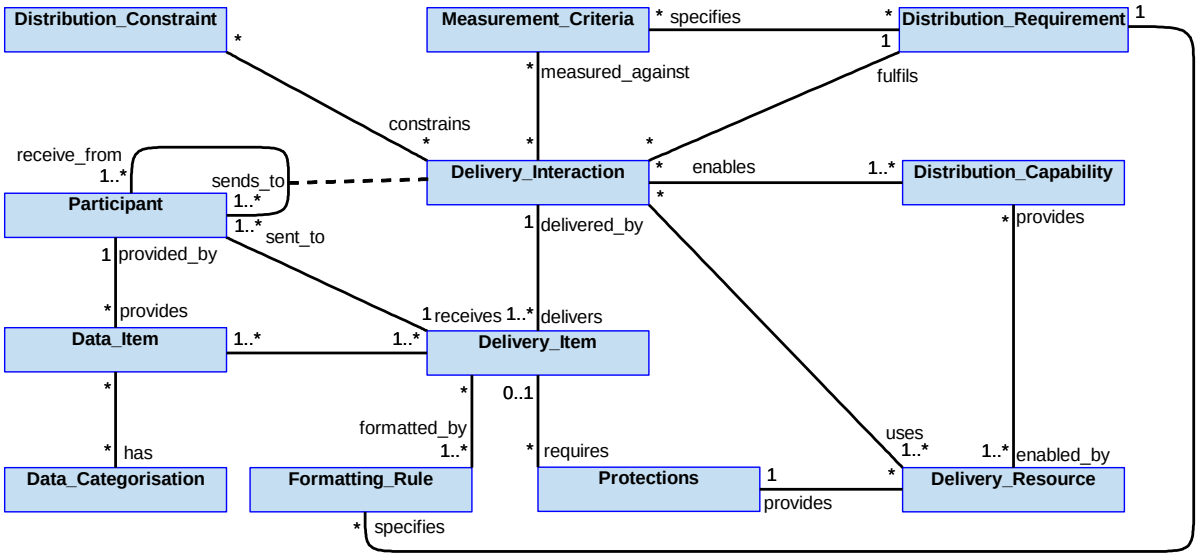

Figure 294: Data Distribution Semantics

B.2.12.5.1 Entities

Delivery_Interaction

An interaction to between Participants to deliver data.

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_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_Requirement

The requirement defining the needs to deliver data.

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.

Data_Item

A specific item of data to be distributed, e.g. a sensor measurement or a TDL message.

Formatting_Rule

The rules for configuring the required Data_Item(s) into a specific format for delivery, e.g. the construction rules for a mission report, the structure of specific IP packet.

Measurement_Criteria

A measurable parameter that can be used to evaluate performance, quality and progress, e.g. quality of service.

Protections

The methods applied in order to protect a data exchange, e.g. the use of secure Delivery_Resources, or encryption of Delivery_Items.

Delivery_Resource

The resources available to the component to perform a data exchange, e.g. communication and network resources, or cryptographic devices.

Data_Categorisation

Information about a Data_Item, e.g. the classification, priority, or whether it is safety critical.

Delivery_Item

The item of collated/decomposed and formatted Data_Items to be delivered to the Participant(s) including any required metadata or overhead, e.g. mission report, release of mission data, Internet Protocol packet.

B.2.12.6 Design Rationale

B.2.12.6.1 Assumptions

- This component is likely to interface with hardware responsible for data distribution activities.
- This component will support extra-nodal and external to the system data distribution.
- 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.
- 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.

B.2.12.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy 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 the different types of the types of communications. 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, STANAG 4586 common messaging for use with STANAG 4817 systems, or DDS for communications with land systems.

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).

B.2.12.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component may result in the inability to 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) 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.
- 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. Therefore, corruption of the data transfer would result in loss of "useable" data.

Therefore, the indicative DAL is C.

B.2.12.6.4 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.

The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about security of data exchange.

B.2.12.7 Services

B.2.12.7.1 Service Definitions

B.2.12.7.1.1 Distribution Requirement

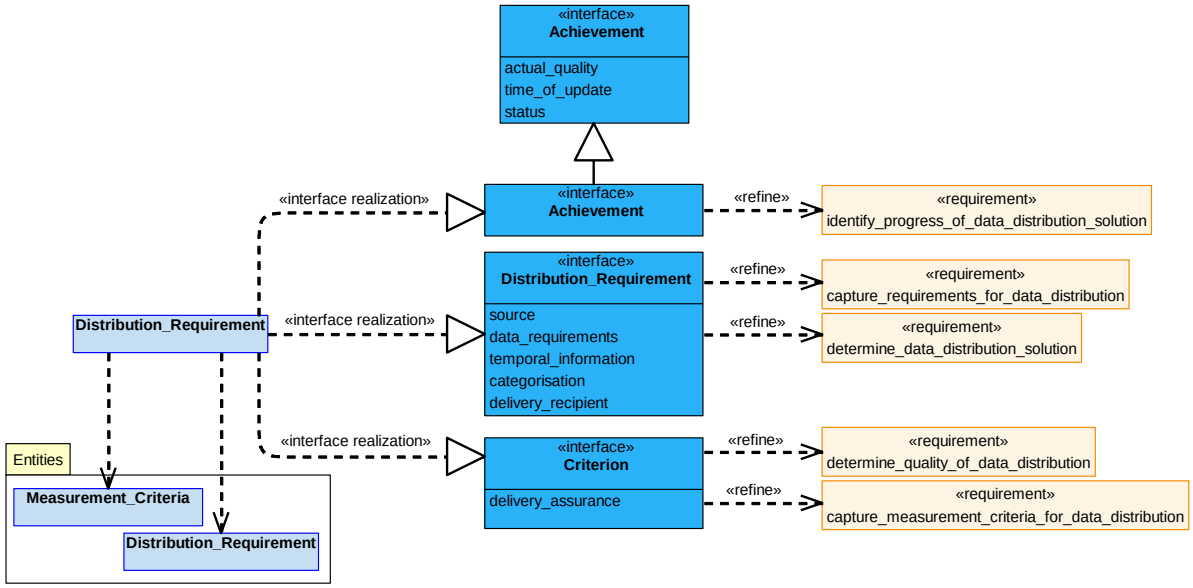

Figure 295: Distribution_Requirement Service Definition

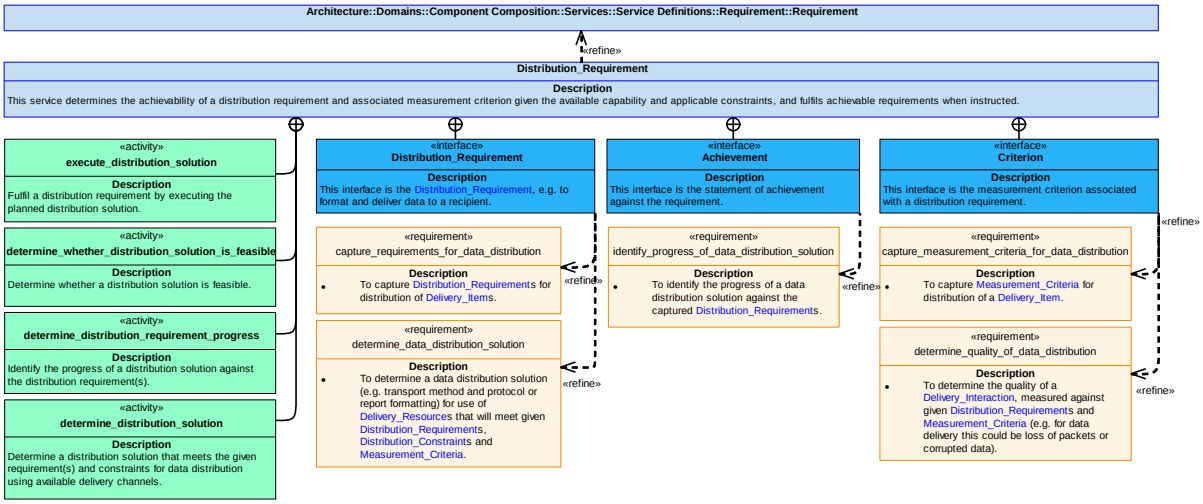

Figure 296: Distribution_Requirement Service Policy

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).

B.2.12.7.1.2 Delivery Dependency

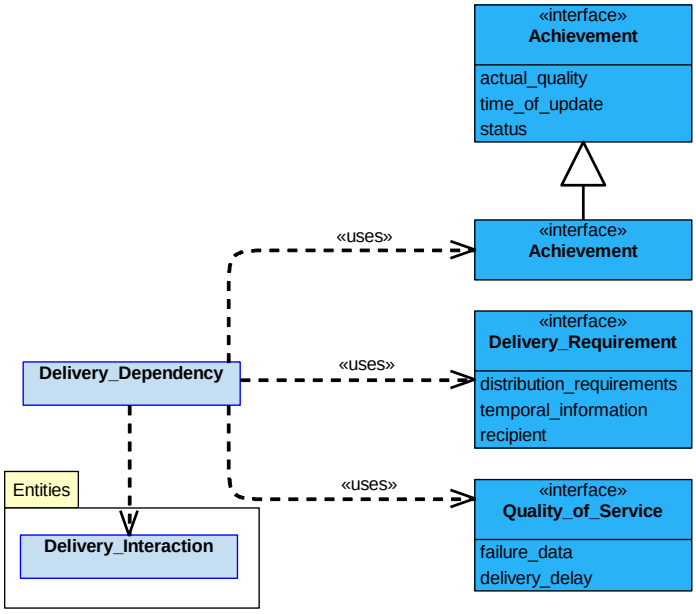

Figure 297: Delivery_Dependency Service Definition

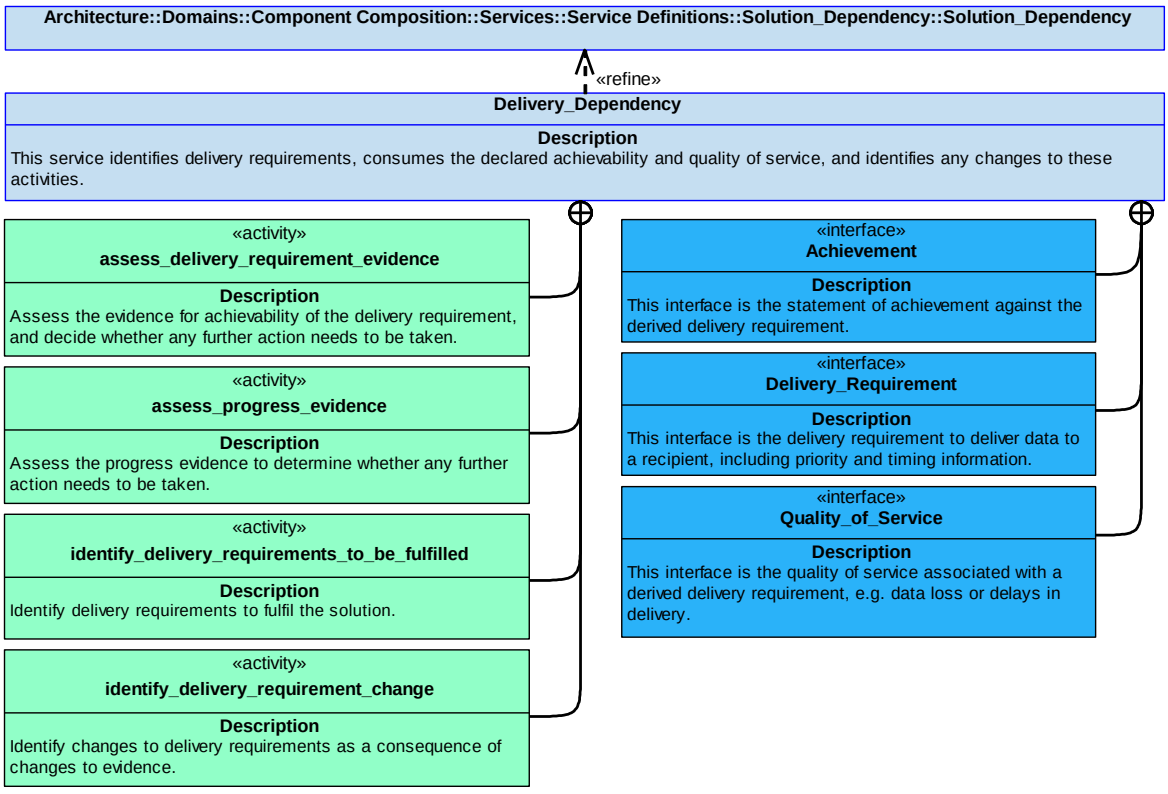

Figure 298: Delivery_Dependency Service Policy

Delivery_Dependency

This service identifies delivery requirements, 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.

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.

B.2.12.7.1.3 Distribution

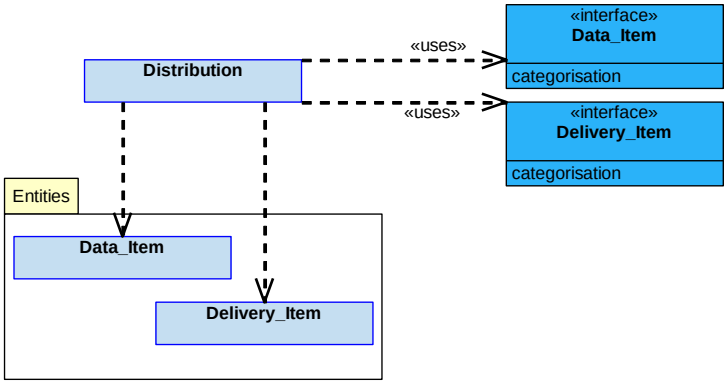

Figure 299: Distribution Service Definition

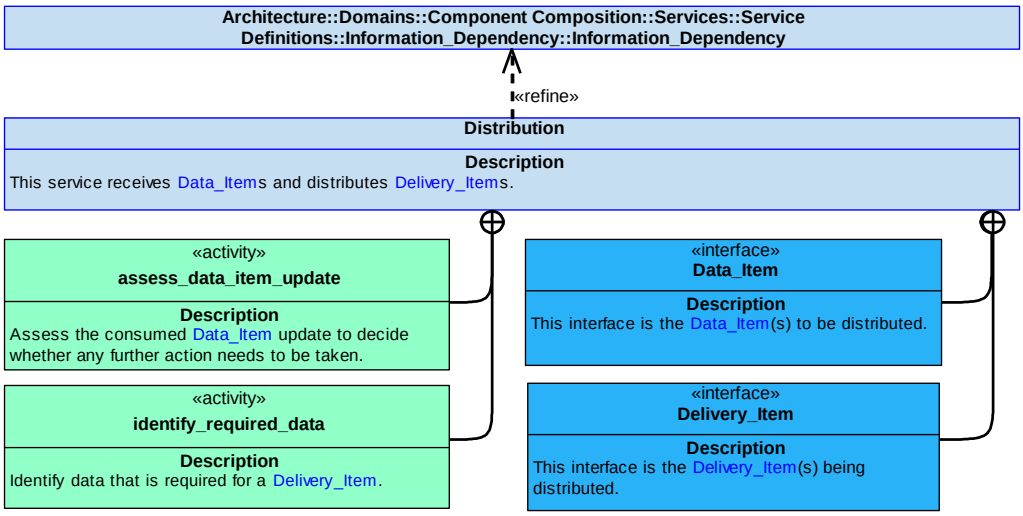

Figure 300: Distribution Service Policy

Distribution

This service receives Data_Items and distributes Delivery_Items.

Interfaces

Data_Item

This interface is the Data_Item(s) to be distributed.

Attribute

categorisation Information about the Data_Item, e.g. the classification, priority, or criticality.

Delivery_Item

This interface is the Delivery_Item(s) being distributed.

Attribute

categorisation Information about the Delivery_Item, e.g. the classification, priority, or criticality.

Activities

assess_data_item_update

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

identify_required_data

Identify data that is required for a Delivery_Item.

B.2.12.7.1.4 Constraint

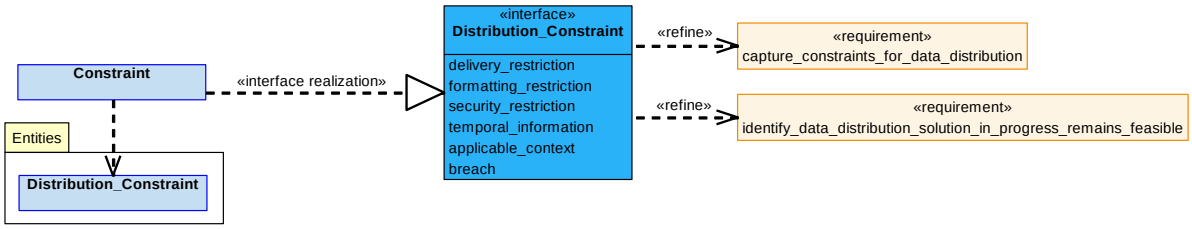

Figure 301: Constraint Service Definition

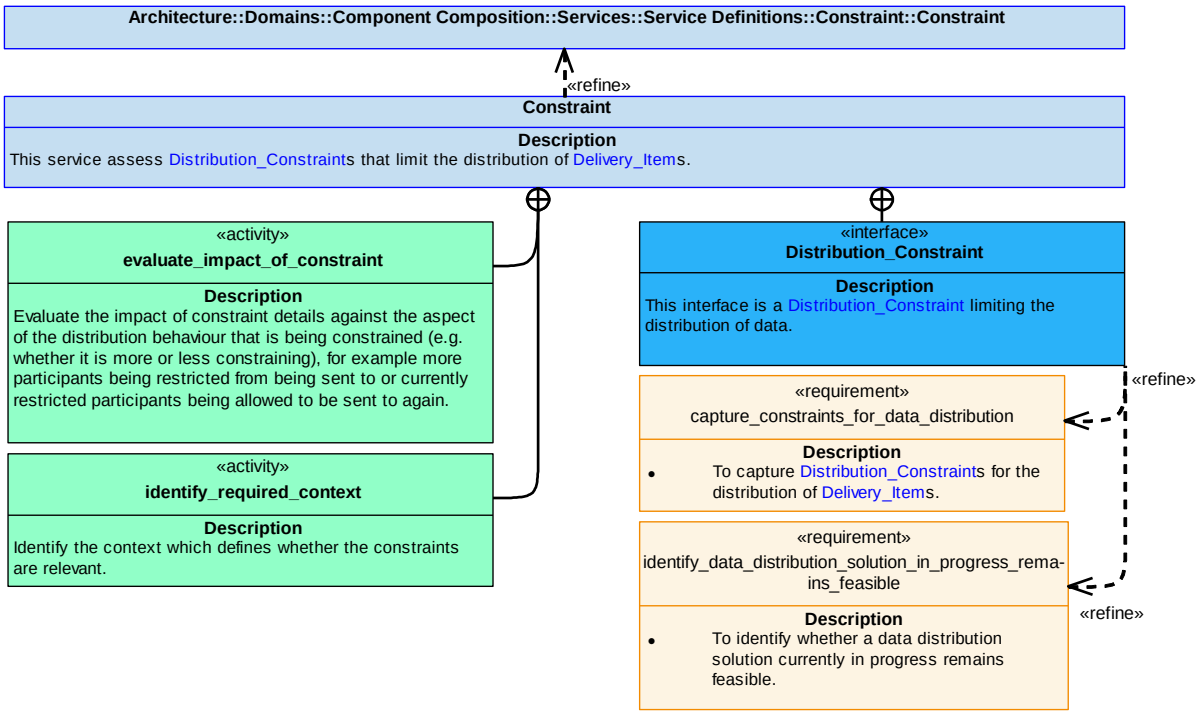

Figure 302: Constraint Service Policy

Constraint

This service assess Distribution_Constraints that limit the distribution of Delivery_Items.

Interface

Distribution_Constraint

This interface is a Distribution_Constraint limiting the distribution of data.

Attributes

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.

B.2.12.7.1.5 Capability

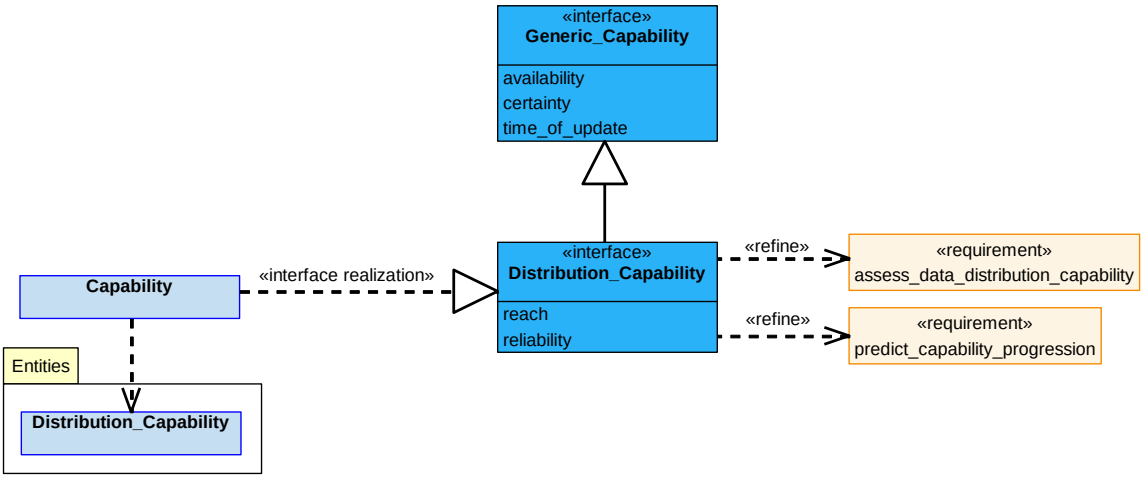

Figure 303: Capability Service Definition

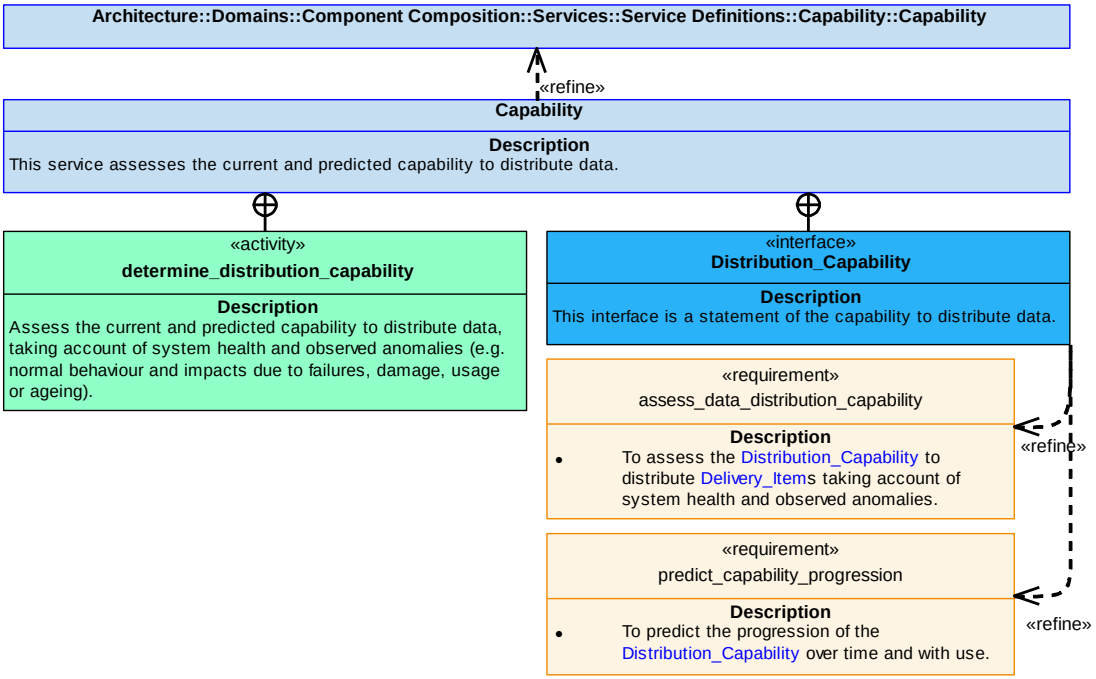

Figure 304: Capability Service Policy

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).

B.2.12.7.1.6 Capability Evidence

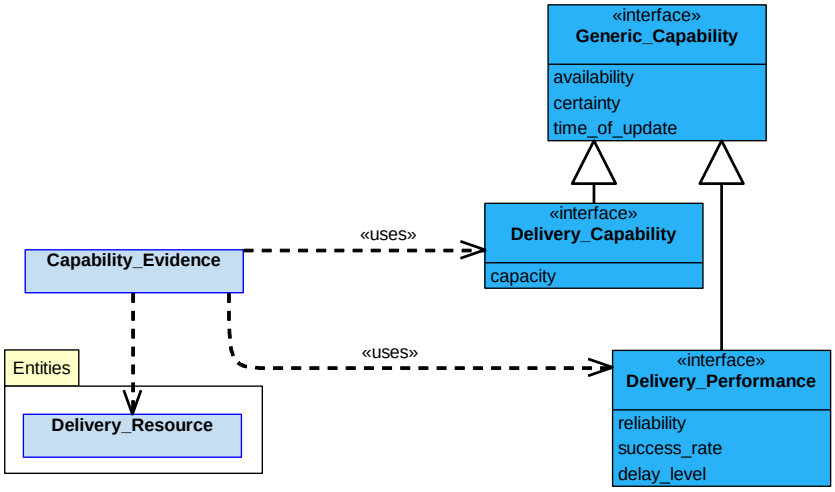

Figure 305: Capability_Evidence Service Definition

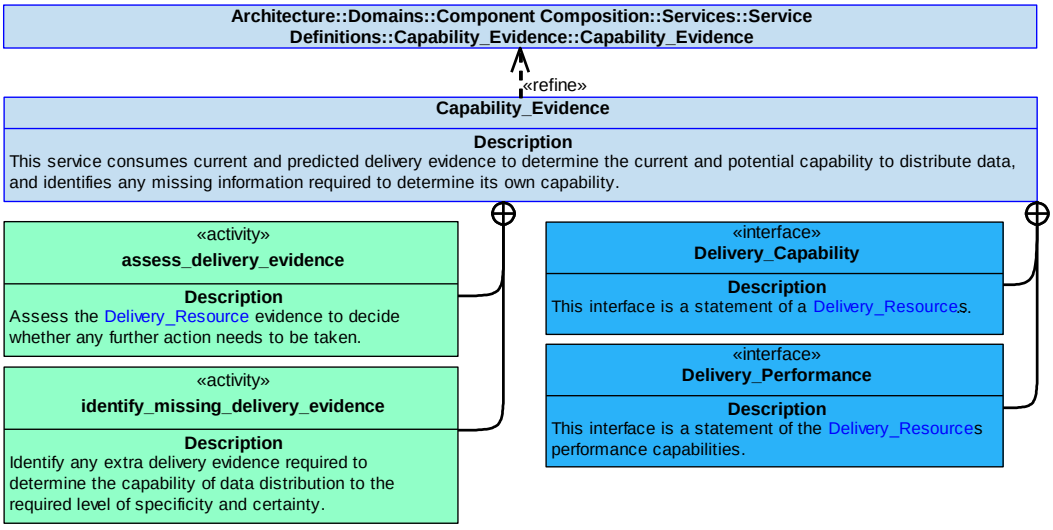

Figure 306: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted delivery evidence to determine the current and potential capability to 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.

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.

B.2.12.7.2 Service Dependencies

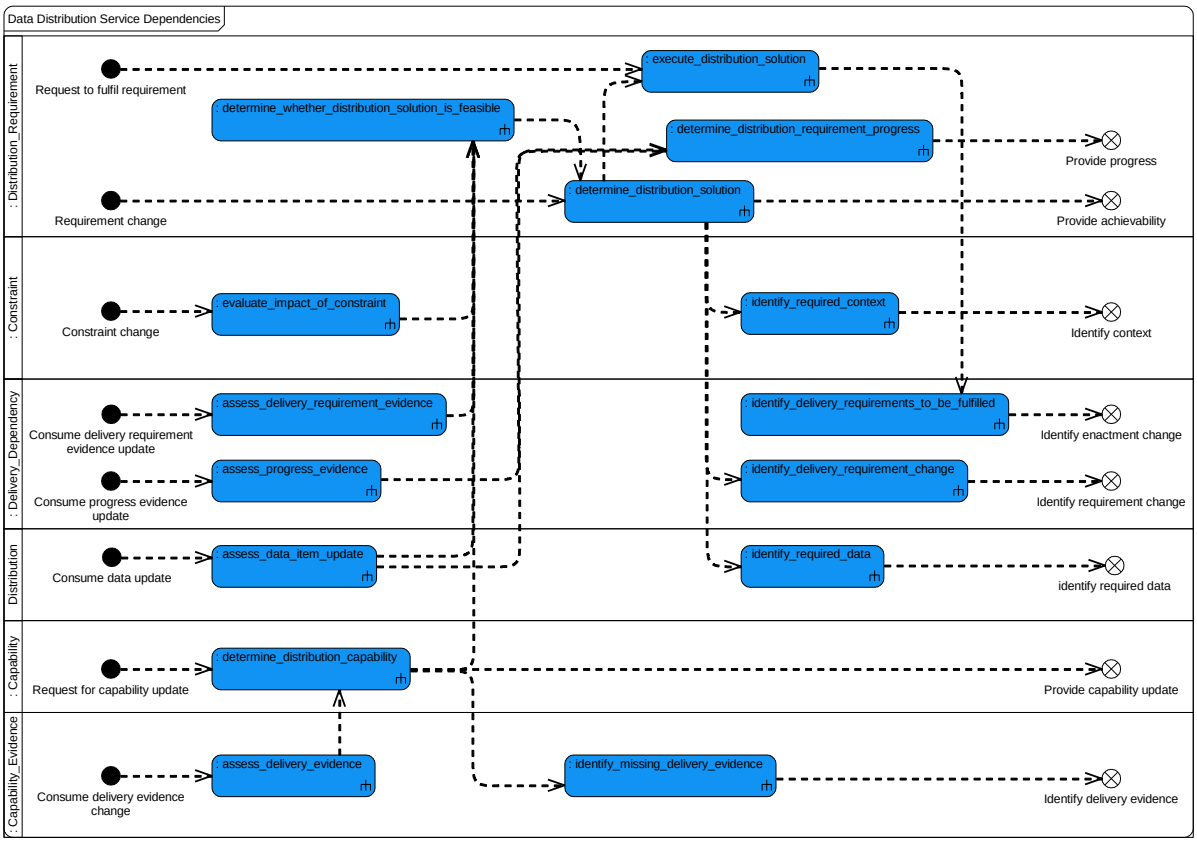

Figure 307: Data Distribution Service Dependencies

B.2.13 Data Fusion

B.2.13.1 Role

The role of Data Fusion is to evaluate and combine evidence of potential objects to provide an understanding of real-world entities.

B.2.13.2 Overview

Control Architecture

Data Fusion is a service component as defined in the Control Architecture policy.

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.

B.2.13.3 Service Summary

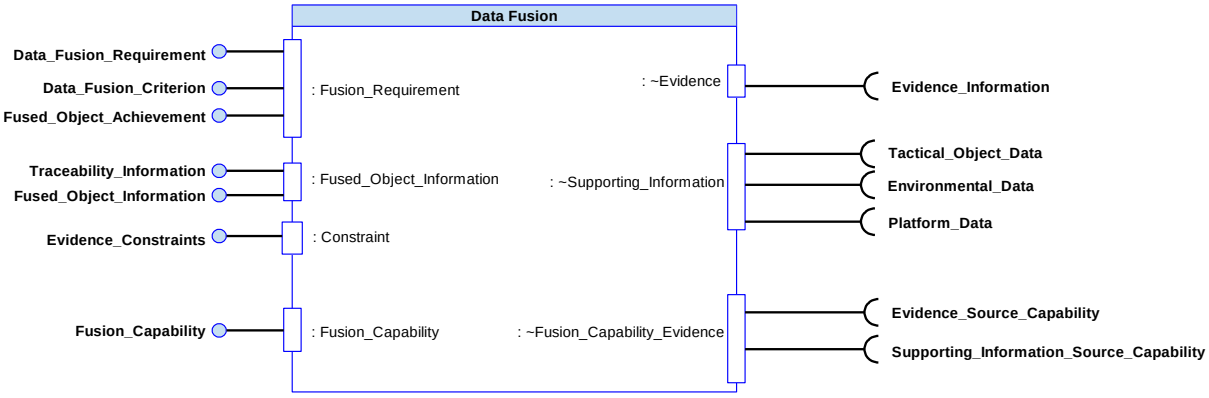

Figure 308: Data Fusion Service Summary

B.2.13.4 Responsibilities

capture_fusion_requirements

- To capture Data_Fusion_Requirements or interpreting Evidence and generating Fused_Objects.

capture_evidence

- To capture provided Evidence along with its lineage.

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_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.

determine_quality_of_fused_objects

- To determine the quality of the Fused_Objects, measured against given Data_Fusion_Requirements and 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)

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.

capture_supporting_information

- To capture provided Supporting_Information (e.g. platform data, weather condition or terrain data).

determine_if_fusion_requirement_is_achievable

- To determine if a Data_Fusion_Requirement is achievable, given current Fusion_Constraints and resources.

B.2.13.5 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.

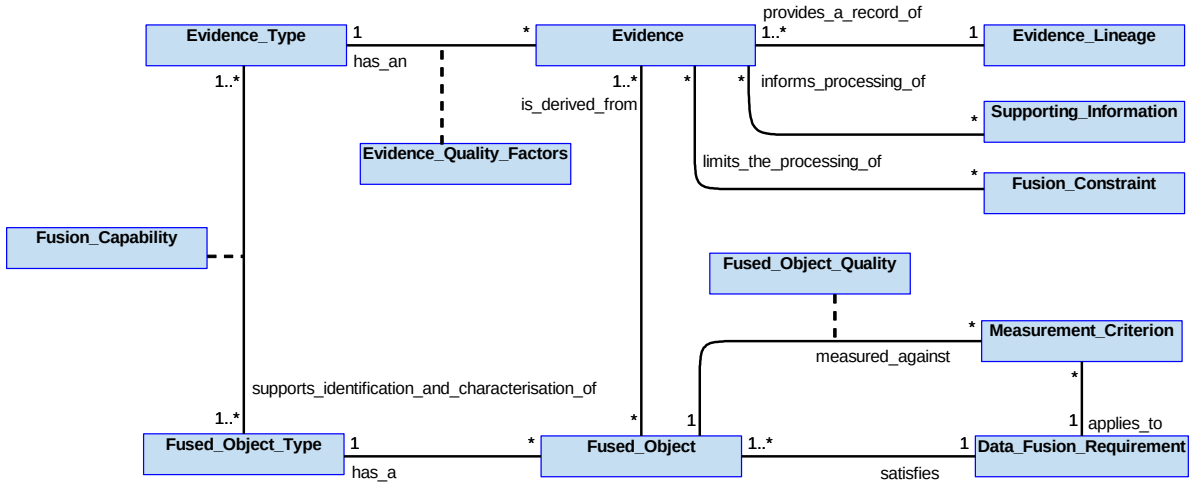

Figure 309: Data Fusion Semantics

B.2.13.5.1 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.

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).

Measurement_Criterion

A measure against which achievement of the fusion requirement can be assessed (e.g. timeliness, reliability, processing completeness or object identification).

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).

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).

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.

Fused_Object_Quality

A measure of the effectiveness or adequacy of the characterisation of a real-world entity (e.g. confidence or accuracy).

B.2.13.6 Design Rationale

B.2.13.6.1 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.

B.2.13.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy is applicable because Data Fusion is classified within the policy 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 policy) to cater for different algorithms.

B.2.13.6.3 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. In accordance with UK MoD direction (see Safety Analysis policy) this drives a DAL B indicative IDAL.

B.2.13.6.4 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.

B.2.13.7 Services

B.2.13.7.1 Service Definitions

B.2.13.7.1.1 Fusion_Requirement

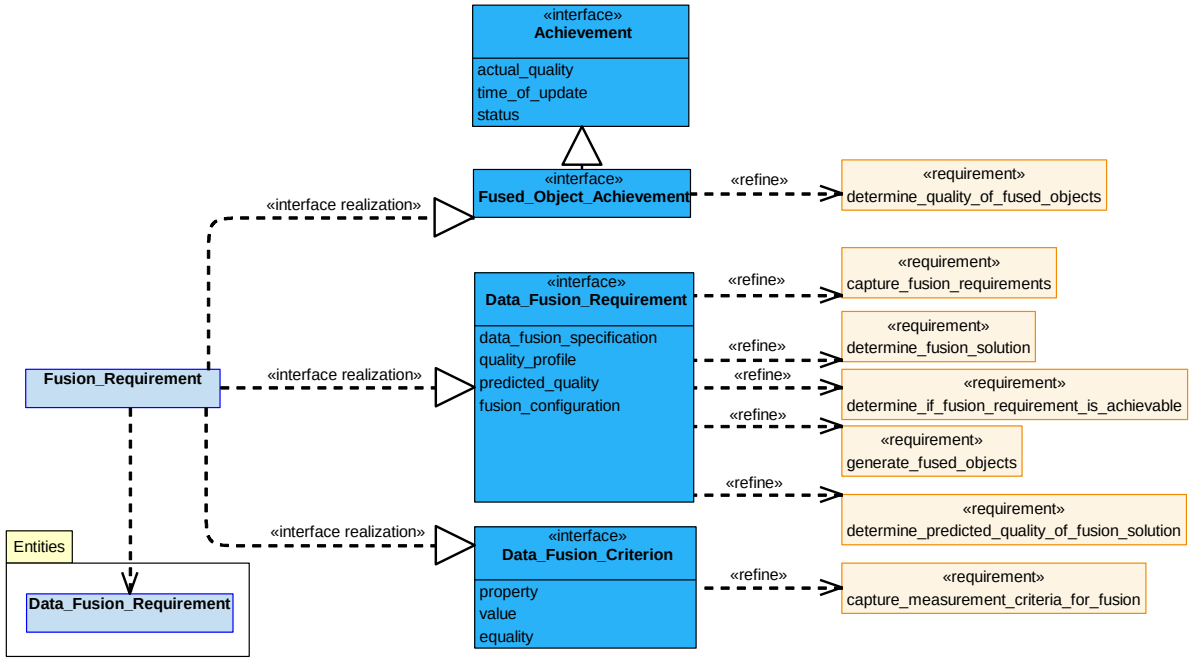

Figure 310: Fusion_Requirement Service Definition

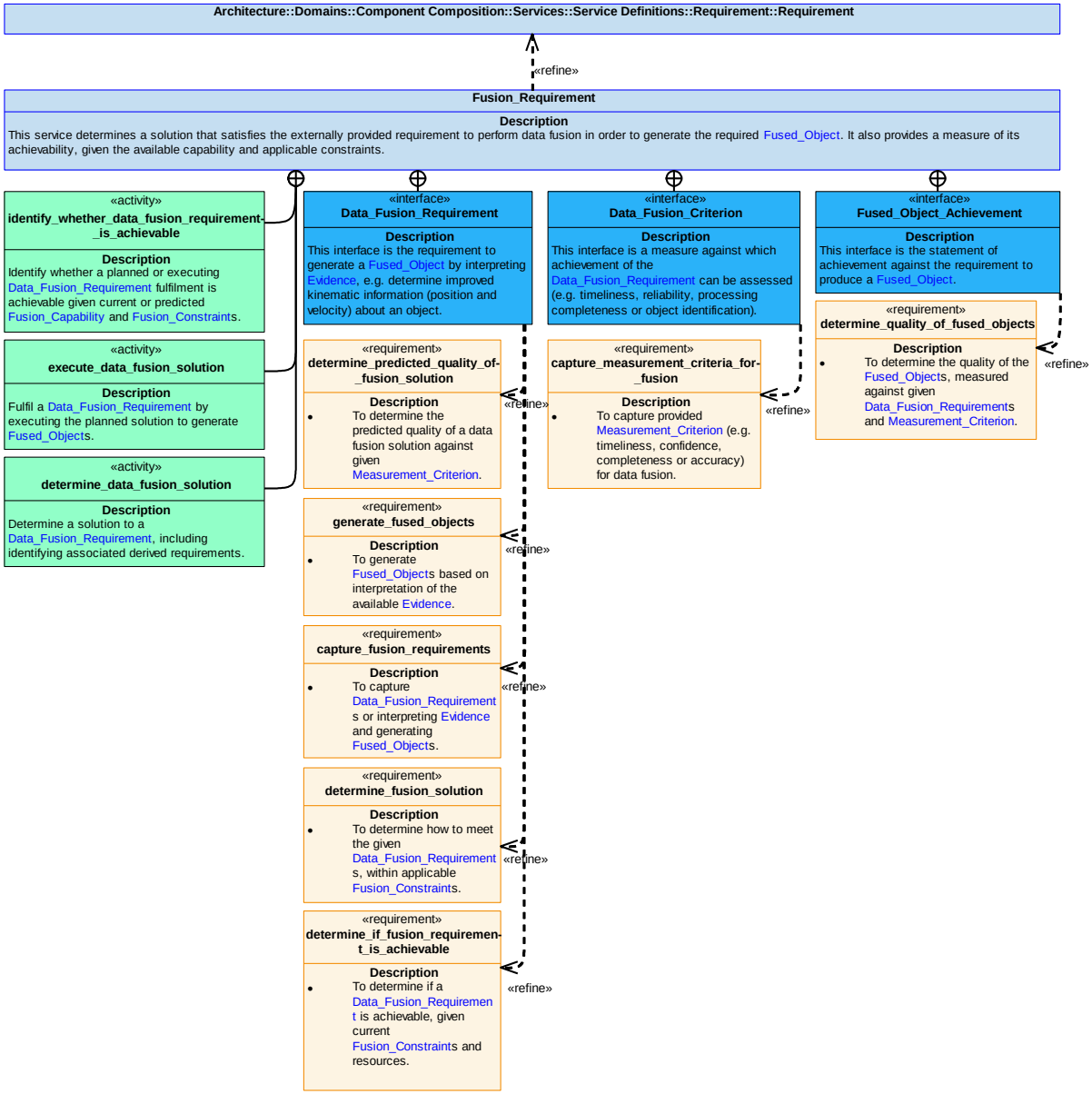

Figure 311: Fusion_Requirement Service Policy

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.

B.2.13.7.1.2 Fused_Object_Information

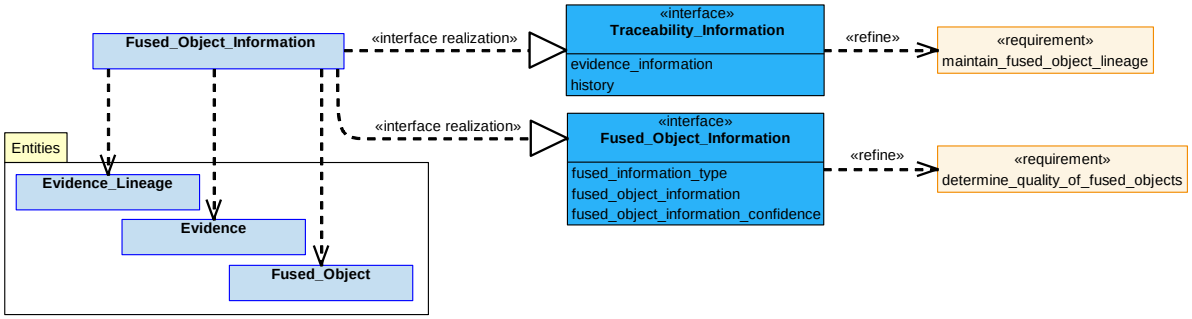

Figure 312: Fused_Object_Information Service Definition

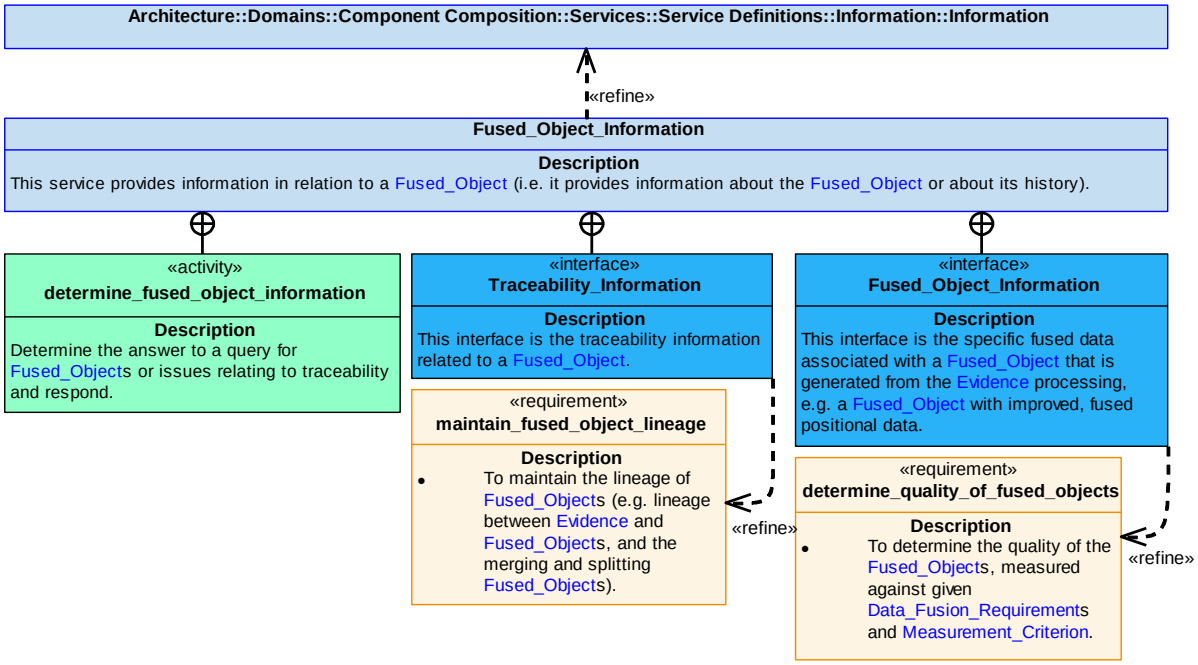

Figure 313: Fused_Object_Information Service Policy

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.

B.2.13.7.1.3 Evidence

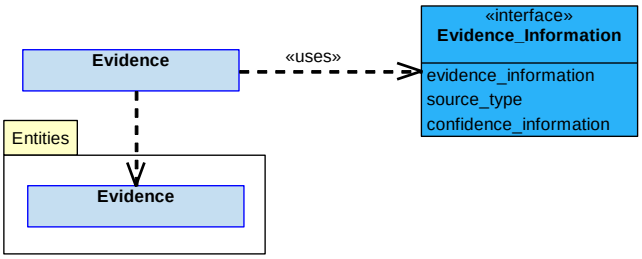

Figure 314: Evidence Service Definition

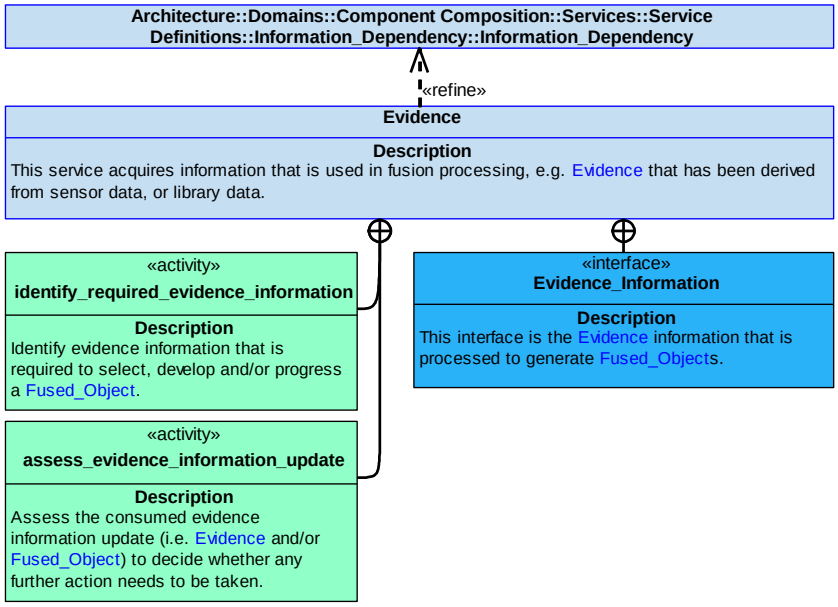

Figure 315: Evidence Service Policy

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.

B.2.13.7.1.4 Supporting_Information

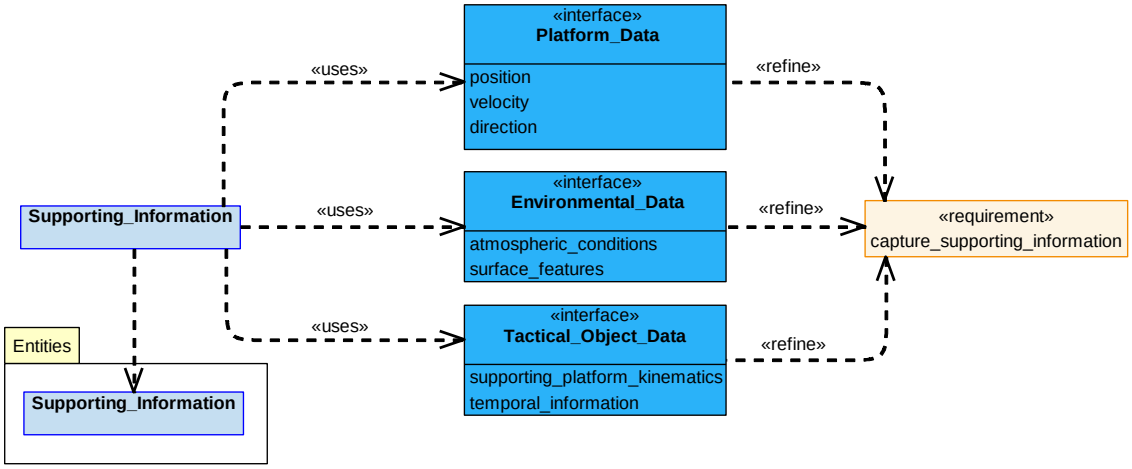

Figure 316: Supporting_Information Service Definition

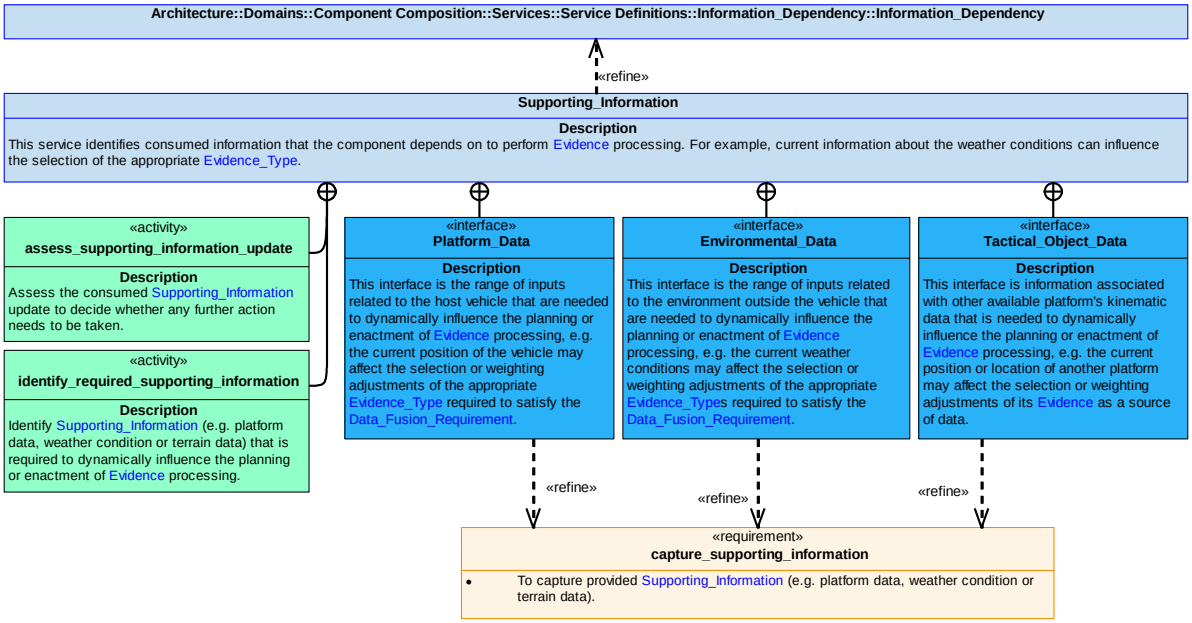

Figure 317: Supporting_Information Service Policy

Supporting_Information

This service identifies consumed information that the component depends on to perform Evidence processing. For example, current information about the weather conditions can influence the selection of the appropriate Evidence_Type.

Interfaces

Platform_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

position	Current platform location and orientation.
velocity	Velocity of the platform.
direction	Direction of travel of the platform.

Environmental_Data

This interface is the range of inputs related to the environment outside the vehicle 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.

Tactical_Object_Data

This interface is information associated with other available platform's kinematic data that is needed to dynamically influence the planning or enactment of Evidence processing, e.g. the current position or location of another platform may affect the selection or weighting adjustments of its Evidence as a source of data.

Attributes

supporting_platform_kinematics	Information relating to the motion of another object (e.g. heading, speed or acceleration).
temporal_information	Timing of supporting platform availability for Evidence updates and low latency synchronization of Evidence.

Activities

identify_required_supporting_information

Identify Supporting_Information (e.g. platform data, weather condition or terrain data) that is required to dynamically influence the planning or enactment of Evidence processing.

assess_supporting_information_update

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

B.2.13.7.1.5 Constraint

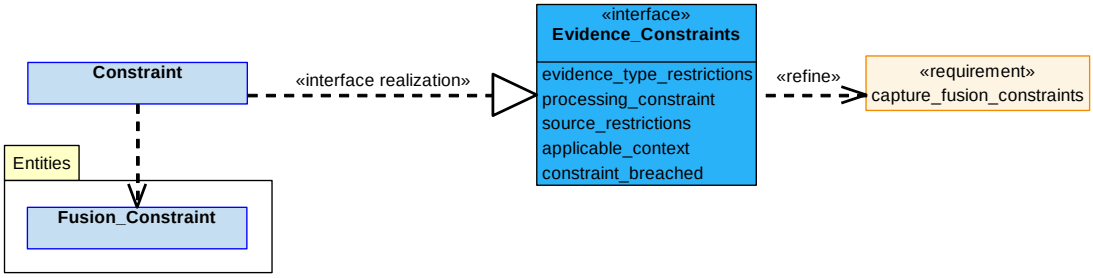

Figure 318: Constraint Service Definition

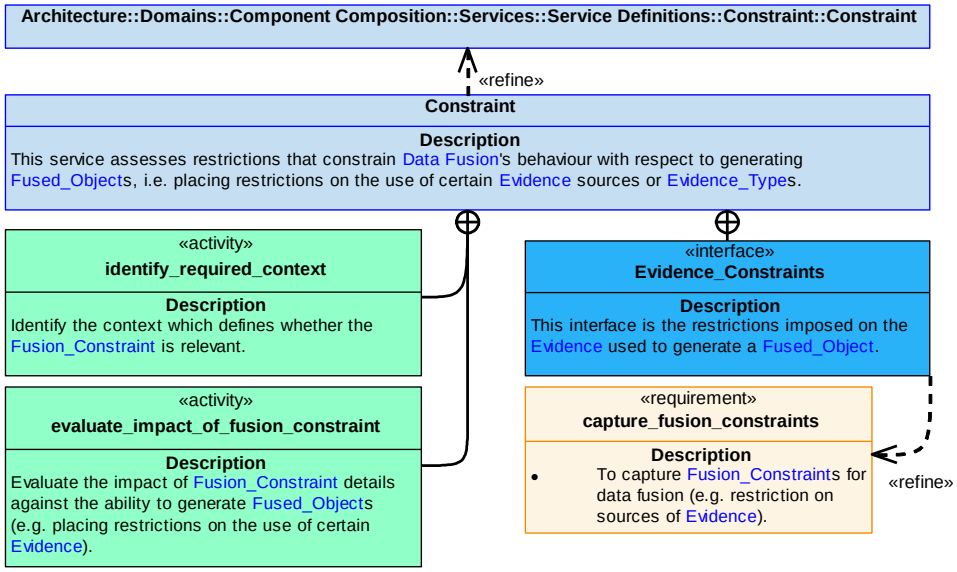

Figure 319: Constraint Service Policy

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.

B.2.13.7.1.6 Fusion_Capability

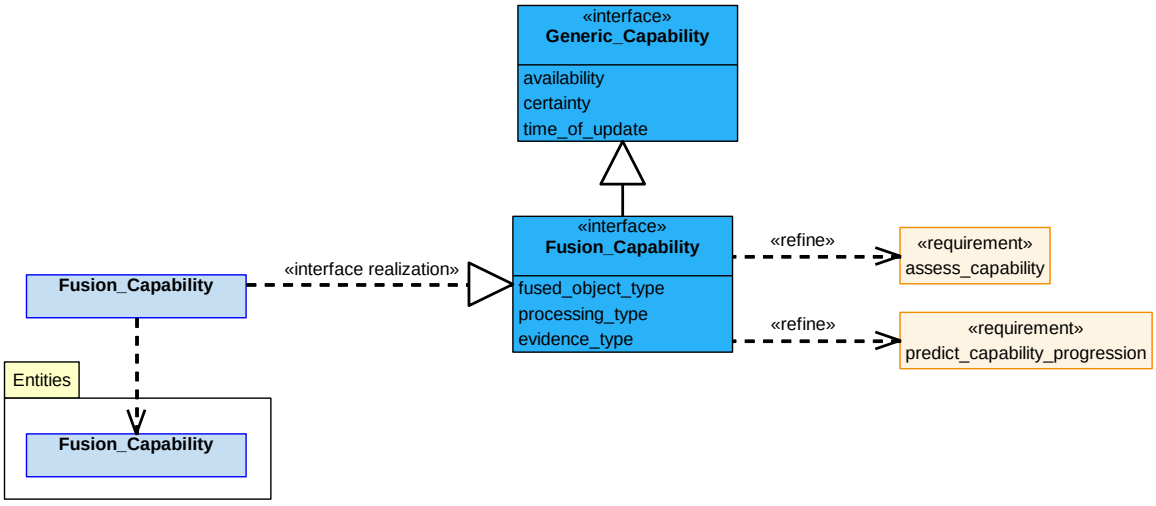

Figure 320: Fusion_Capability Service Definition

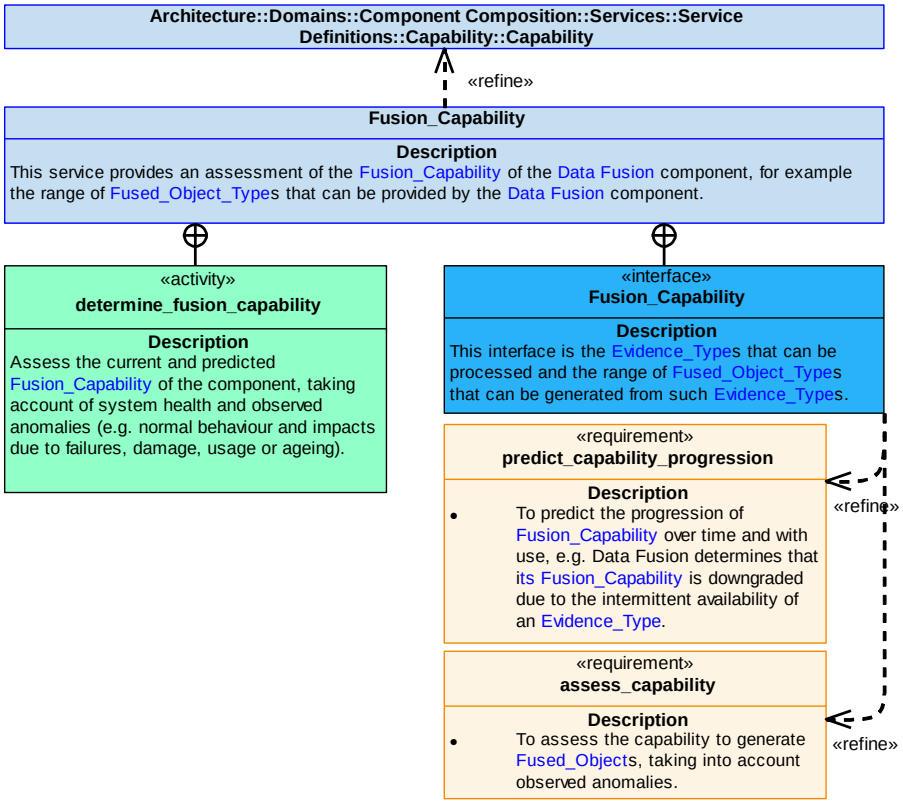

Figure 321: Fusion_Capability Service Policy

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).

B.2.13.7.1.7 Fusion_Capability_Evidence

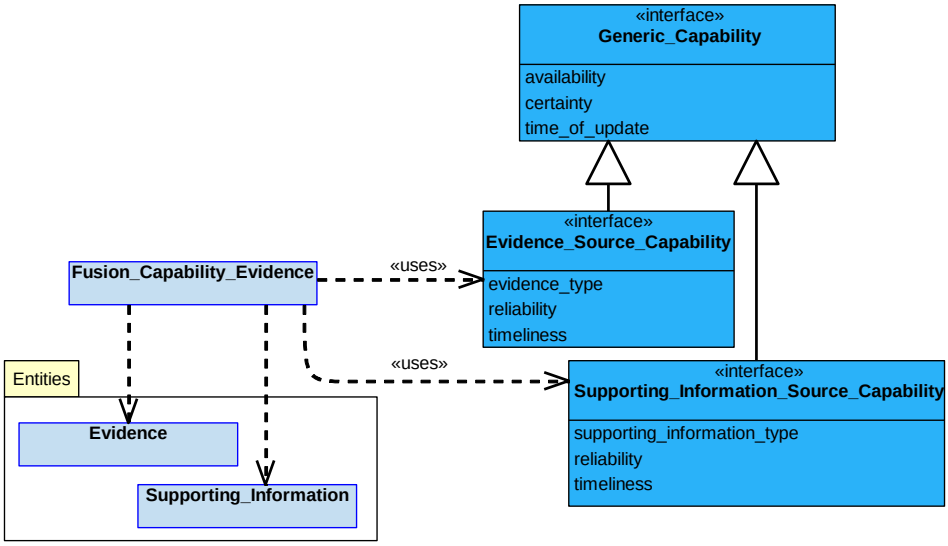

Figure 322: Fusion_Capability_Evidence Service Definition

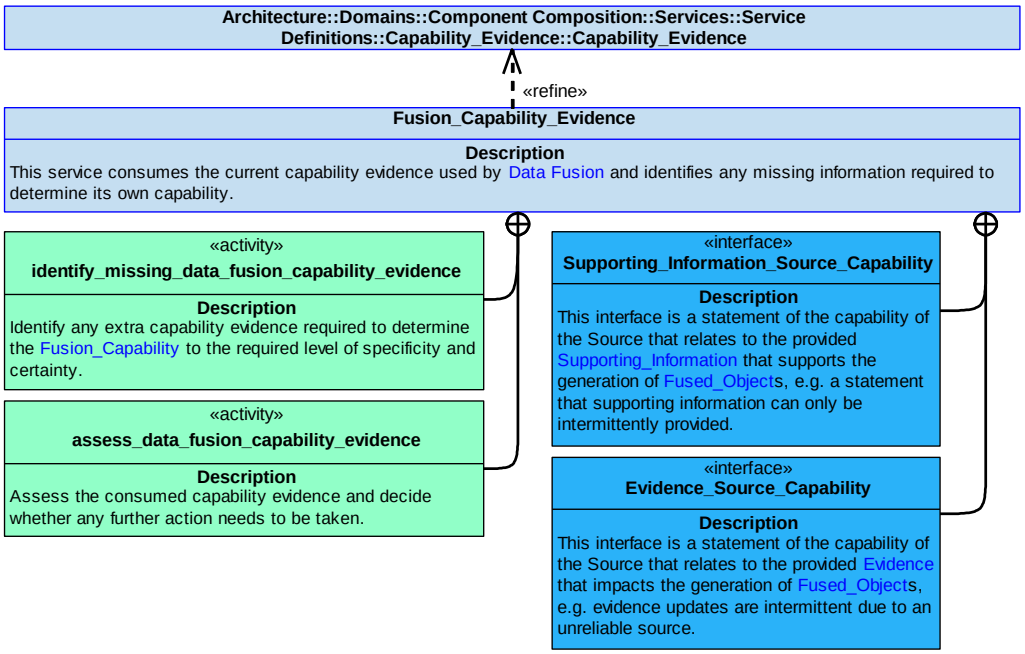

Figure 323: Fusion_Capability_Evidence Service Policy

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 the Source that relates to the provided 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.

Supporting_Information_Source_Capability

This interface is a statement of the capability of the Source that relates to the provided Supporting_Information that supports the generation of Fused_Objects, e.g. a statement that supporting information can only be intermittently provided.

Attributes

supporting_information_type	The type of Supporting_Information (e.g. platform data, weather condition report or terrain data) that the source is capable of providing.
reliability	A capability measure that indicates whether the Supporting_Information source is able to provide consistent/repeatable Supporting_Information outputs.
timeliness	A capability measure that indicates whether the Supporting_Information source is able to provide accessible and available Supporting_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.

B.2.13.7.2 Service Dependencies

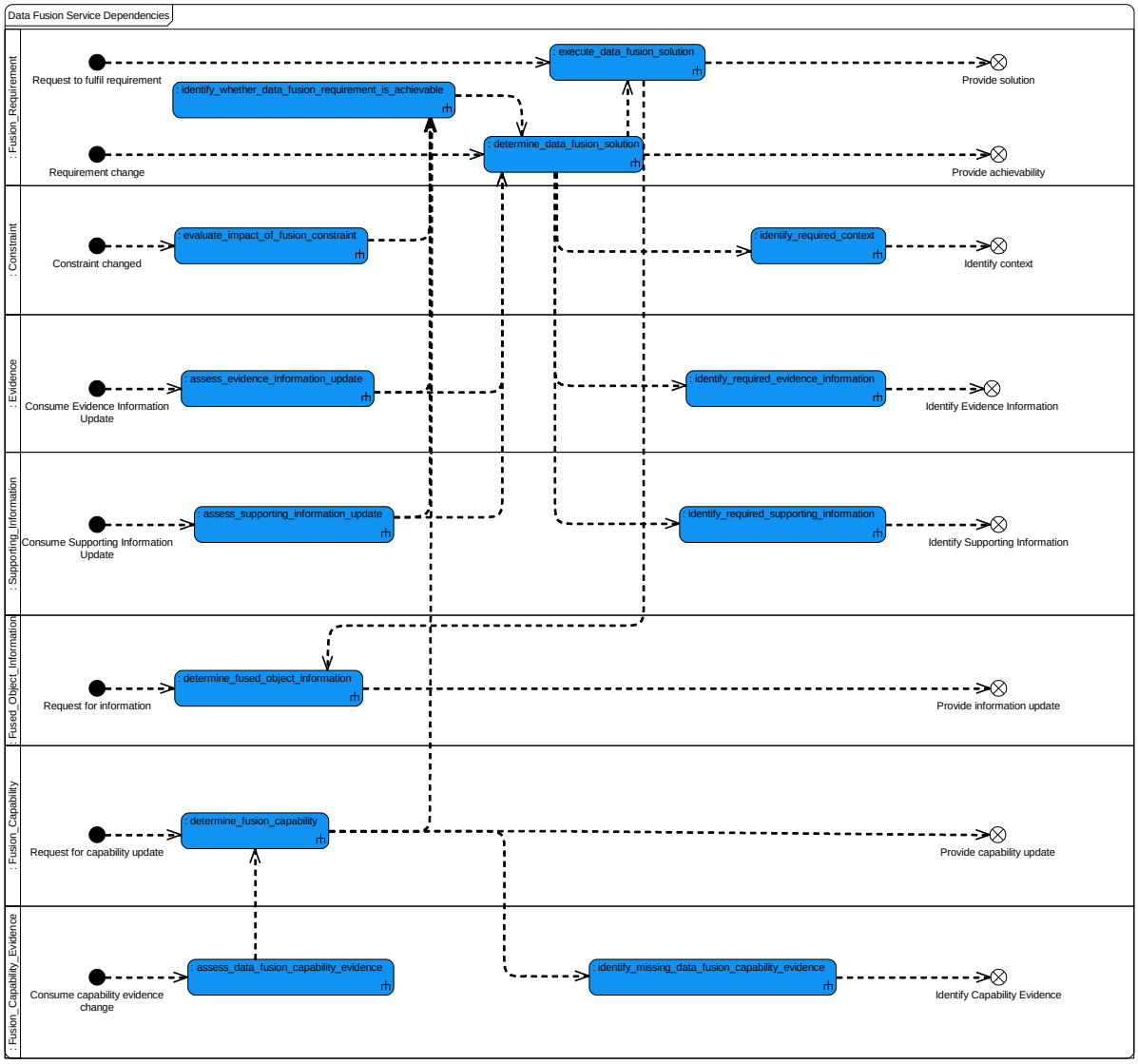

Figure 324: Data Fusion Service Dependencies

B.2.14 Destructive Effects

B.2.14.1 Role

The role of Destructive Effects is to determine the destructive effects achievable by weapons and the settings necessary to achieve the selected effect.

B.2.14.2 Overview

Control Architecture

Destructive Effects is a service component as defined in the Control Architecture policy.

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. 2 * 500lb bomb or 1 * 1000lb bomb) 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.

B.2.14.3 Service Summary

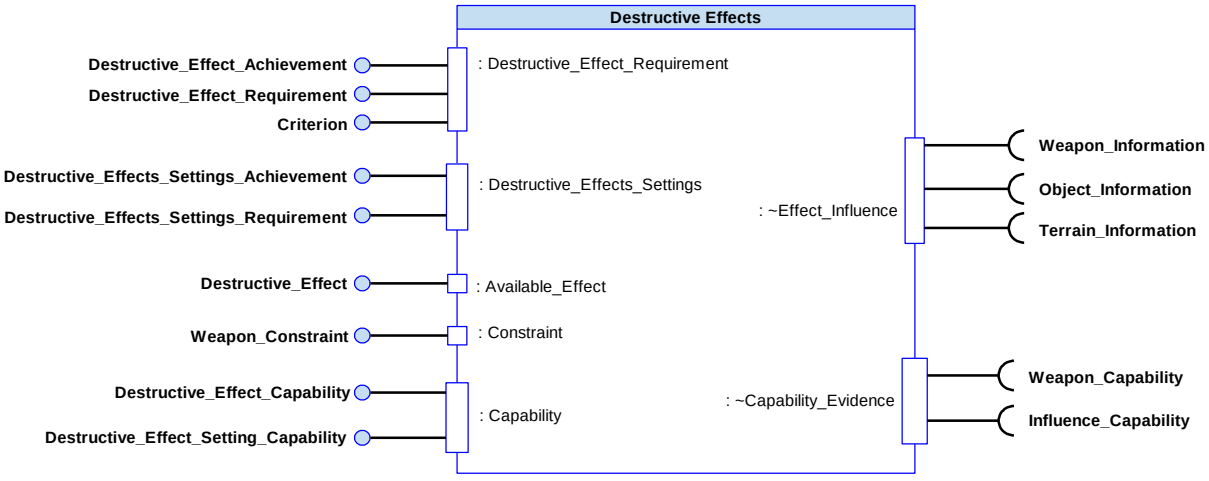

Figure 325: Destructive Effects Service Summary

B.2.14.4 Responsibilities

capture_requirements_for_destructive_effects

- To capture provided Destructive_Effect_Requirements.

control_destructive_effect_settings

- To control the Destructive_Effect_Settings, e.g. set fusing mode.

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.

capture_constraints_for_destructive_effects

- To capture provided Constraints that apply to the use of Weapon_Resources.

predict_capability_progression

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

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.

capture_measurement_criteria_for_destructive_effects

- To capture provided Measurement_Criterion for the use of the Destructive_Effects.

identify_pre-conditions

- To identify Pre-conditions required to achieve a Destructive_Effect.

determine_quality_of_destructive_effects_solution

- To determine the quality of a Destructive_Effect against given Measurement_Criterion.

B.2.14.5 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.

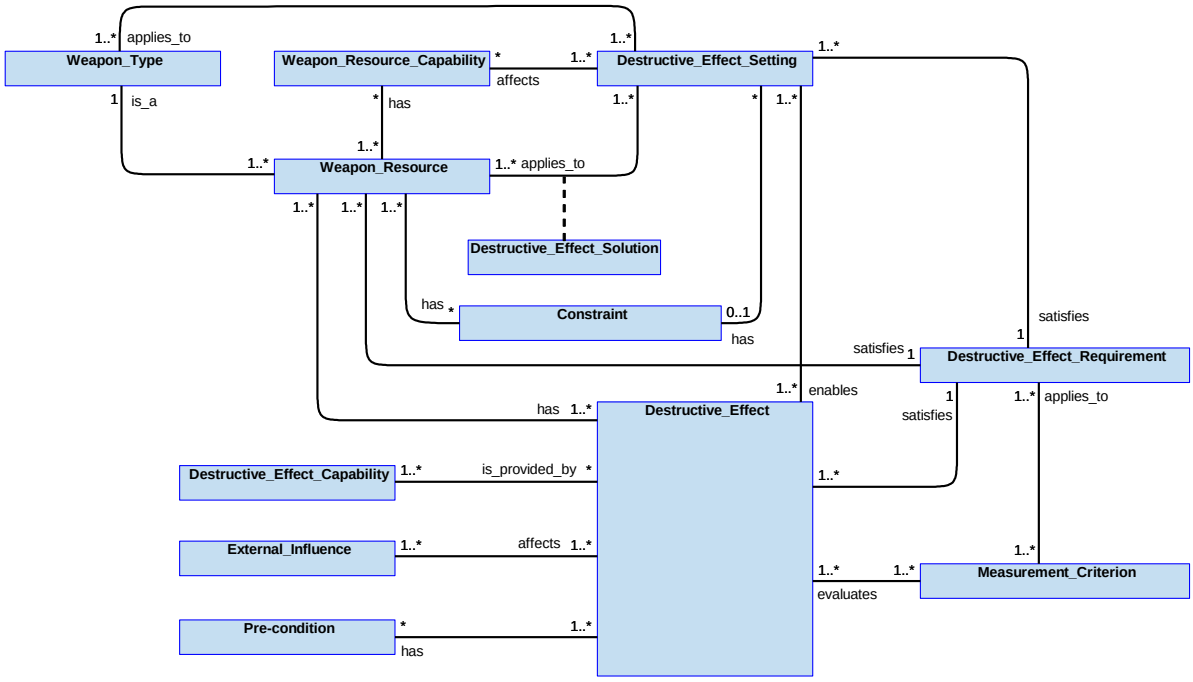

Figure 326: Destructive Effects Semantics

B.2.14.5.1 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_Setting

A selection of required settings to produce a Destructive_Effect, e.g. setting a bomb to air burst mode.

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. a set impact angle must be achievable.

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_Solution

A solution to a Destructive_Effect_Requirement determined by selection of Weapon_Resources and the associated Destructive_Effect_Settings.

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.

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.

B.2.14.6 Design Rationale

B.2.14.6.1 Assumptions

None.

B.2.14.6.2 Design Considerations

Directly Applicable Policies

These policies 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 PRA deployment.
- Data Driving - This is applicable as the weapon description items should be data driveable as per the WIUK Reference Architecture Framework, Ref. [19]. There are numerous Weapon_Types each of which has 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 exploitation 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 exploitation 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.

B.2.14.6.3 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.

B.2.14.6.4 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.

B.2.14.7 Services

B.2.14.7.1 Service Definitions

B.2.14.7.1.1 Destructive_Effect_Requirement

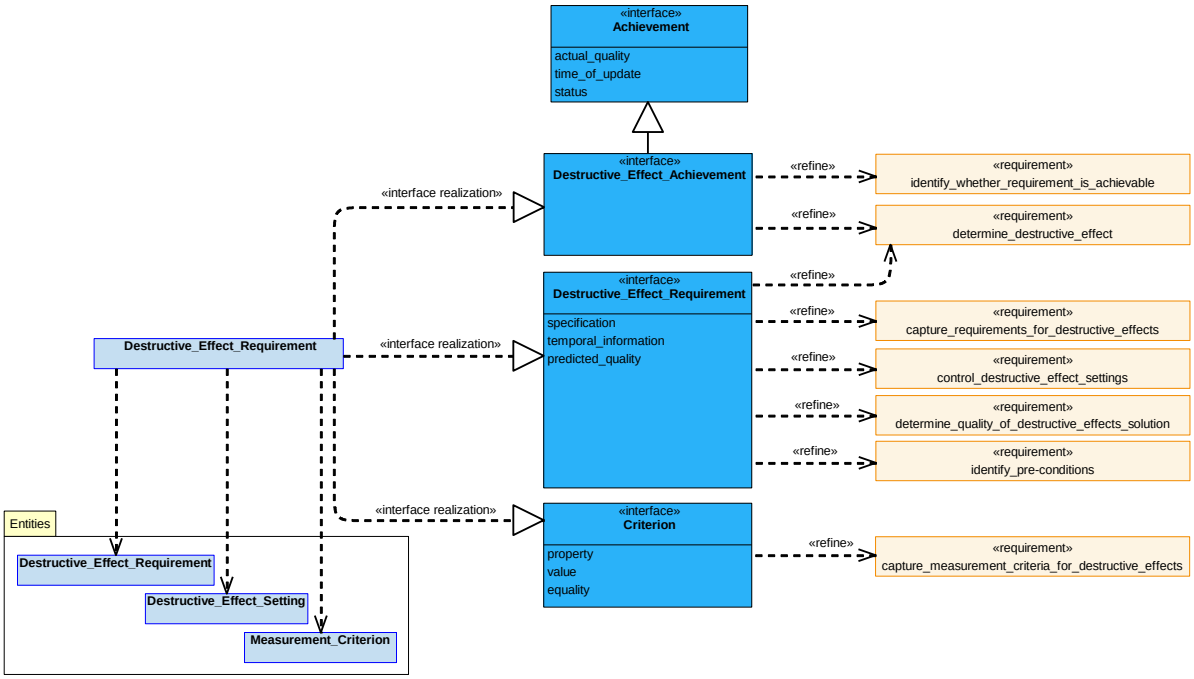

Figure 327: Destructive_Effect_Requirement Service Definition

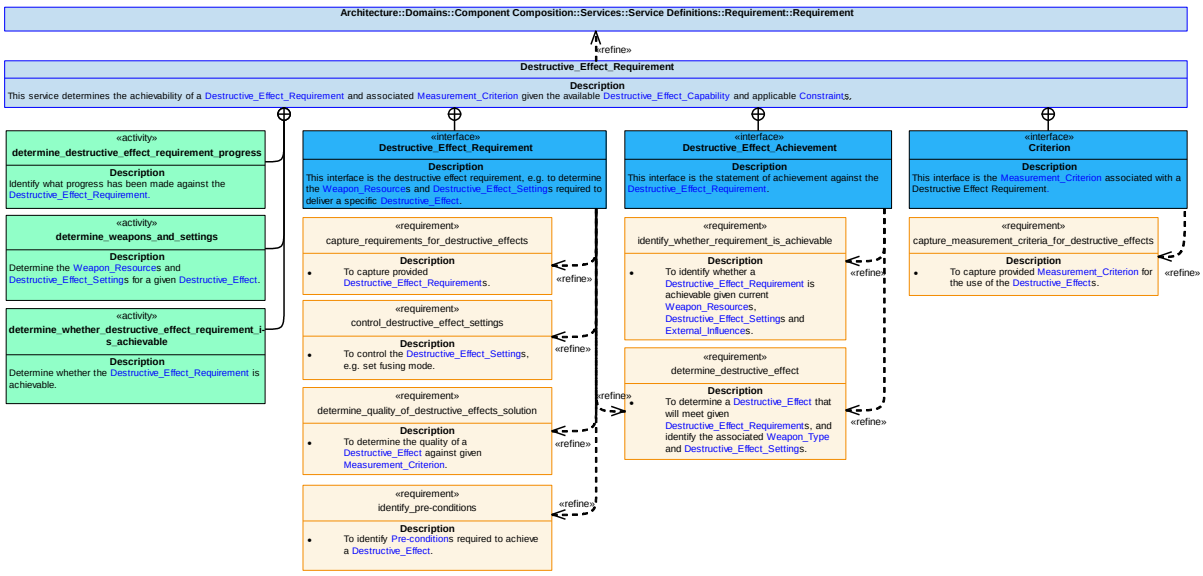

Figure 328: Destructive_Effect_Requirement Service Policy

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.

B.2.14.7.1.2 Destructive_Effects_Settings

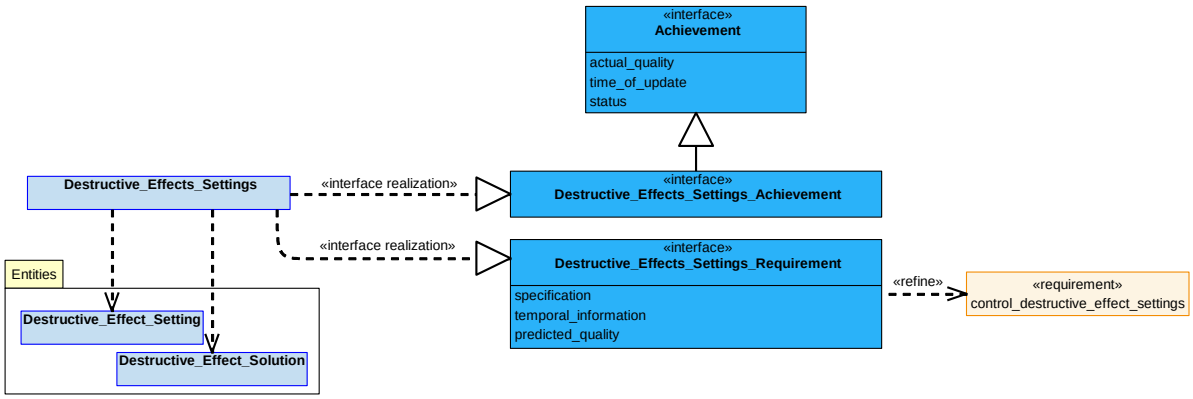

Figure 329: Destructive_Effects_Setting Service Definition

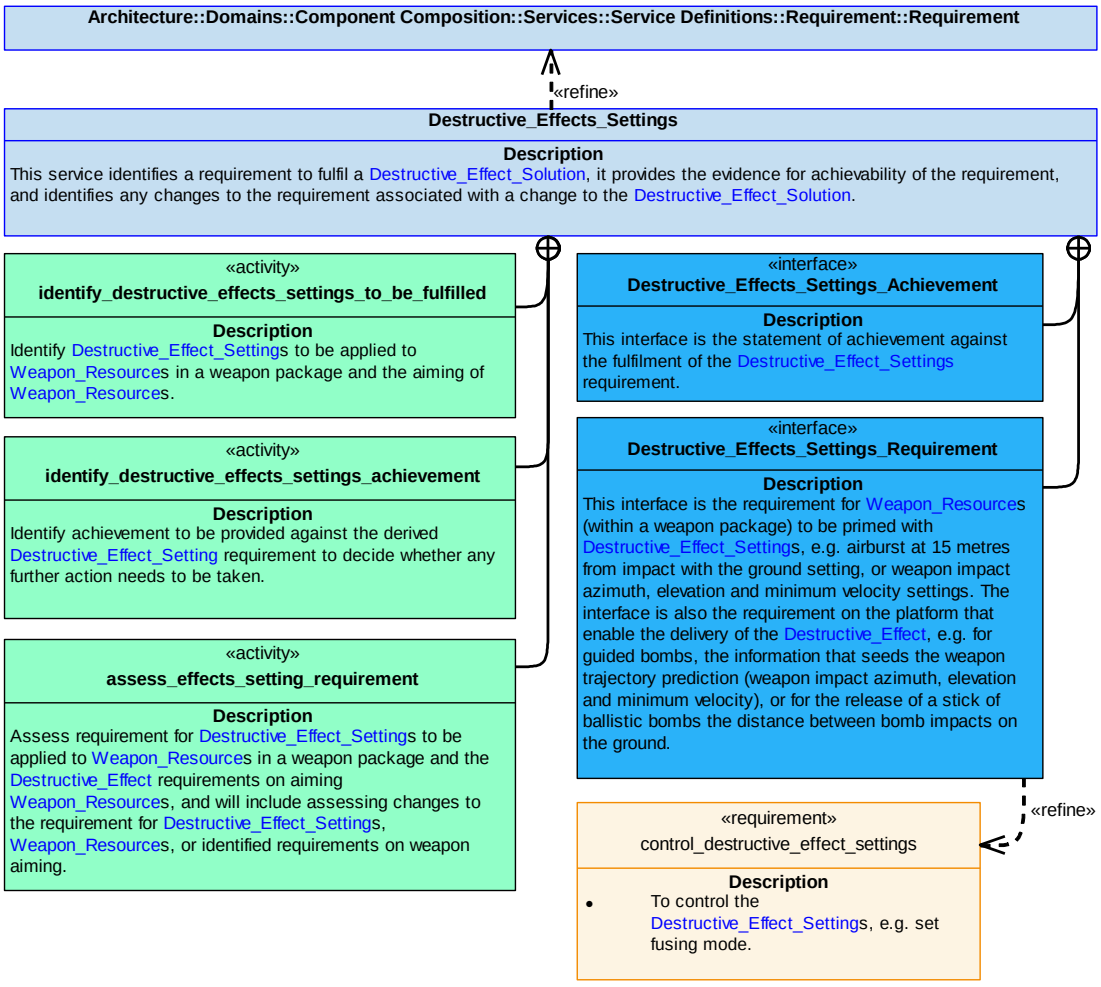

Figure 330: Destructive_Effects_Settings Service Policy

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 enable 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 the release of a stick of ballistic bombs the distance between bomb impacts on the ground.

Attributes

specification	This is the requirement for a Destructive_Effect_Solution, 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 a 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 the release of a stick of 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 requirement for Destructive_Effect_Settings to be applied to Weapon_Resources in a weapon package and the Destructive_Effect requirements on aiming Weapon_Resources, and will include assessing changes to the requirement for Destructive_Effect_Settings, Weapon_Resources, or identified requirements on weapon aiming.

B.2.14.7.1.3 Available_Effect

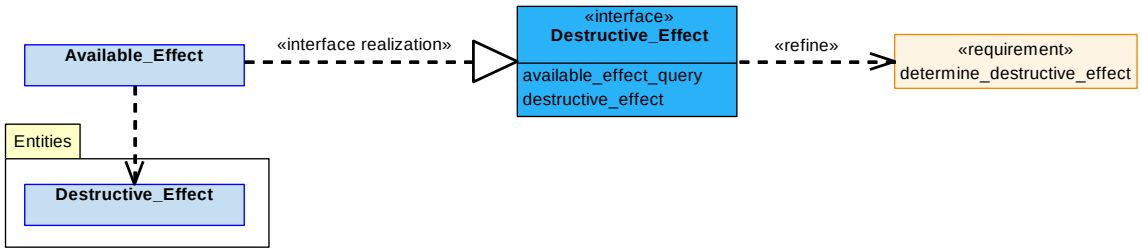

Figure 331: Available_Effect Service Definition

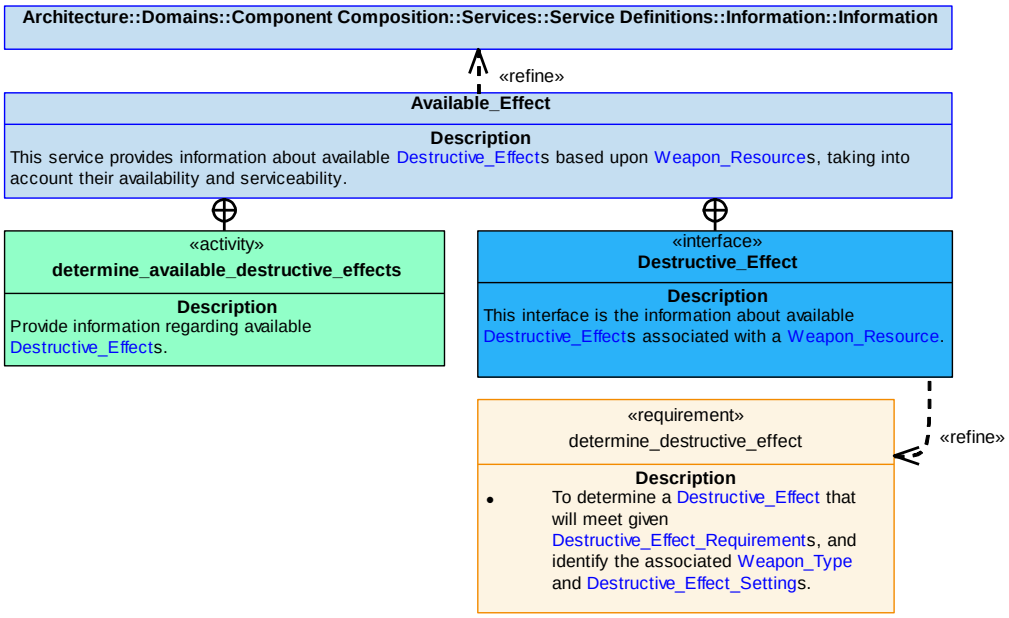

Figure 332: Available_Effect Service Policy

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.

B.2.14.7.1.4 Effect_Influence

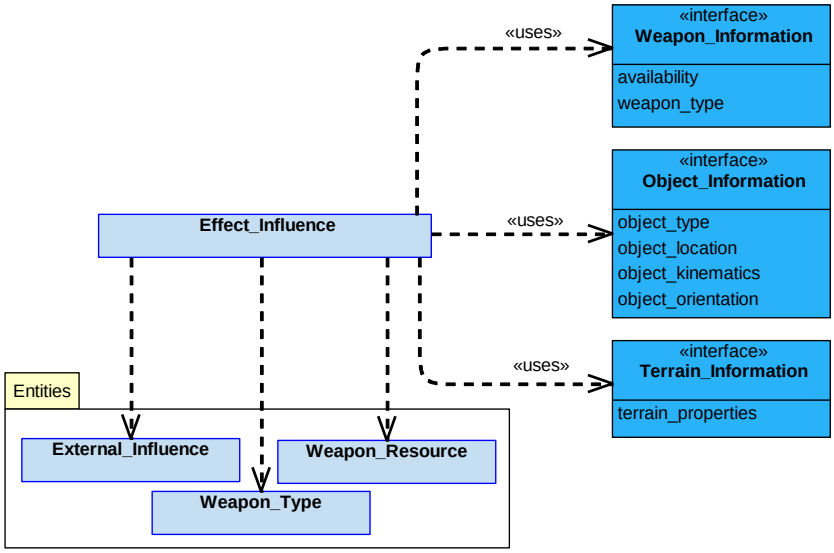

Figure 333: Effect_Influence Service Definition

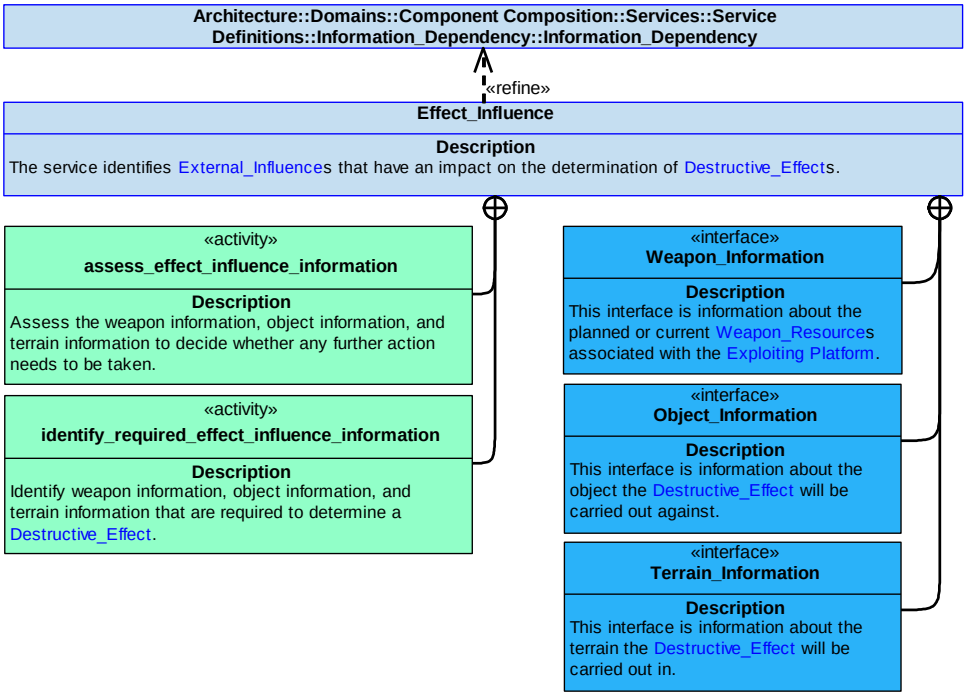

Figure 334: Effect_Influence Service Policy

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.

B.2.14.7.1.5 Constraint

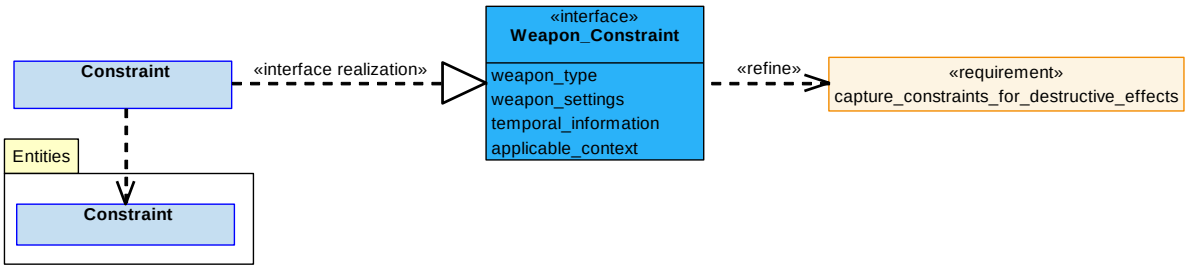

Figure 335: Constraint Service Definition

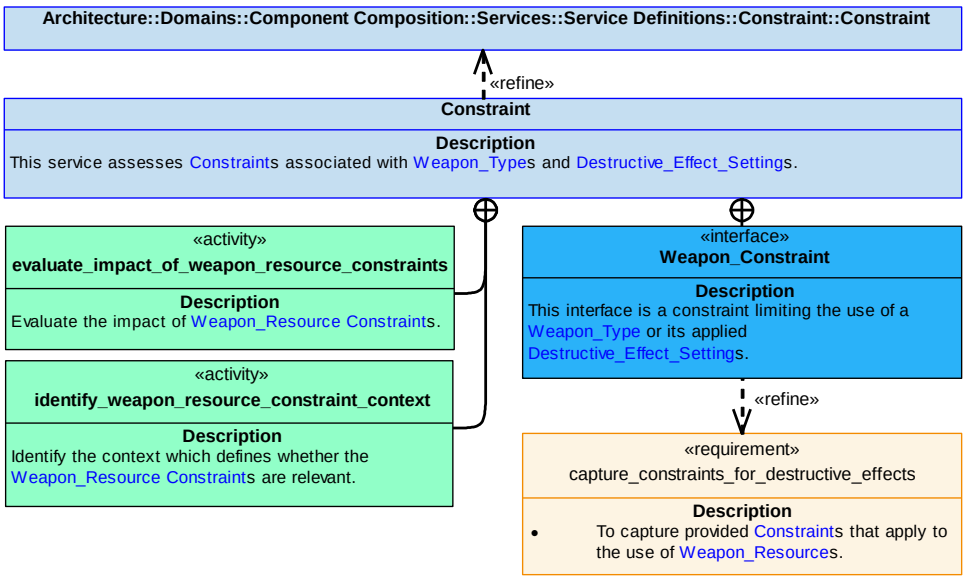

Figure 336: Constraint Service Policy

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.

B.2.14.7.1.6 Capability

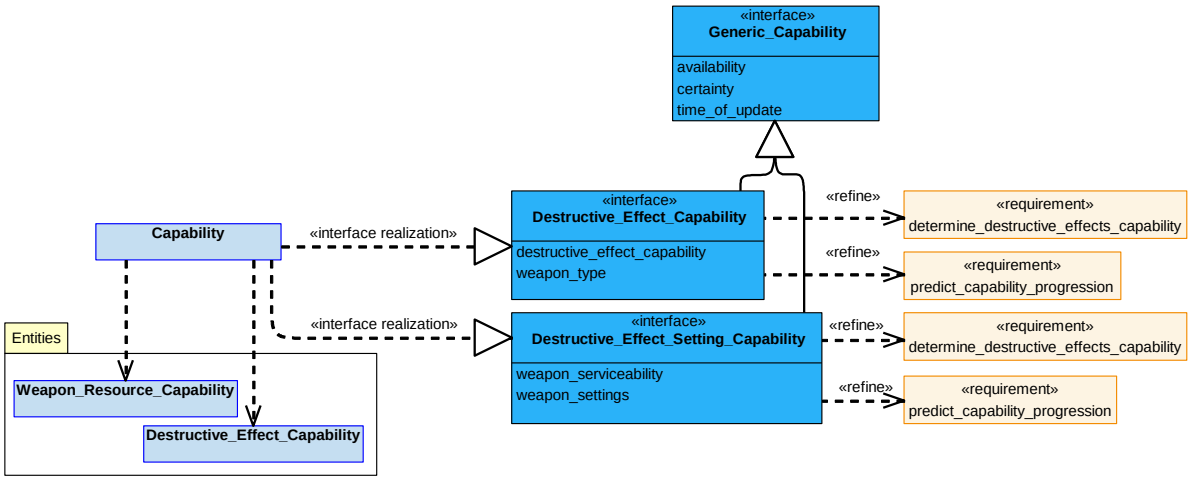

Figure 337: Capability Service Definition

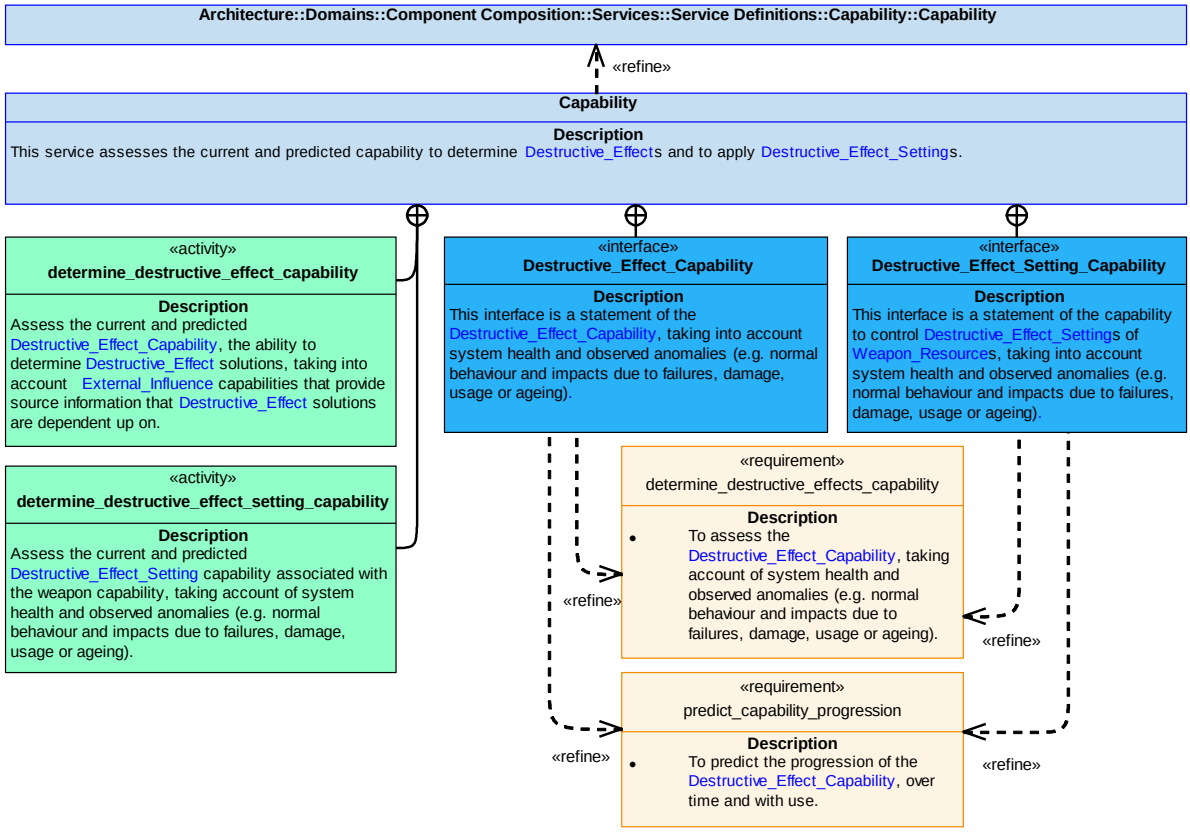

Figure 338: Capability Service Policy

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).

B.2.14.7.1.7 Capability_Evidence

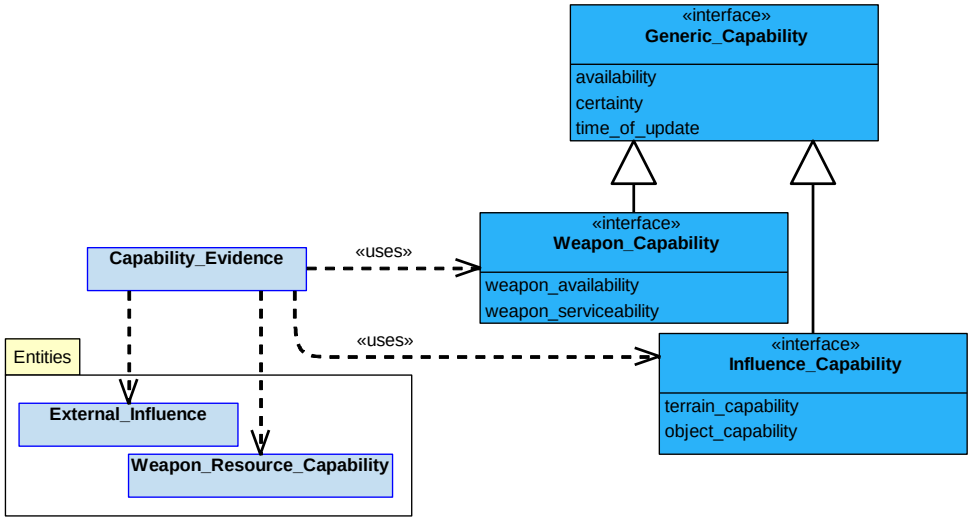

Figure 339: Capability_Evidence Service Definition

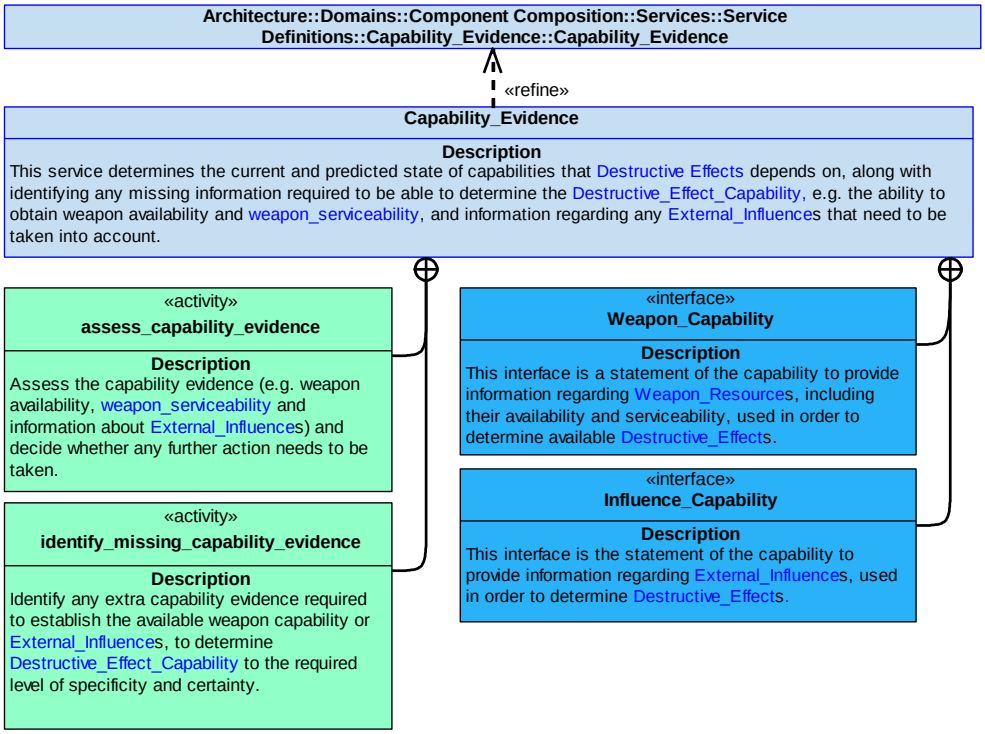

Figure 340: Capability_Evidence Service Policy

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.

B.2.14.7.2 Service Dependencies

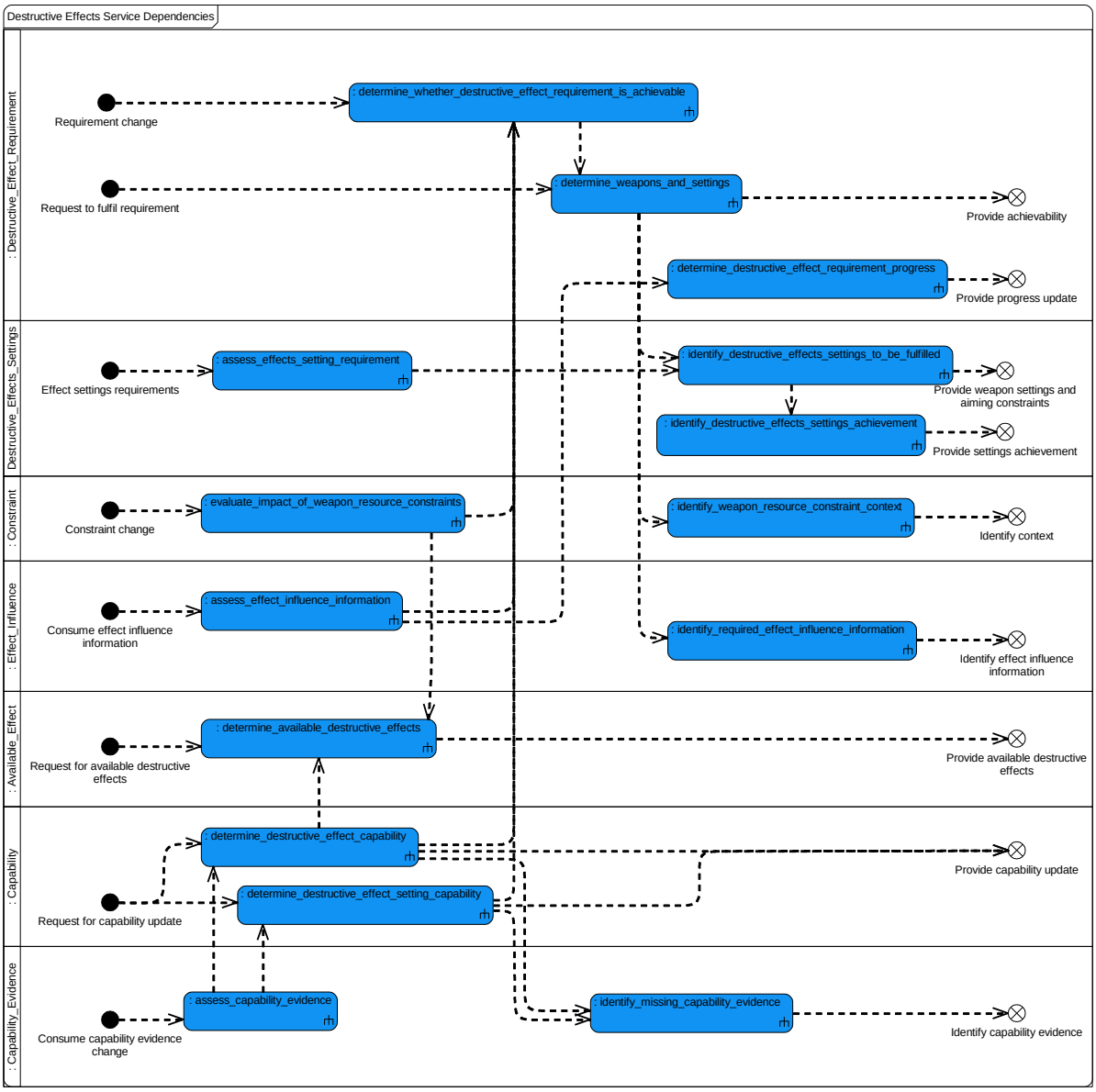

Figure 341: Destructive Effects Service Dependencies

B.2.15 Effectors

B.2.15.1 Role

The role of Effectors is to provide an interface to manage effector devices.

B.2.15.2 Overview

Control Architecture

Effectors is a resource component as defined in the Control Architecture policy.

Standard Pattern of Use

In response to a demand, an Effector_Resource will provide an Effect.

Examples of Use

This component will be required for control of Effector_Resources to provide effects such as:

- Heating for ice protection.
- Extinguishing for fire protection.
- Aileron actuation.
- A jamming pod with a simple interface (on/off).
- Illumination of a target or search area.

B.2.15.3 Service Summary

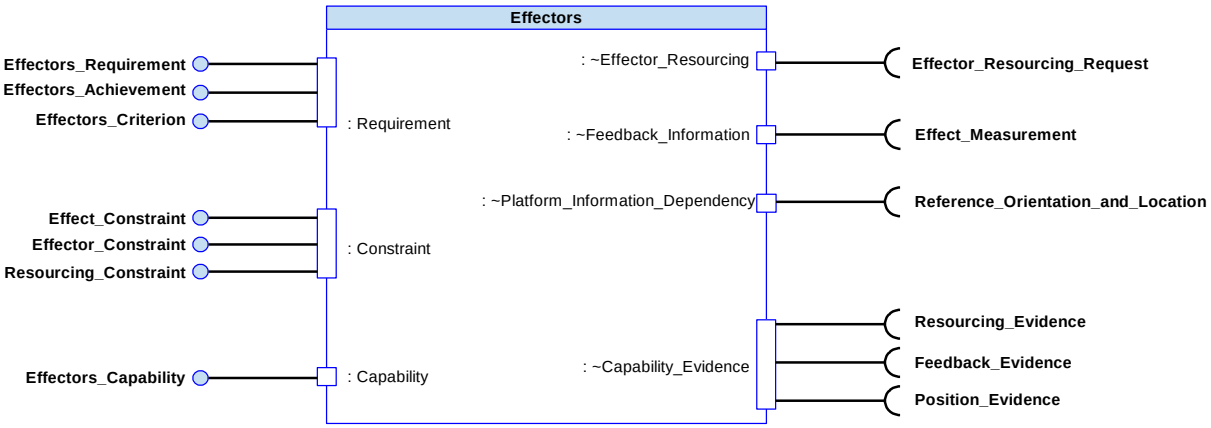

Figure 342: Effectors Service Summary

B.2.15.4 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).

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).

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.

predict_capability_progression

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

determine_effector_solution

- To determine a solution for use of Effector_Resources that will meet given Requirements.

capture_measurement_criteria

- To capture Measurement_Criterion for an Effect.

determine_resources_used_by_the_effector

- To determine information about effectors use of the resources.

capture_effector_constraints

- To capture provided constraints, e.g. EMCON.

determine_if_solution_remains_feasible

- To determine if a planned or ongoing Effector_Solution remains feasible.

B.2.15.5 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.

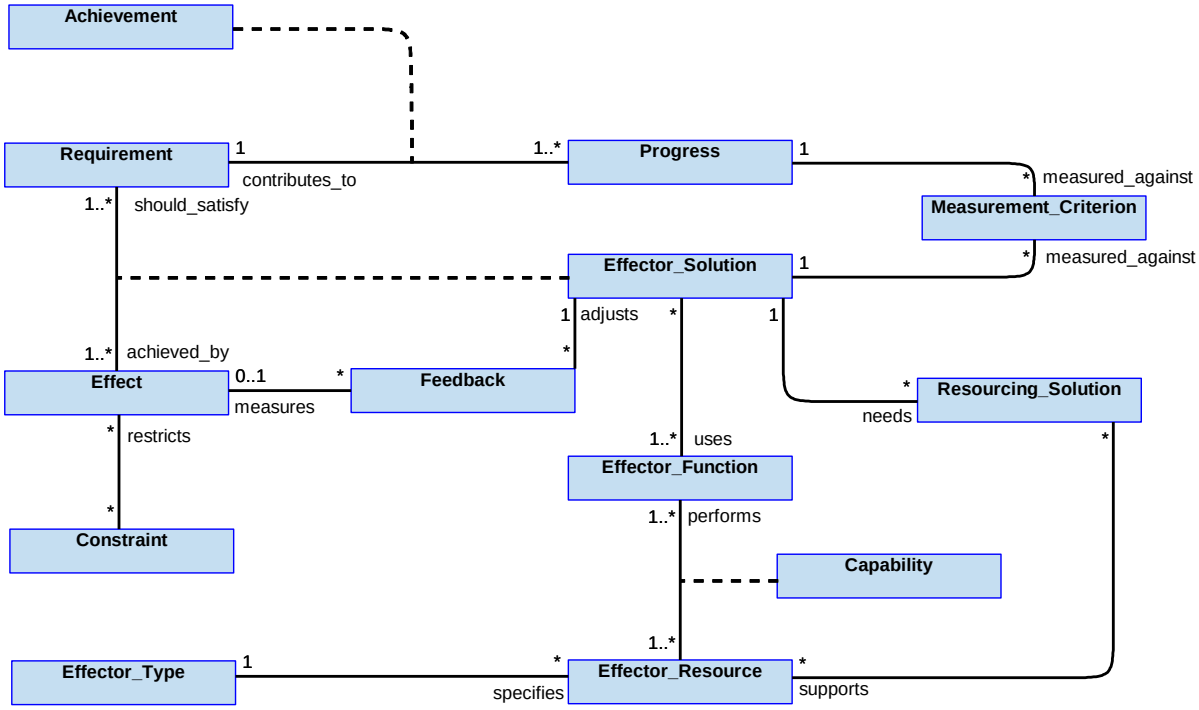

Figure 343: Effectors Semantics

B.2.15.5.1 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.

Effect

A change to the physical environment caused by an Effector_Resource.

Effector_Function

An effector operation that can be performed by an effector, e.g. heating or jamming.

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 an Effect.

Effector_Type

A type of effector that can be utilised, e.g. an aileron actuator.

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 control an effector to 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.

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 platform process which inhibits the Effector_Solution from supporting its Effect.

Feedback

The information needed to adjust the Effector_Solution to achieve the intended Effect.

B.2.15.6 Design Rationale

B.2.15.6.1 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 policy.
- An Effector_Resource produces an effect on an aspect of the physical environment (e.g. an actuator that moves a rudder, 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.

B.2.15.6.2 Design Considerations

Directly Applicable Policies

These policies 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 is direct (i.e. not via a service interface), therefore it will be specific 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. The PRA accommodates this by having 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.

B.2.15.6.3 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.

B.2.15.6.4 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.

B.2.15.7 Services

B.2.15.7.1 Service Definitions

B.2.15.7.1.1 Requirement

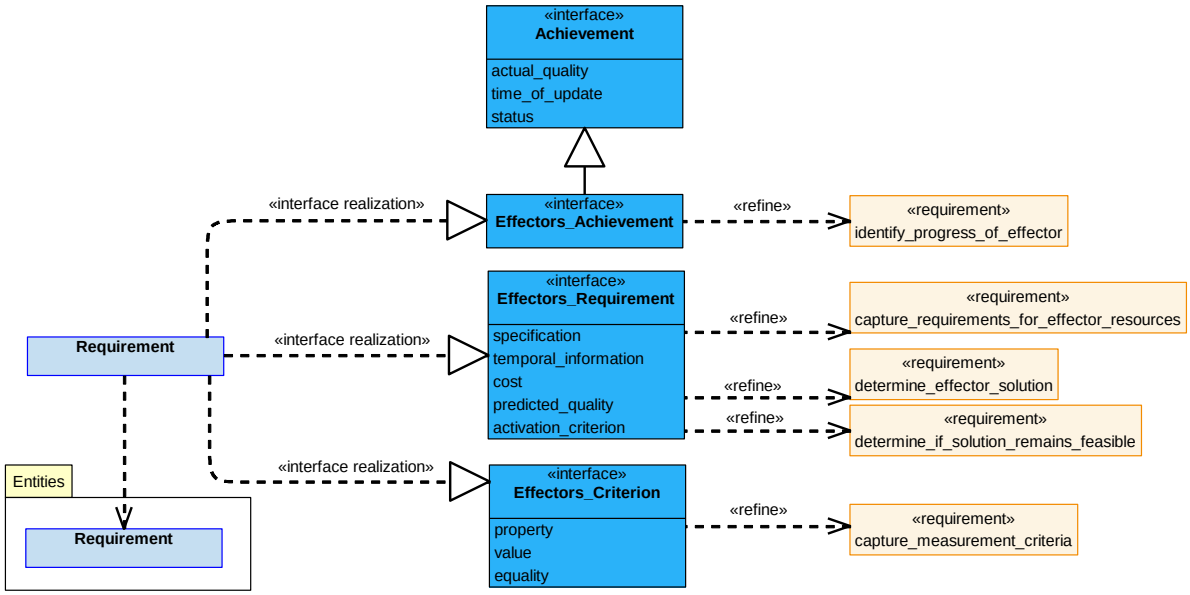

Figure 344: Requirement Service Definition

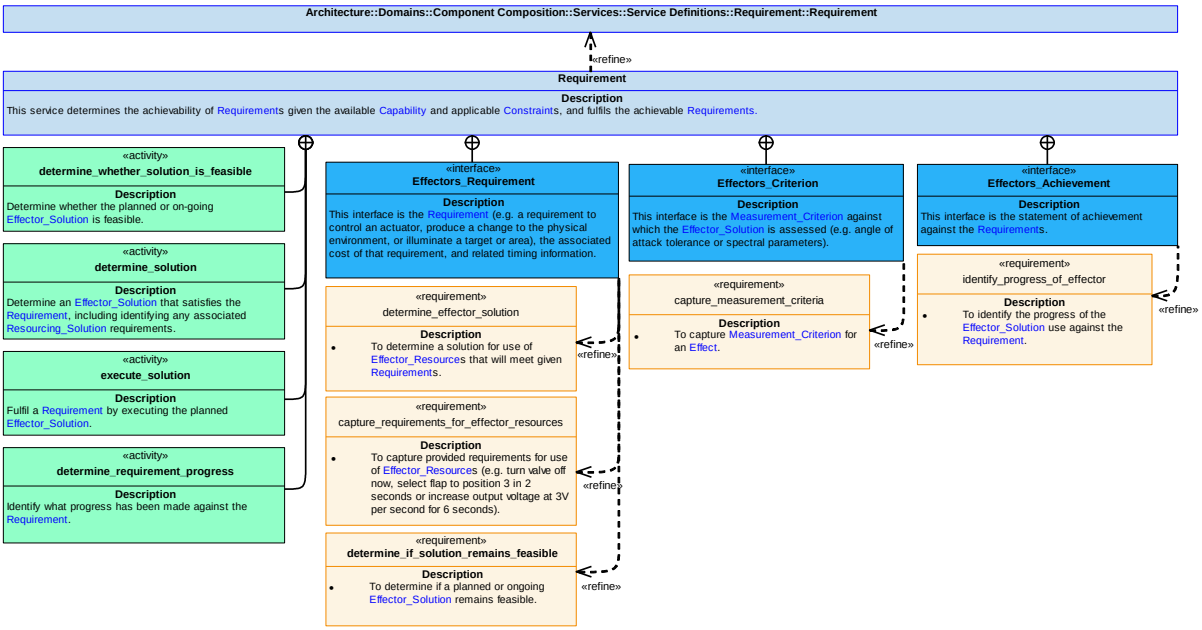

Figure 345: Requirement Service Policy

Requirement

This service determines the achievability of Requirements 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, should the Effector_Solution be enacted immediately, or at a particular time or location.

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.

B.2.15.7.1.2 Effector_Resourcing

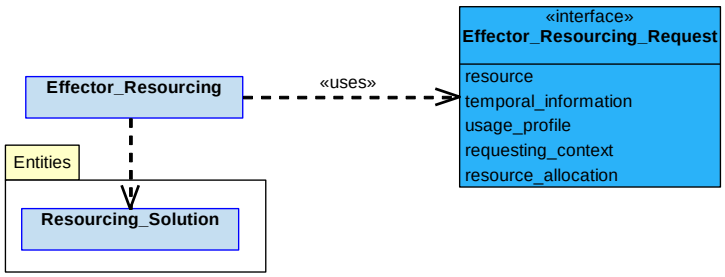

Figure 346: Effector_Resourcing Service Definition

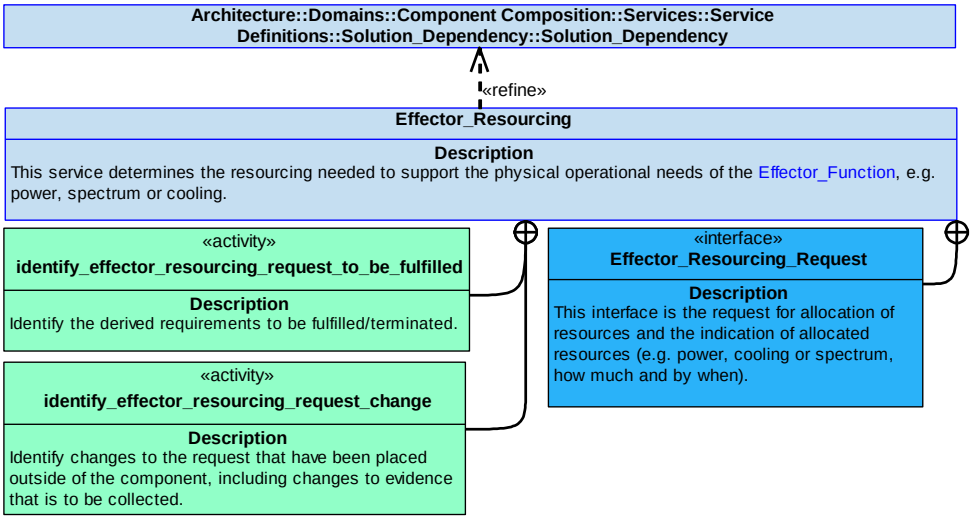

Figure 347: Effector_Resourcing Service Policy

Effector_Resourcing

This service determines the resourcing needed to support the physical operational needs of the Effector_Function, e.g. power, spectrum or cooling.

Interface

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, cooling or spectrum
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.

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.

B.2.15.7.1.3 Feedback_Information

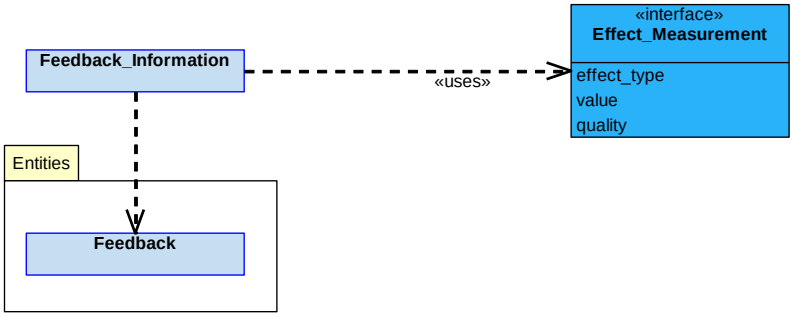

Figure 348: Feedback_Information Service Definition

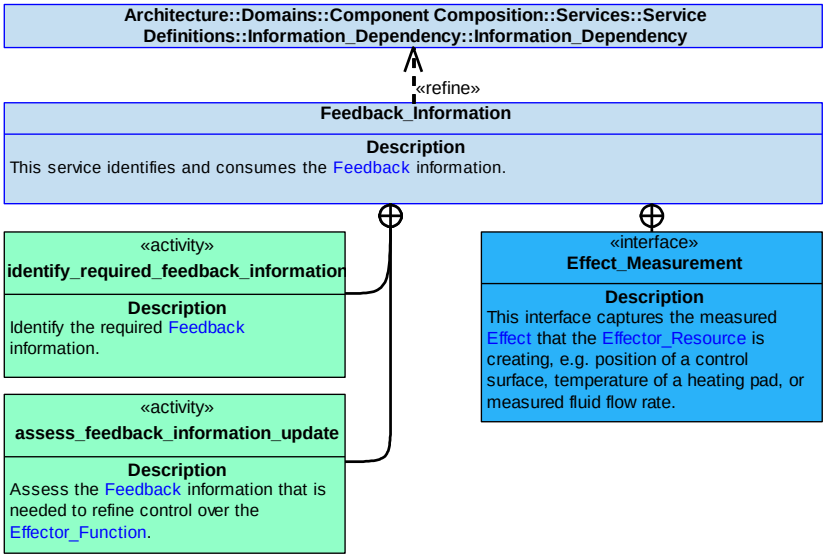

Figure 349: Feedback_Information Policy Diagram

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. position of a control surface, temperature of a heating pad, or measured fluid flow rate.

Attributes

effect_type	The effect that is being measured, e.g. elevator position or fuel flow.
value	The measured value of the effect, e.g. a deflection in degrees, or a temperature in 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.

B.2.15.7.1.4 Platform_Information_Dependency

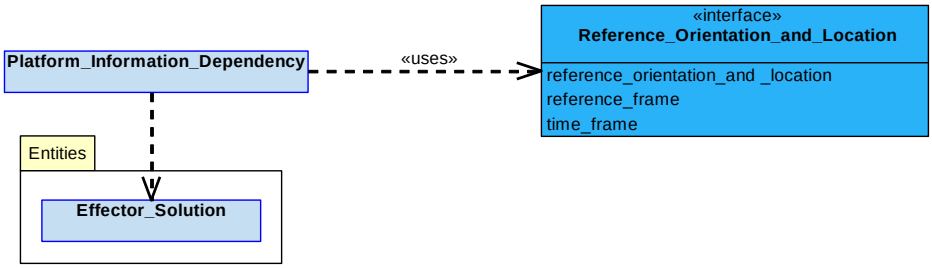

Figure 350: Platform_Information_Dependency Service Definition

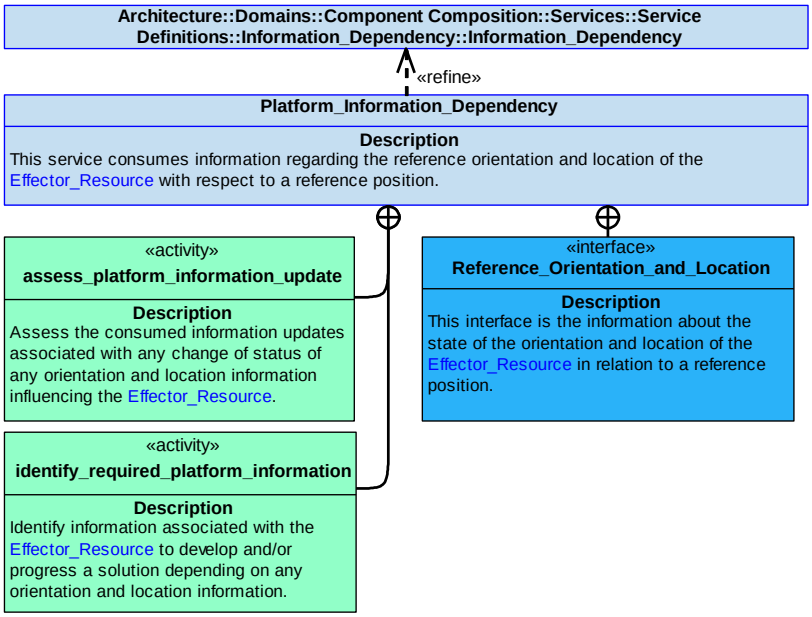

Figure 351: Platform_Information_Dependency Service Policy

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.

B.2.15.7.1.5 Constraint

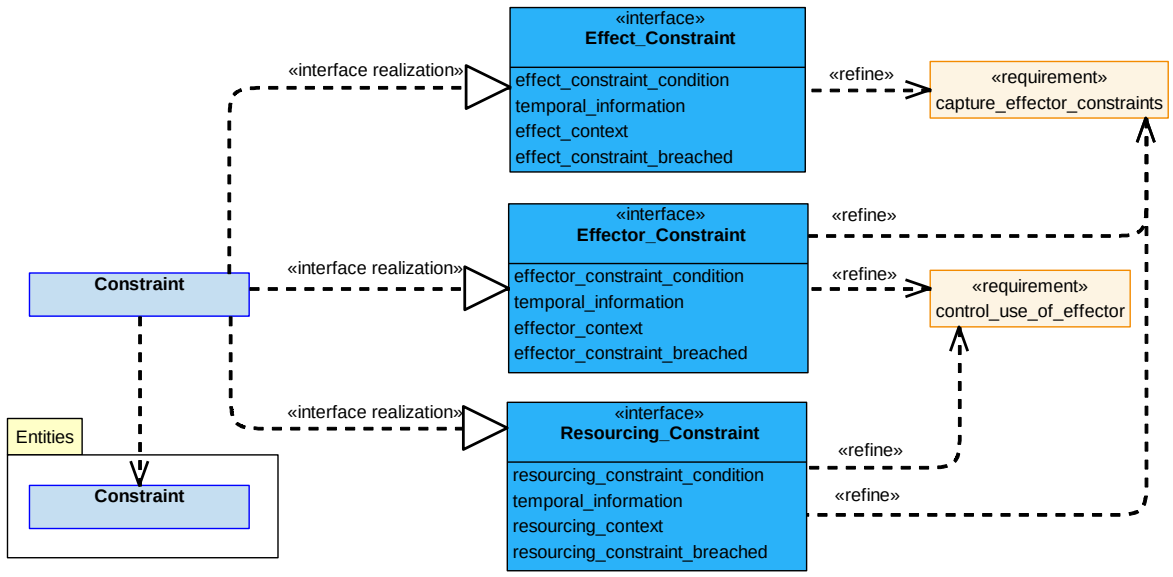

Figure 352: Constraint Service Definition

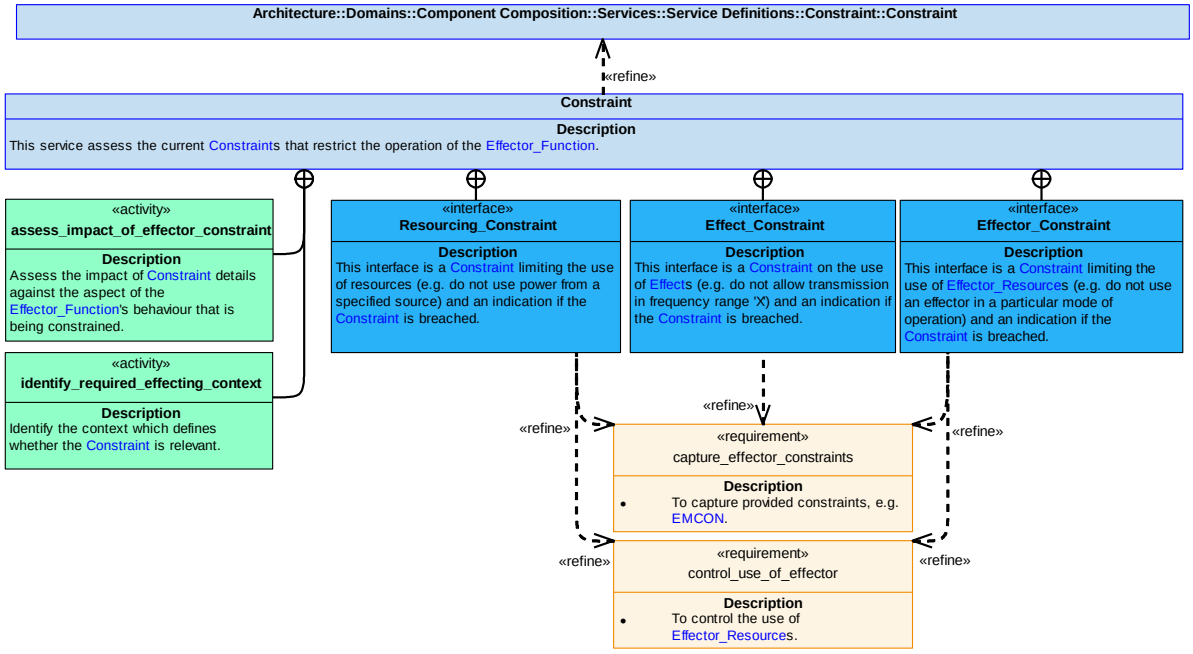

Figure 353: Constraint Service Policy

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.

B.2.15.7.1.6 Capability

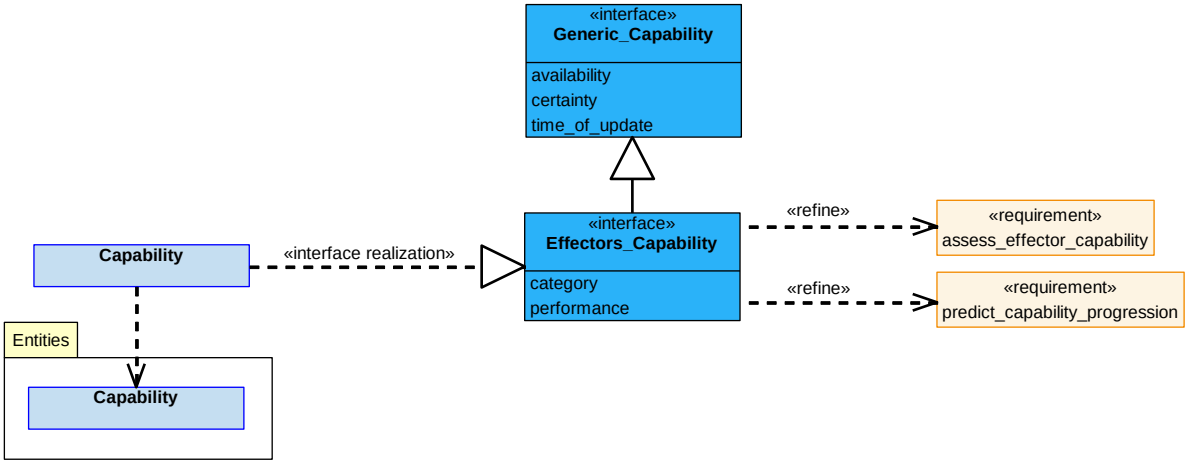

Figure 354: Capability Service Definition

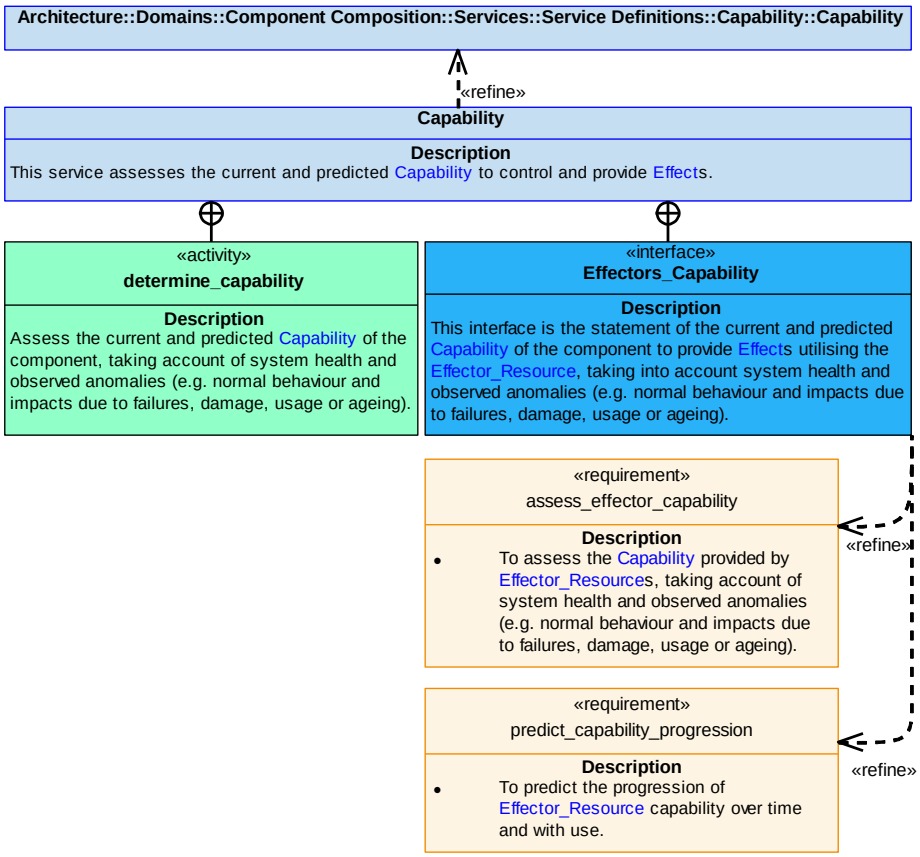

Figure 355: Capability Service Policy

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).

B.2.15.7.1.7 Capability_Evidence

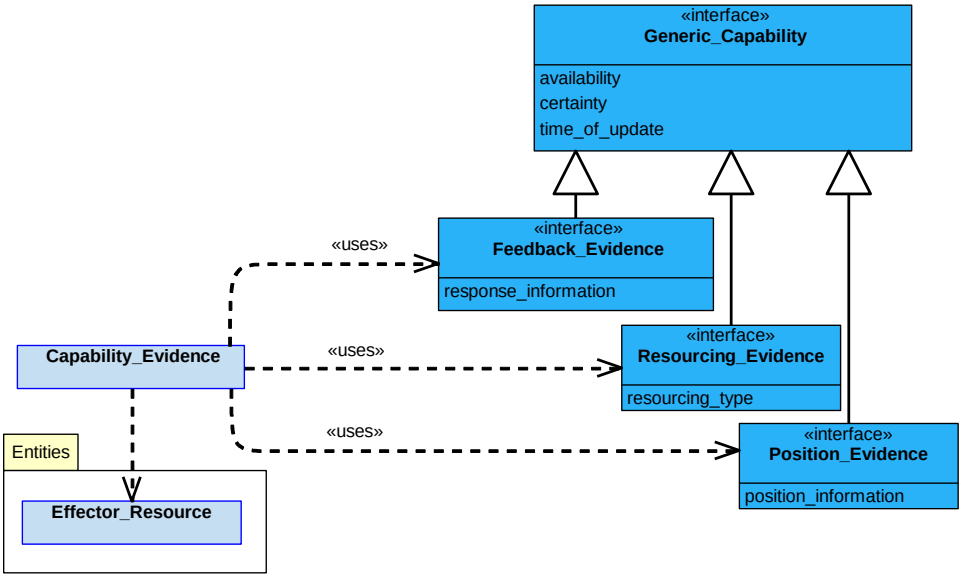

Figure 356: Capability_Evidence Service Definition

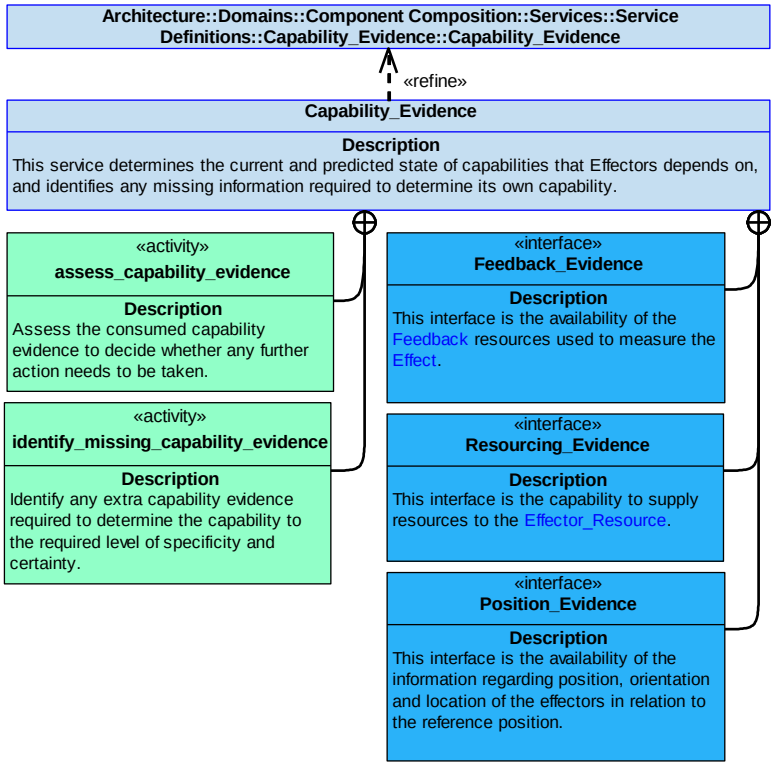

Figure 357: Capability_Evidence Service Policy

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.

Interfaces

Feedback_Evidence

This interface is the availability of the Feedback resources used to measure the Effect.

Attribute

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.

Attribute

resourcing_type The definition of the resource.

Position_Evidence

This interface is the availability of the information regarding position, orientation and location of the effectors in relation to the reference position.

Attribute

position_information The availability of information relating to position, orientation and location.

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.

B.2.15.7.2 Service Dependencies

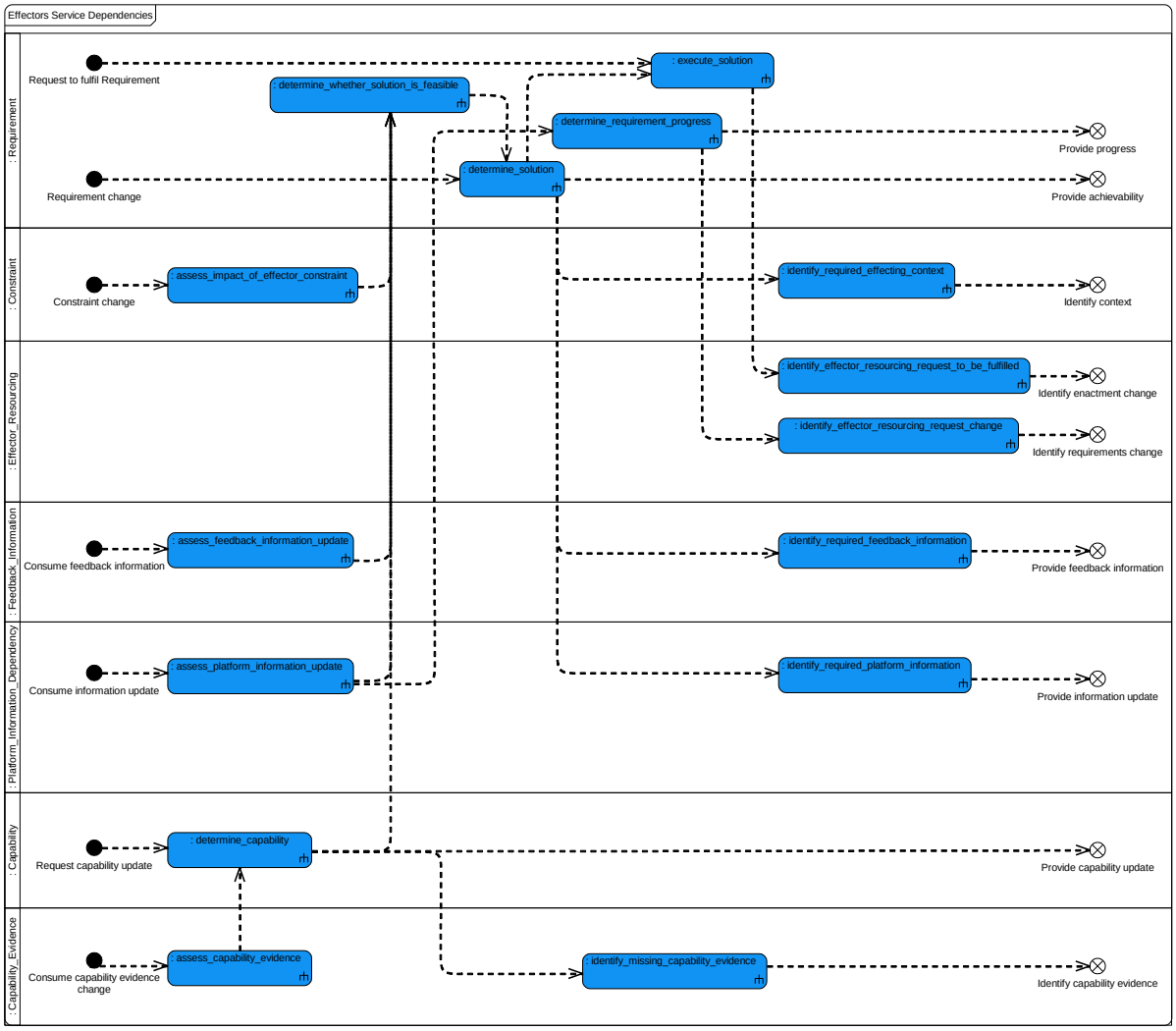

Figure 358: Effectors Service Dependencies

B.2.16 Environment Infrastructure

B.2.16.1 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.

B.2.16.2 Overview

Control Architecture

Environment Infrastructure is a service component as defined in the Control Architecture policy.

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.

B.2.16.3 Service Summary

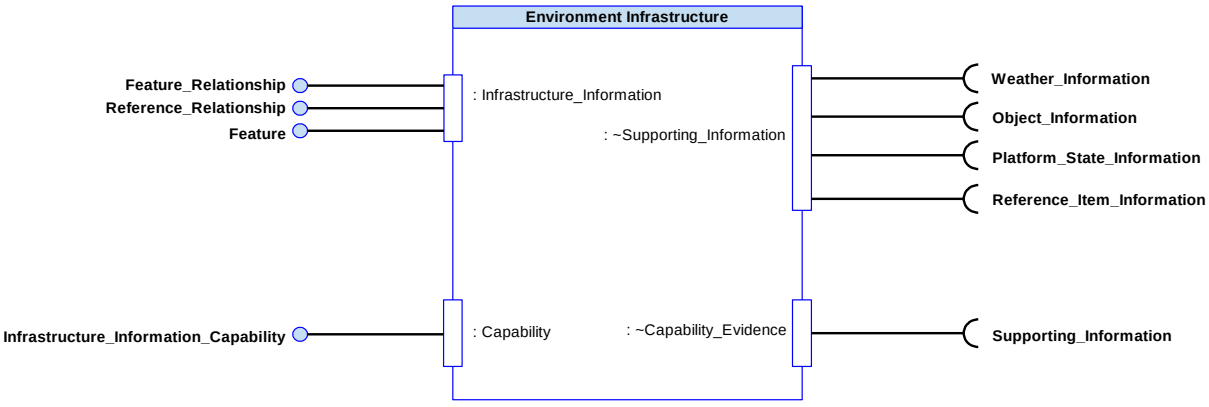

Figure 359: Environment Infrastructure Service Summary

B.2.16.4 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).

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).

determine_feature_relationship

- To determine information on how Infrastructure_Features relate to each other.

B.2.16.5 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.

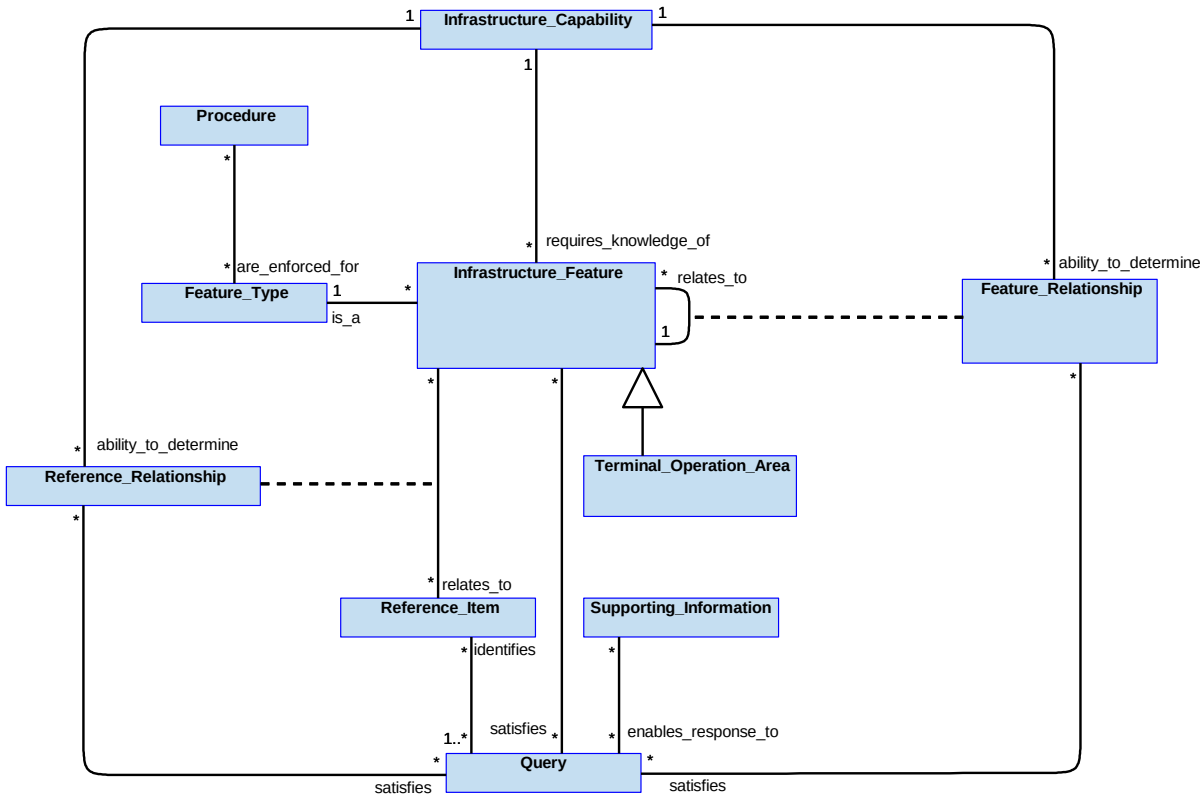

Figure 360: Environment Infrastructure Semantics

B.2.16.5.1 Entities

Infrastructure_Capability

The ability to provide information on Infrastructure_Features, Feature_Relationships and Reference_Relationships, and the ability to determine such relationships.

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_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.

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.

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 and helicopters.

Query

A request to determine a set of properties relating to an Infrastructure_Feature, a Feature_Relationship, or a Reference_Relationship.

Supporting_Information

Information used to determine the response to a Query, e.g. information about the operating environment, or relating to a Reference_Item.

B.2.16.6 Design Rationale

B.2.16.6.1 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.

B.2.16.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.16.6.3 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.

B.2.16.6.4 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 unmanned aircraft around a TOA.

The component is not expected to directly implement security enforcing functions but relies on the integrity of external data sources.

B.2.16.7 Services

B.2.16.7.1 Service Definitions

B.2.16.7.1.1 Infrastructure_Information

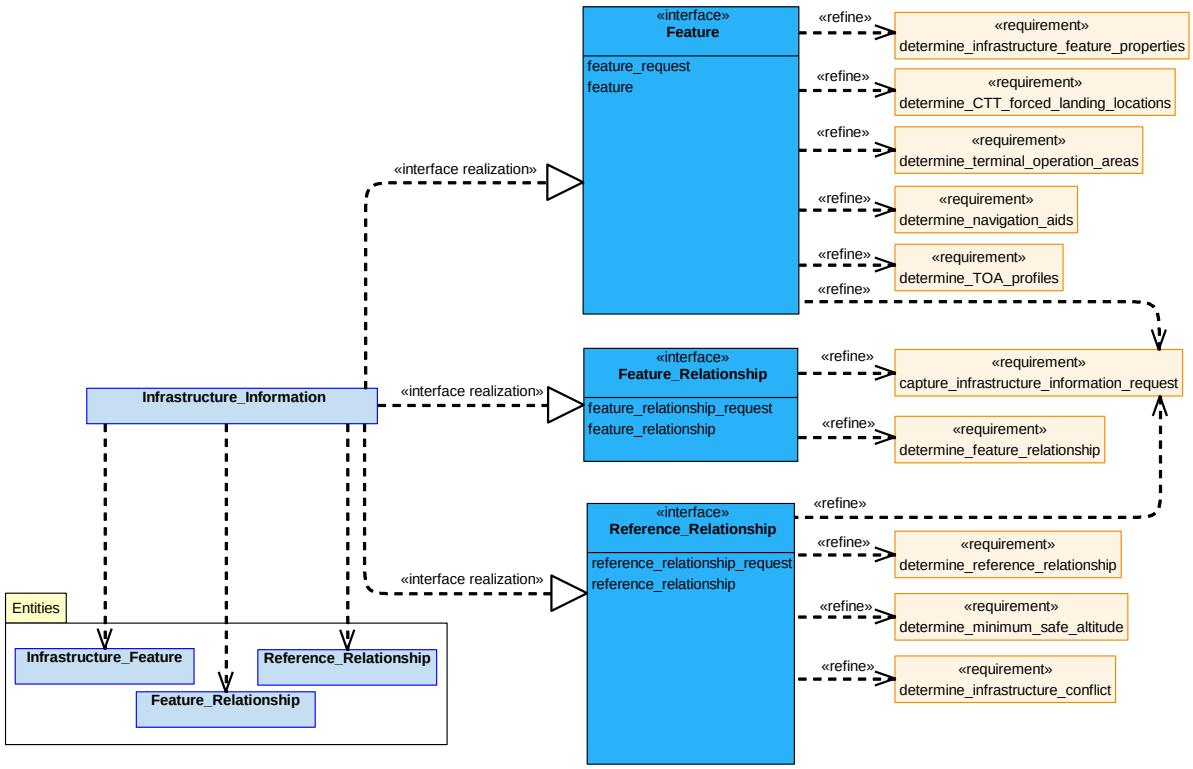

Figure 361: Infrastructure_Information Service Definition

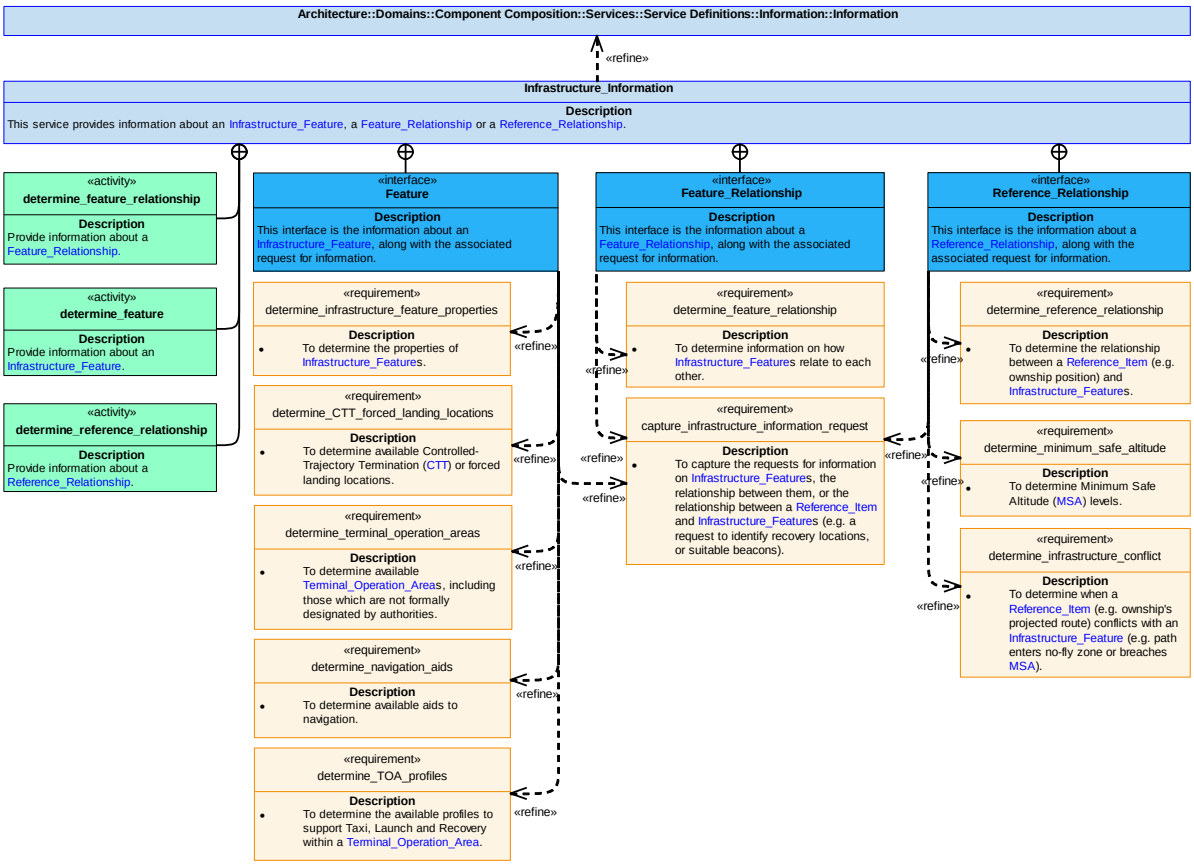

Figure 362: Infrastructure_Information Service Policy

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.

B.2.16.7.1.2 Supporting_Information

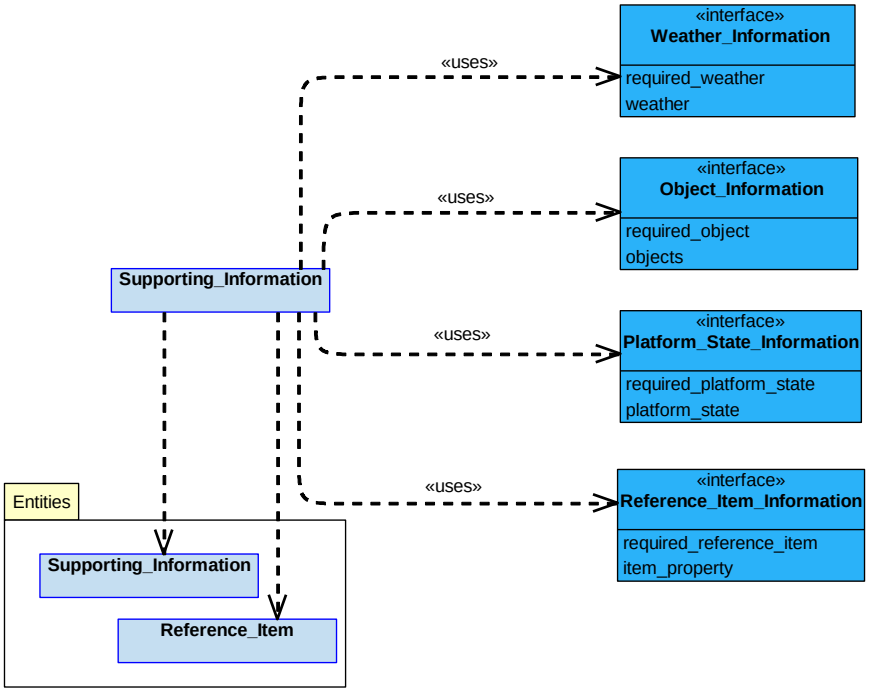

Figure 363: Supporting_Information Service Definition

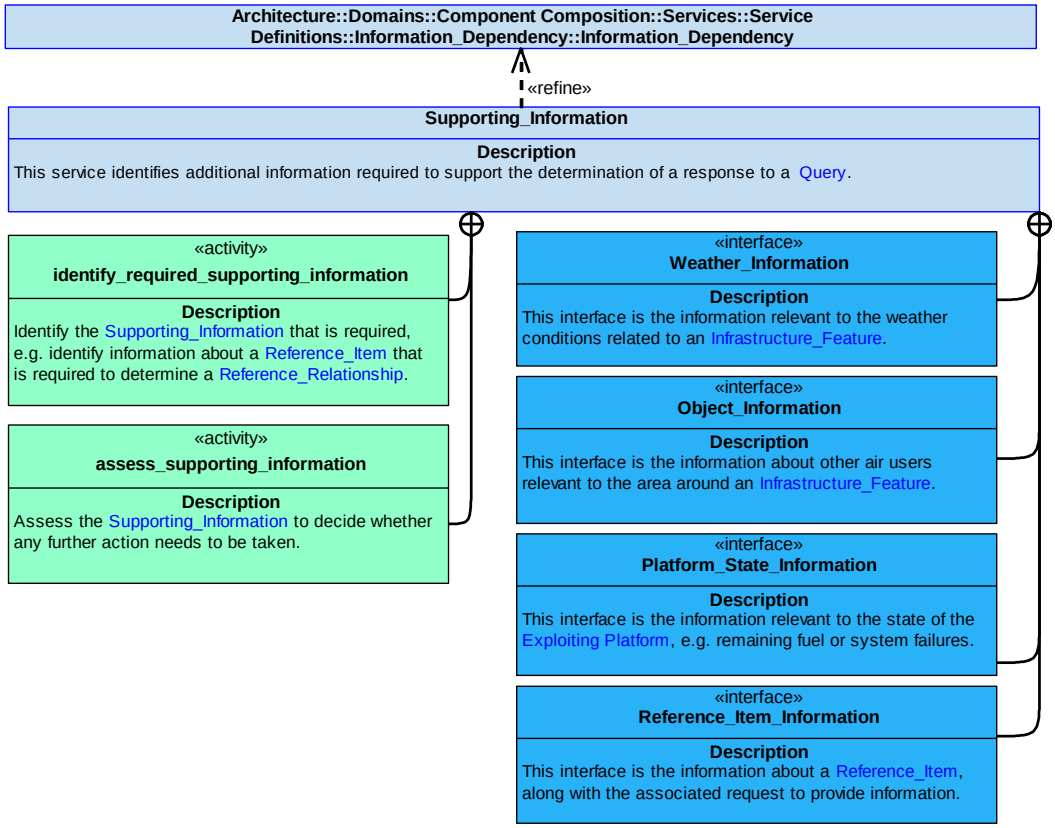

Figure 364: Supporting_Information Service Policy

Supporting_Information

This service identifies additional information required to support the determination of a response to a Query.

Interfaces

Weather_Information

This interface is the information relevant to the weather conditions related to an Infrastructure_Feature.

Attributes

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.

B.2.16.7.1.3 Capability

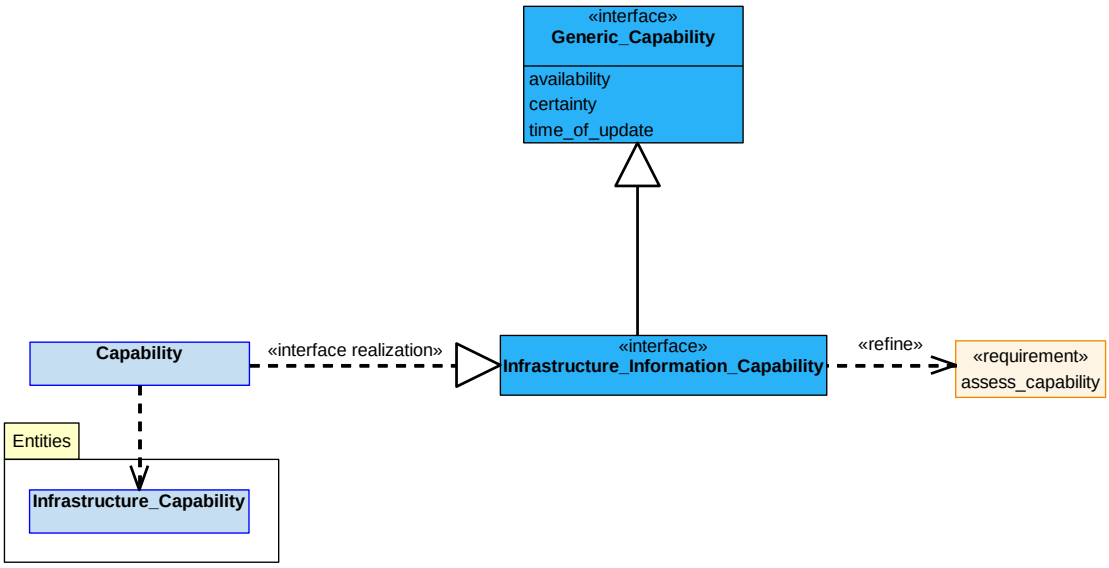

Figure 365: Capability Service Definition

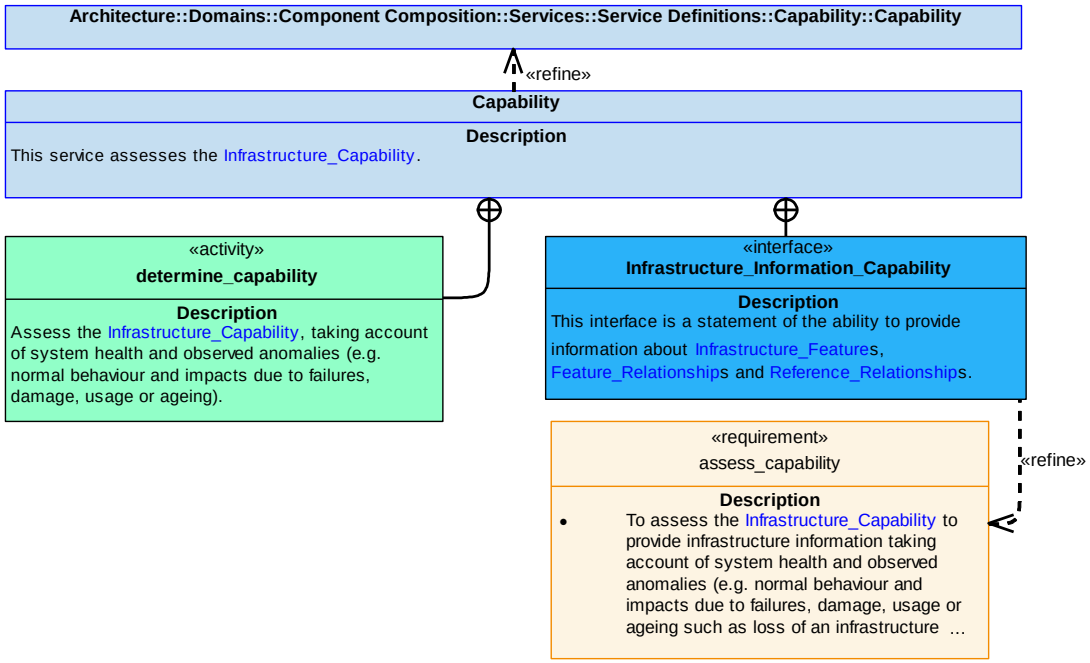

Figure 366: Capability Service Policy

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).

B.2.16.7.1.4 Capability_Evidence

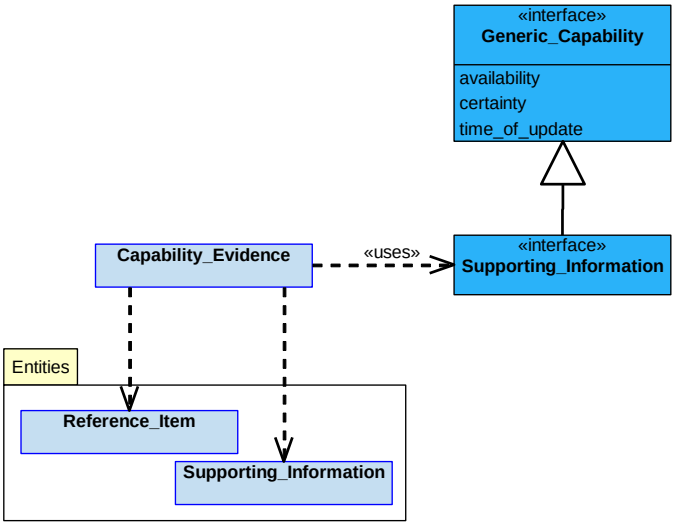

Figure 367: Capability_Evidence Service Definition

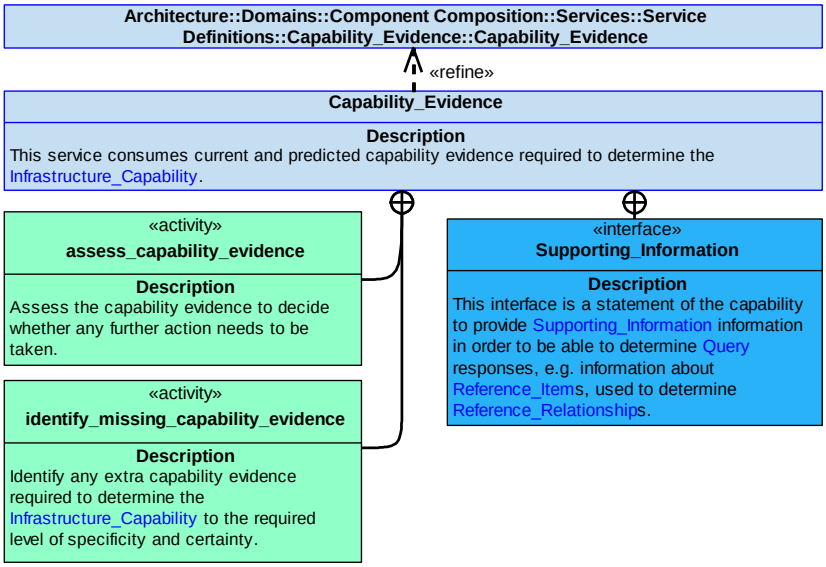

Figure 368: Capability_Evidence Service Policy

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 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.

B.2.16.7.2 Service Dependencies

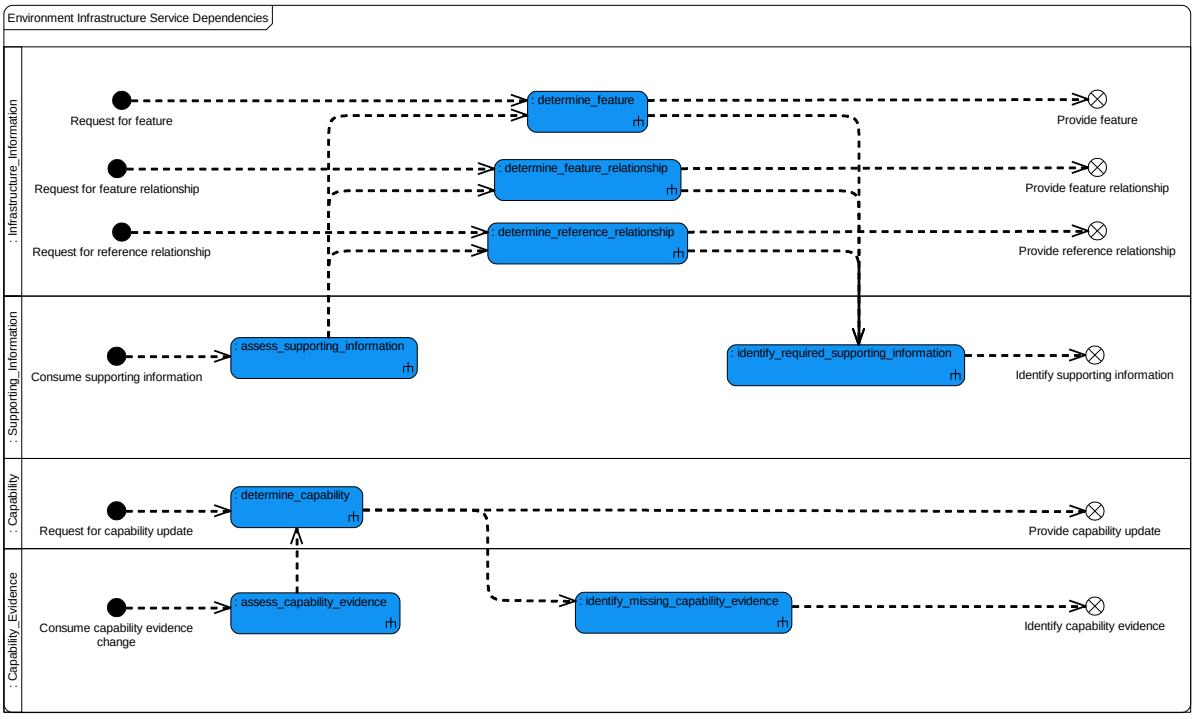

Figure 369: Environment Infrastructure Service Dependencies

B.2.17 Environment Integration

B.2.17.1 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.

B.2.17.2 Overview

Control Architecture

Environment Integration is an action component as defined in the Control Architecture policy.

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.

B.2.17.3 Service Summary

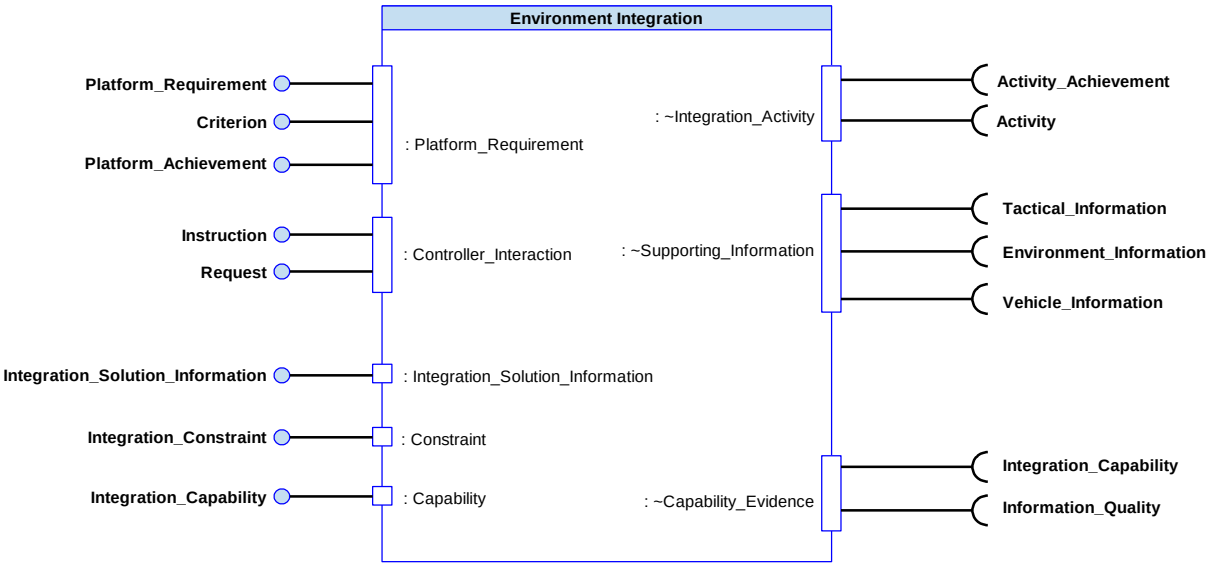

Figure 370: Environment Integration Service Summary

B.2.17.4 Responsibilities

capture_requirements_for_environment_integration_actions

- To capture provided Requirements (e.g. coordinate ATC transition) for environment integration actions.

capture_environment_integration_constraints

- To capture provided Constraints for environment integration actions.

determine_applicable_environment_integration_rules

- To determine the currently applicable Protocols for integrating into the operating environment.

determine_environment_integration_solution

- To determine an Integration_Solution that meets the given Requirements within provided Constraints using the available System_Capability.

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).

determine_integration_settings

- To determine the Integration_Settings for the operating environment.

coordinate_interactions_with_controlling_service

- To generate and interpret automated interactions with the Controlling_Service.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Integration_Capability and Constraints.

capture_measurement_criteria_for_environment_integration_actions

- To capture provided Measurement_Criterion/criteria that an Integration_Solution and its Outcomes will be measured against.

coordinate_environment_integration_solution

- To coordinate the execution of an Integration_Solution.

determine_predicted_quality_of_environment_integration_solution

- To determine the predicted quality of the Integration_Solution against given Measurement_Criterion/criteria.

determine_actual_quality_of_environment_integration_deliverables

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

identify_environment_integration_pre_conditions

- To identify Pre-conditions required to support the Integration_Solution or an Action_Step of the Integration_Solution.

identify_progress_of_environment_integration_solution

- To identify the progress of an Integration_Solution against the Requirements.

predict_capability_progression

- To predict the progression of Environment Integration's Integration_Capability over time and with use.

provide_integration_solution_information

- To provide information relating to an Integration_Solution.

B.2.17.5 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.

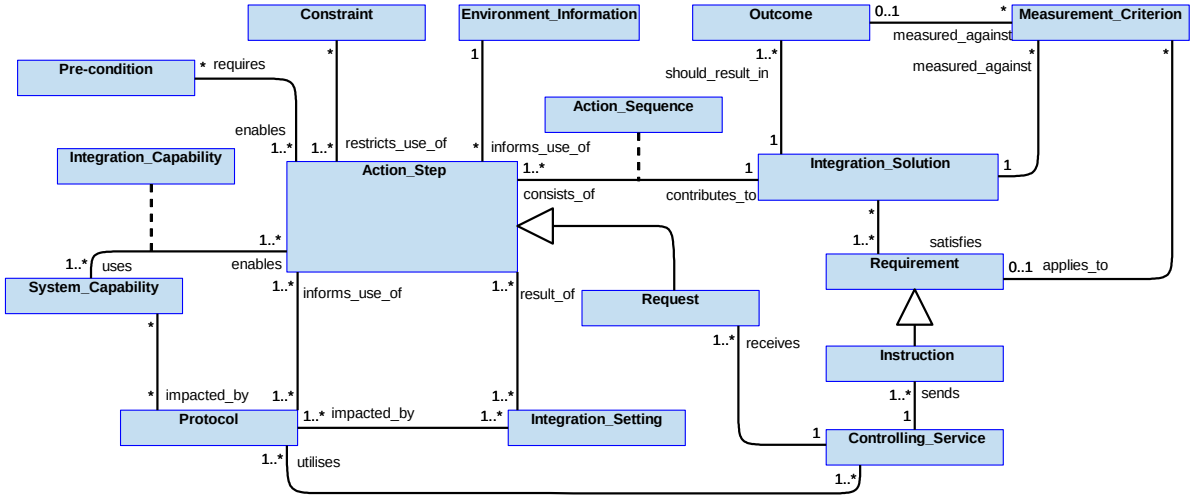

Figure 371: Environment Integration Semantics

B.2.17.5.1 Entities

Action_Sequence

The order in which Action_Steps must be performed to achieve an Integration_Solution.

Action_Step

An activity 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 with, e.g. obstacles, vehicles, 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 (taxi, launch, and 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.

B.2.17.6 Design Rationale

B.2.17.6.1 Assumptions

- Environment Integration does not have knowledge of the structure of the environment.
- Environment Integration does not have knowledge of the communications plan etc. only the protocols necessary for integration with the local environment.
- While Environment Integration may impose a behavioural rule on the Exploiting Platform depending on the current operating environment, it has no knowledge of the current location of the Exploiting Platform.

B.2.17.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Environment Integration:

- Data Driving - This policy 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 instance(s) 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.

B.2.17.6.3 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 unmanned air vehicles - for a manned air vehicle the additional situational awareness of the crew may be sufficient to reduce the DAL.

B.2.17.6.4 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 area). 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.

B.2.17.7 Services

B.2.17.7.1 Service Definitions

B.2.17.7.1.1 Platform_Requirement

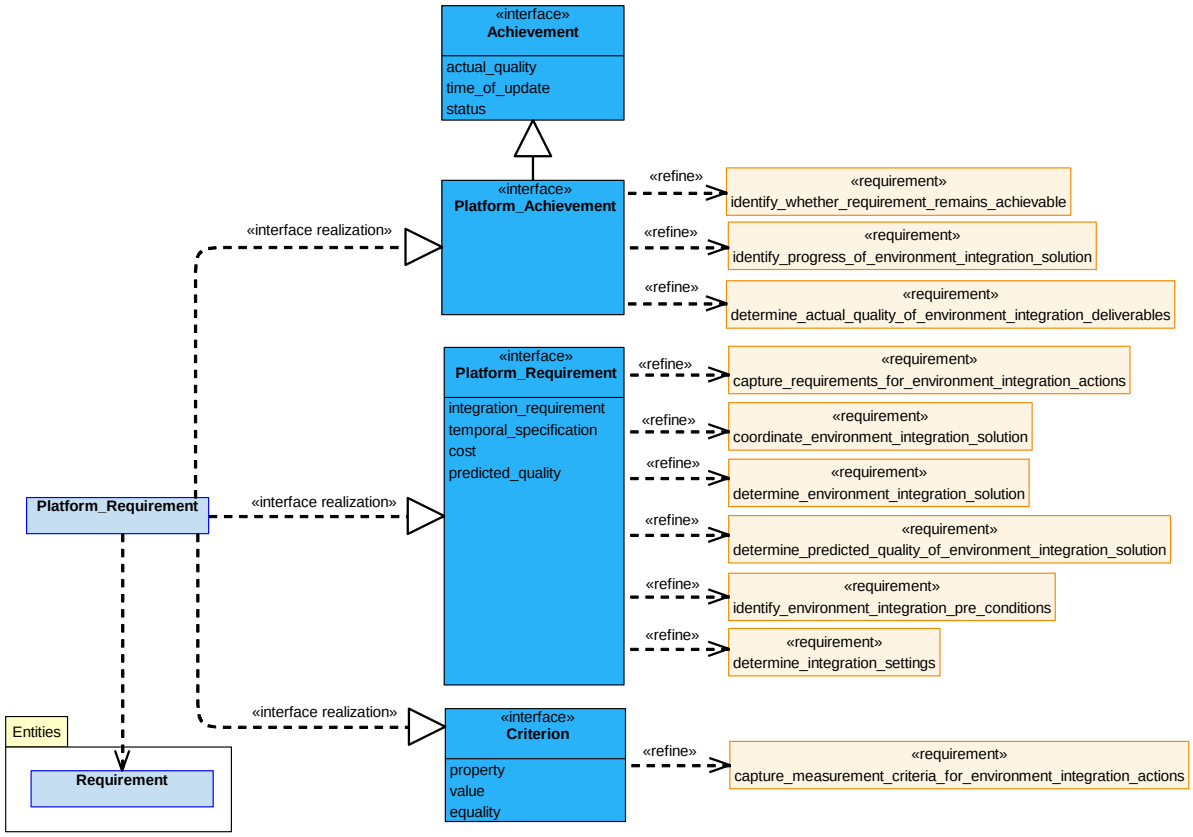

Figure 372: Platform_Requirement Service Definition

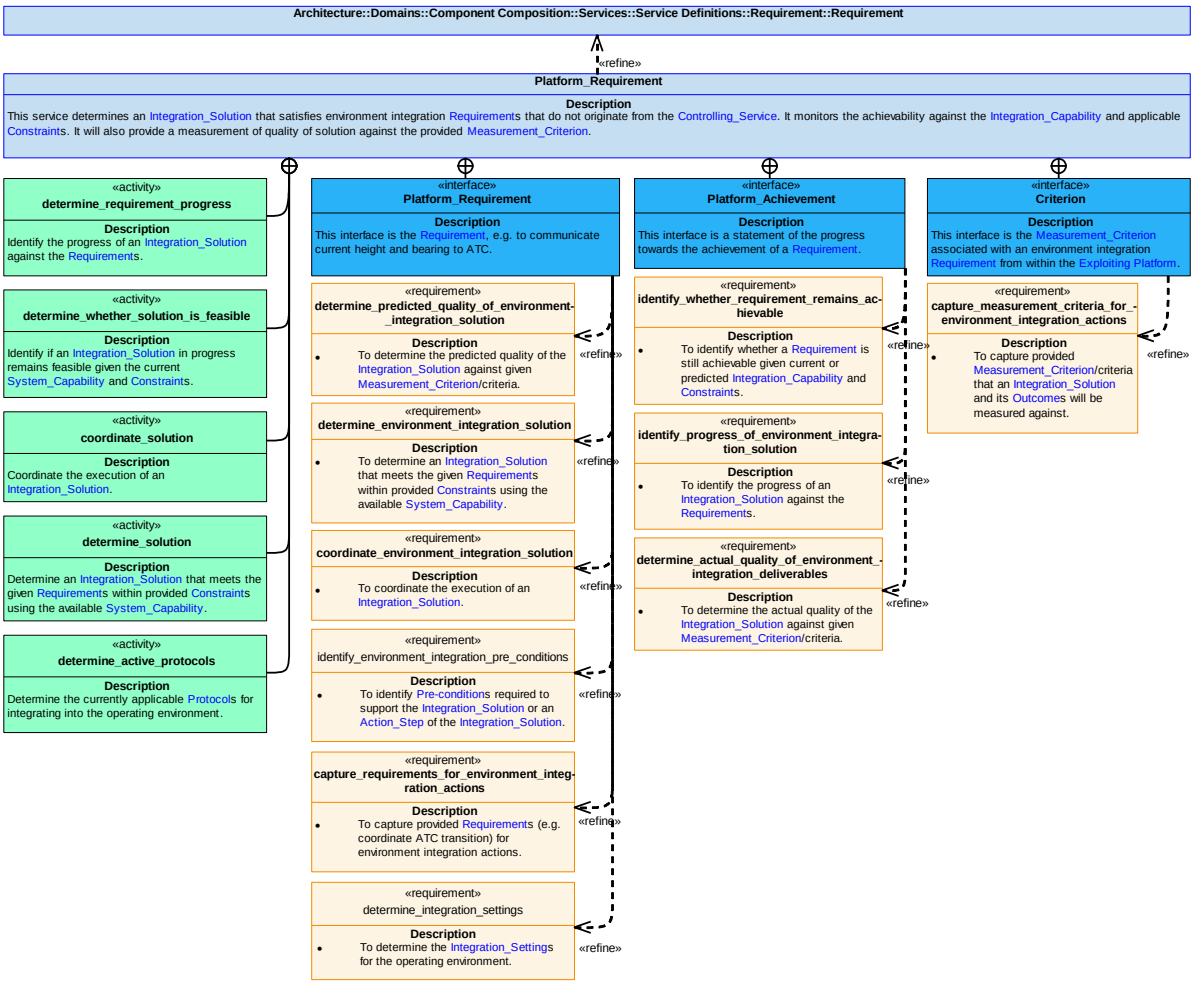

Figure 373: Platform_Requirement Service Policy

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.

B.2.17.7.1.2 Controller_Interaction

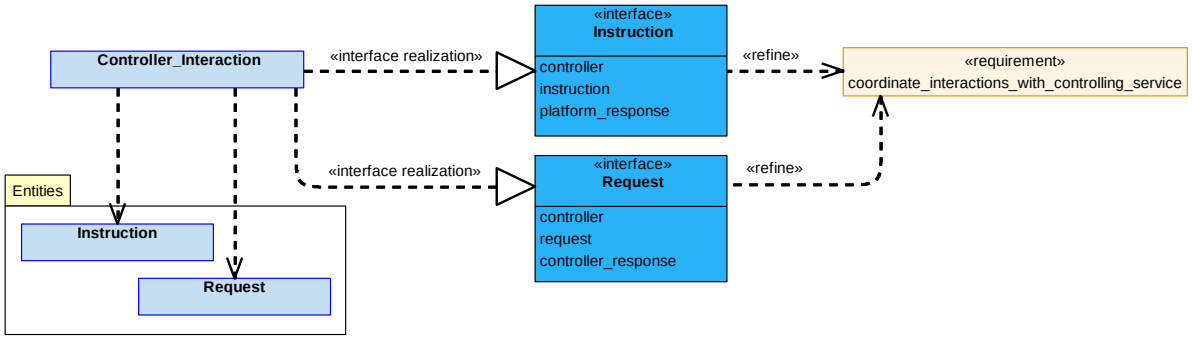

Figure 374: Controller_Interaction Service Definition

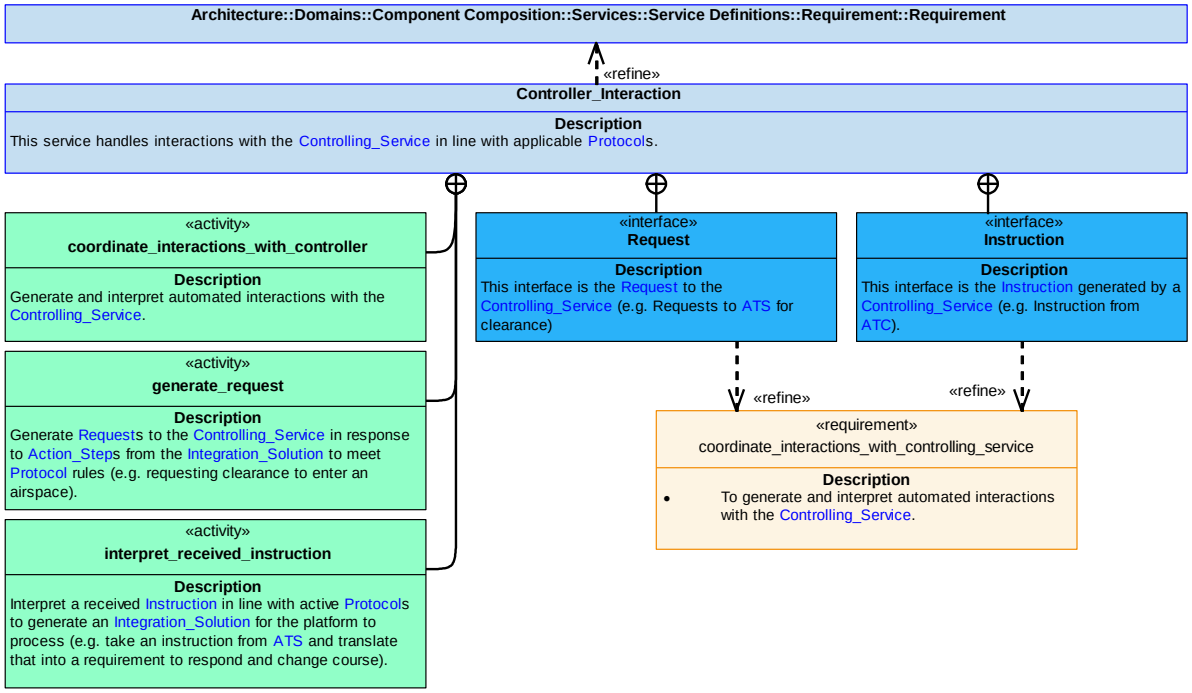

Figure 375: Controller_Interaction Service Policy

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).

B.2.17.7.1.3 Integration_Activity

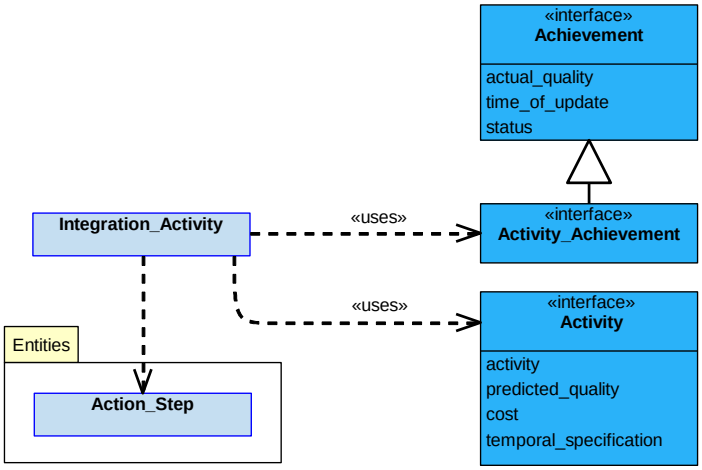

Figure 376: Integration_Activity Service Definition

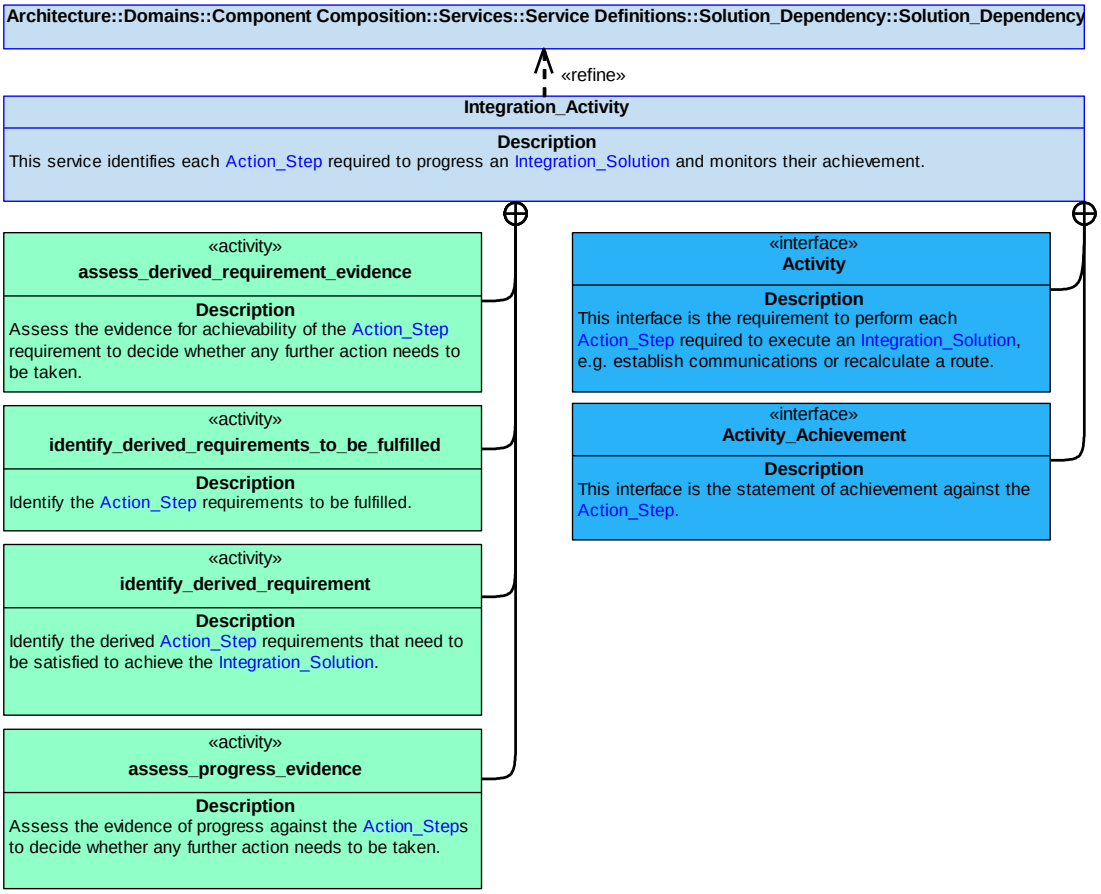

Figure 377: Integration_Activity Service Policy

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.

B.2.17.7.1.4 Integration_Solution_Information

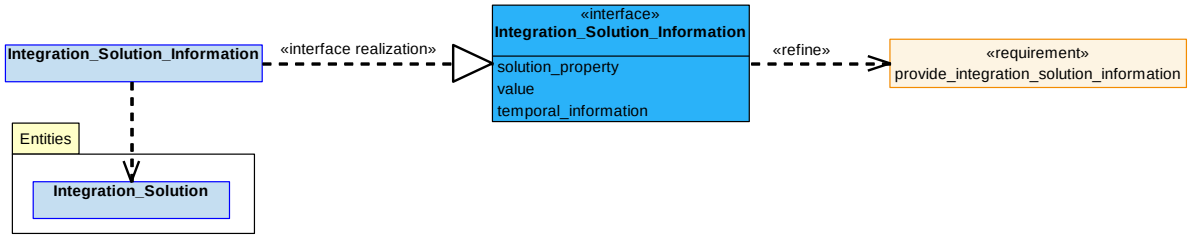

Figure 378: Integration_Solution_Information Service Definition

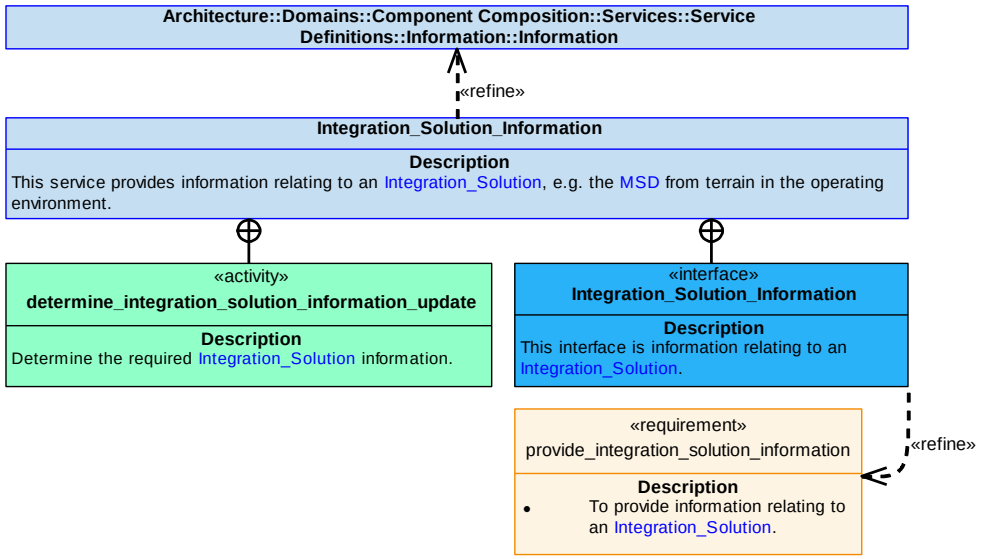

Figure 379: Integration_Solution_Information Service Policy

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.

B.2.17.7.1.5 Supporting_Information

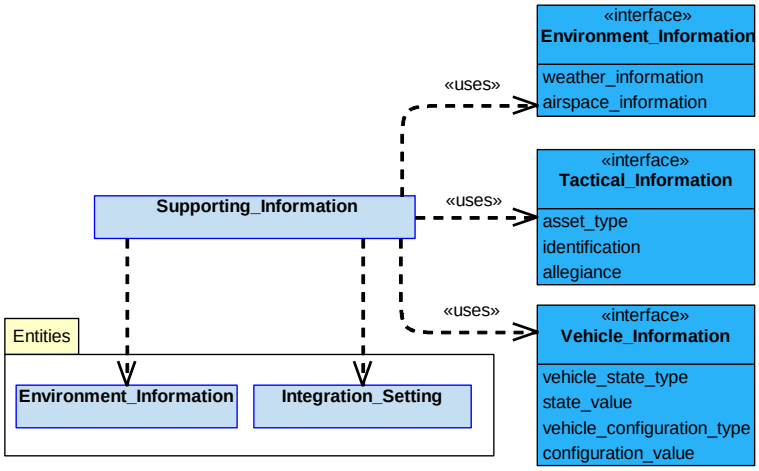

Figure 380: Supporting_Information Service Definition

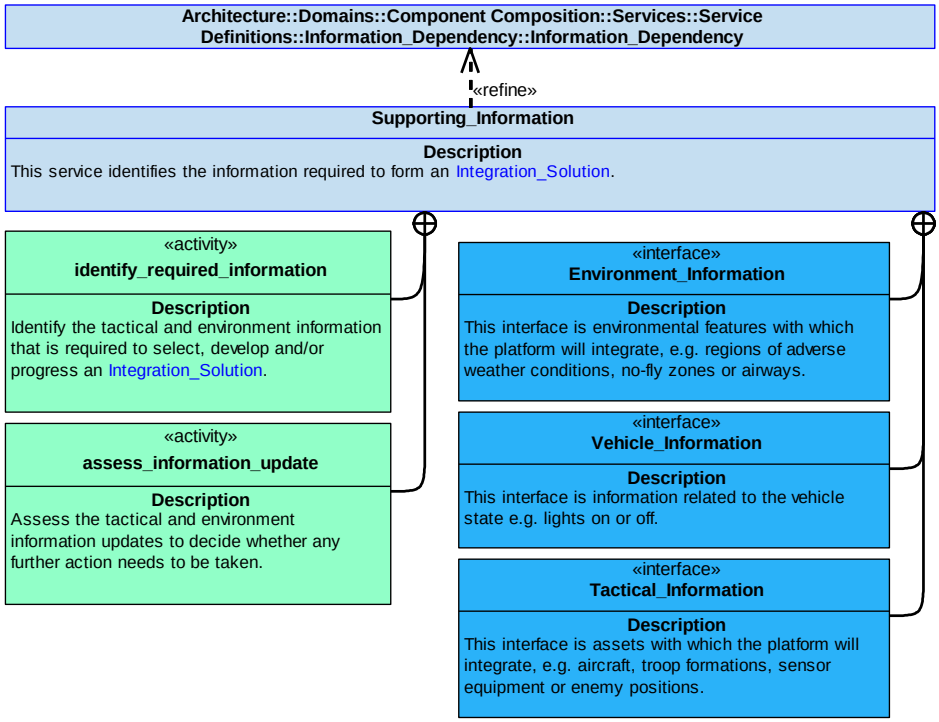

Figure 381: Supporting_Information Service Policy

Supporting_Information

This service identifies the information required to form an Integration_Solution.

Interfaces

Tactical_Information

This interface is assets with which the platform will integrate, e.g. aircraft, troop formations, sensor equipment or enemy positions.

Attributes

asset_type	Asset type, e.g. fuel tanker, cargo plane, enemy jet, 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, Unknown.

Environment_Information

This interface is environmental features with which the platform will integrate, e.g. regions of adverse weather conditions, no-fly zones or airways.

Attributes

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.

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

assess_information_update

Assess the tactical and environment information updates to decide whether any further action needs to be taken.

identify_required_information

Identify the tactical and environment information that is required to select, develop and/or progress an Integration_Solution.

B.2.17.7.1.6 Constraint

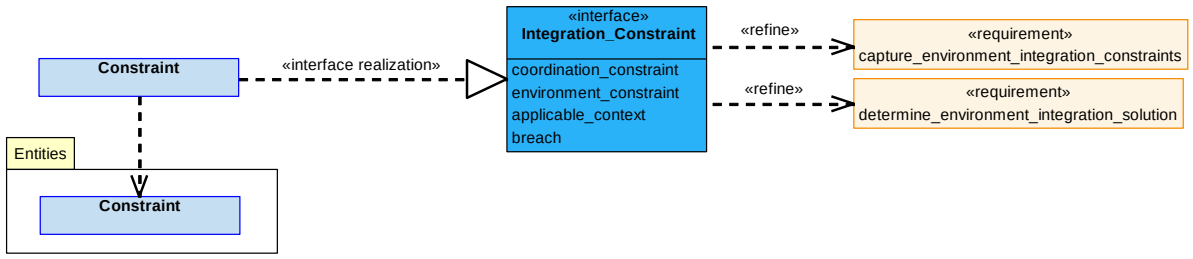

Figure 382: Constraint Service Definition

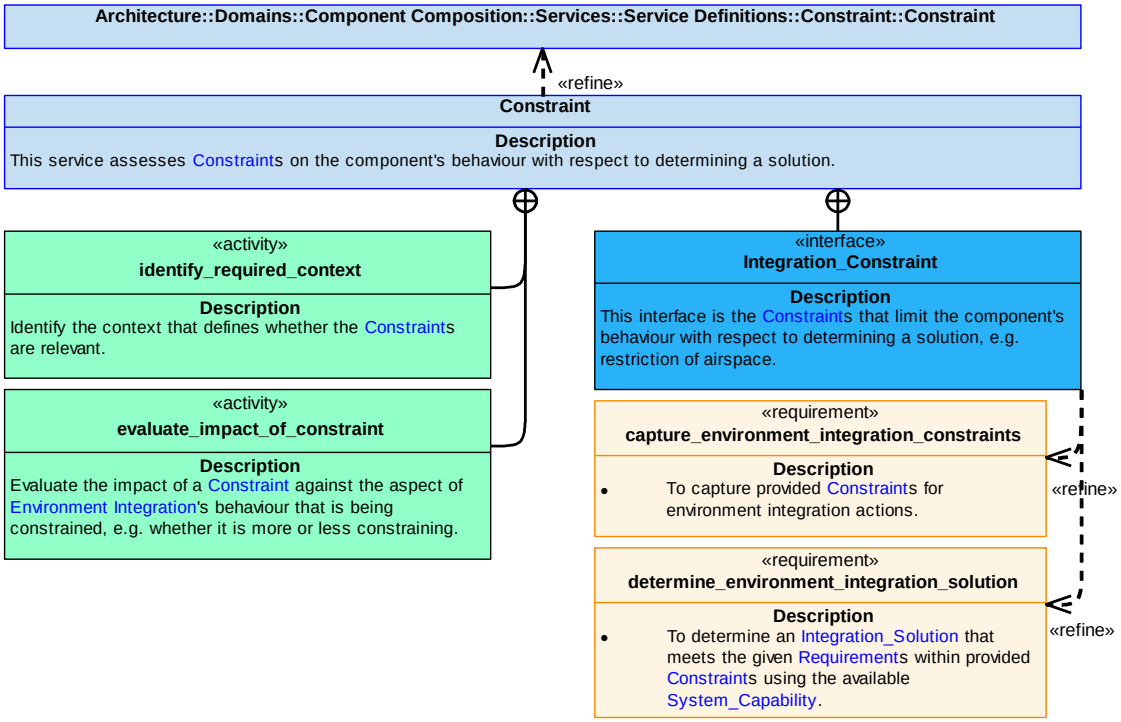

Figure 383: Constraint Service Policy

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.

B.2.17.7.1.7 Capability

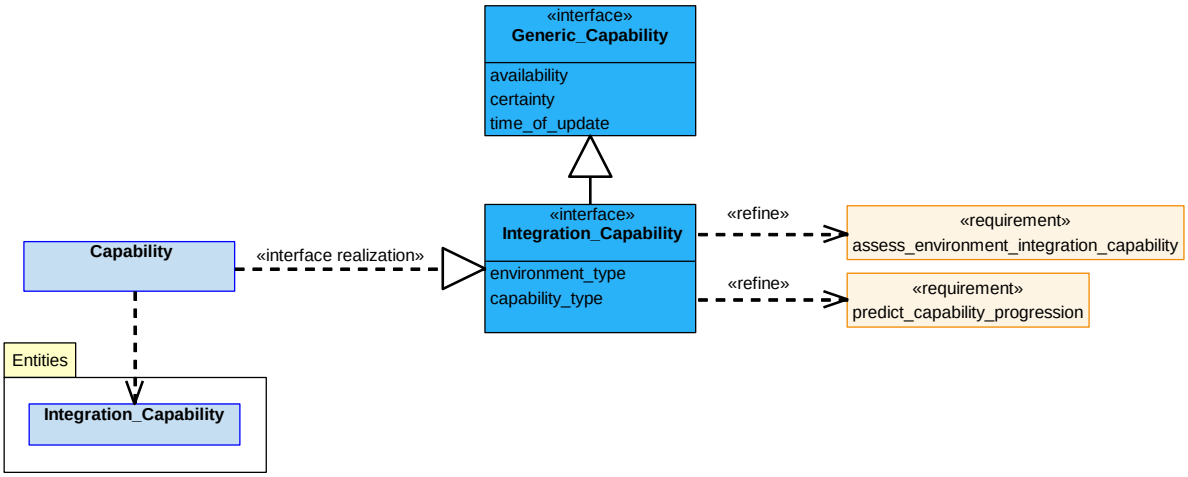

Figure 384: Capability Service Definition

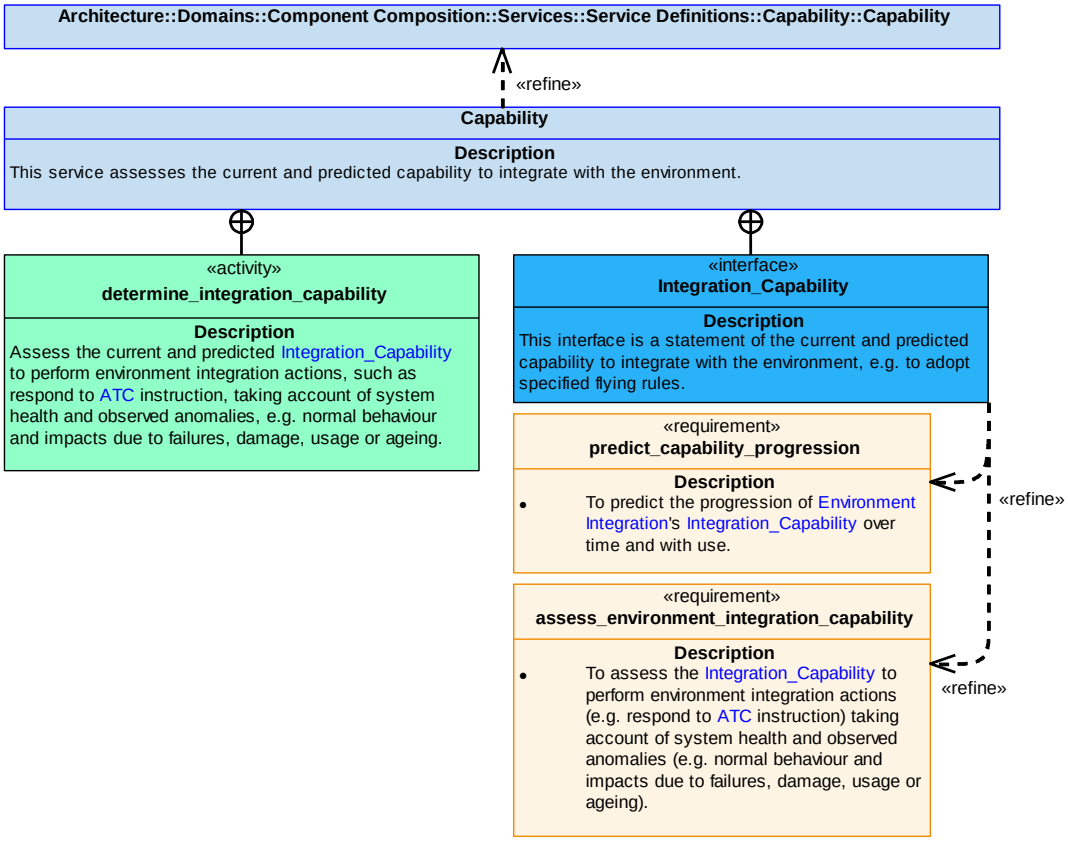

Figure 385: Capability Service Policy

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.

B.2.17.7.1.8 Capability_Evidence

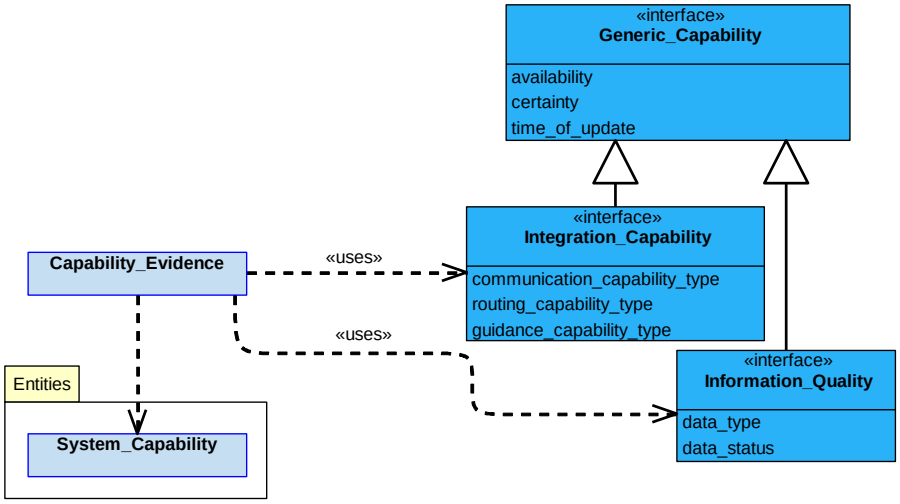

Figure 386: Capability_Evidence Service Definition

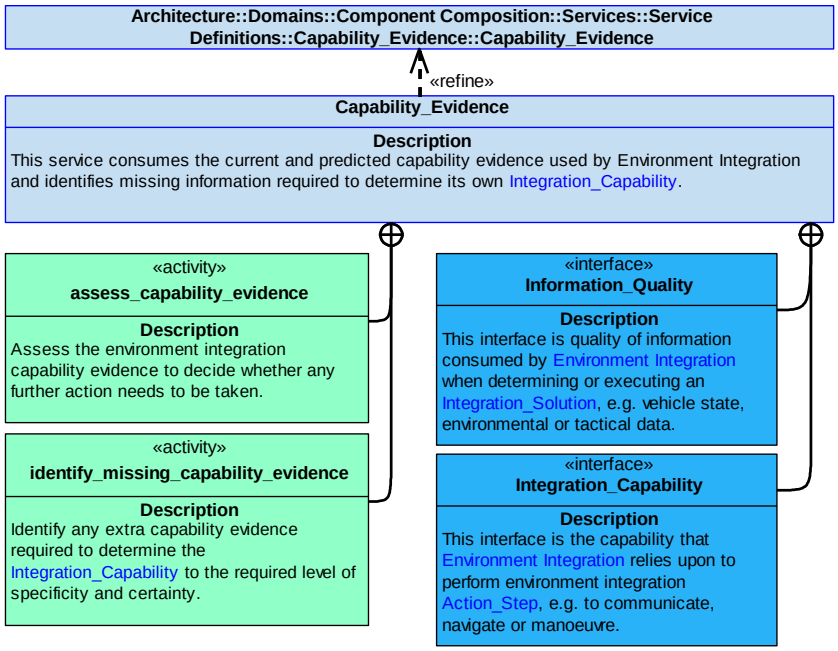

Figure 387: Capability_Evidence Service Policy

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

Information_Quality

This interface is quality of information consumed by Environment Integration when determining or executing an Integration_Solution, e.g. vehicle state, environmental or tactical data.

Attributes

data_type	The type of data being reported on, e.g. airport information, aircraft positions, or internal vehicle state.
data_status	Indication of the quality of the data being provided.

Integration_Capability

This interface is the capability that Environment Integration relies upon to perform environment integration Action_Step, 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 platform speed).

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.

B.2.17.7.2 Service Dependencies

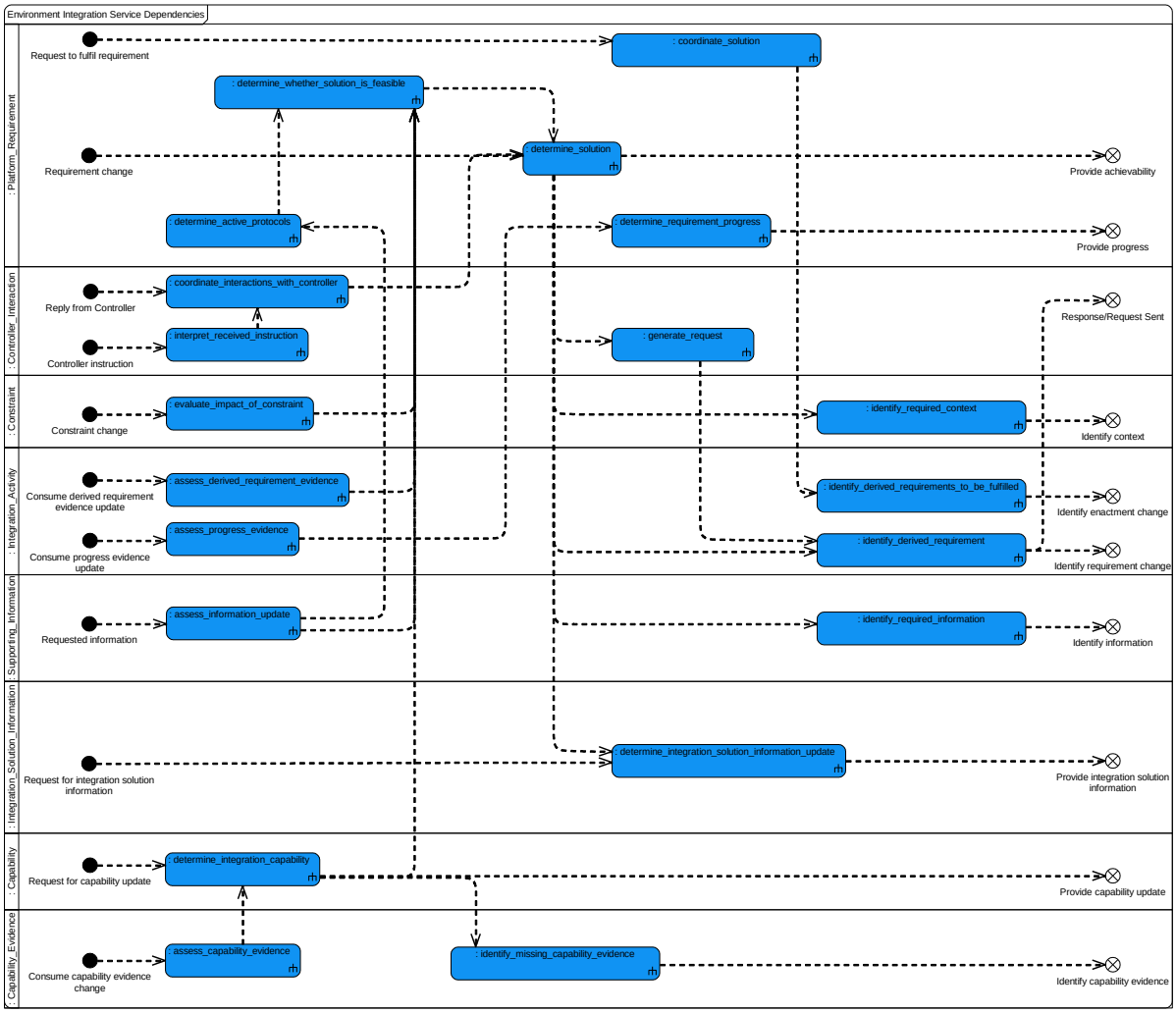

Figure 388: Environment Integration Service Dependencies

B.2.18 Environmental Conditioning

B.2.18.1 Role

The role of Environmental Conditioning is to control the environmental properties of environmental zones.

B.2.18.2 Overview

Control Architecture

Environmental Conditioning is an action component as defined in the Control Architecture policy.

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_Medium used to control those properties (e.g. pressure of engine bleed air), Environmental Conditioning controls Effector_Capabilitys (e.g. valves) 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.

B.2.18.3 Service Summary

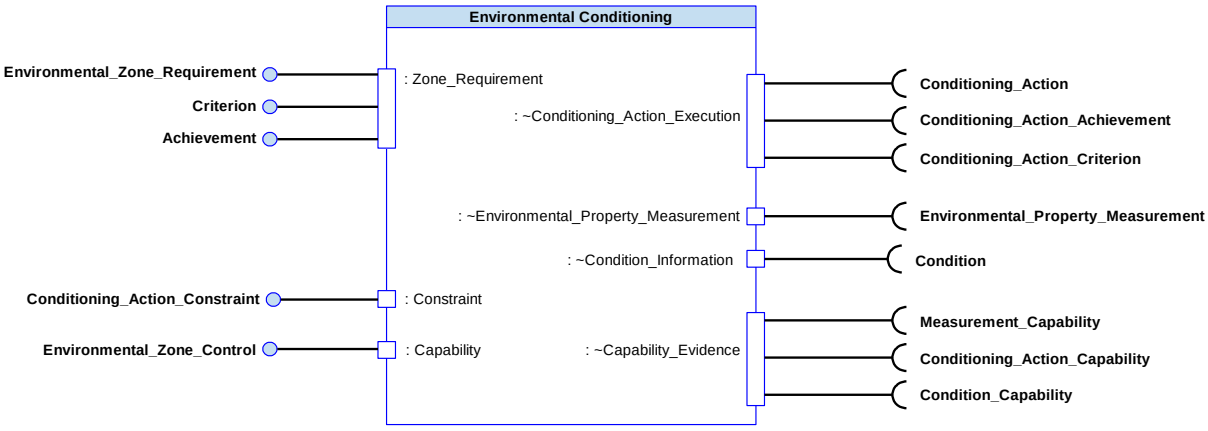

Figure 389: Environmental Conditioning Service Summary

B.2.18.4 Responsibilities

capture_zone_requirements

- To capture given Zone_Requirements for an Environmental_Zone (e.g. temperature range).

capture_constraints

- To capture provided Constraints for Conditioning_Actions (e.g. opening of doors is not permitted).

capture_measurement_criteria

- To capture each provided Measurement_Criterion for Conditioning_Actions (e.g. response time).

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).

predict_capability_progression

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

identify_missing_information

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

determine_solution

- To determine a Solution that meets Zone_Requirements and Constraints using a Conditioning_Medium.

identify_pre-conditions

- To identify Pre-conditions required to support a Solution or a step of that solution.

determine_quality_of_solution

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

coordinate_solution

- To coordinate the execution of a Solution via the use of Effector_Capability.

determine_quality_of_deliverables

- To determine the quality of the Environmental_Property/properties controlled by executing a Solution, measured against given Zone_Requirements and Measurement_Criterion/criteria.

identify_progress_of_solution

- To identify the progress of a Solution against the Zone_Requirements.

identify_solution_in_progress_remains_feasible

- To identify whether a Solution in progress remains feasible given current resources.

B.2.18.5 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.

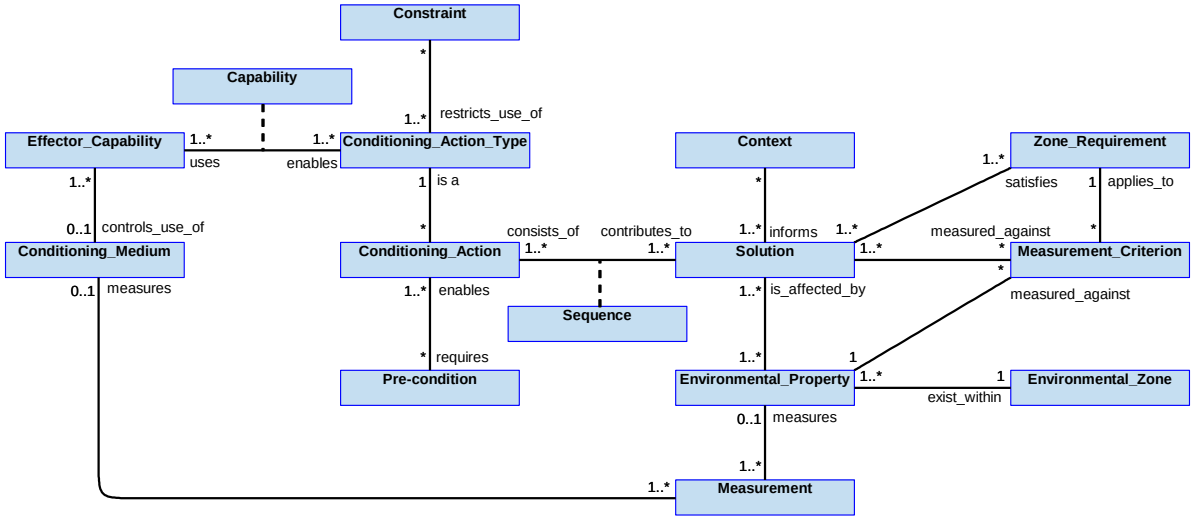

Figure 390: Environmental Conditioning Semantics

B.2.18.5.1 Entities

Capability

The capability to execute Conditioning_Actions via the control of a Conditioning_Medium.

Conditioning_Action

An action that affects an Environmental_Property (or properties) of an Environmental_Zone.

Conditioning_Medium

A physical resource used as the medium to provide Conditioning_Actions. It could be used for other purposes, e.g. engine bleed air being used to provide heating or cooling.

Sequence

The order in which Conditioning_Actions must be performed to achieve a Solution.

Solution

A solution to satisfy one or more Zone_Requirements through the use of Conditioning_Actions.

Conditioning_Action_Type

A type of Conditioning_Action, e.g. a heating, cooling or de-humidifying action.

Constraint

A constraint on the execution of Conditioning_Actions. The constraint could be a limitation on a specific Effector_Capability or could be a wider constraint relating to the Solution.

Context

Information that Environmental Conditioning needs to know to determine a solution, e.g. information relating to current operating conditions or vehicle configuration.

Effector_Capability

The capability by which a Conditioning_Medium is controlled in order to execute Conditioning_Actions.

Environmental_Property

A property of the physical environment (e.g. temperature or pressure) that is required to be controlled.

Environmental_Zone

An area, volume or region whose Environmental_Property (or properties) are required to be controlled, e.g. cargo bay, UAV Control System cabin or aerofoil surface.

Measurement

A sensor measurement. This includes data on the measurement quality and may include data on the capability to provide measurement information.

Measurement_Criterion

A criterion used to evaluate the effectiveness of a Conditioning_Action on an Environmental_Property.

Pre-condition

A condition that must be true before a Conditioning_Action can take place.

Zone_Requirement

A requirement to control environmental aspects of an Environmental_Zone.

B.2.18.6 Design Rationale

B.2.18.6.1 Assumptions

- Environmental Conditioning can be used to control any type of Environmental_Property, not just temperature.

B.2.18.6.2 Design Considerations

Directly Applicable Policies

These policies 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 Effector_Capabilitys are likely to require a significant proportion of available system resources such as electrical power and the Conditioning_Medium used to provide environmental conditioning 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.

B.2.18.6.3 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.

B.2.18.6.4 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).

B.2.18.7 Services

B.2.18.7.1 Service Definitions

B.2.18.7.1.1 Zone_Requirement

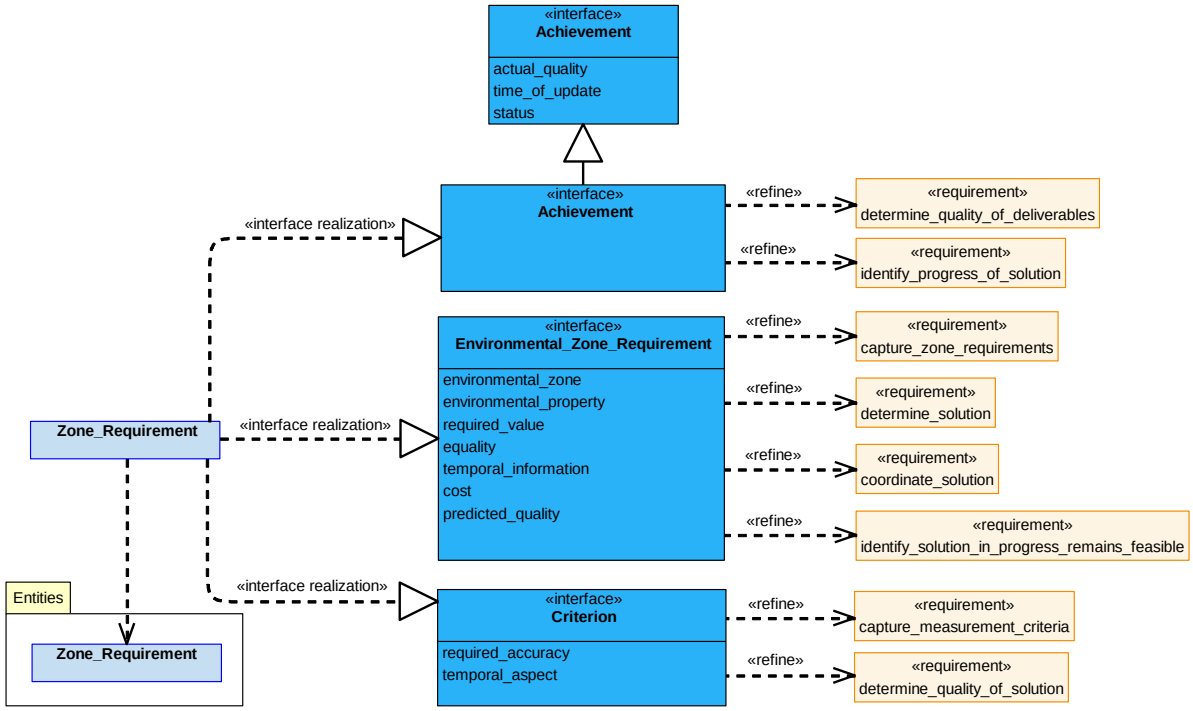

Figure 391: Zone_Requirement Service Definition

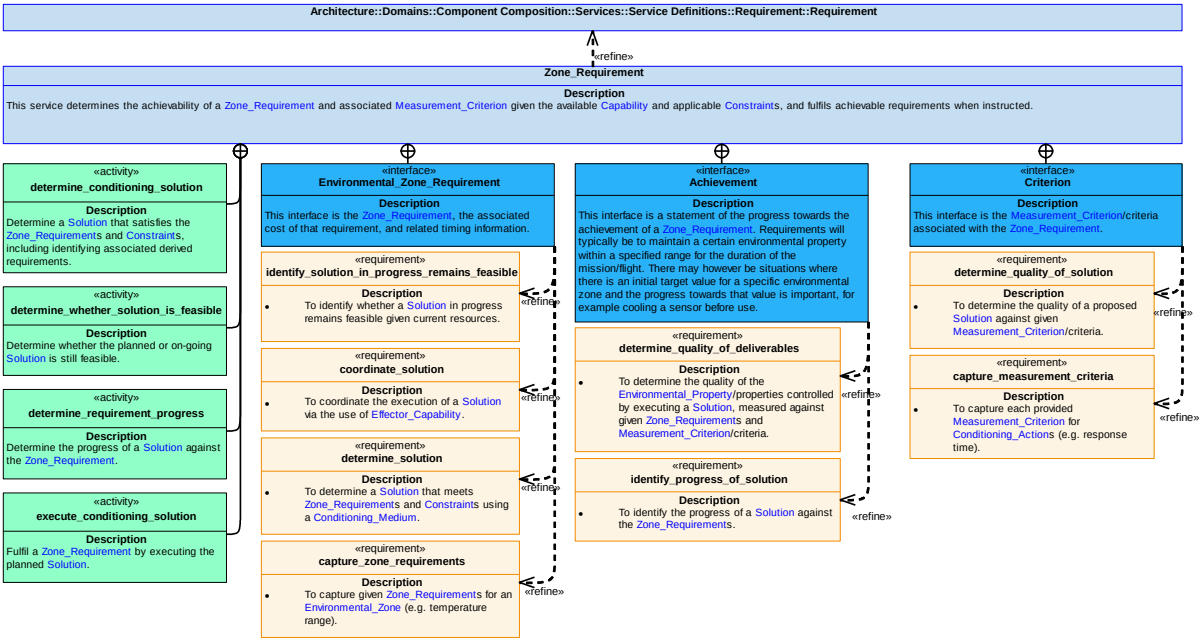

Figure 392: Zone_Requirement Service Policy

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 the Solution, e.g. resources used or time taken.
predicted_quality	How well the planned Solution 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 Solution, e.g. the amount of time that a value strays outside of a desired range.

Activities

determine_requirement_progress

Determine the progress of a Solution against the Zone_Requirement.

determine_conditioning_solution

Determine a Solution that satisfies the Zone_Requirements and Constraints, including identifying associated derived requirements.

execute_conditioning_solution

Fulfil a Zone_Requirement by executing the planned Solution.

determine_whether_solution_is_feasible

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

B.2.18.7.1.2 Conditioning_Action_Execution

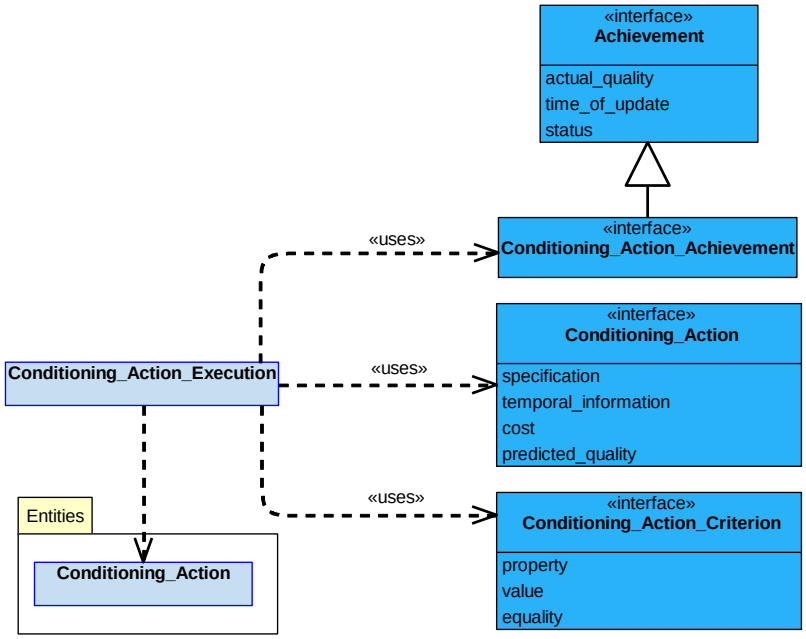

Figure 393: Conditioning_Action_Execution Service Definition

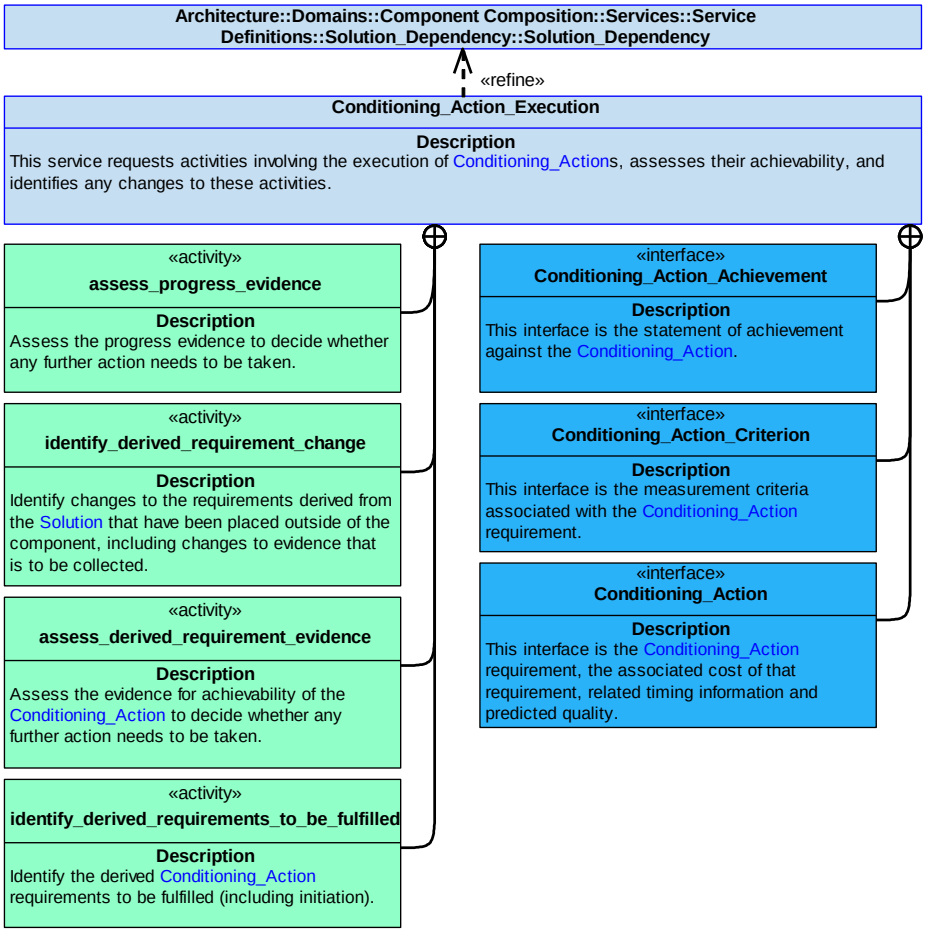

Figure 394: Conditioning_Action_Execution Service Policy

Conditioning_Action_Execution

This service requests activities involving the execution of Conditioning_Actions, assesses their achievability, and identifies any changes to these activities.

Interfaces

Conditioning_Action

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

Attributes

specification	The definition of the Conditioning_Action requirement.
temporal_information	Information covering timing, for example the length of time required to keep an aperture open.
cost	The cost of executing the solution, for example: resources used or time taken.
predicted_quality	How well the planned Conditioning_Action solution is predicted to satisfy the requirement.

Conditioning_Action_Criterion

This interface is the measurement criteria associated with the Conditioning_Action requirement.

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 Conditioning_Action.

Activities

assess_derived_requirement_evidence

Assess the evidence for achievability of the Conditioning_Action 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 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 Conditioning_Action requirements to be fulfilled (including initiation).

B.2.18.7.1.3 Environmental_Property_Measurement

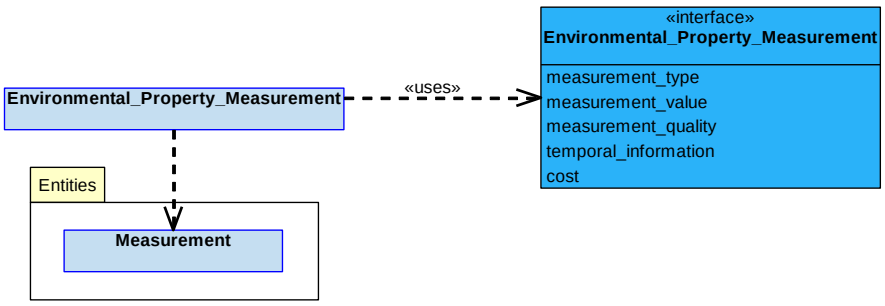

Figure 395: Environmental_Property_Measurement Service Definition

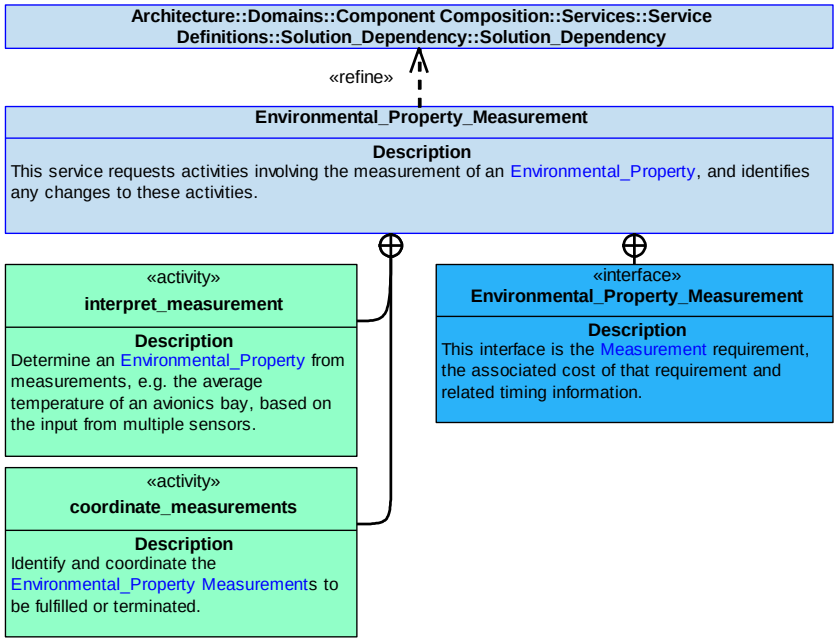

Figure 396: Environmental_Property_Measurement Service Policy

Environmental_Property_Measurement

This service requests activities involving the measurement of an Environmental_Property, and identifies any changes to these activities.

Interface

Environmental_Property_Measurement

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

Attributes

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.

B.2.18.7.1.4 Condition_Information

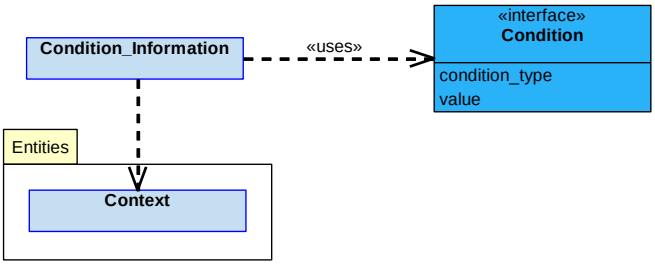

Figure 397: Condition_Information Service Definition

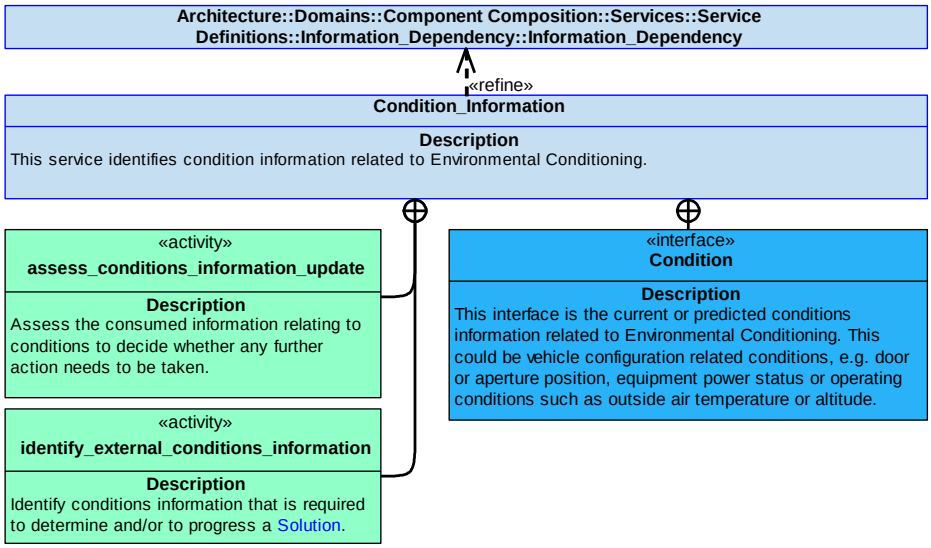

Figure 398: Condition_Information Service Policy

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 Solution.

B.2.18.7.1.5 Constraint

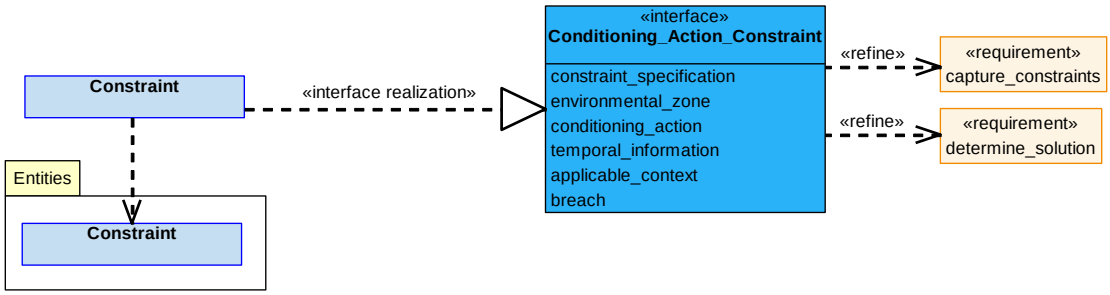

Figure 399: Constraint Service Definition

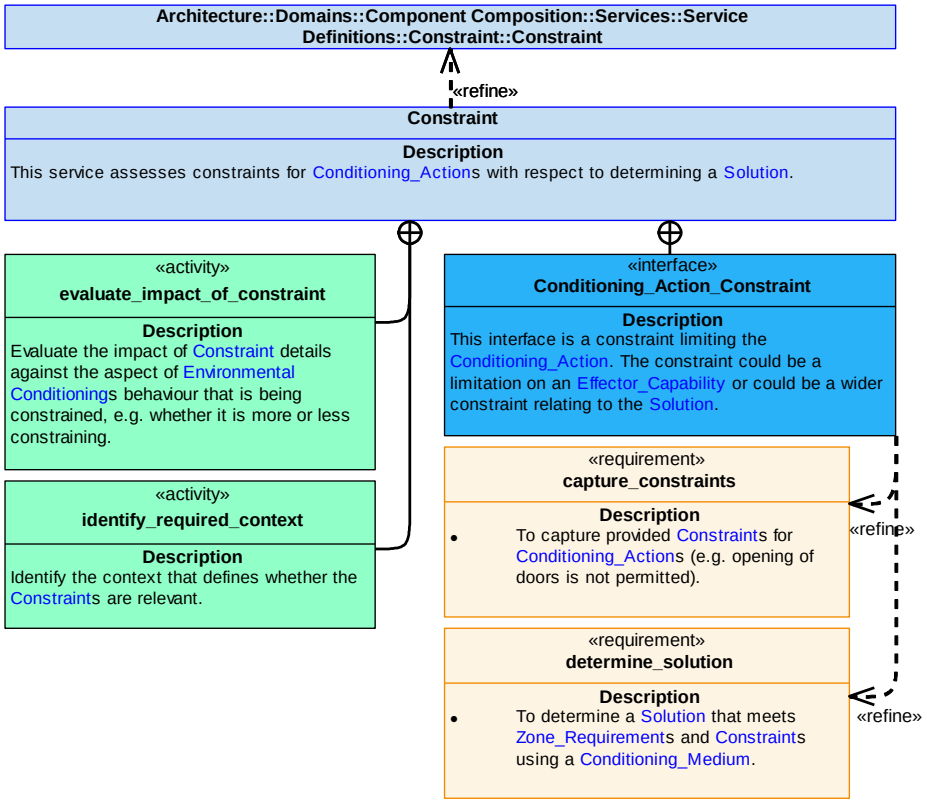

Figure 400: Constraint Service Policy

Constraint

This service assesses constraints for Conditioning_Actions with respect to determining a Solution.

Interface

Conditioning_Action_Constraint

This interface is a constraint limiting the Conditioning_Action. The constraint could be a limitation on an Effector_Capability or could be a wider constraint relating to the Solution.

Attributes

constraint_specification	A constraint that will restrict the Conditioning_Action.
environmental_zone	The Environmental_Zone (or zones) that the constraint relates to.
conditioning_action	The Conditioning_Action (or actions) that the constraint relates to.
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 Environmental Conditionings 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.

B.2.18.7.1.6 Capability

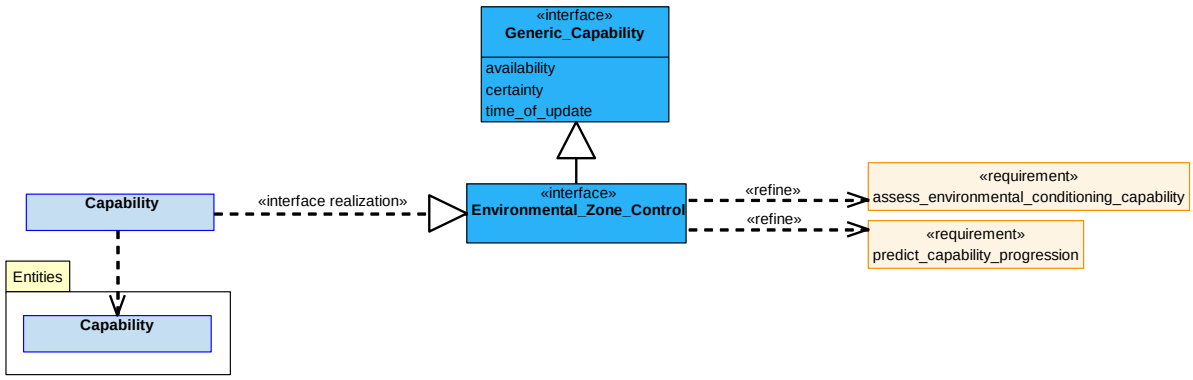

Figure 401: Capability Service Definition

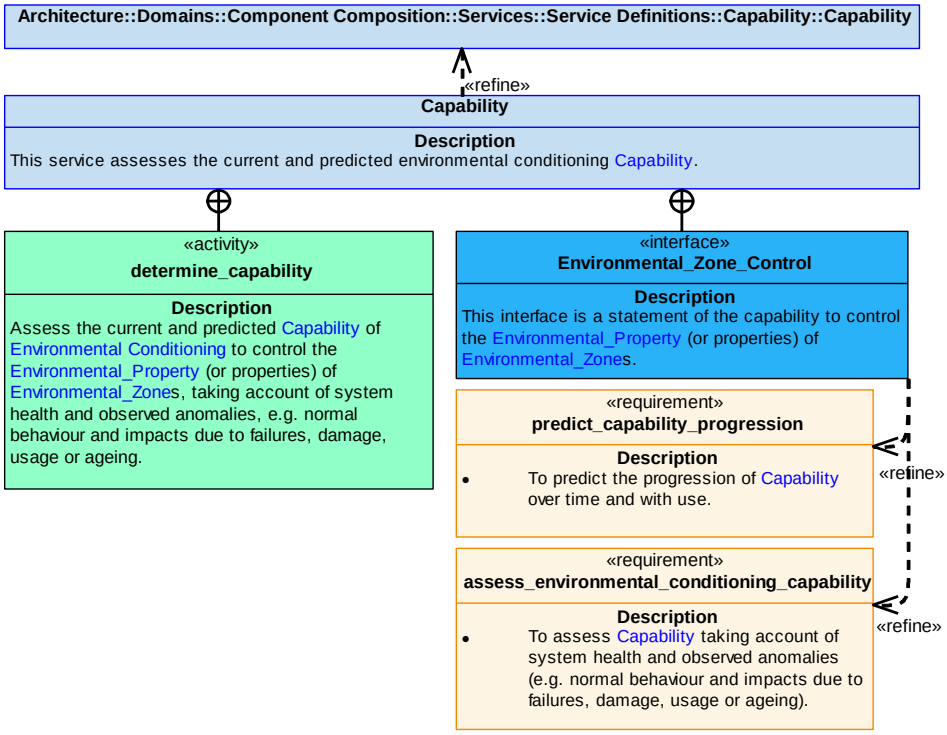

Figure 402: Capability Service Policy

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.

B.2.18.7.1.7 Capability_Evidence

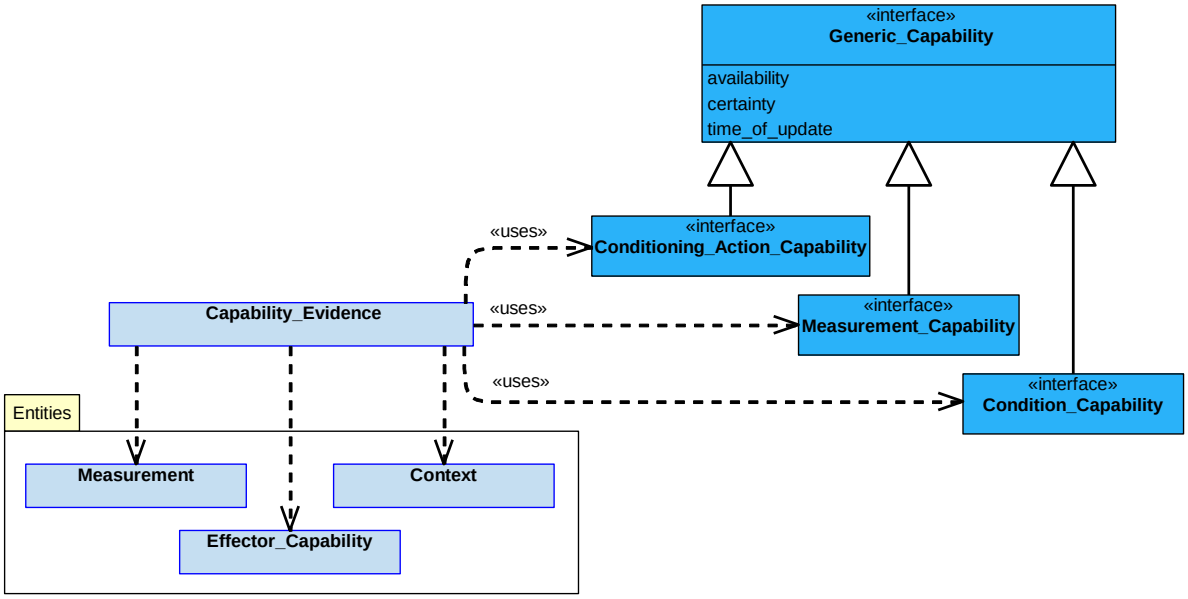

Figure 403: Capability_Evidence Service Definition

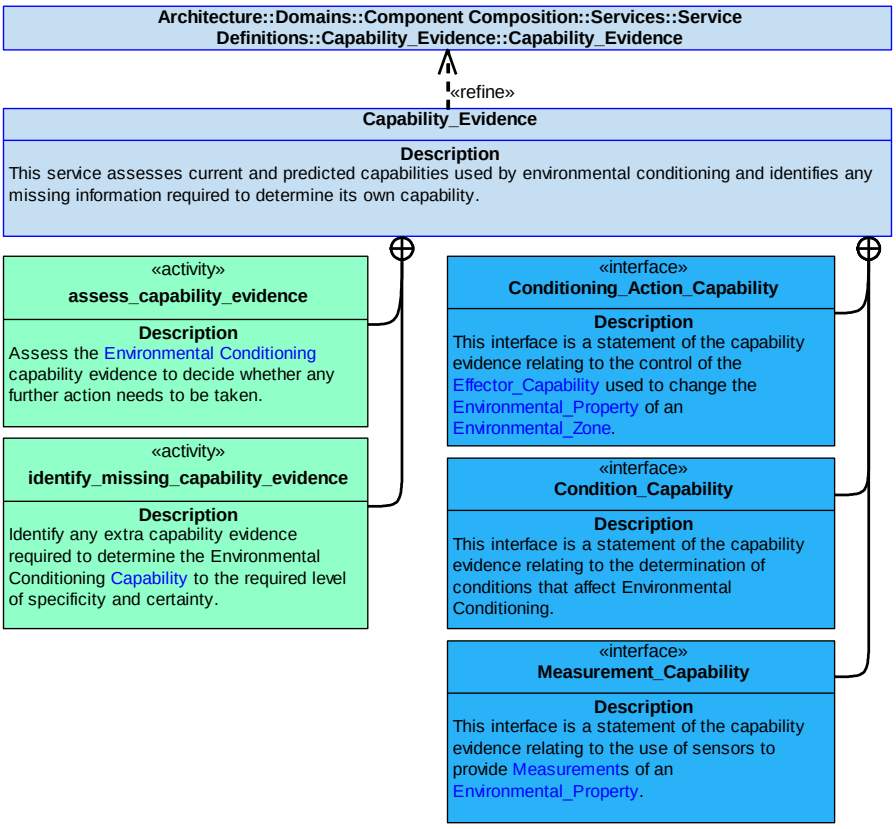

Figure 404: Capability_Evidence Service Policy

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 Effector_Capability 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.

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.

B.2.18.7.2 Service Dependencies

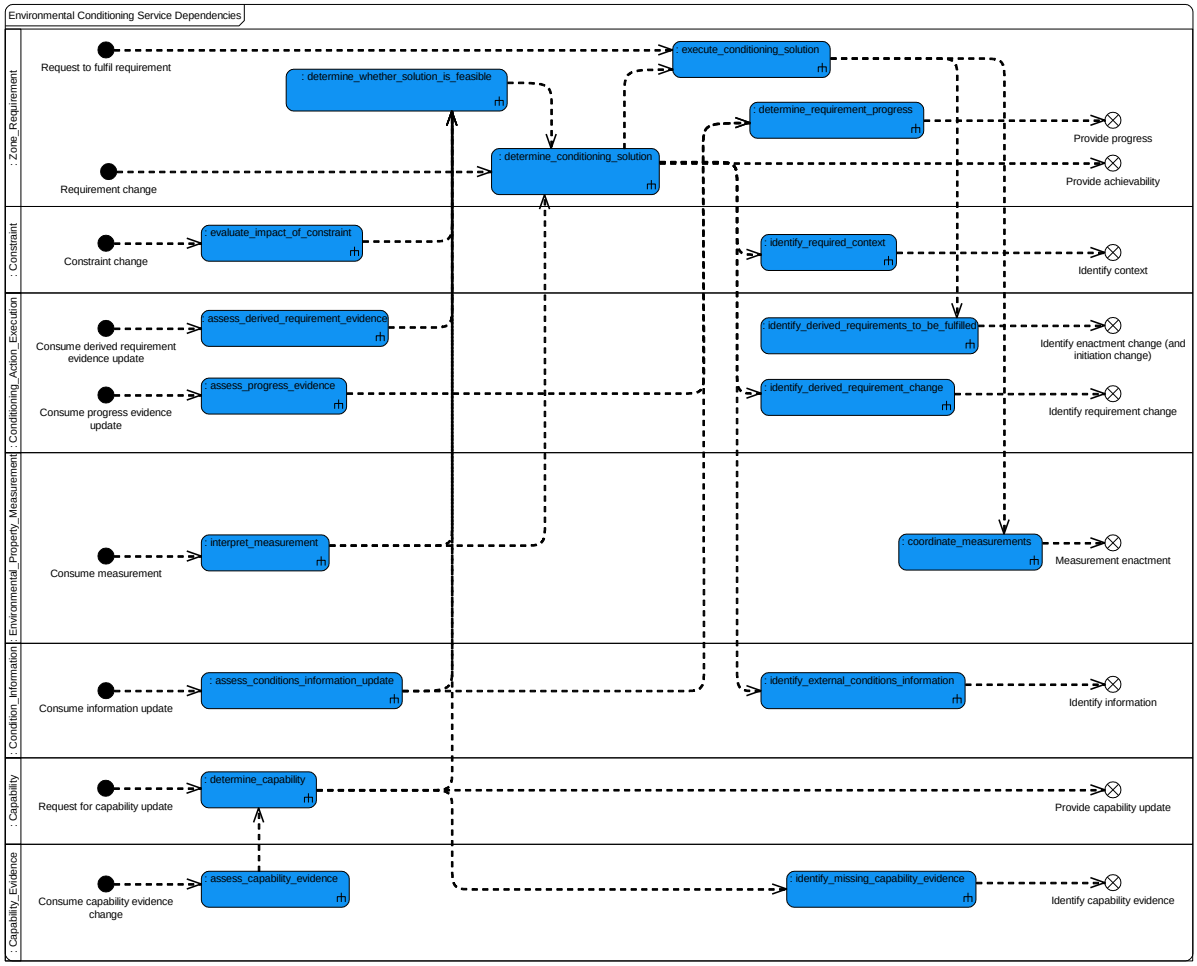

Figure 405: Environmental Conditioning Service Dependencies

B.2.19 Flights

B.2.19.1 Role

The role of Flights is to manage the composition of flights that Exploiting Platforms participate in.

B.2.19.2 Overview

Control Architecture

Flights is an action component as defined in the Control Architecture policy.

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.

B.2.19.3 Service Summary

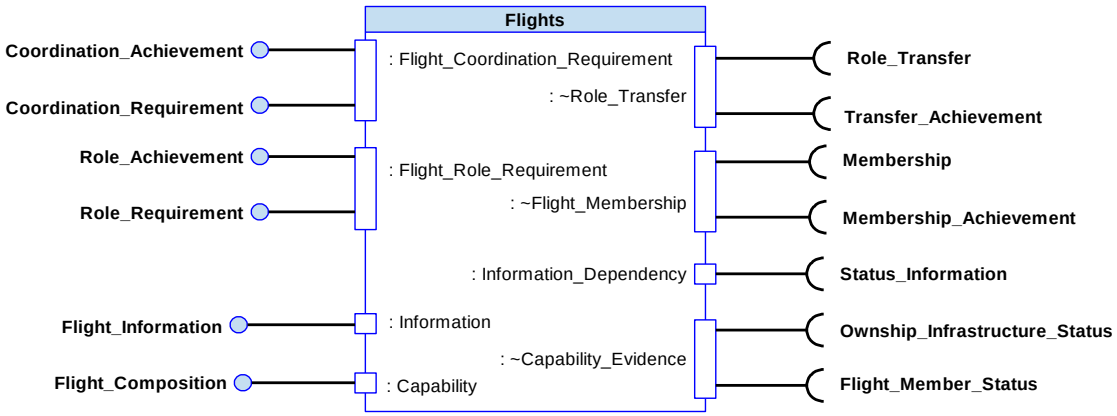

Figure 406: Flights Service Summary

B.2.19.4 Responsibilities

manage_according_to_structure

- To manage the Flight in accordance with the Control_Structure (e.g. maintain the hierarchy).

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.

identify_flight_members

- To identify the Members that make up a Flight.

assess_capability

- To assess the Capability to manage Flight composition taking account of system health and observed anomalies.

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_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).

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.

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).

identify_achievement

- To identify what has been achieved against the Requirement.

identify_whether_requirement_remains_achievable

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

B.2.19.5 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.

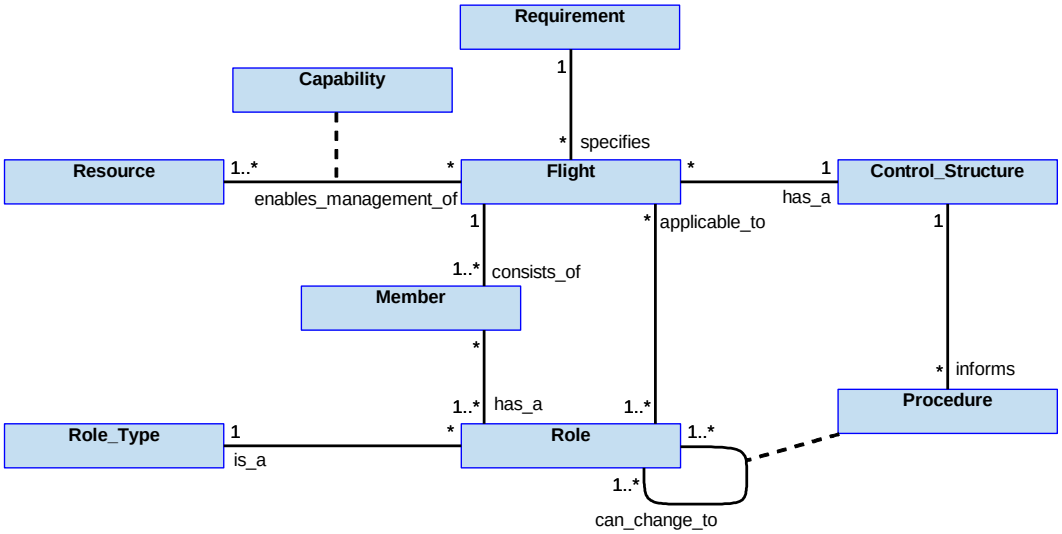

Figure 407: Flights Semantics

B.2.19.5.1 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).

B.2.19.6 Design Rationale

B.2.19.6.1 Assumptions

- For a Flight to be managed automatically by the system all participants are assumed to be PRA based, each with an instance of this component.
- A Member will not be part of more than one Flight at a time.

B.2.19.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy is applicable.

B.2.19.6.3 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. the Contingency extension to 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.

B.2.19.6.4 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 leader, 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.

B.2.19.7 Services

B.2.19.7.1 Service Definitions

B.2.19.7.1.1 Flight_Coordination_Requirement

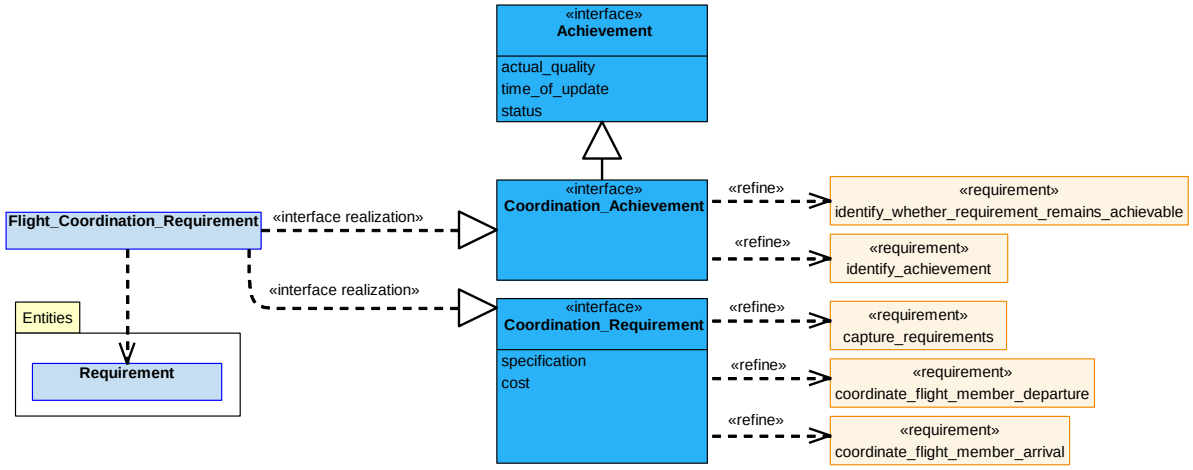

Figure 408: Flight_Coordination_Requirement Service Definition

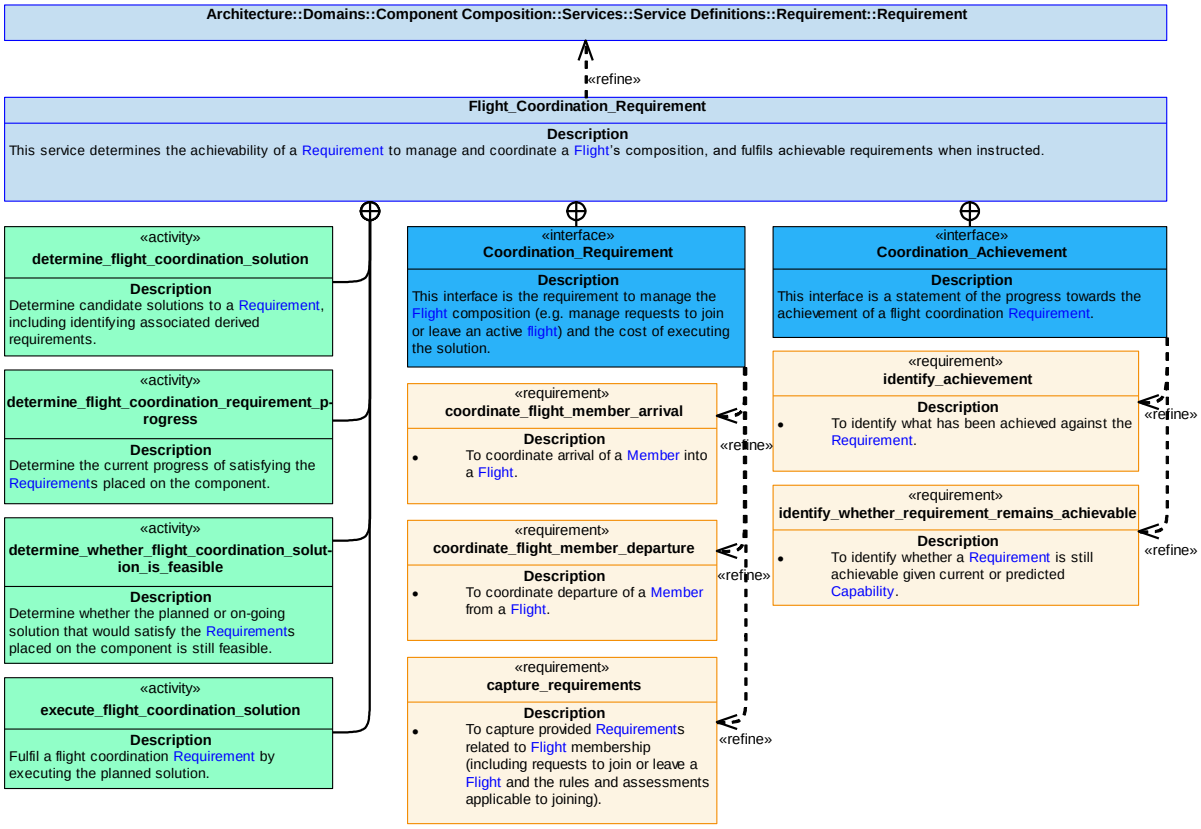

Figure 409: Flight_Coordination_Requirement Service Policy

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.

B.2.19.7.1.2 Flight_Role_Requirement

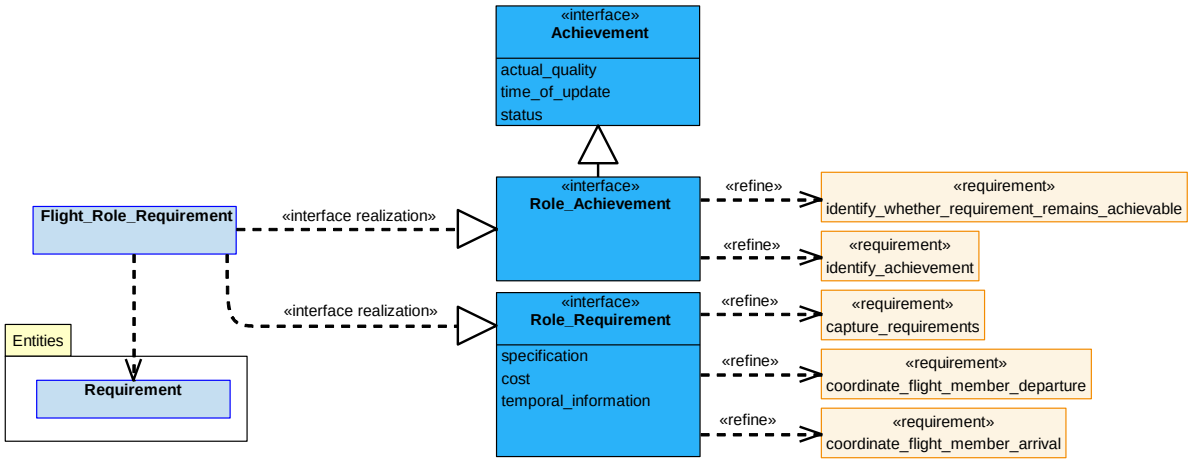

Figure 410: Flight_Role_Requirement Service Definition

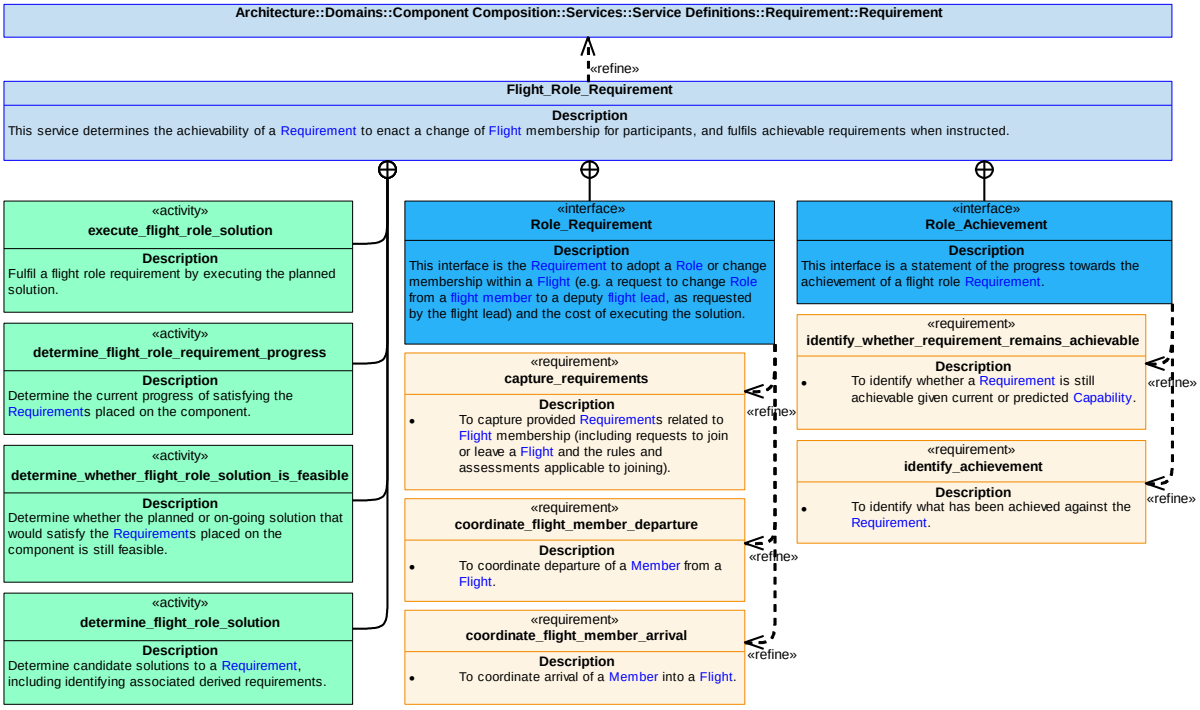

Figure 411: Flight_Role_Requirement Service Policy

Flight_Role_Requirement

This service determines the achievability of a Requirement to enact a change of Flight membership for participants, and fulfils achievable requirements when instructed.

Interfaces

Role_Requirement

This interface is the Requirement to adopt a Role or change membership 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.

B.2.19.7.1.3 Role_Transfer

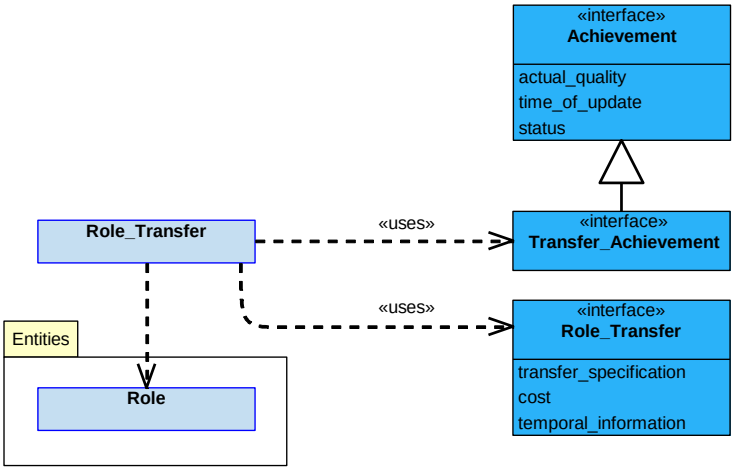

Figure 412: Role_Transfer Service Definition

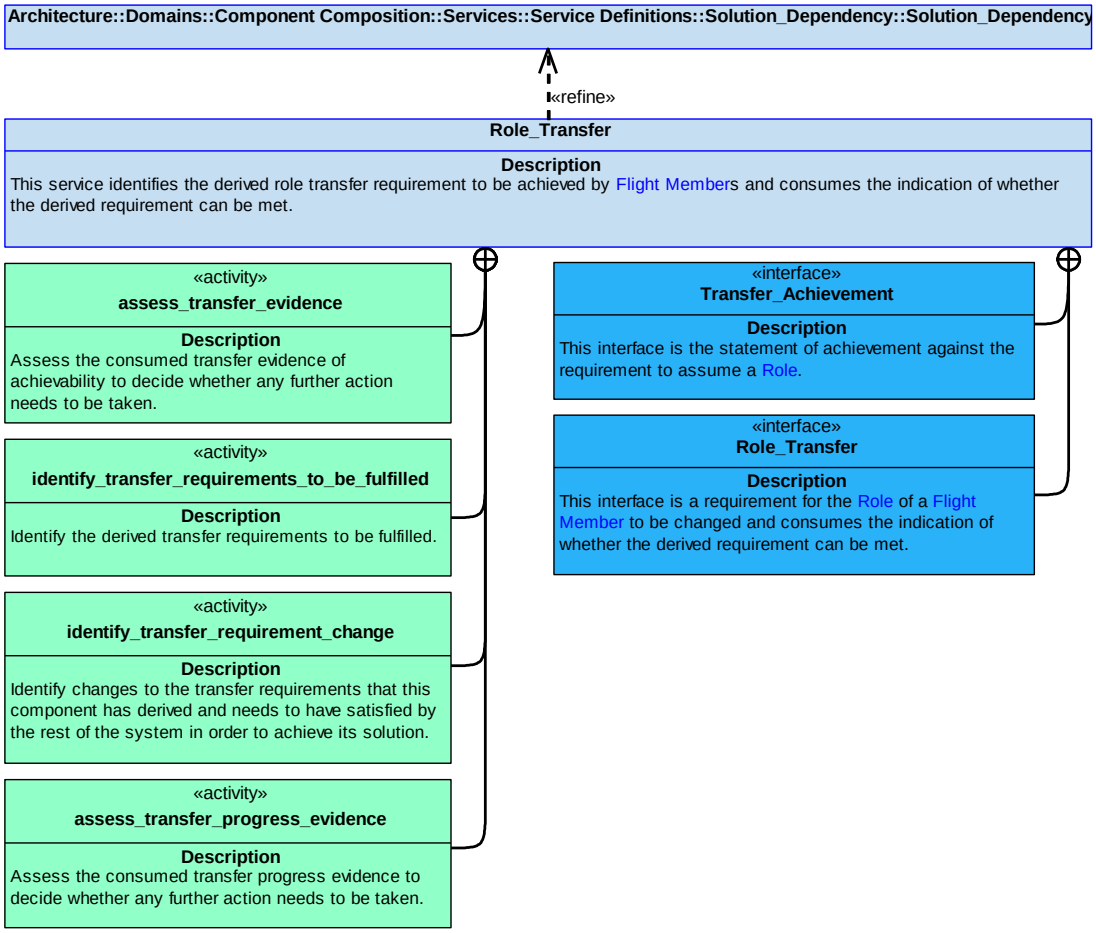

Figure 413: Role_Transfer Service Policy

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 assume 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.

B.2.19.7.1.4 Flight_Membership

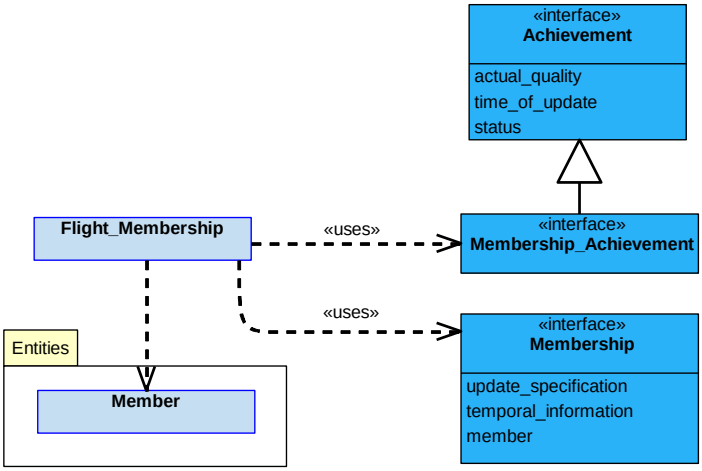

Figure 414: Flight_Membership Service Definition

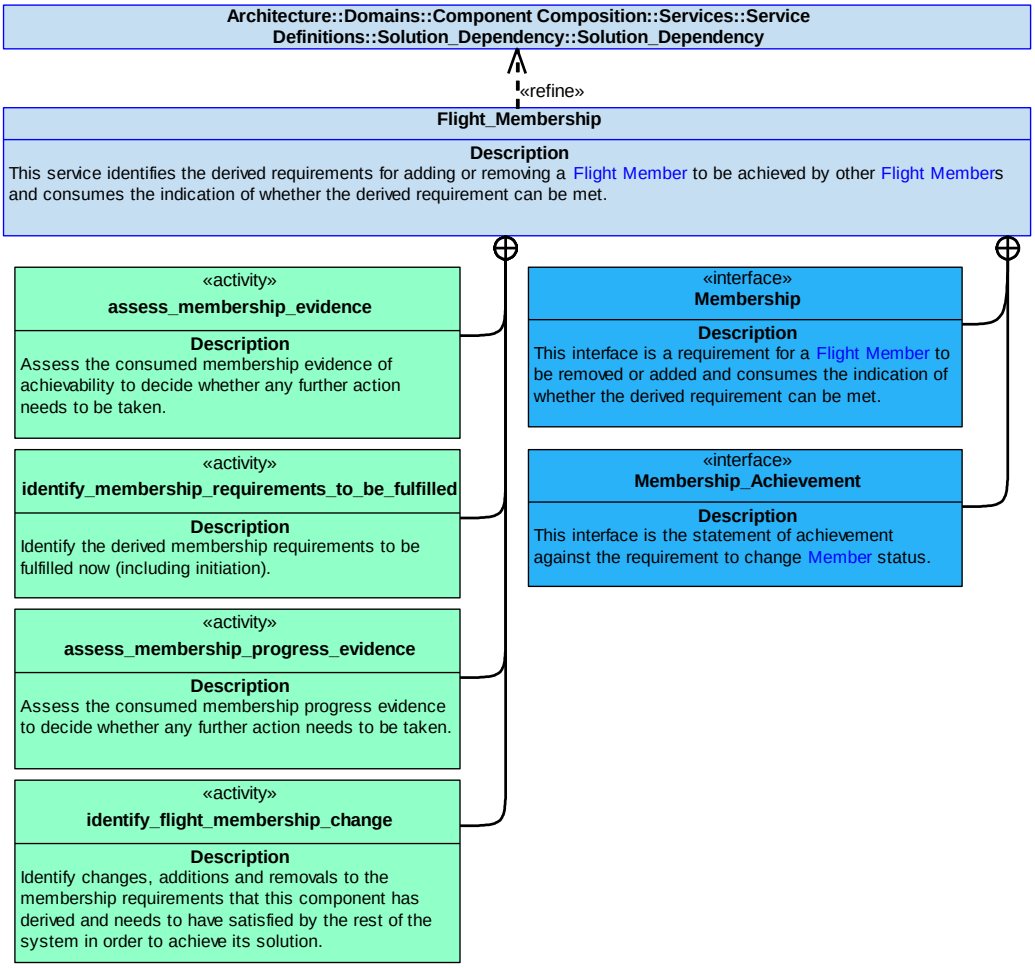

Figure 415: Flight_Membership Service Policy

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.

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).

B.2.19.7.1.5 Information

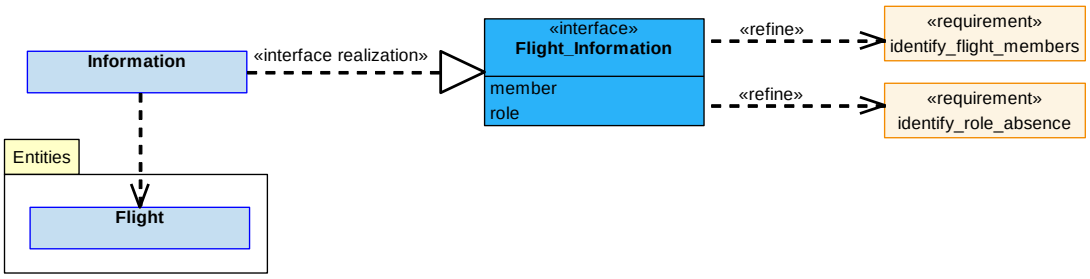

Figure 416: Information Service Definition

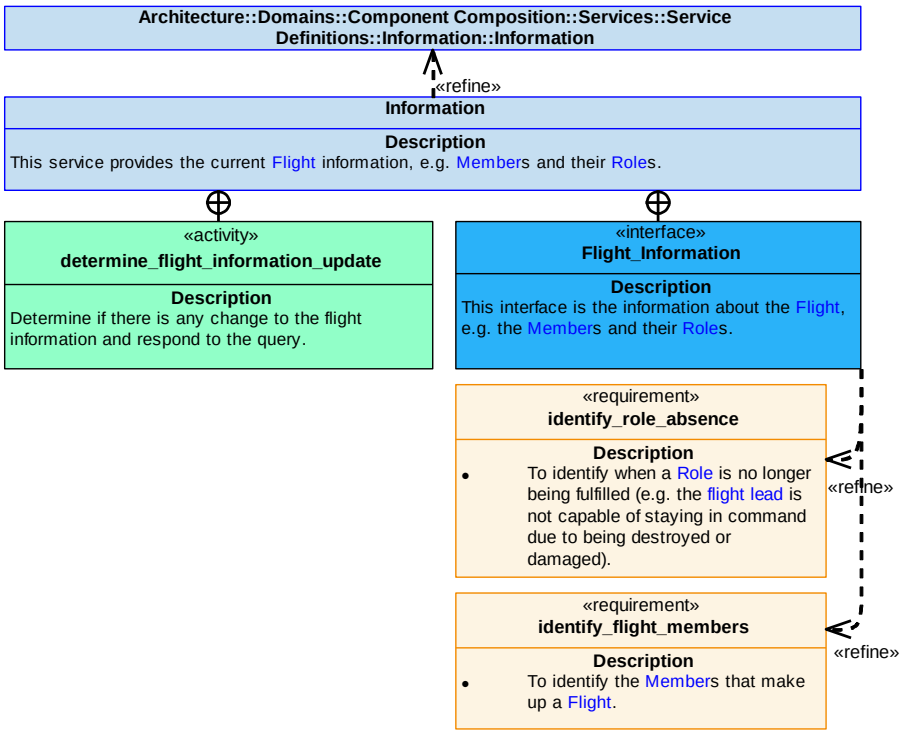

Figure 417: Information Service Policy

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 an asset 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.

B.2.19.7.1.6 Capability

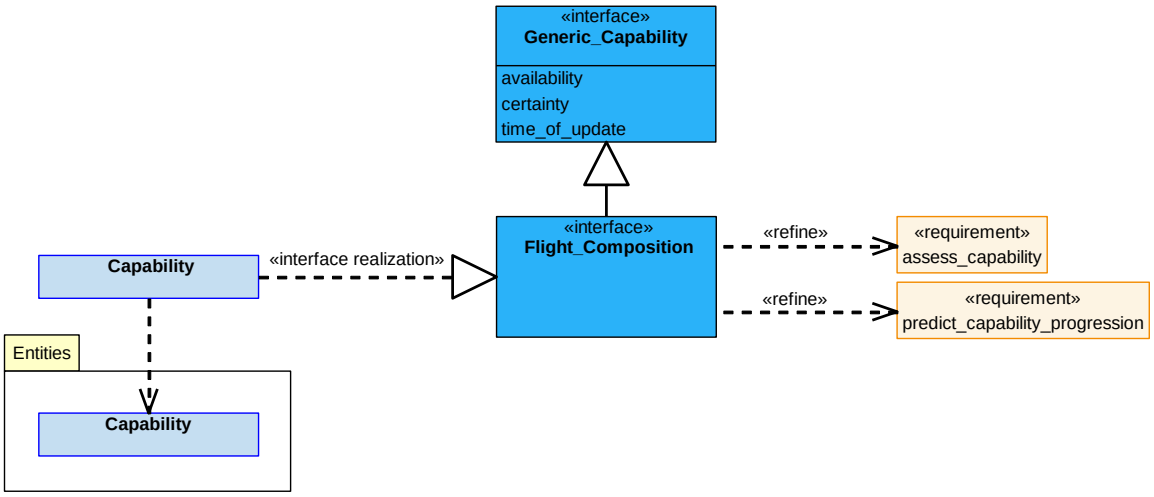

Figure 418: Capability Service Definition

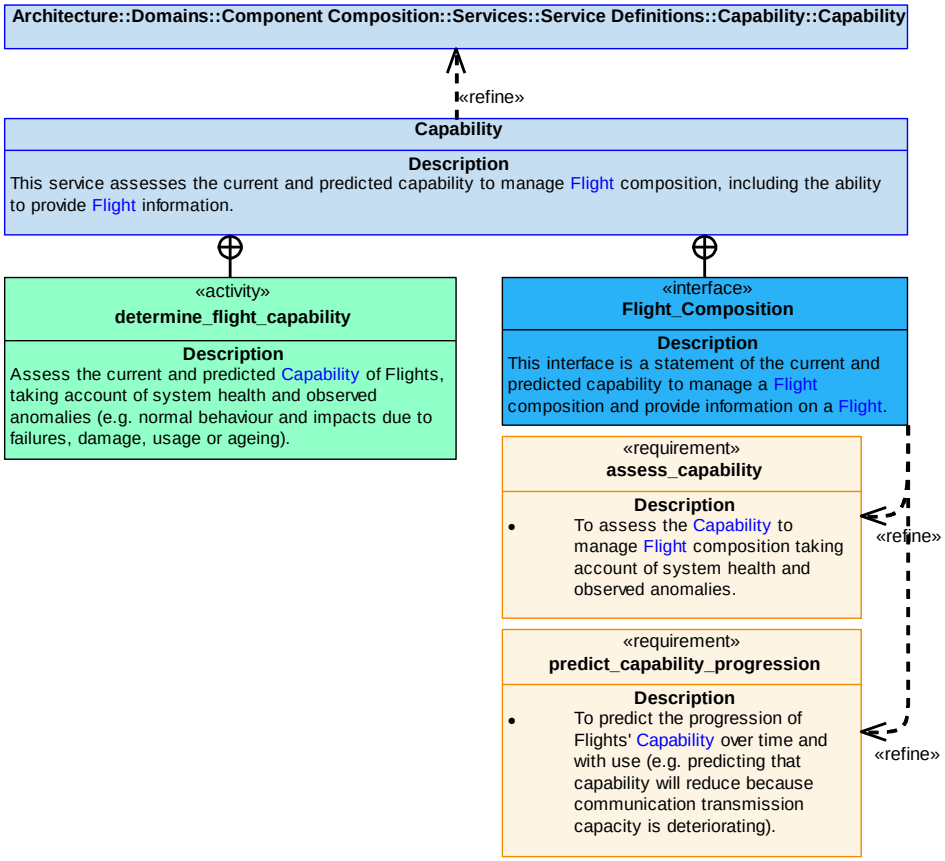

Figure 419: Capability Service Policy

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).

B.2.19.7.1.7 Capability_Evidence

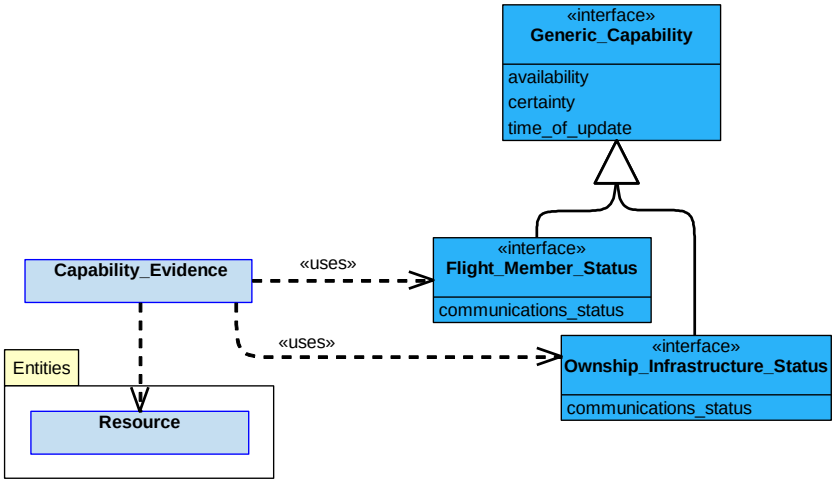

Figure 420: Capability_Evidence Service Definition

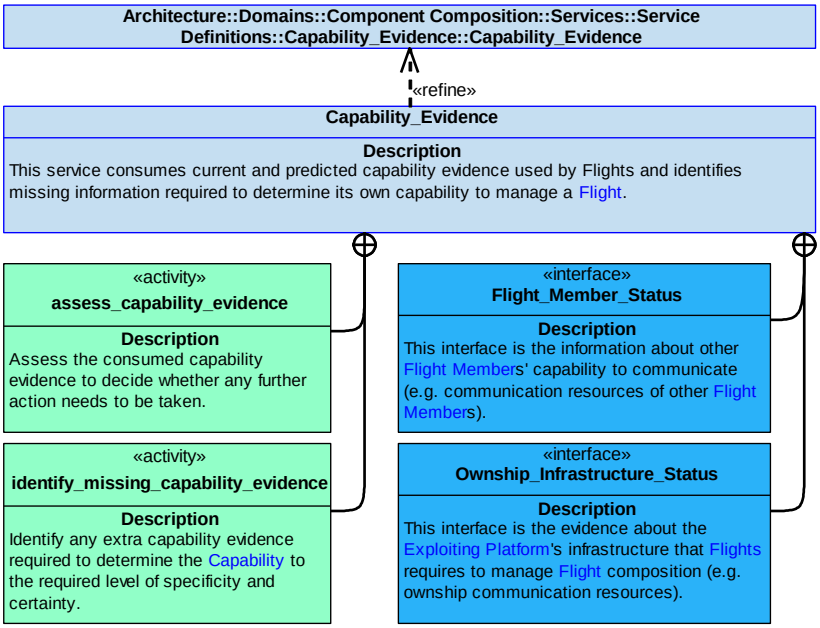

Figure 421: Capability_Evidence Service Policy

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).

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.

B.2.19.7.1.8 Information_Dependency

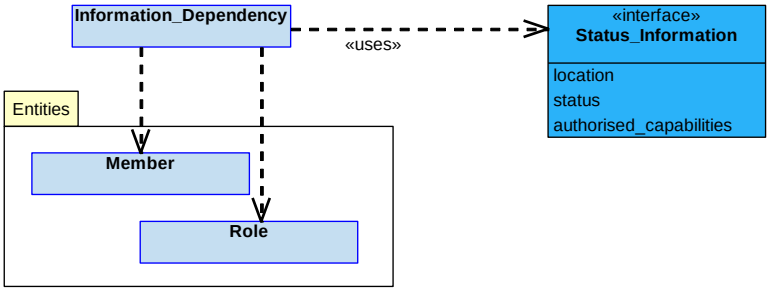

Figure 422: Information_Dependency Service Definition

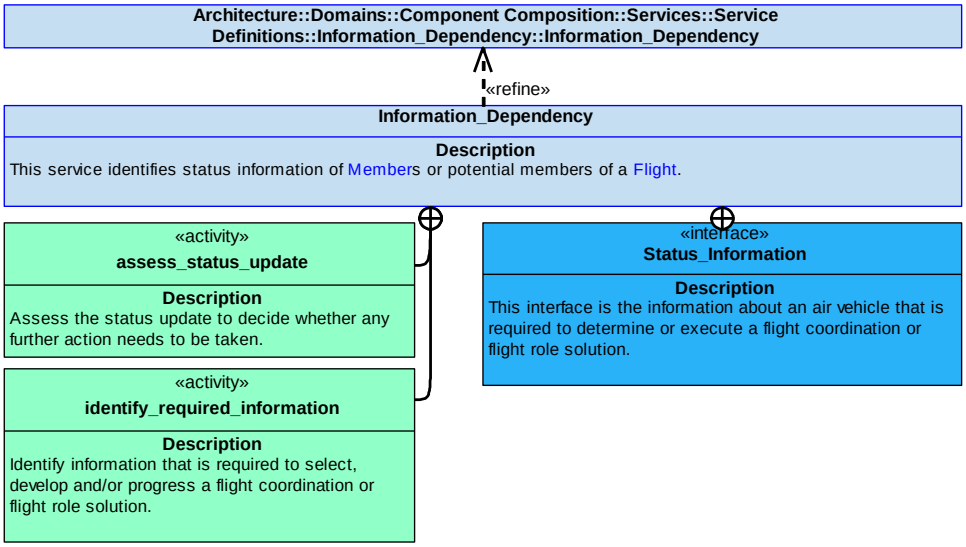

Figure 423: Information_Dependency Service Policy

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.

B.2.19.7.2 Service Dependencies

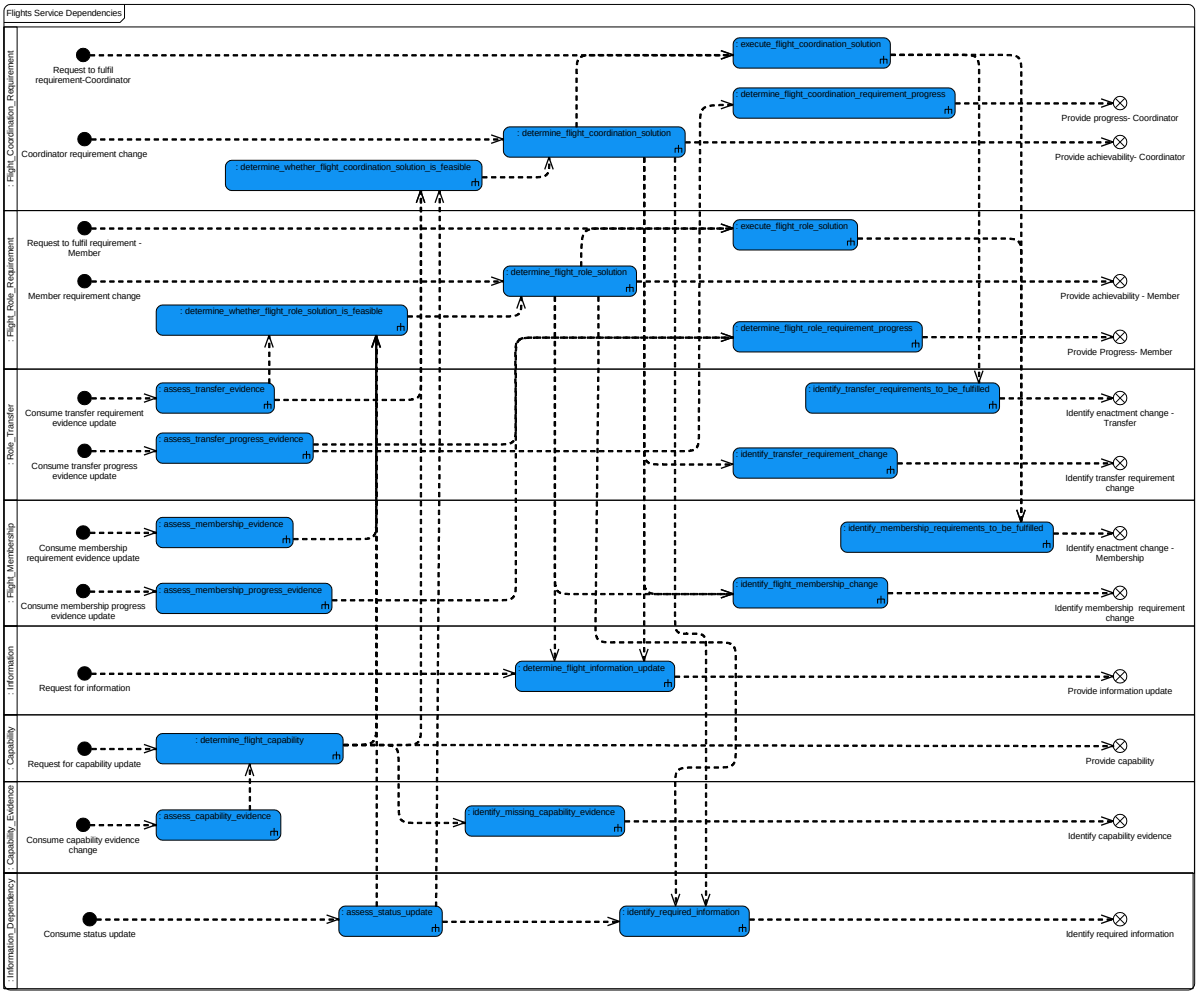

Figure 424: Flights Service Dependencies

B.2.20 Fluids

B.2.20.1 Role

The role of Fluids is to manage the storage and transfer of fluids.

B.2.20.2 Overview

Control Architecture

Fluids is a resource component as defined in the Control Architecture policy.

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.

B.2.20.3 Service Summary

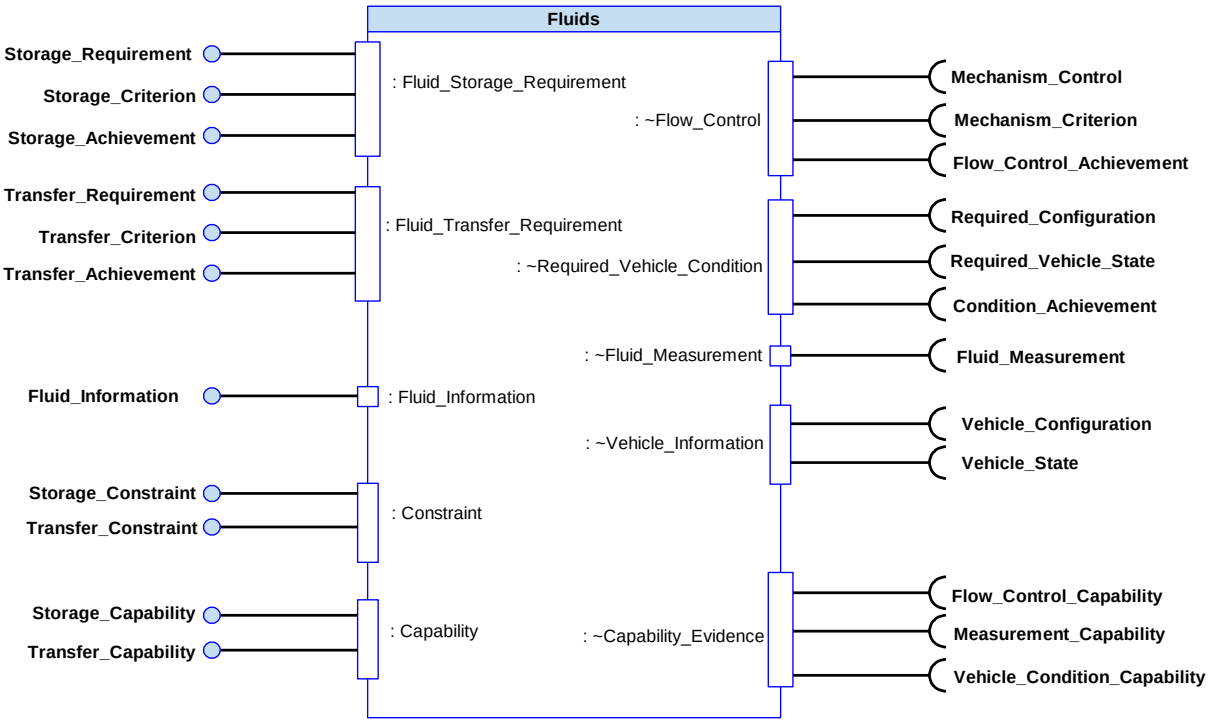

Figure 425: Fluids Service Summary

B.2.20.4 Responsibilities

capture_transfer_requirements

- To capture Transfer_Requirements to transfer fluid.

capture_constraints

- To capture Constraints related to fluid storage and transfer.

determine_transfer_solution

- To determine a fluid transfer solution.

determine_storage_solution

- To determine a fluid storage solution.

monitor_fluid_properties

- To monitor the properties of fluid (e.g. quantity or temperature).

coordinate_fluid_storage

- To coordinate a solution for the storage of fluid.

coordinate_fluid_transfer

- To coordinate a solution for the transfer of fluid.

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.

identify_fluid_distribution_progress

- To identify the progress of a fluid distribution solution against a Fluid_Management_Requirement.

capture_storage_requirements

- To capture Storage_Requirements to store fluid.

determine_predicted_fluid_management_quality

- To determine the predicted quality of a proposed Fluid_Management_Solution against given measurement criteria.

capture_fluid_management_criteria

- To capture provided measurement criteria for fluid management.

update_achievability

- To identify whether a Fluid_Management_Requirement is still achievable given current or predicted Capability and Constraints.

determine_actual_fluid_management_quality

- To determine the actual quality of a proposed Fluid_Management_Solution against given measurement criteria.

identify_distribution_pre-conditions

- To identify Vehicle_States required to support fluid distribution.

B.2.20.5 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).

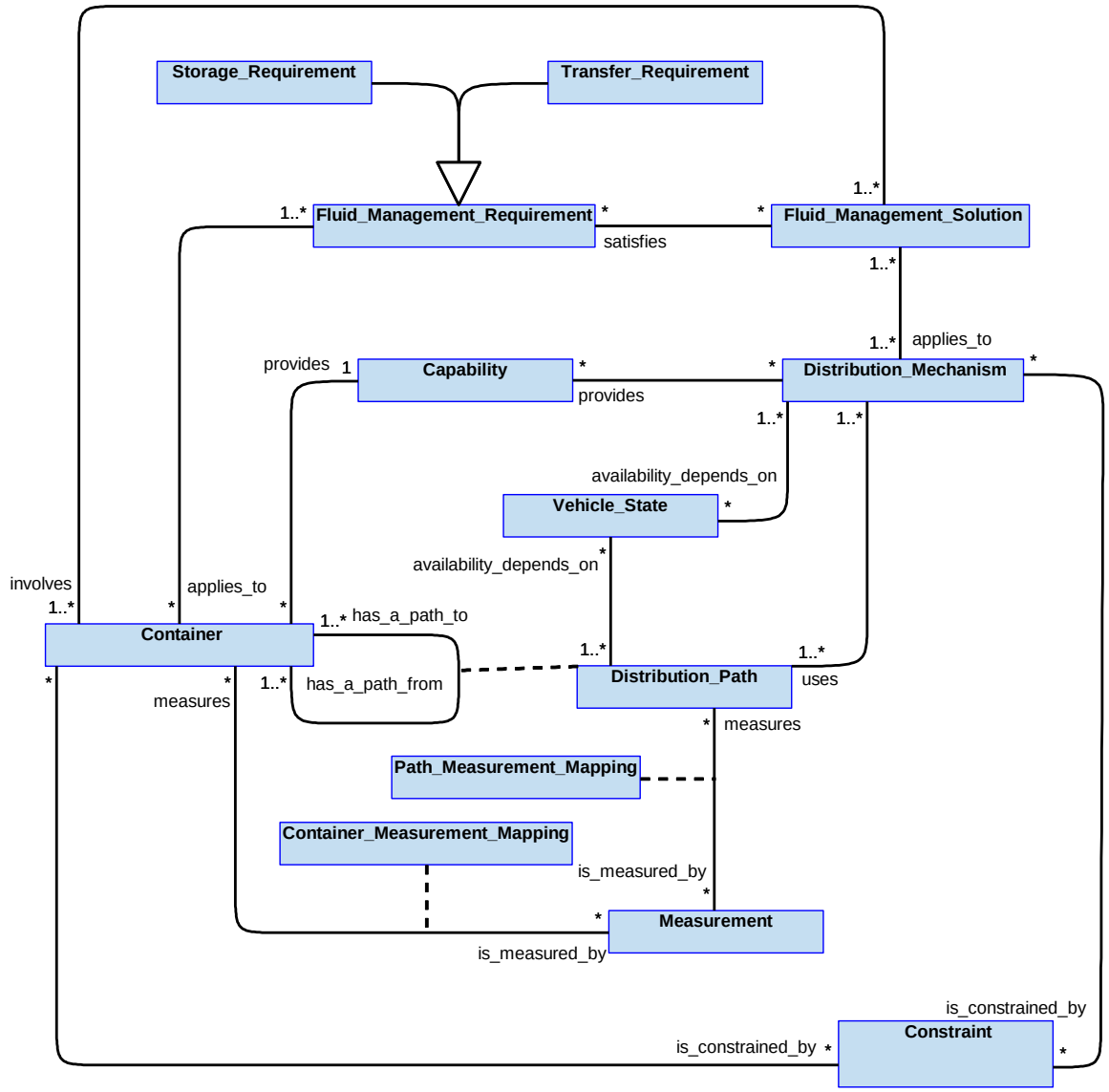

Figure 426: Fluids Semantics

B.2.20.5.1 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.

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).

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).

B.2.20.6 Design Rationale

B.2.20.6.1 Assumptions

- The Container contents that concern this component are fluids, i.e. substances that have no fixed shape and yield easily to external pressure (typically a liquid, gas or particle-laden flow).
- The component is not responsible for 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.

B.2.20.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.20.6.3 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).

B.2.20.6.4 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).

B.2.20.7 Services

B.2.20.7.1 Service Definitions

B.2.20.7.1.1 Fluid_Storage_Requirement

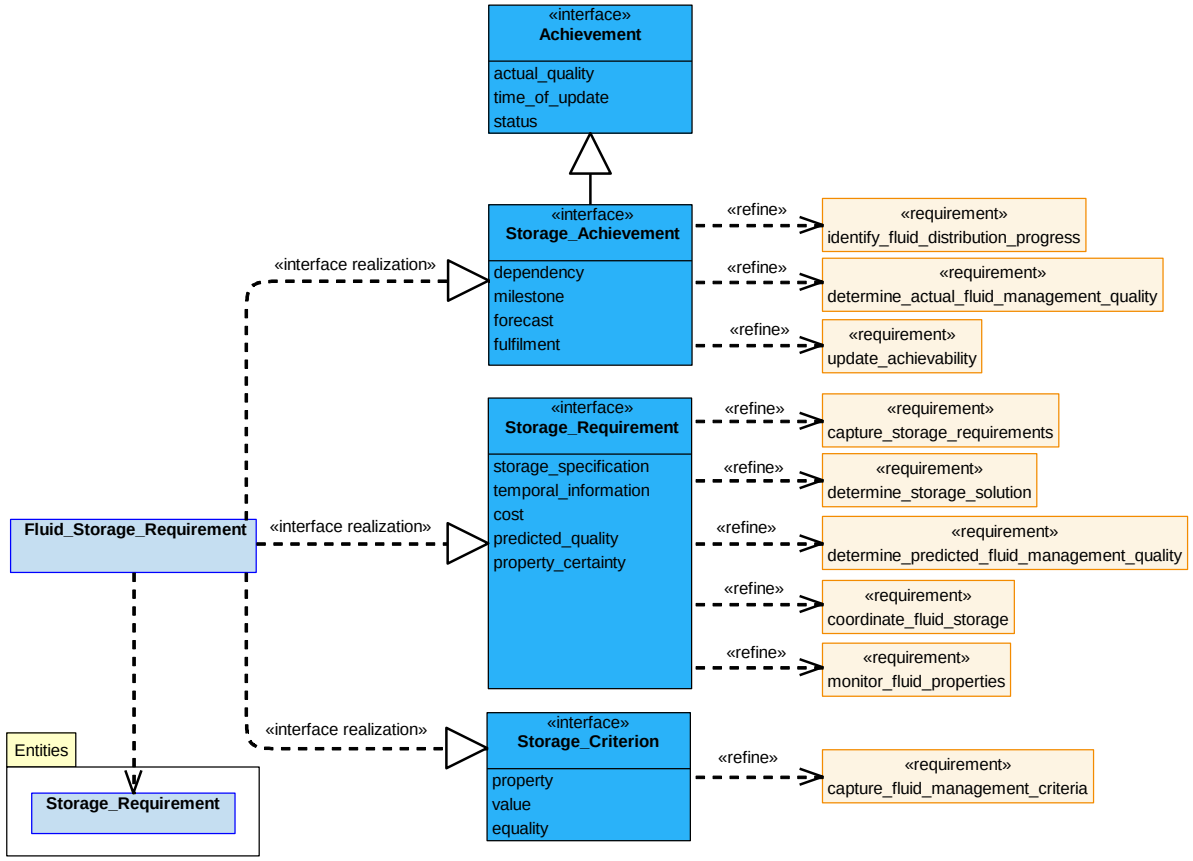

Figure 427: Fluid_Storage_Requirement Service Definition

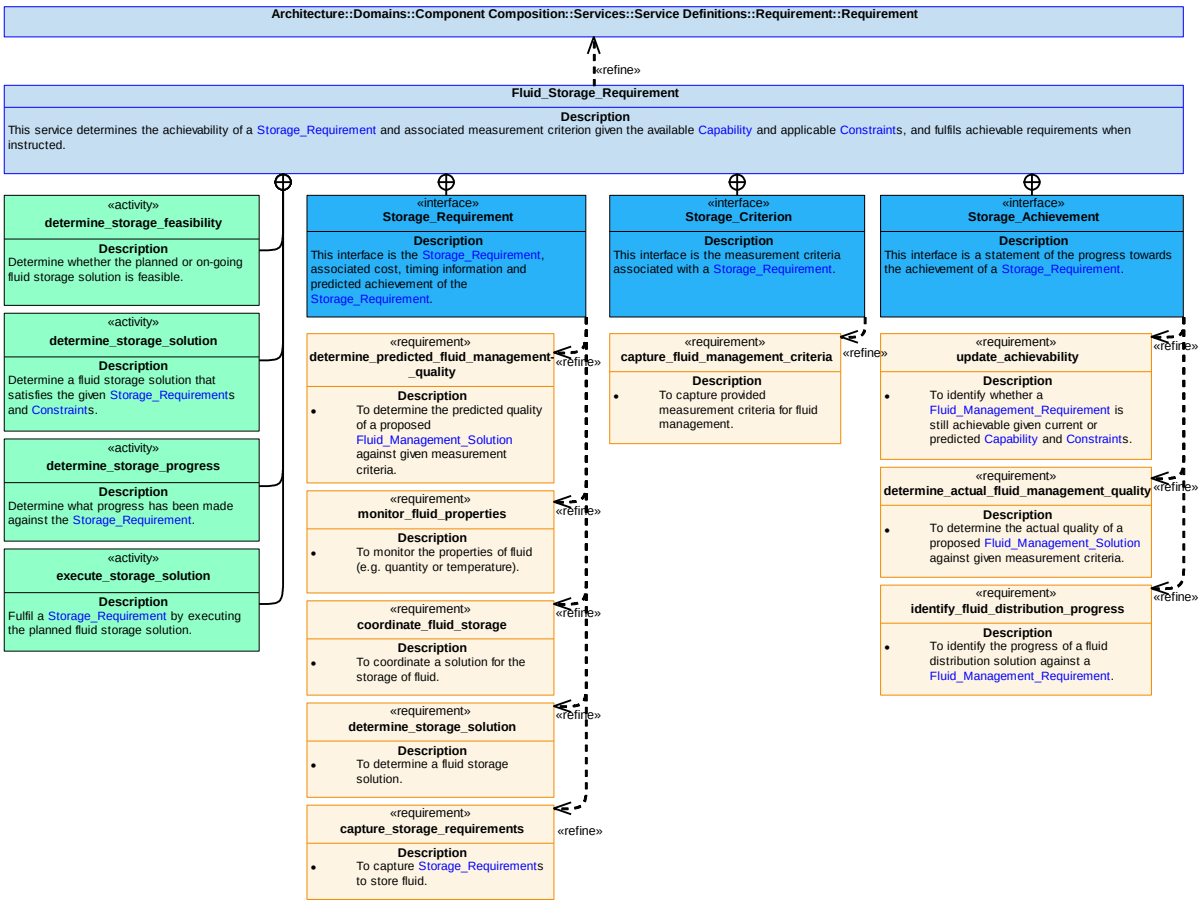

Figure 428: Fluid_Storage_Requirement Service Policy

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.

Interfaces

Storage_Requirement

This interface is the Storage_Requirement, associated cost, timing information and predicted achievement of the Storage_Requirement.

Attributes

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, 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.

B.2.20.7.1.2 Fluid_Transfer_Requirement

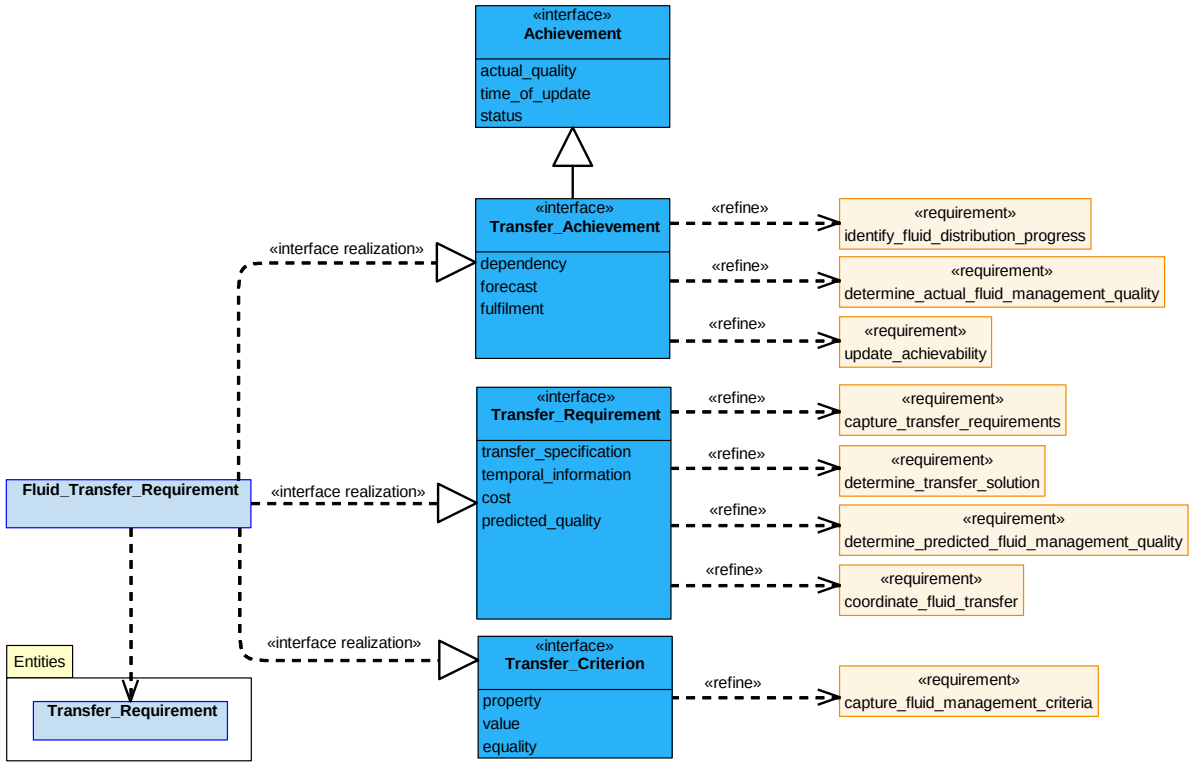

Figure 429: Fluid_Transfer_Requirement Service Definition

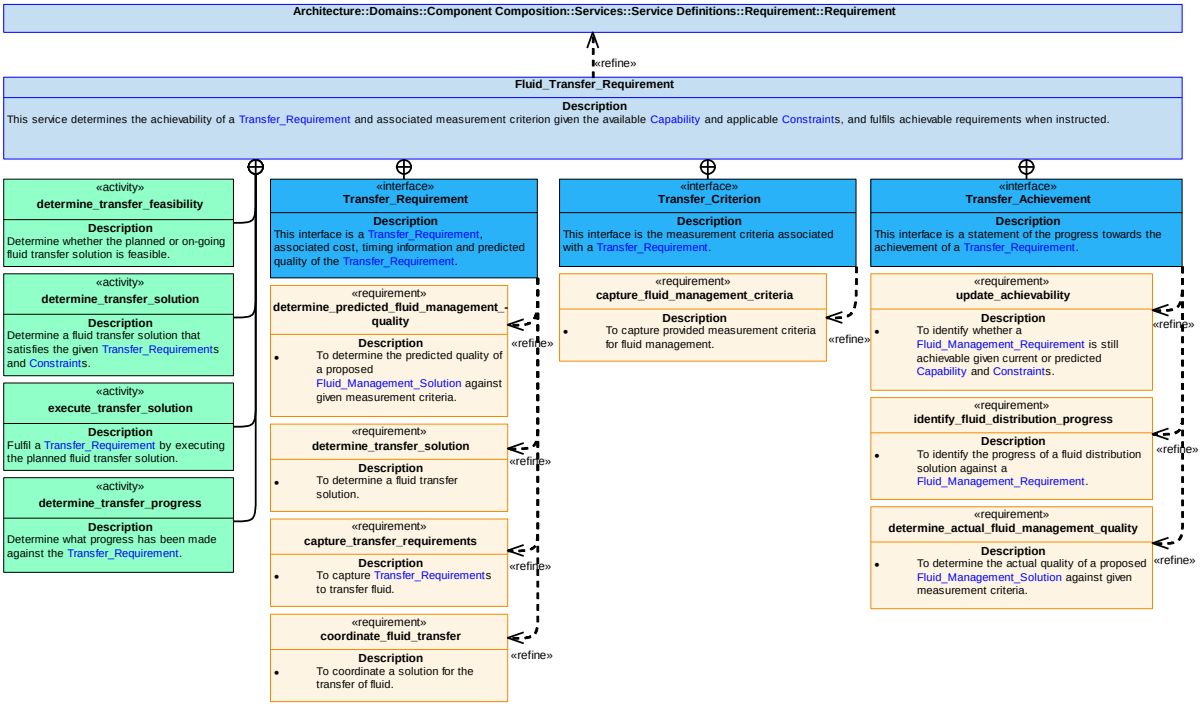

Figure 430: Fluid_Transfer_Requirement Service Policy

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.

B.2.20.7.1.3 Flow_Control

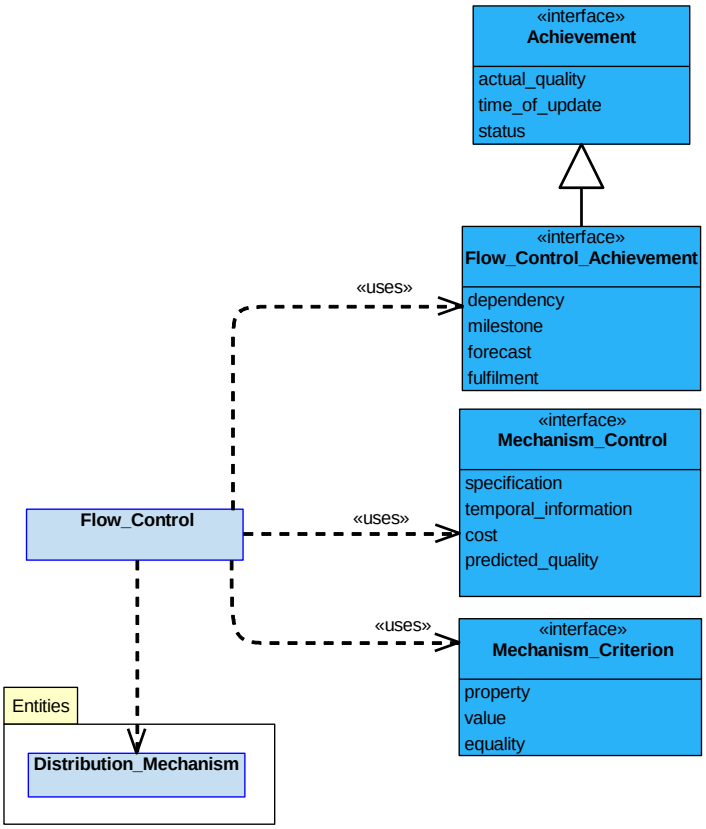

Figure 431: Flow_Control Service Definition

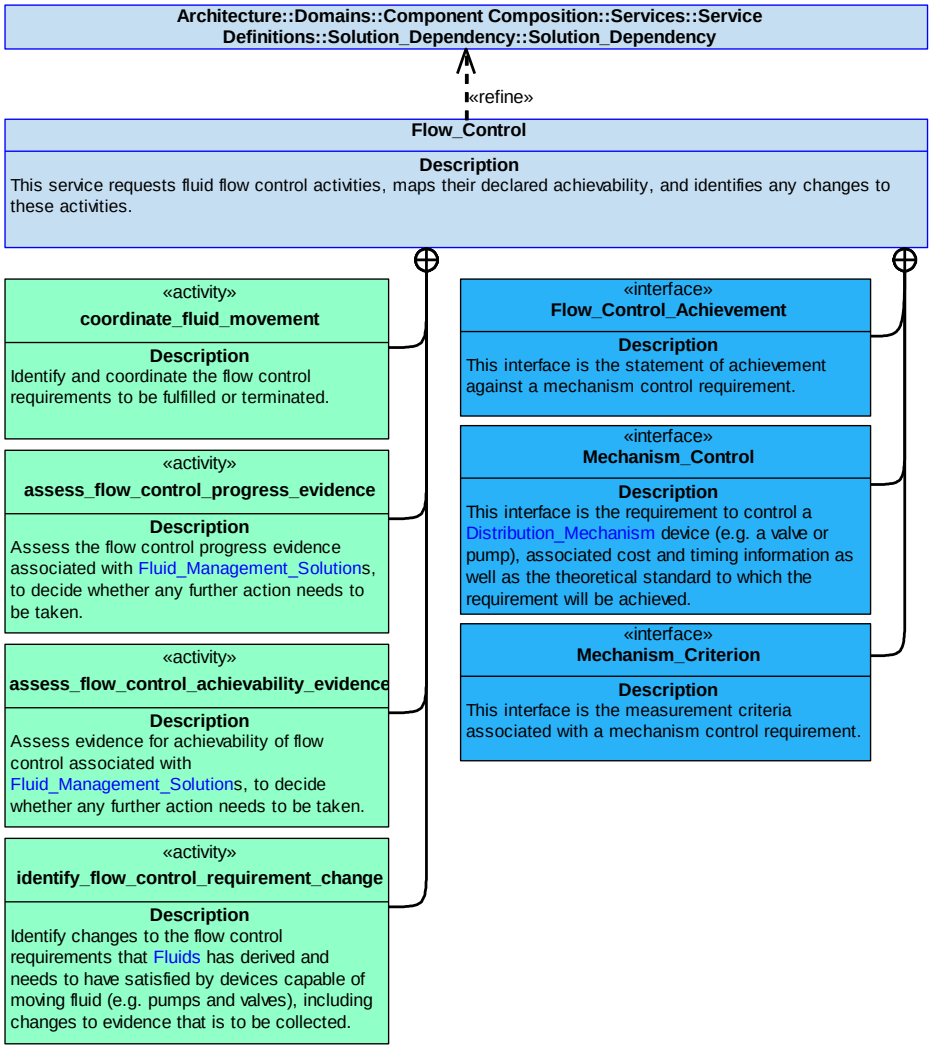

Figure 432: Flow_Control Service Policy

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.

B.2.20.7.1.4 Required_Vehicle_Condition

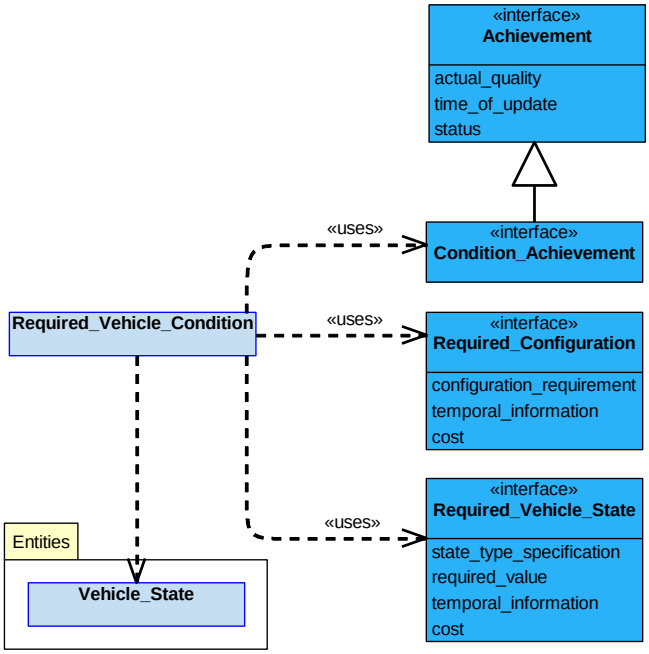

Figure 433: Required_Vehicle_Condition Service Definition

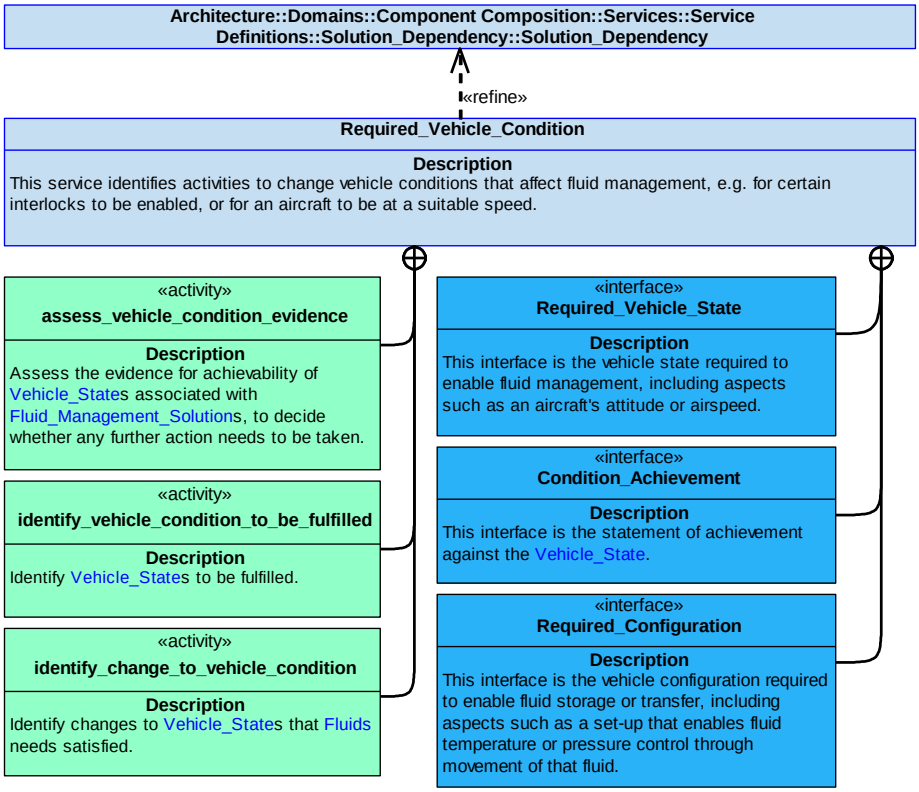

Figure 434: Required_Vehicle_Condition Service Policy

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.

B.2.20.7.1.5 Fluid_Information

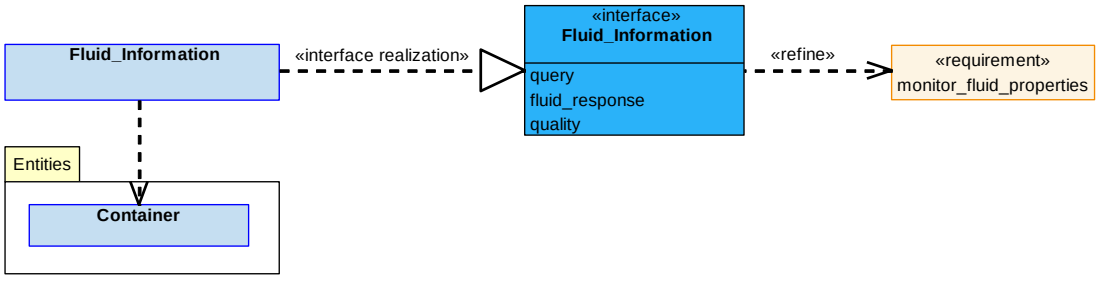

Figure 435: Fluid_Information Service Definition

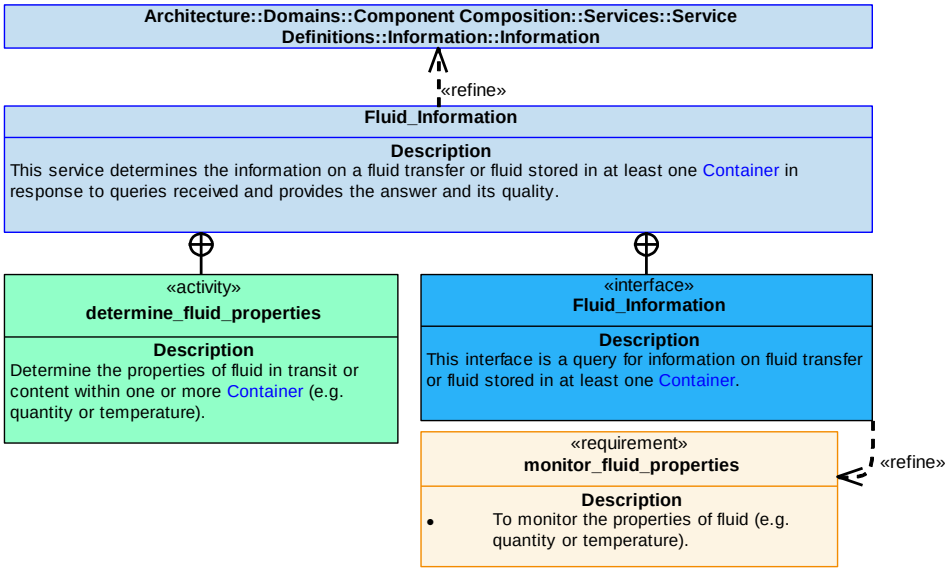

Figure 436: Fluid_Information Service Policy

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.

Interface

Fluid_Information

This interface is a query for information on fluid transfer or fluid stored in at least one Container.

Attributes

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.

Activity

determine_fluid_properties

Determine the properties of fluid in transit or content within one or more Container (e.g. quantity or temperature).

B.2.20.7.1.6 Fluid_Measurement

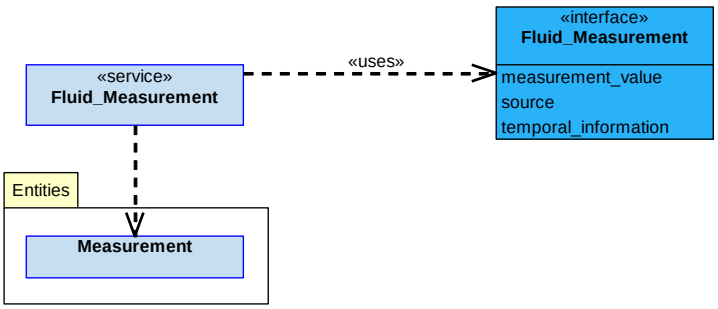

Figure 437: Fluid_Measurement Service Definition

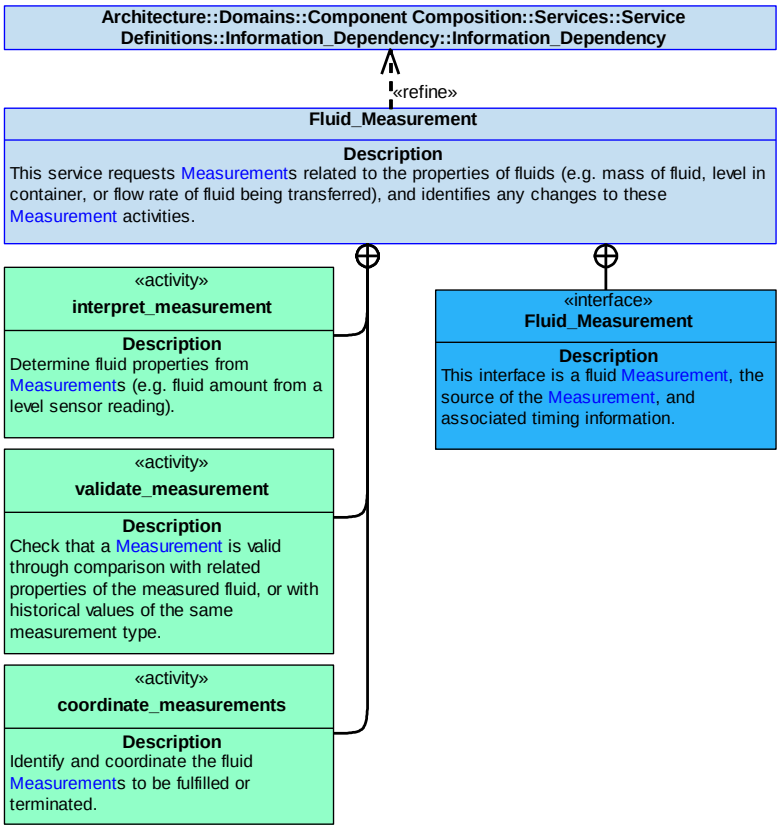

Figure 438: Fluid_Measurement Service Policy

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.

B.2.20.7.1.7 Vehicle_Information

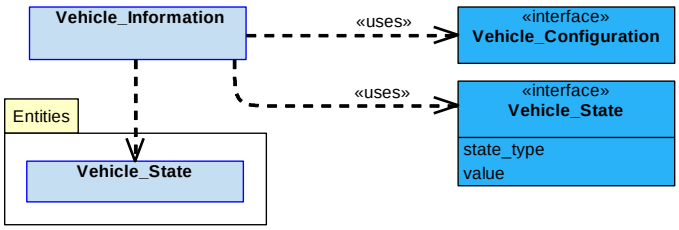

Figure 439: Vehicle_Information Service Definition

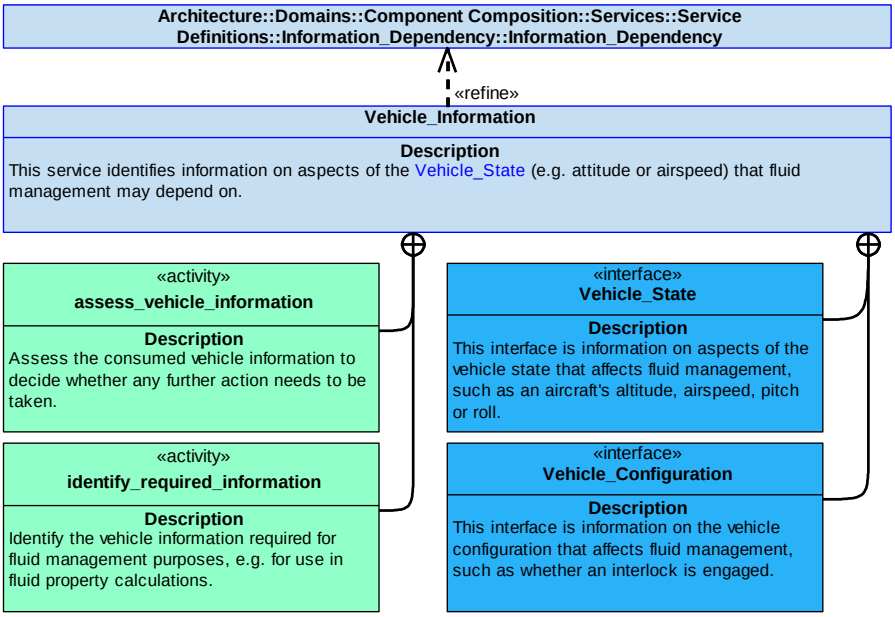

Figure 440: Vehicle_Information Service Policy

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.

B.2.20.7.1.8 Constraint

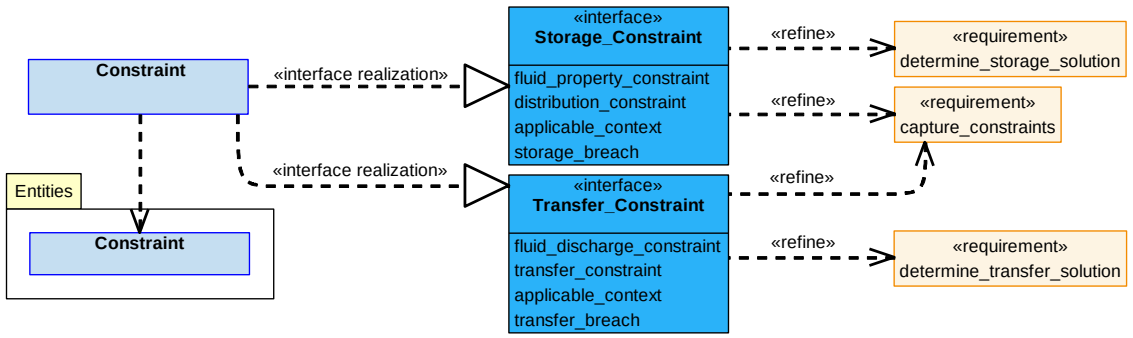

Figure 441: Constraint Service Definition

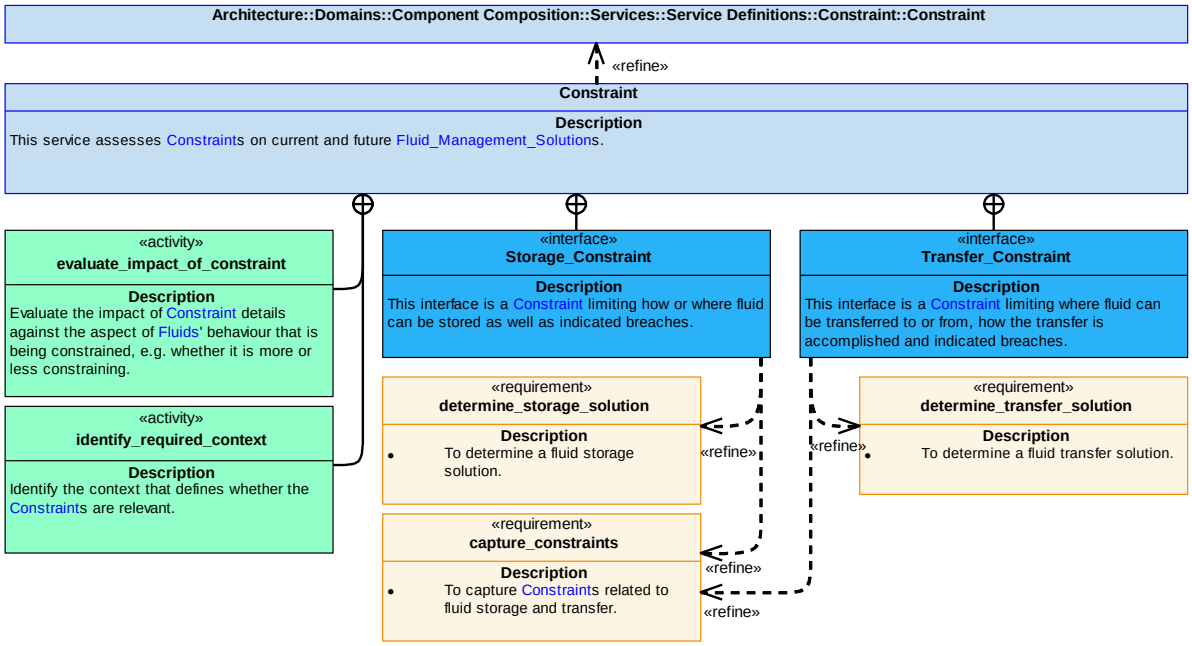

Figure 442: Constraint Service Policy

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.

B.2.20.7.1.9 Capability

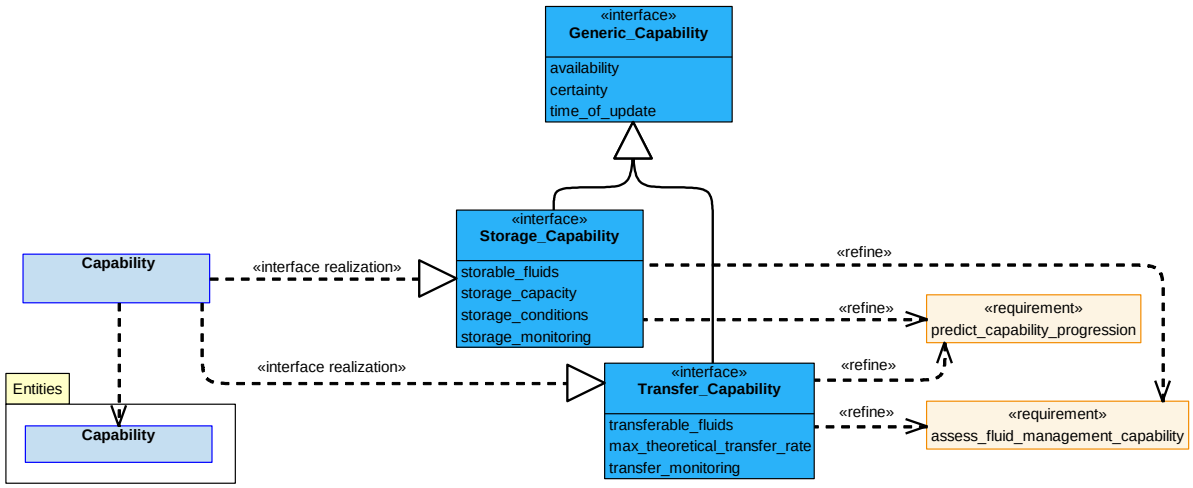

Figure 443: Capability Service Definition

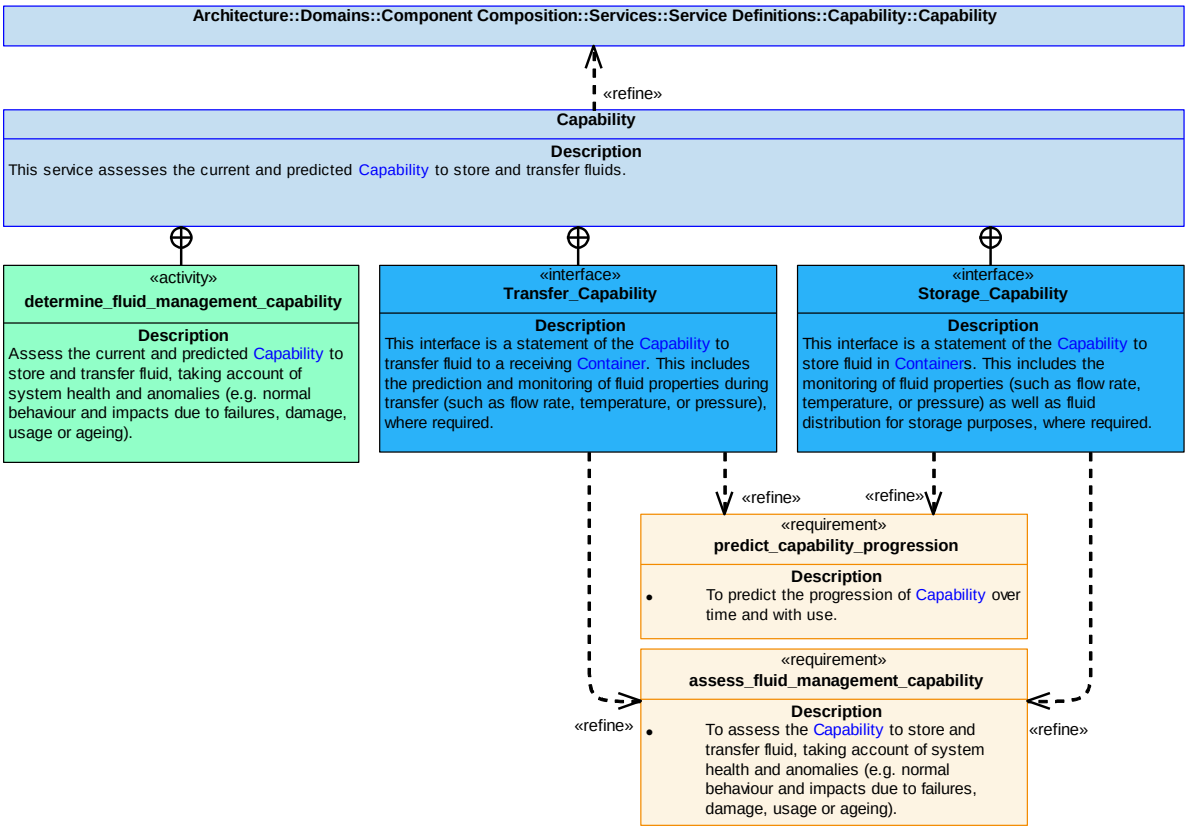

Figure 444: Capability Service Policy

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).

B.2.20.7.1.10 Capability_Evidence

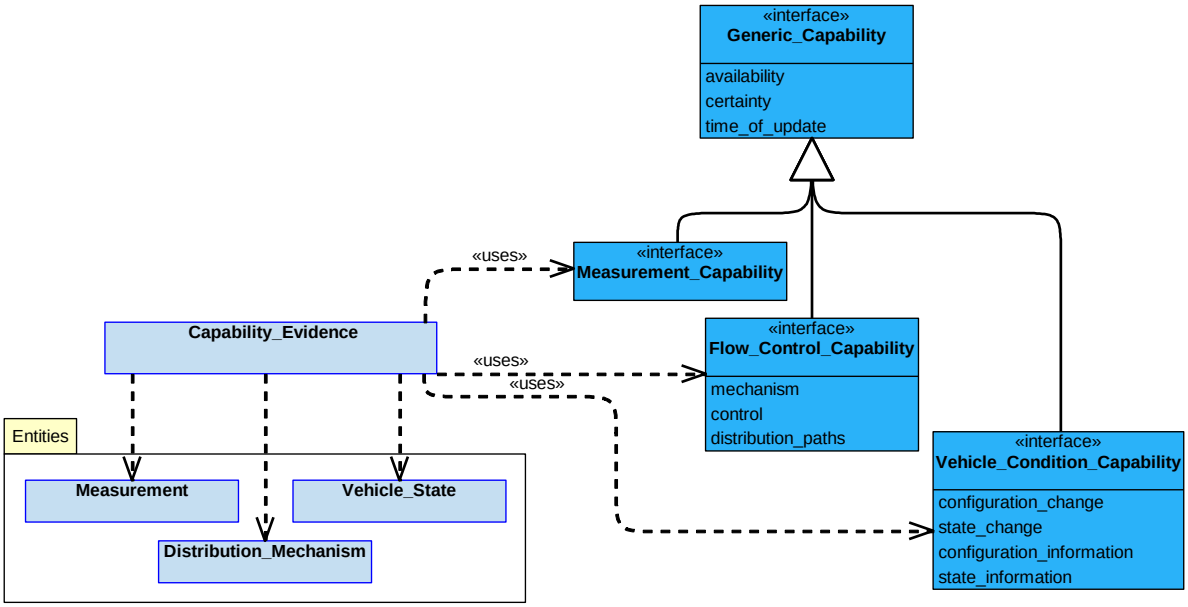

Figure 445: Capability_Evidence Service Definition

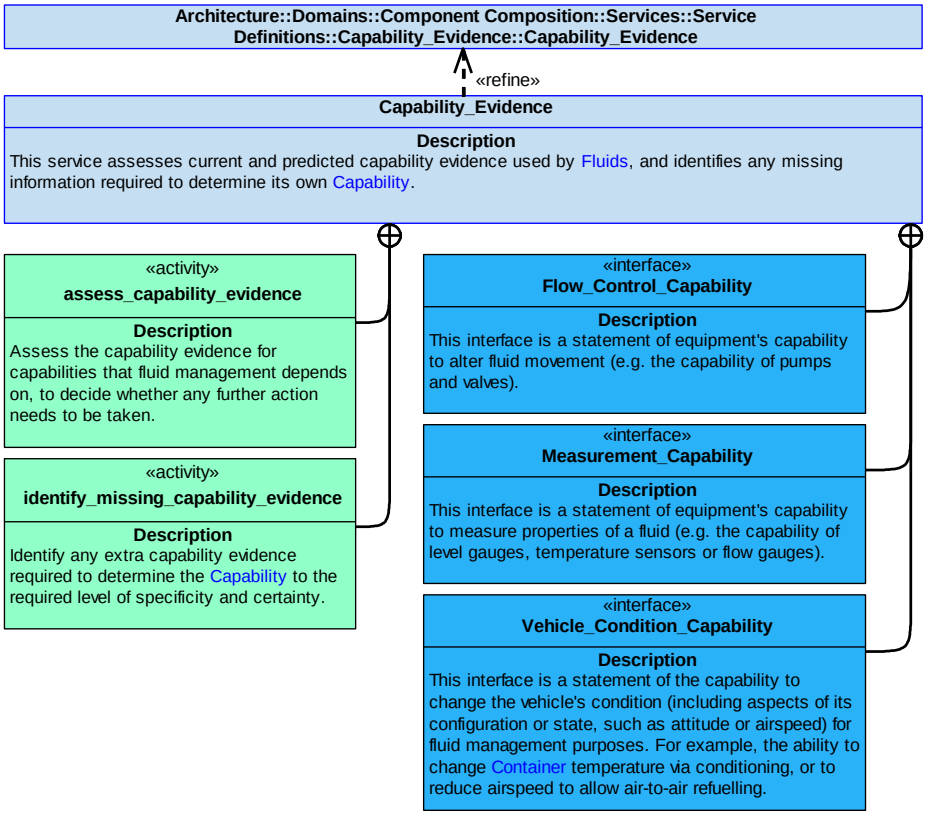

Figure 446: Capability Evidence Service Policy

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, or to reduce airspeed to allow air-to-air refuelling.

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.
configuration_information	An indication of how well the vehicle configuration can be established.
state_information	An indication of how well an aspect of the vehicle state 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.

B.2.20.7.2 Service Dependencies

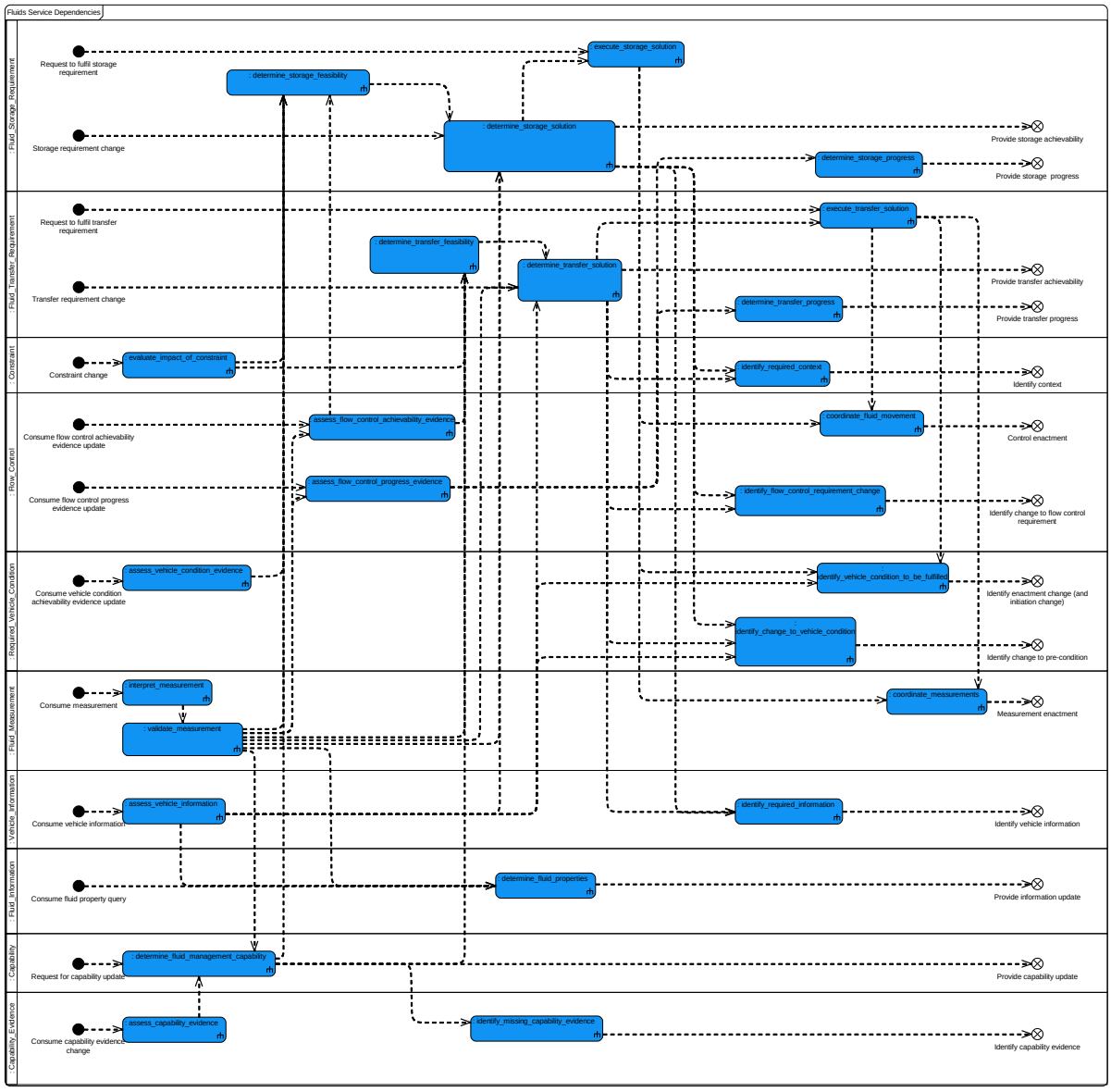

Figure 447: Fluids Service Dependencies

B.2.21 Formations

B.2.21.1 Role

The role of Formations is to coordinate and execute changes in the relative positions of vehicles moving in a formation.

B.2.21.2 Overview

Control Architecture

Formations is an action component as defined in the Control Architecture policy.

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).

B.2.21.3 Service Summary

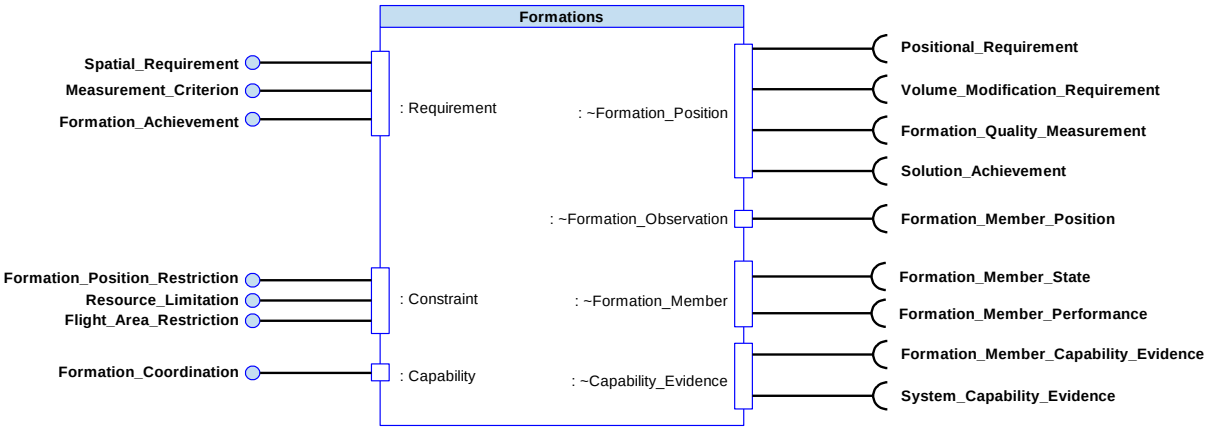

Figure 448: Formations Service Summary

B.2.21.4 Responsibilities

capture_formation_requirements

- To capture given formation Requirements (e.g. number of Formation_Members, required positional relationships between Formation_Members, lead vehicle identification).

capture_formation_constraints

- To capture given Constraints affecting formations (e.g. EMCON).

capture_measurement_criteria_for_formations

- To capture provided Measurement_Criterion/criteria for Formation_Solutions and Delivered_Formations.

assess_formation_capability

- To assess the Capability to plan and execute Formation_Solutions taking account of Formation_Members' health and observed anomalies.

predict_capability_progression

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

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.

identify_missing_information

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

identify_whether_requirement_remains_achievable

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

B.2.21.5 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.

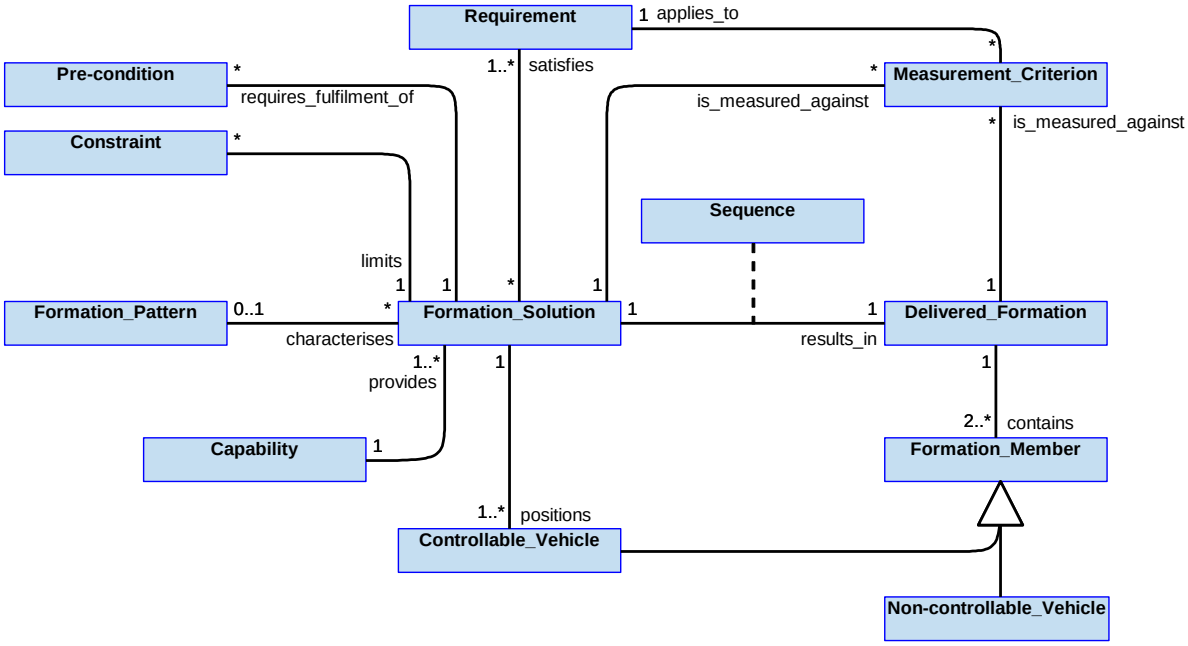

Figure 449: Formations Semantics

B.2.21.5.1 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_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).

Formation_Member

A vehicle belonging to the group of vehicles that are part of the formation.

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.

B.2.21.6 Design Rationale

B.2.21.6.1 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.

B.2.21.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy.
- Autonomy - Maintenance of Formations could involve a degree of autonomous movement of Formation_Member vehicles, which must be in accordance with the Autonomy policy.
- 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 policy. 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.

B.2.21.6.3 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.

B.2.21.6.4 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 unmanned 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.

B.2.21.7 Services

B.2.21.7.1 Service Definitions

B.2.21.7.1.1 Requirement

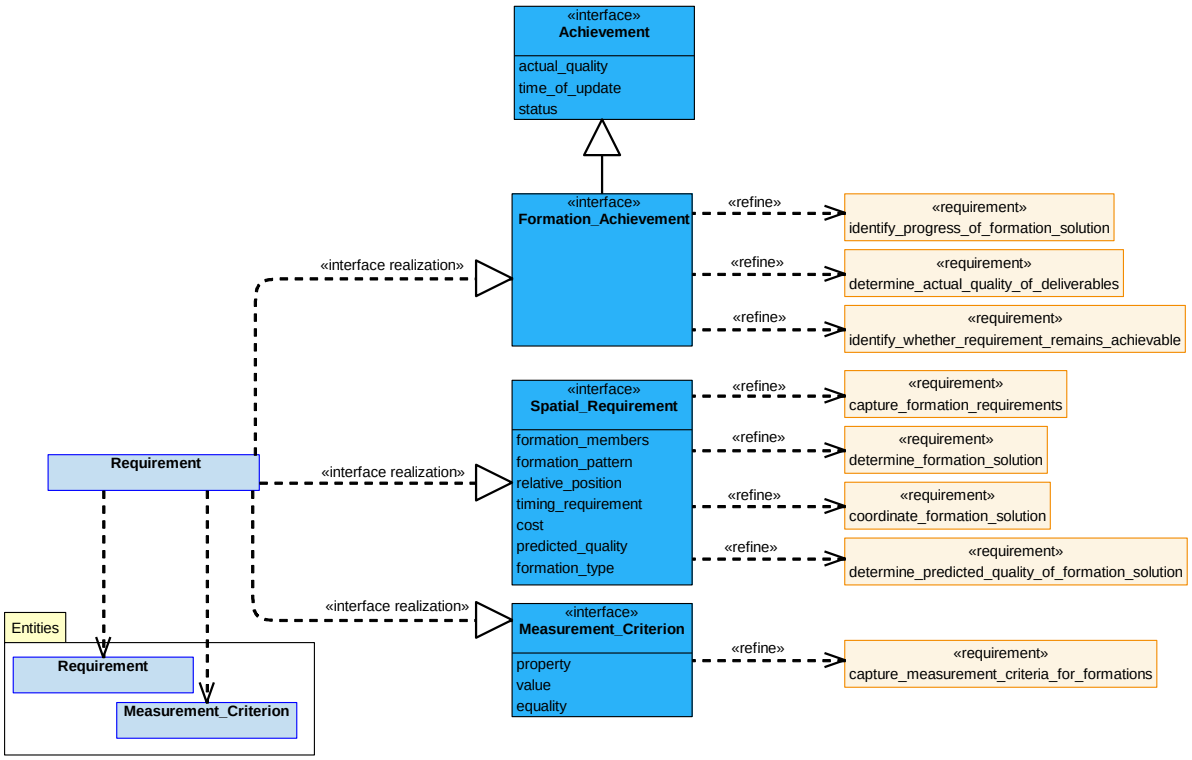

Figure 450: Requirement Service Definition

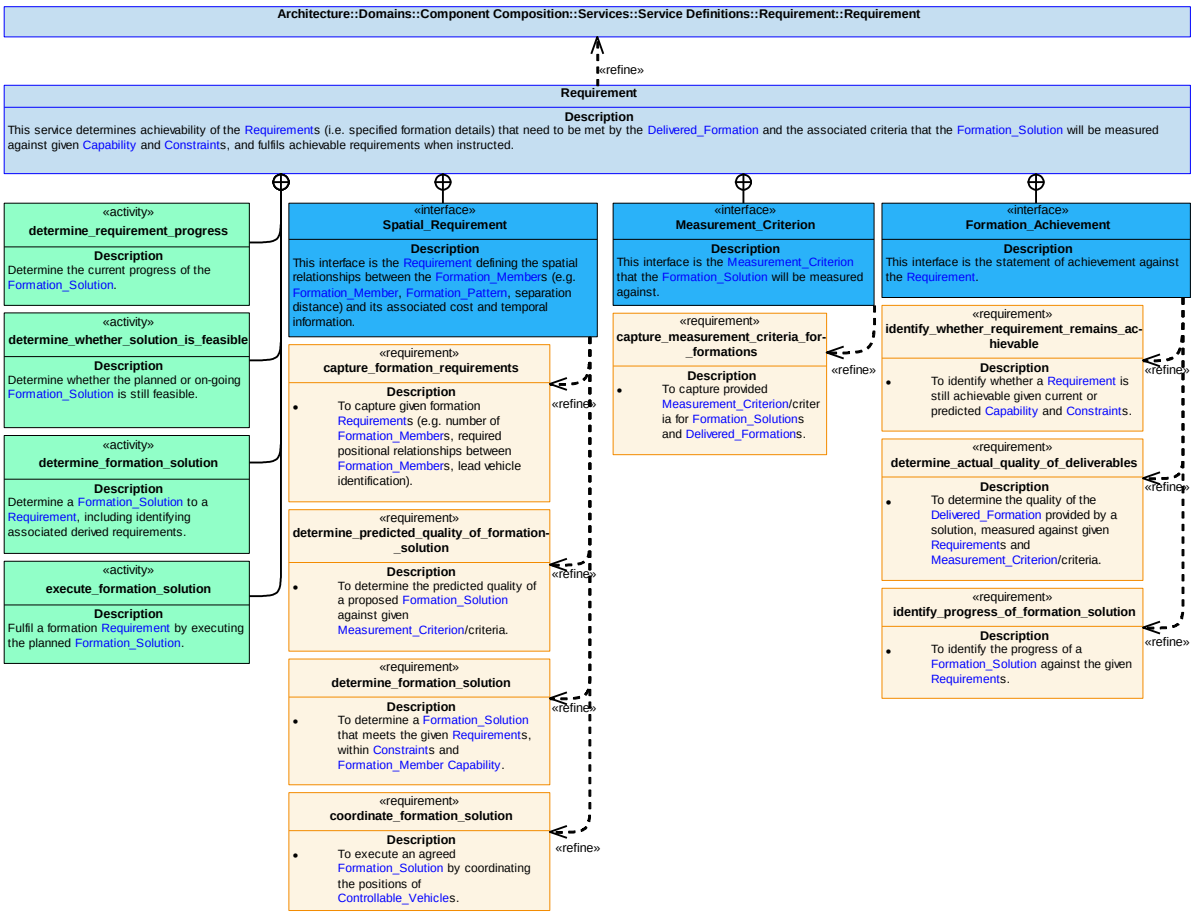

Figure 451: Requirement Service Policy

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 nm.
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.

B.2.21.7.1.2 Formation_Position

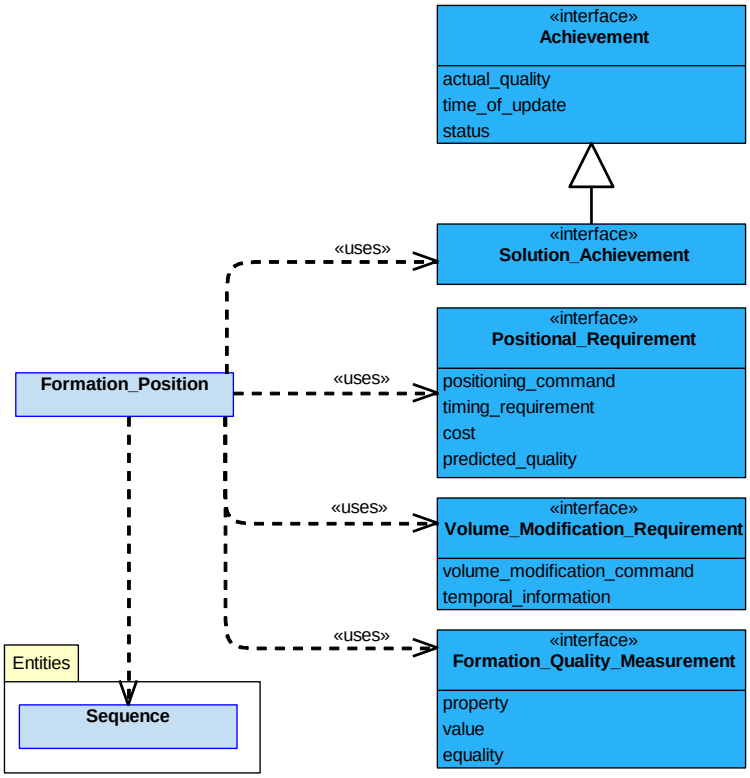

Figure 452: Formation_Position Service Definition

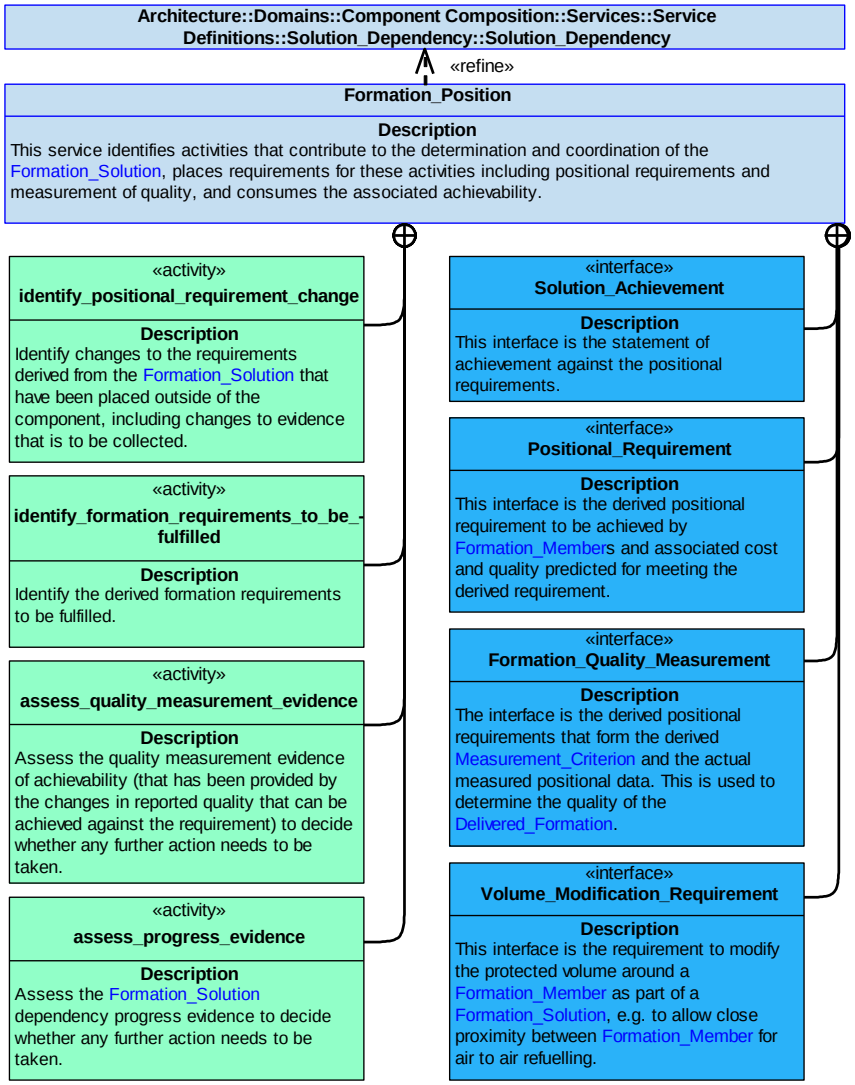

Figure 453: Formation_Position Service Policy

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.

B.2.21.7.1.3 Formation_Member

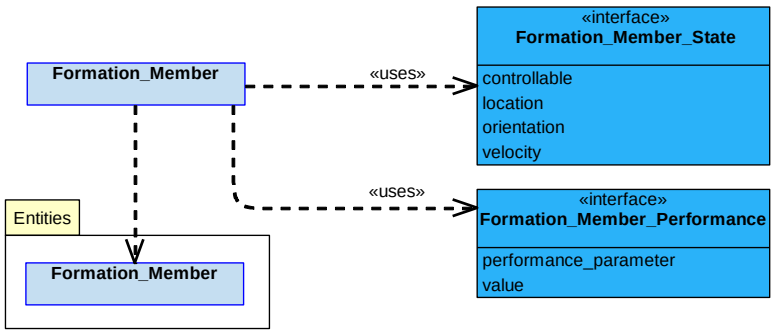

Figure 454: Formation_Member Service Definition

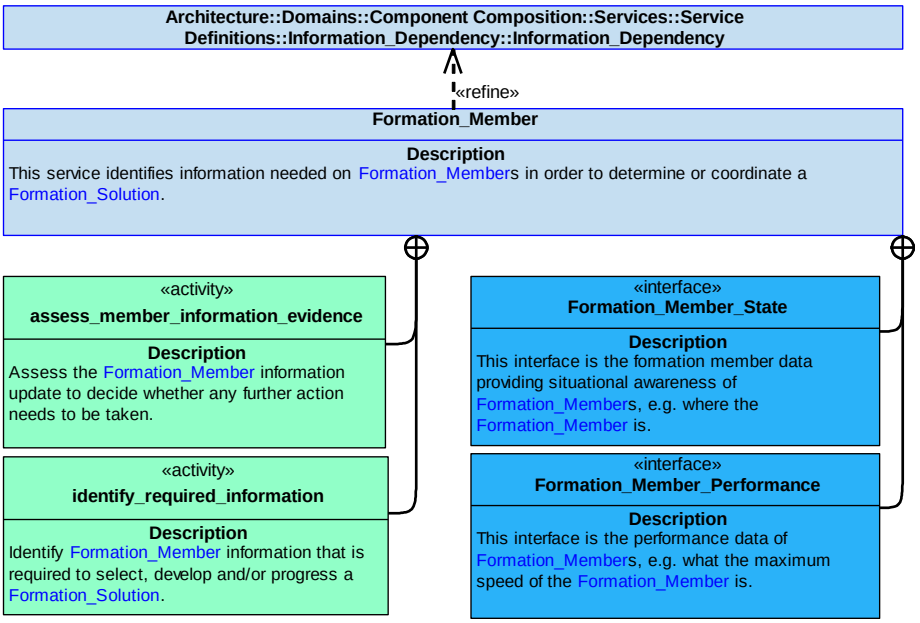

Figure 455: Formation_Member Service Policy

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 situational 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.

B.2.21.7.1.4 Constraint

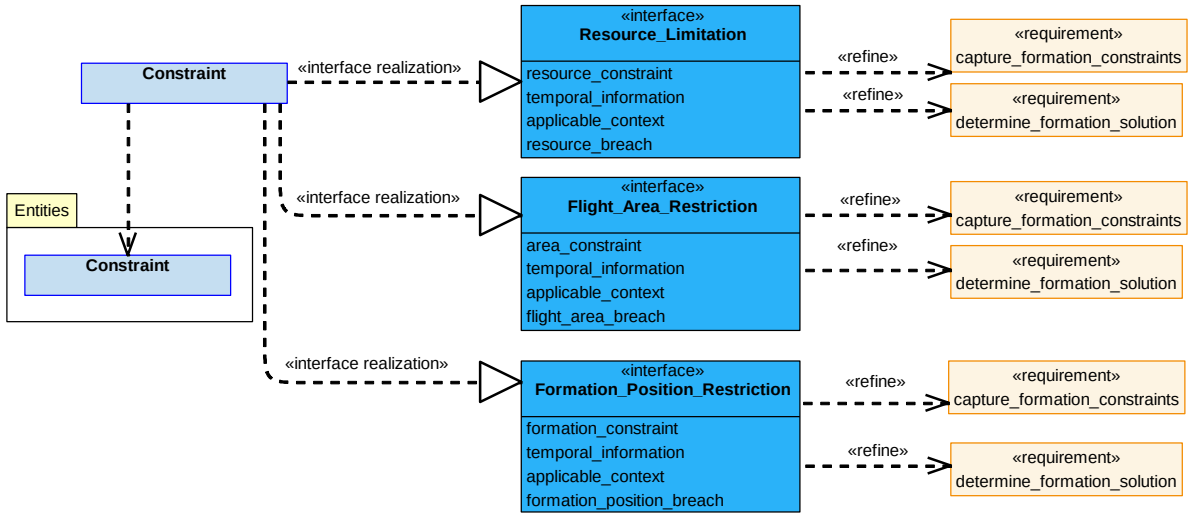

Figure 456: Constraint Service Definition

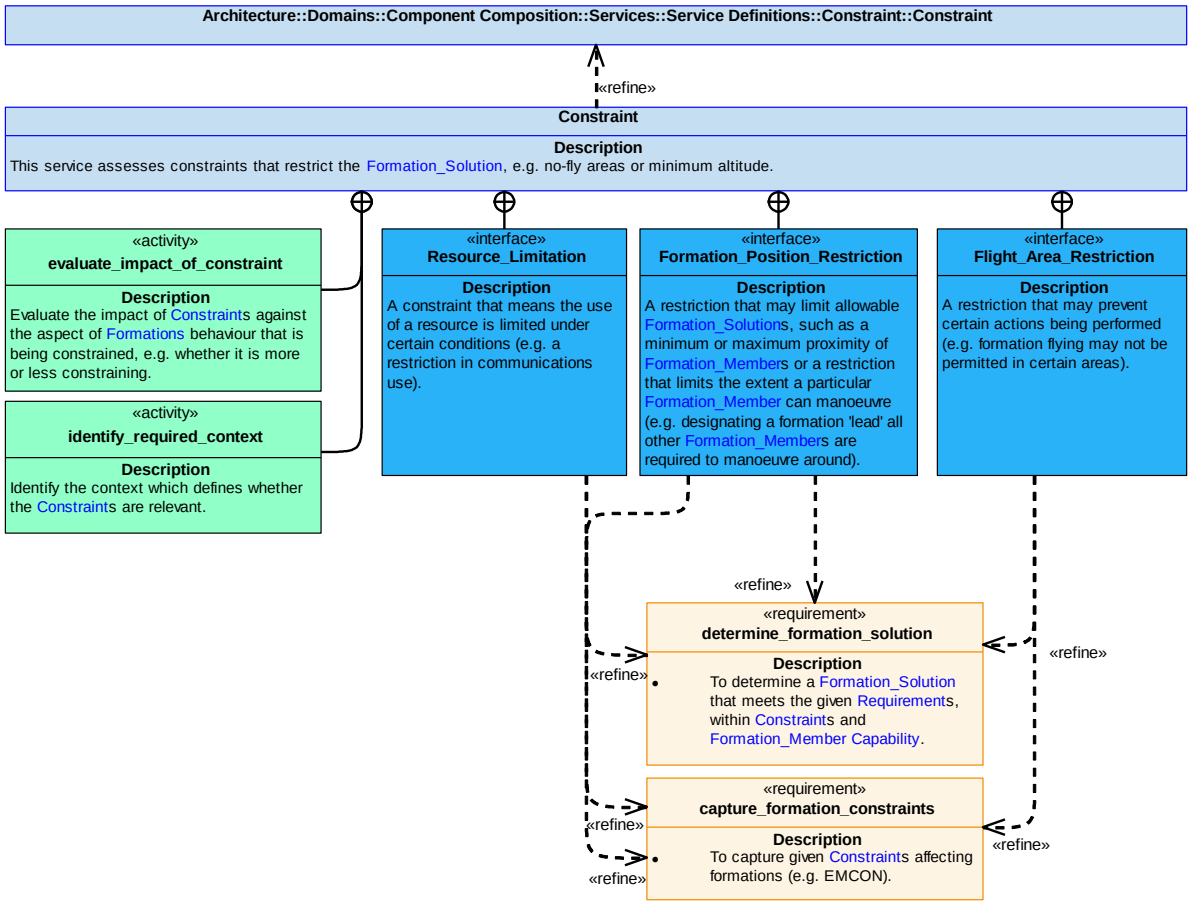

Figure 457: Constraint Service Policy

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.

B.2.21.7.1.5 Capability

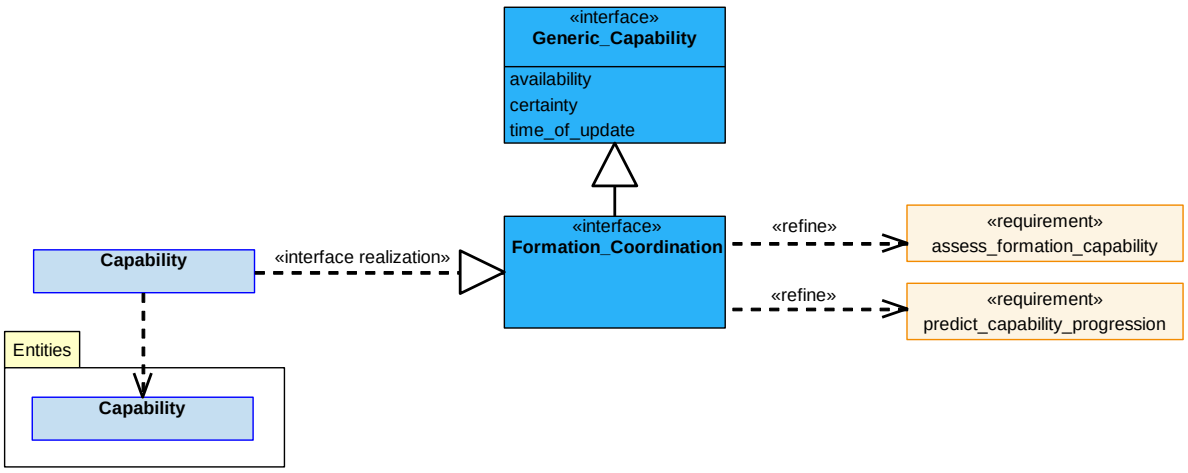

Figure 458: Capability Service Definition

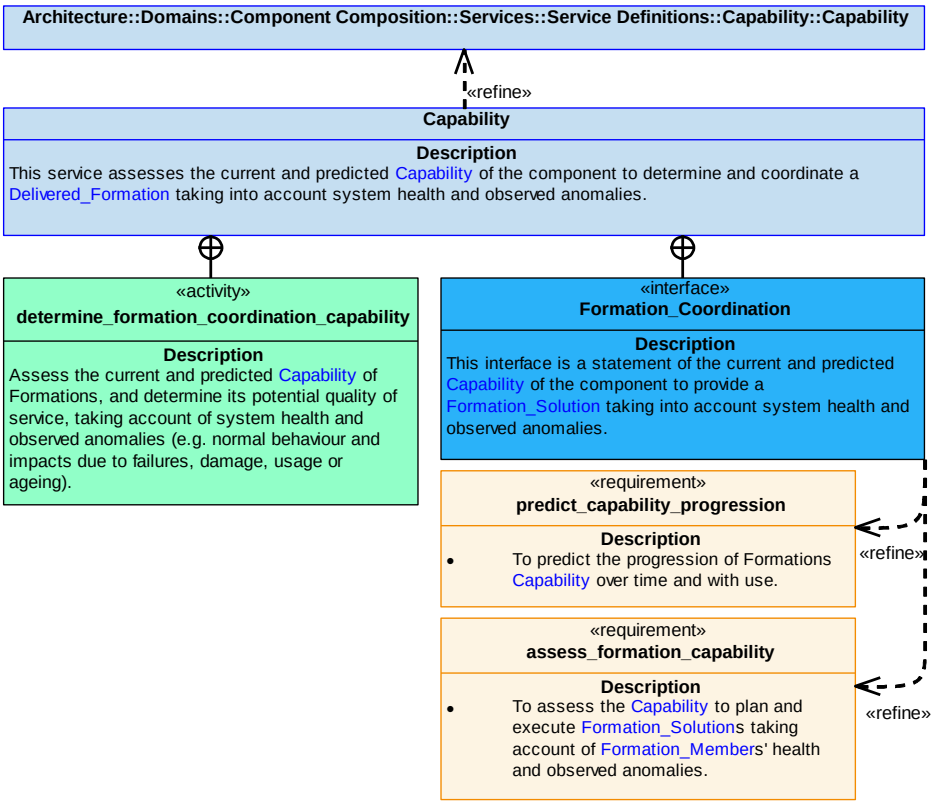

Figure 459: Capability Service Policy

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).

B.2.21.7.1.6 Capability_Evidence

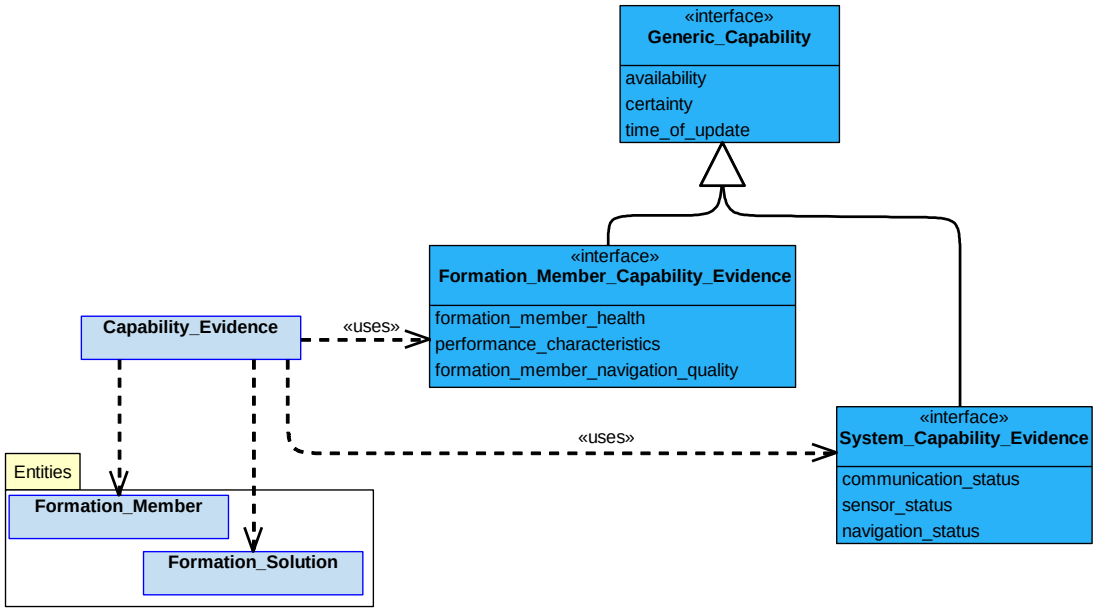

Figure 460: Capability_Evidence Service Definition

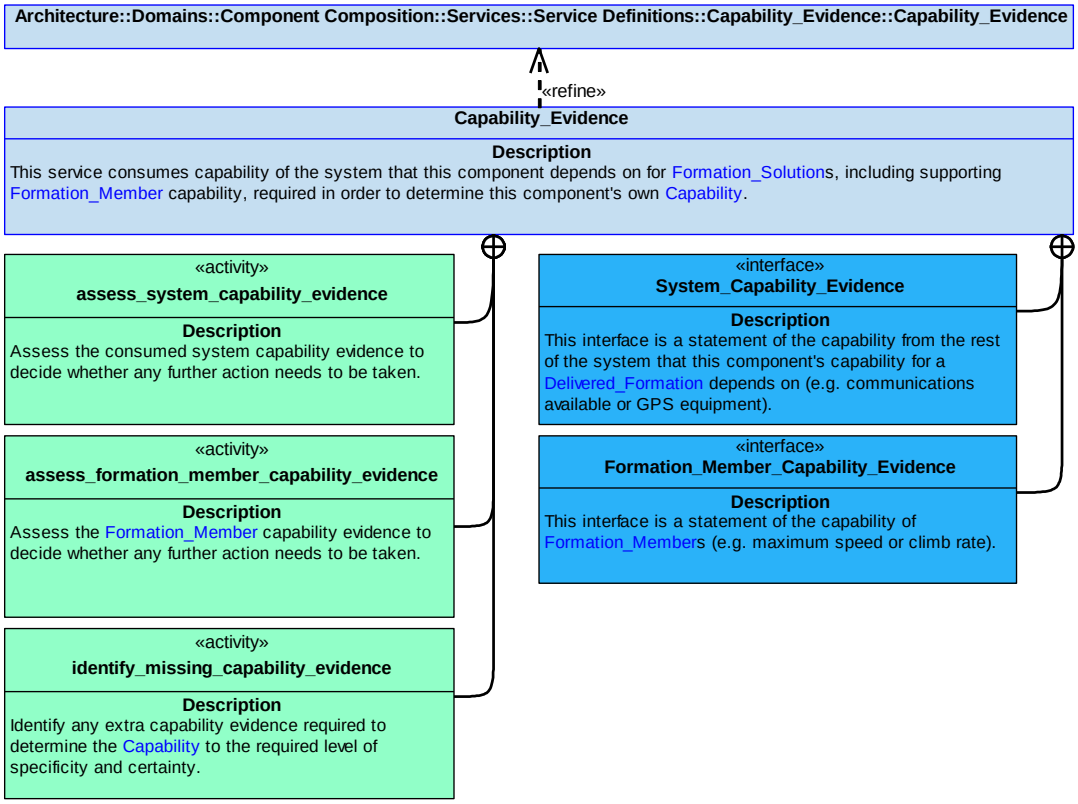

Figure 461: Capability_Evidence Service Policy

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, 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.

B.2.21.7.1.7 Formation_Observation

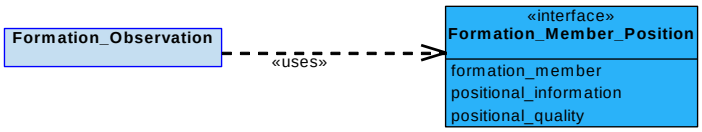

Figure 462: Formation_Observation Service Definition

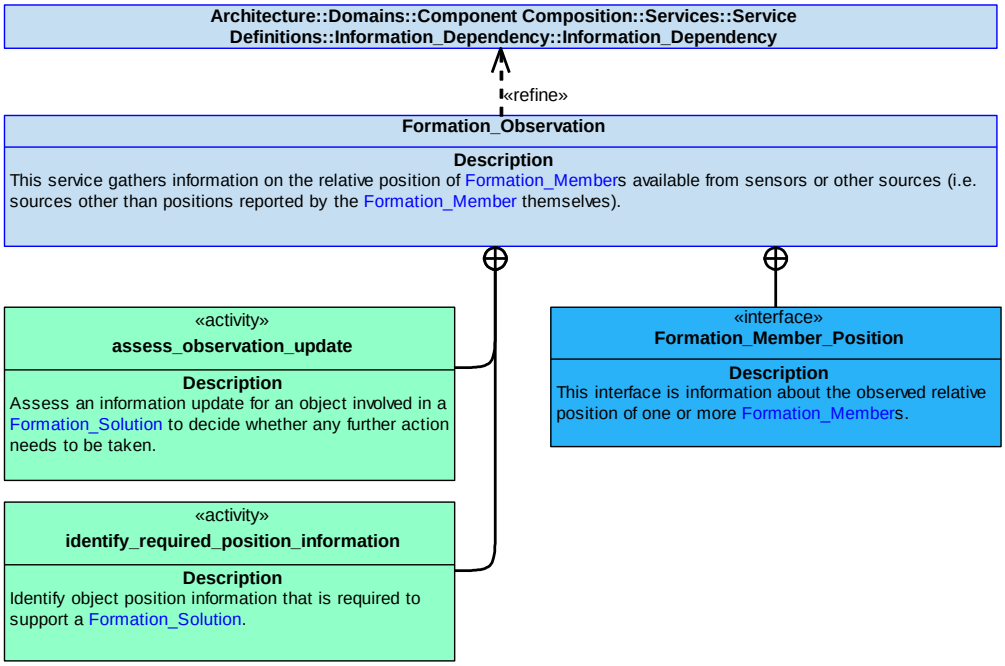

Figure 463: Formation_Observation Service Policy

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.

B.2.21.7.2 Service Dependencies

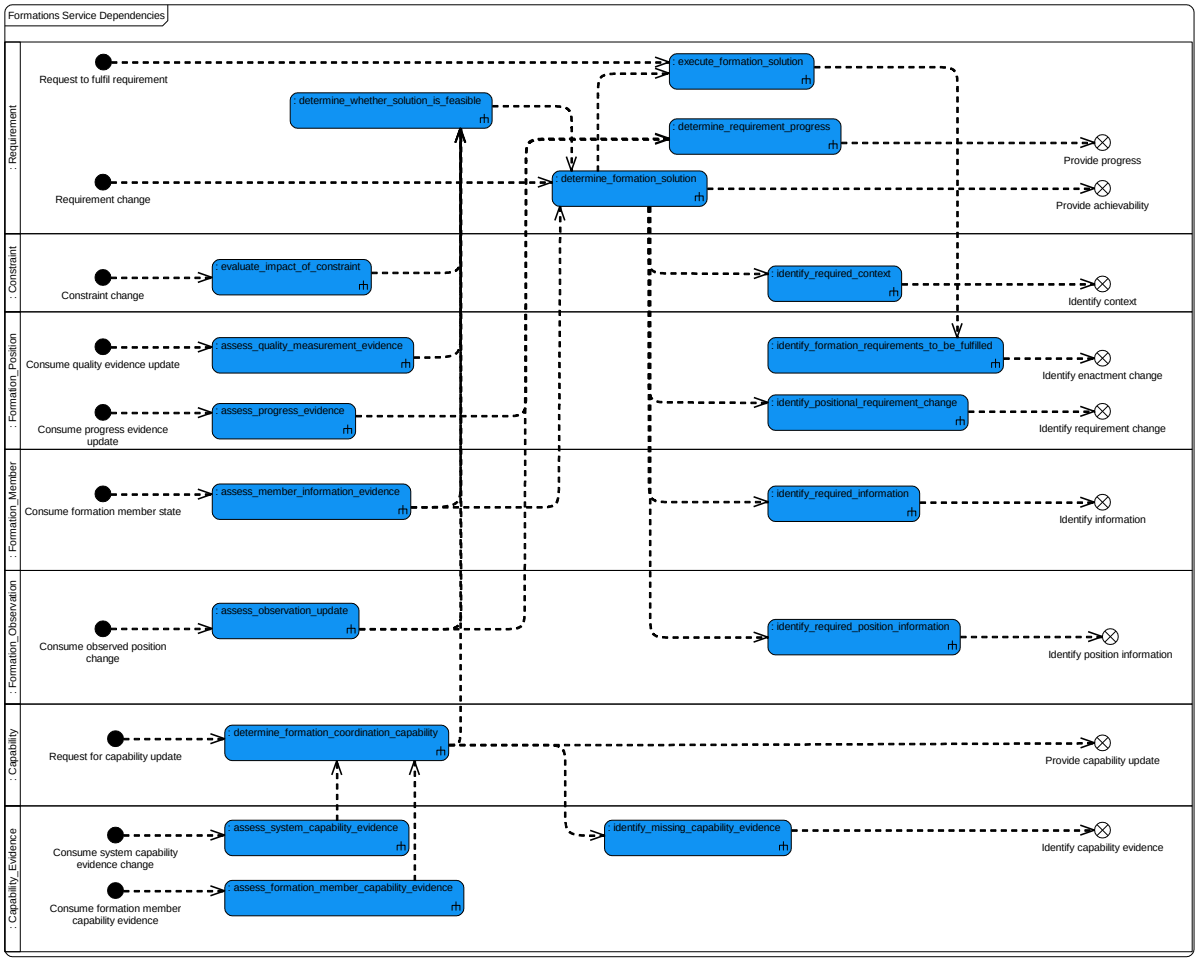

Figure 464: Formations Service Dependencies

B.2.22 Geography

B.2.22.1 Role

The role of Geography is to represent information about geographical features, including their location and the relationships between them.

B.2.22.2 Overview

Control Architecture

Geography is a service component as defined in the Control Architecture policy.

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.

B.2.22.3 Service Summary

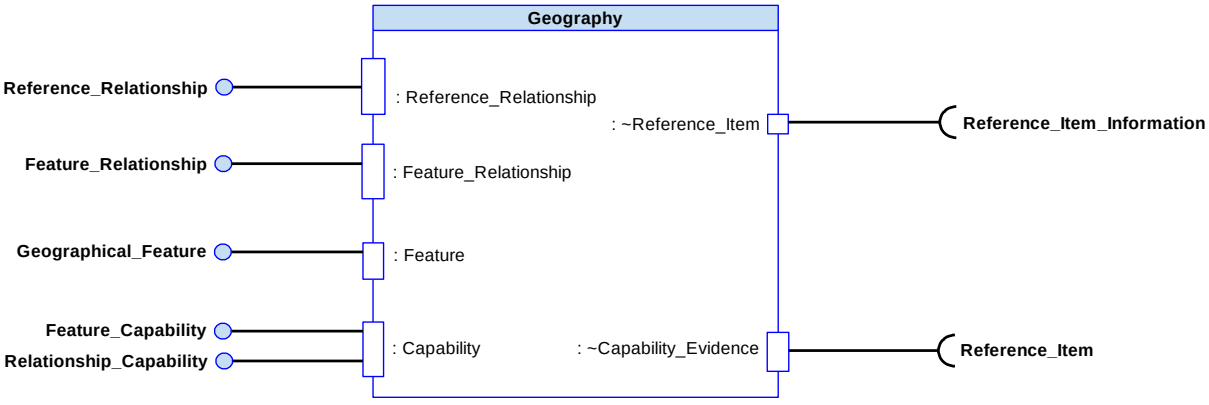

Figure 465: Geography Service Summary

B.2.22.4 Responsibilities

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.

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.

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.

B.2.22.5 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.

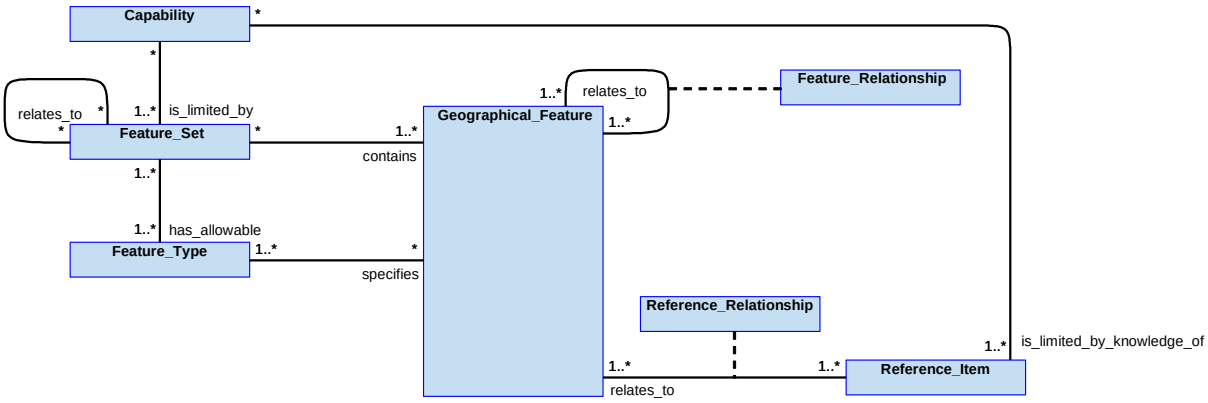

Figure 466: Geography Semantics

B.2.22.5.1 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_Set

A set of information on Geographical_Features, e.g. a terrain map or a military city map.

Feature_Relationship

The relationship between Geographical_Features, such as the distance between two buildings.

Feature_Type

A specific type of Geographical_Feature, e.g. a building, terrain, country border or territorial water border.

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.

Geographical_Feature

A geographical feature of the Earth, e.g. a specific river or the land border between two countries.

Reference_Item

A specific item, or a reference to, that is the subject of a request, e.g. ownship or projected route.

B.2.22.6 Design Rationale

B.2.22.6.1 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).

B.2.22.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Geography:

- Data Driving - This policy 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.

B.2.22.6.3 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.

B.2.22.6.4 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.

B.2.22.7 Services

B.2.22.7.1 Service Definitions

B.2.22.7.1.1 Reference_Relationship

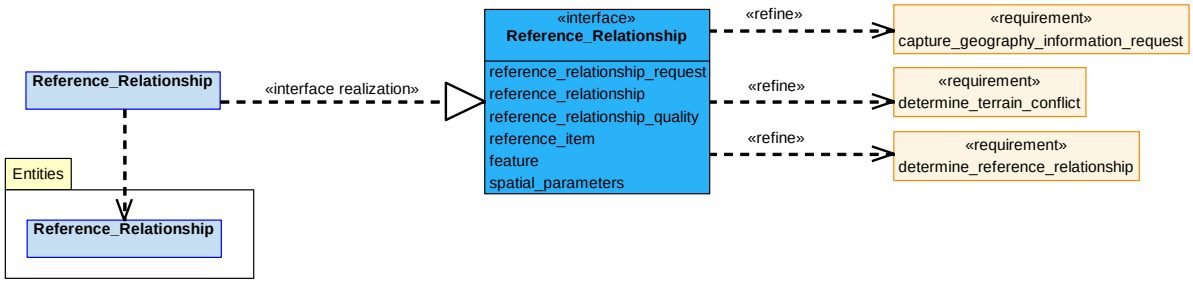

Figure 467: Reference_Relationship Service Definition

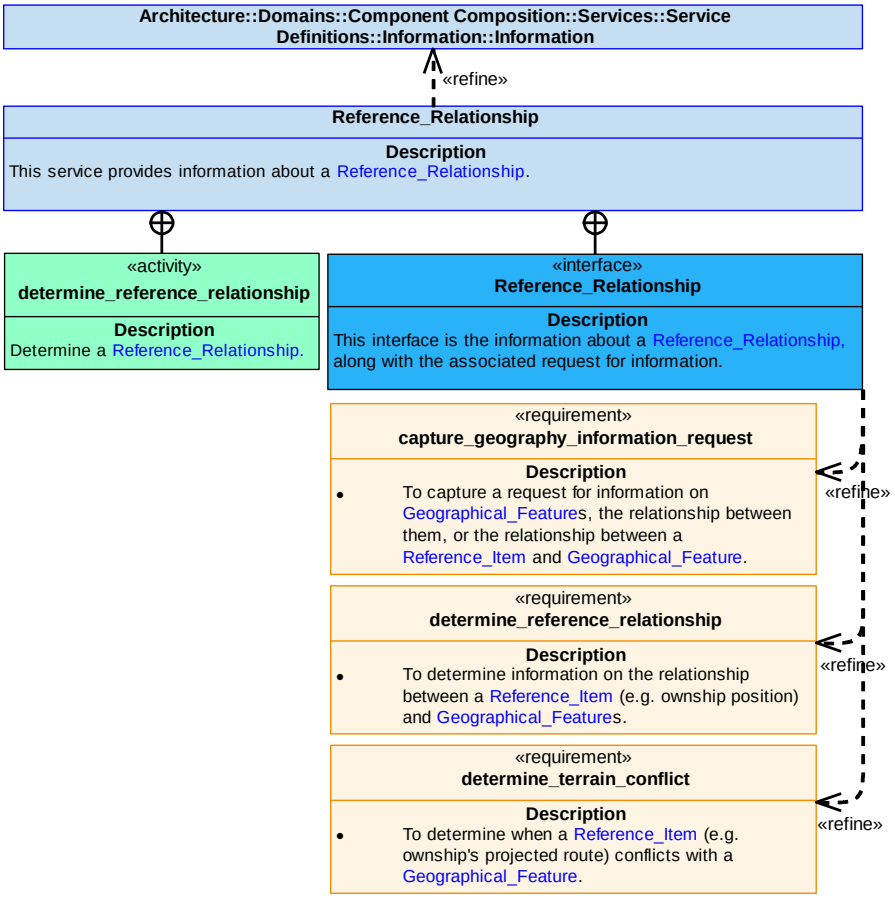

Figure 468: Reference_Relationship Service Policy

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.

B.2.22.7.1.2 Feature_Relationship

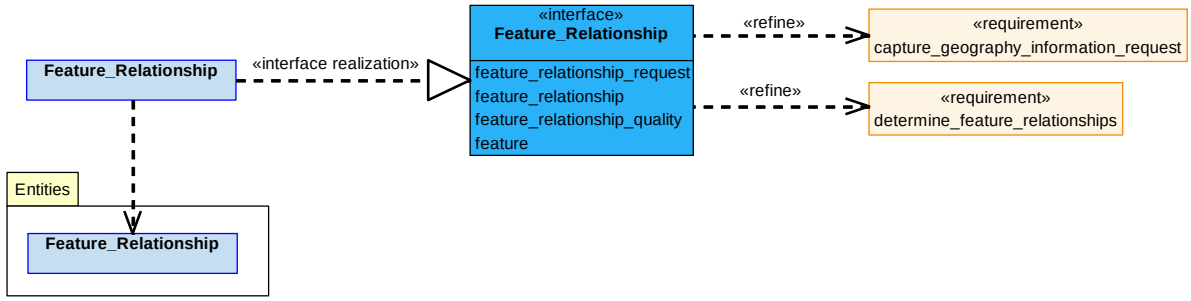

Figure 469: Feature_Relationship Service Definition

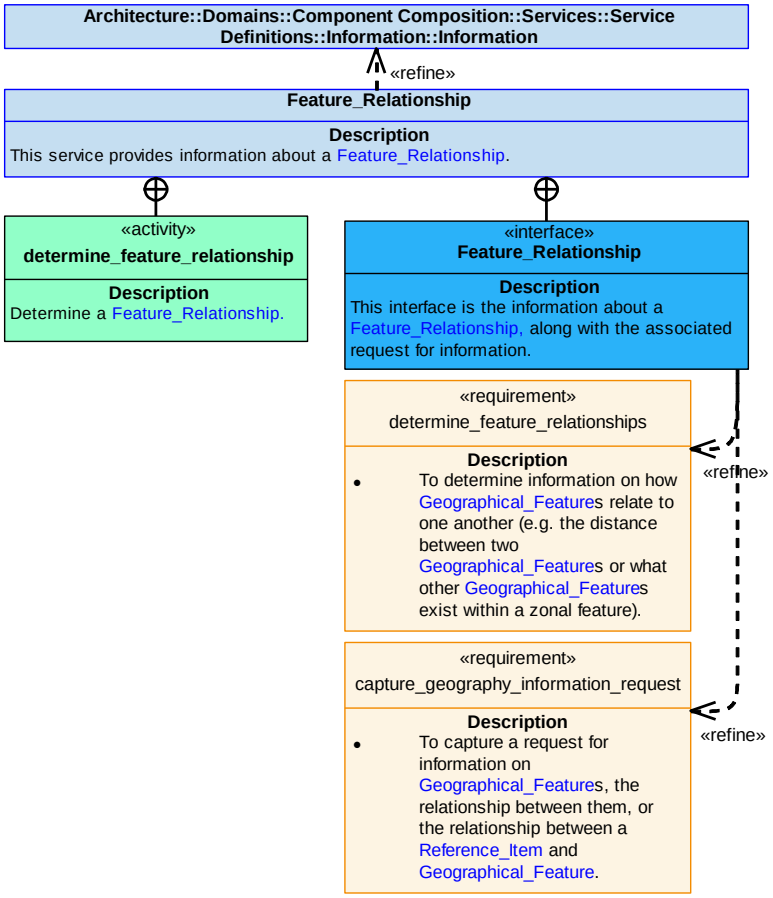

Figure 470: Feature_Relationship Service Policy

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.

B.2.22.7.1.3 Feature

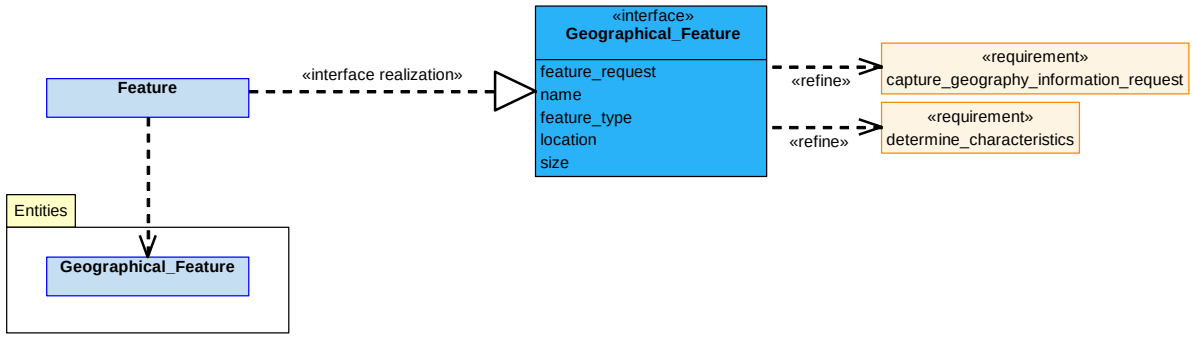

Figure 471: Feature Service Definition

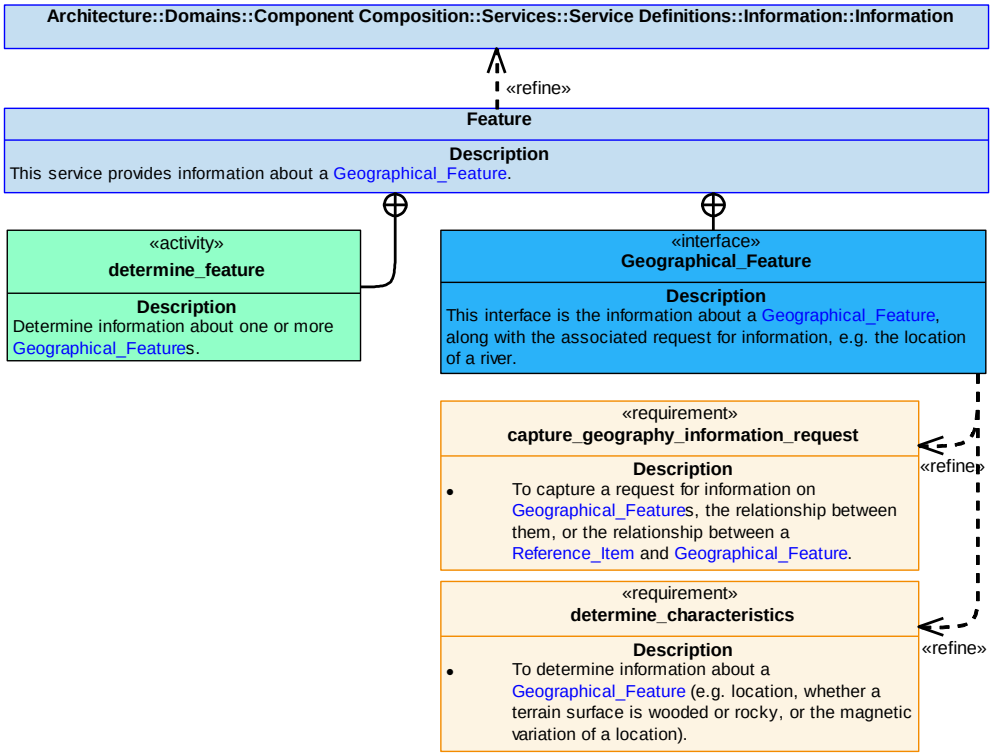

Figure 472: Feature Service Policy

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.

B.2.22.7.1.4 Reference_Item

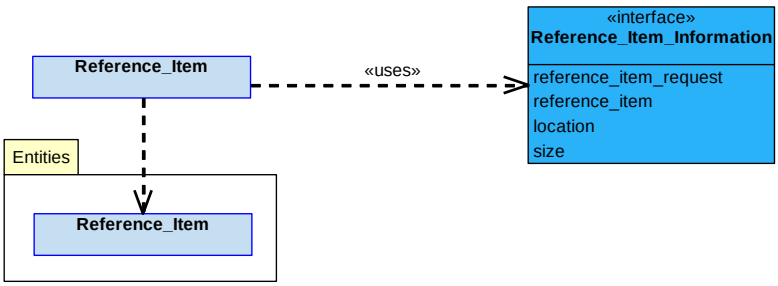

Figure 473: Reference_Item Service Definition

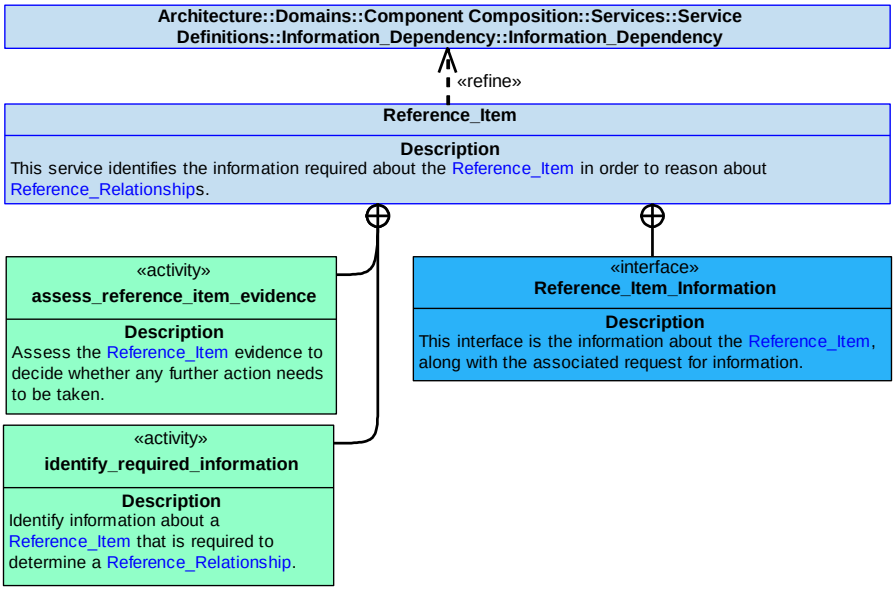

Figure 474: Reference_Item Service Policy

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.

B.2.22.7.1.5 Capability

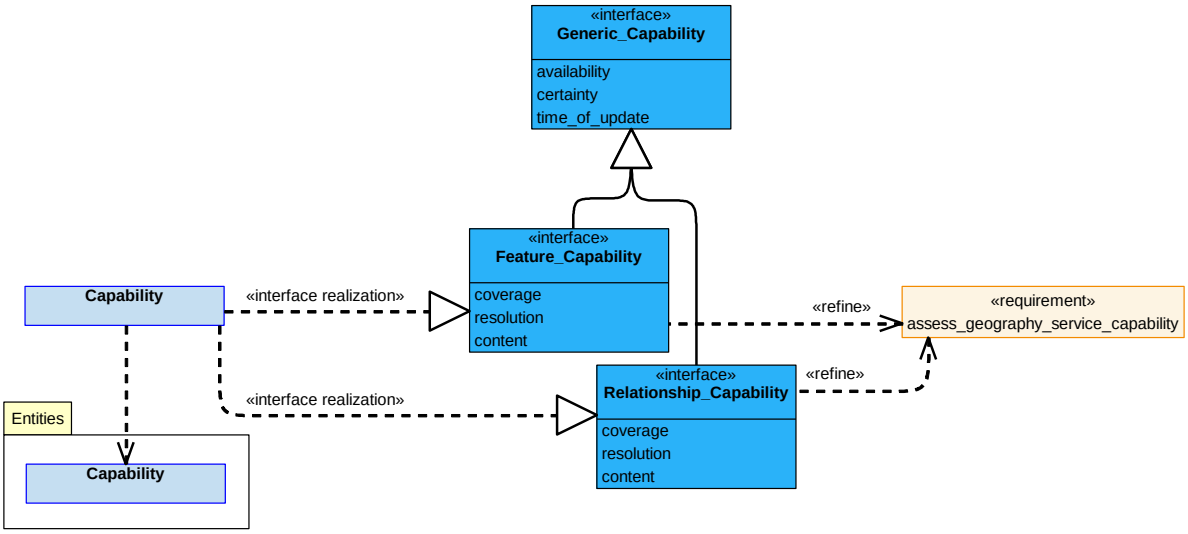

Figure 475: Capability Service Definition

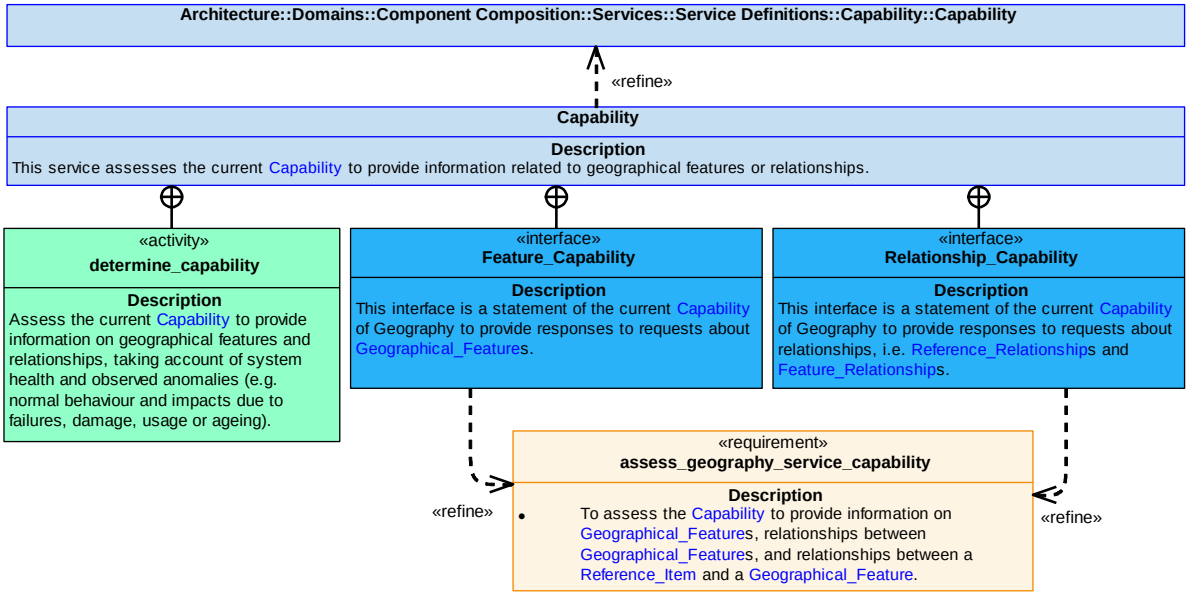

Figure 476: Capability Service Policy

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).

B.2.22.7.1.6 Capability_Evidence

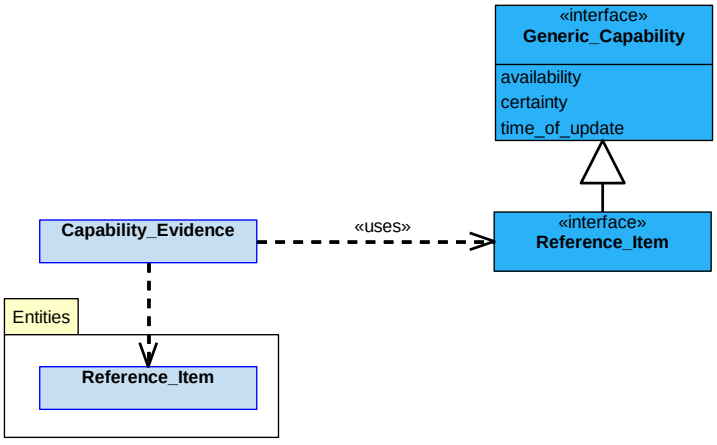

Figure 477: Capability_Evidence Service Definition

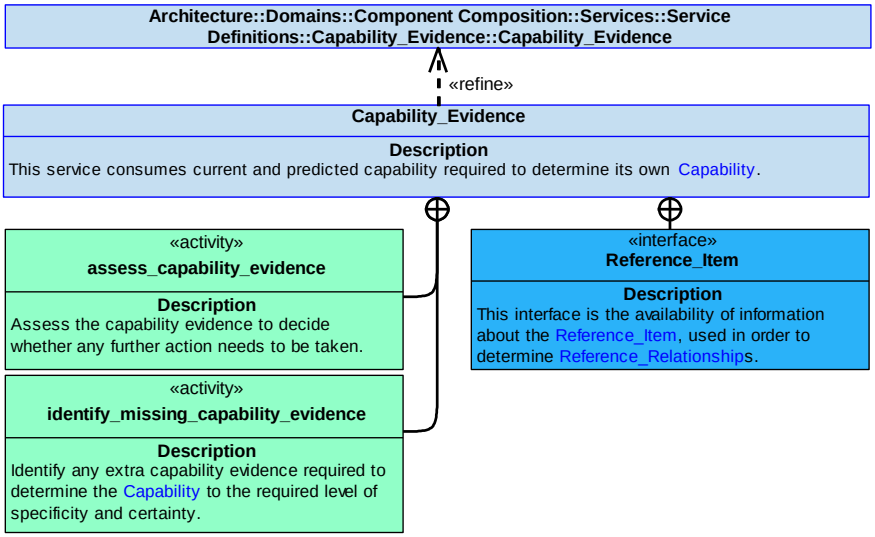

Figure 478: Capability_Evidence Service Policy

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.

B.2.22.7.2 Service Dependencies

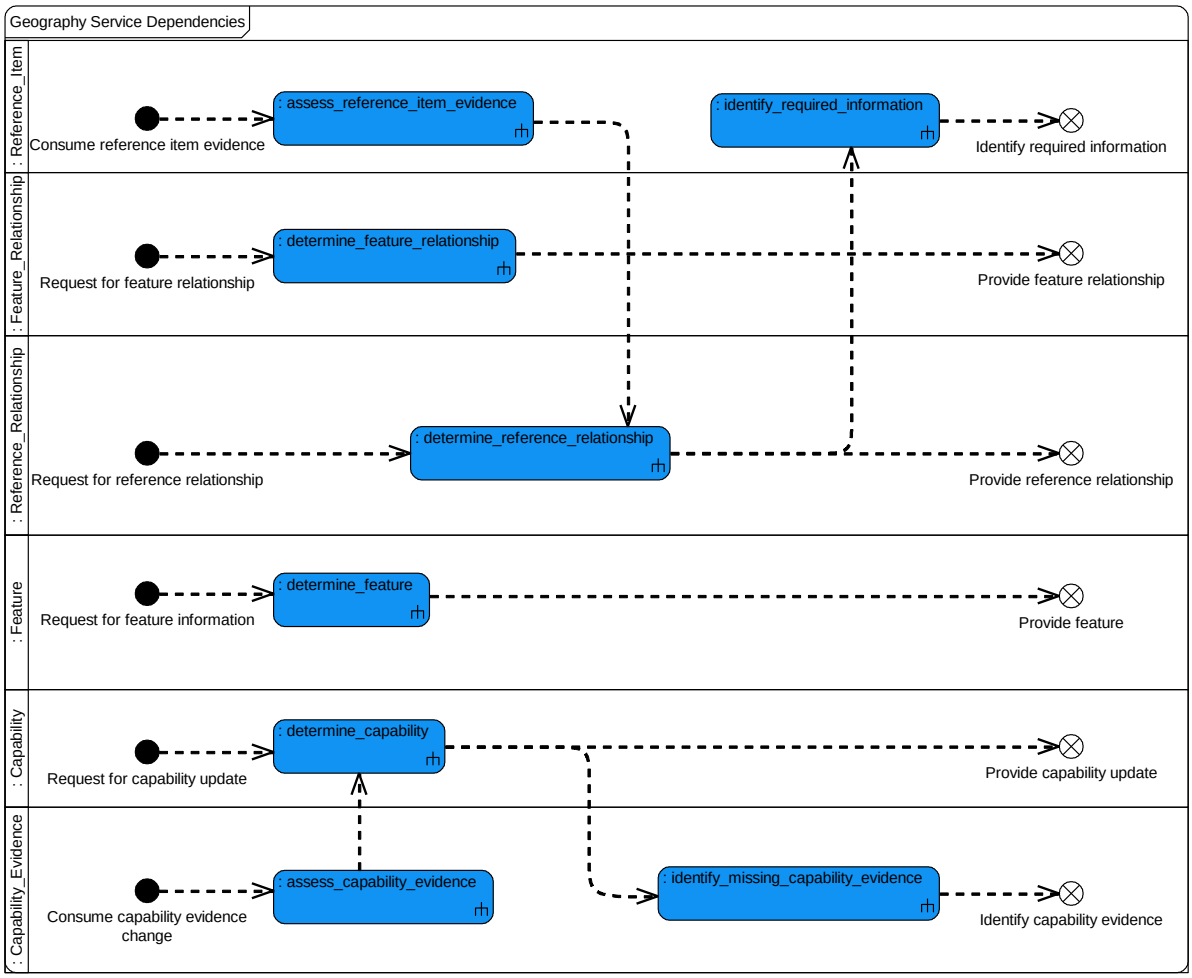

Figure 479: Geography Service Dependencies

B.2.23 Health Assessment

B.2.23.1 Role

The role of Health Assessment is to identify when a system hardware item has been degraded due to failure, damage, usage or ageing.

B.2.23.2 Overview

Control Architecture

Health Assessment is a service component as defined in the Control Architecture policy.

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.

B.2.23.3 Service Summary

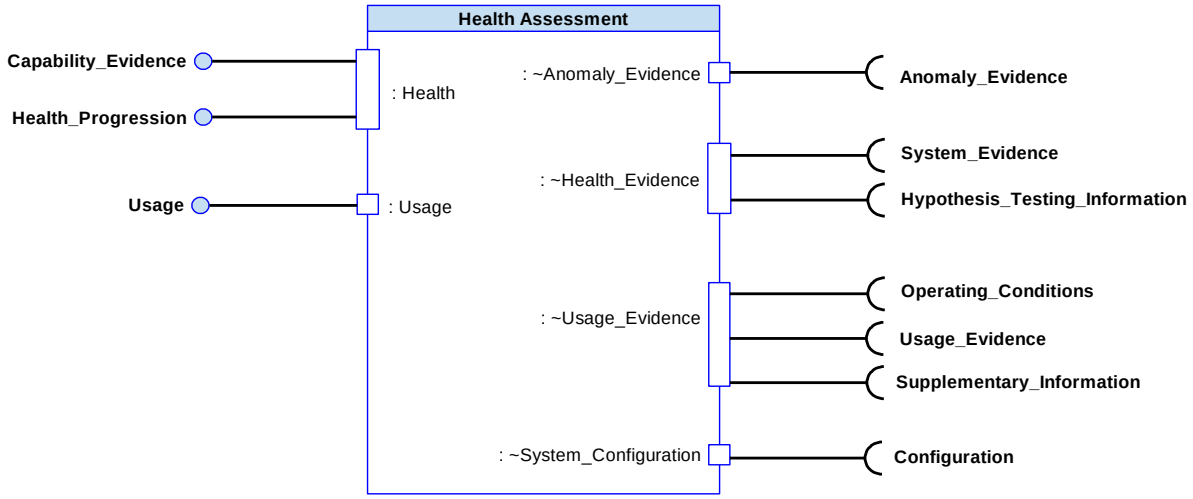

Figure 480: Health Assessment Service Summary

B.2.23.4 Responsibilities

identify_extent_of_degradation

- To identify the extent to which Hardware has been degraded by Failure, Damage, Usage or ageing.

predict_degradation

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

identify_missing_information_to_improve_health_solution

- To identify Missing_Health_Information which could improve the certainty or specificity of the health assessment.

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.).

B.2.23.5 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.

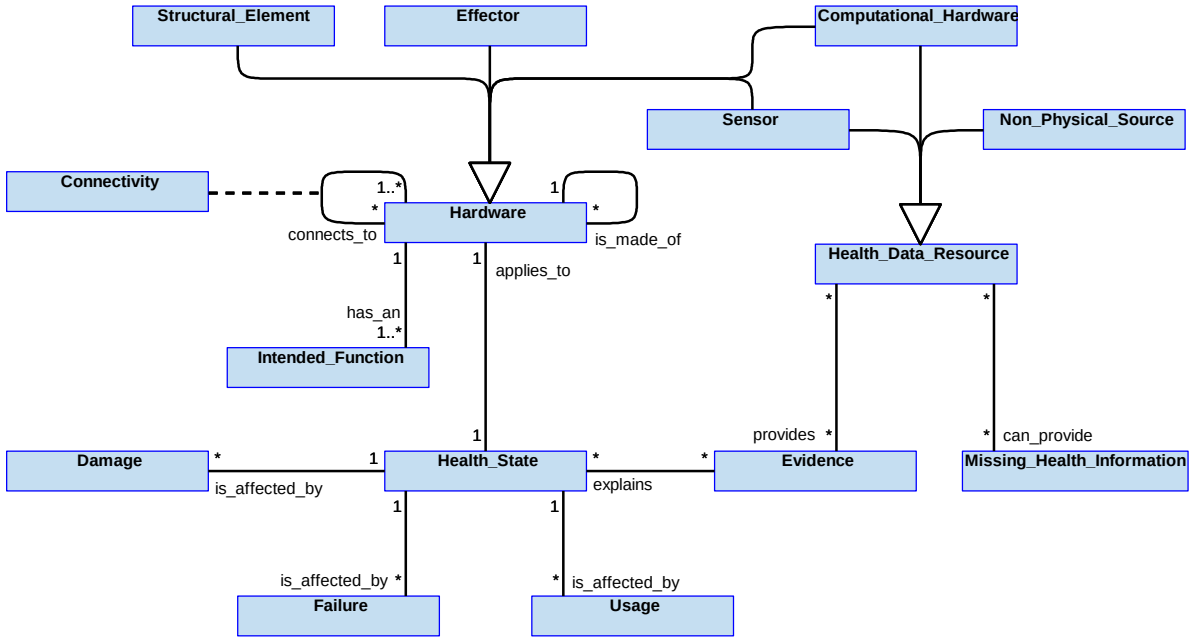

Figure 481: Health Assessment Semantics

B.2.23.5.1 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.

B.2.23.6 Design Rationale

B.2.23.6.1 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.

B.2.23.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Health Assessment:

- Health Management - This policy is important in understanding how this component is used.
- Capability Assessment - This component supports the capability assessment of other components but does not provide an evolving view of its own capabilities.
- Recording and Logging - This policy defines how data retention will be managed, especially for audit purposes.

Extensions

- Instead of using extensions, it is likely, except for very simple deployments, that Health Assessment components will be deployed hierarchically, with each instance being responsible for assessing the health of a particular area of the system. The Health Assessment components will 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.
- It is expected that instances of this component will be required to comply with ISO13374 (Condition Monitoring and Diagnostics of Machines) Ref. [18]. See Health Management policy.

B.2.23.6.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component may result a "significant reduction in safety margins", which has a major severity.

Therefore, the indicative DAL is C.

The rationale for this is that the capability assessment functions of each component in conjunction with the related instance of the Anomaly Detection component would detect and report the relevant failures, such that they can be appropriately mitigated, without any reliance on Health Assessment. This component may result in the cause of a fault to be misdiagnosed but at worst this may result in unnecessary mission abort or maintainer workload.

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.

B.2.23.6.4 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.

B.2.23.7 Services

B.2.23.7.1 Service Definitions

B.2.23.7.1.1 Health

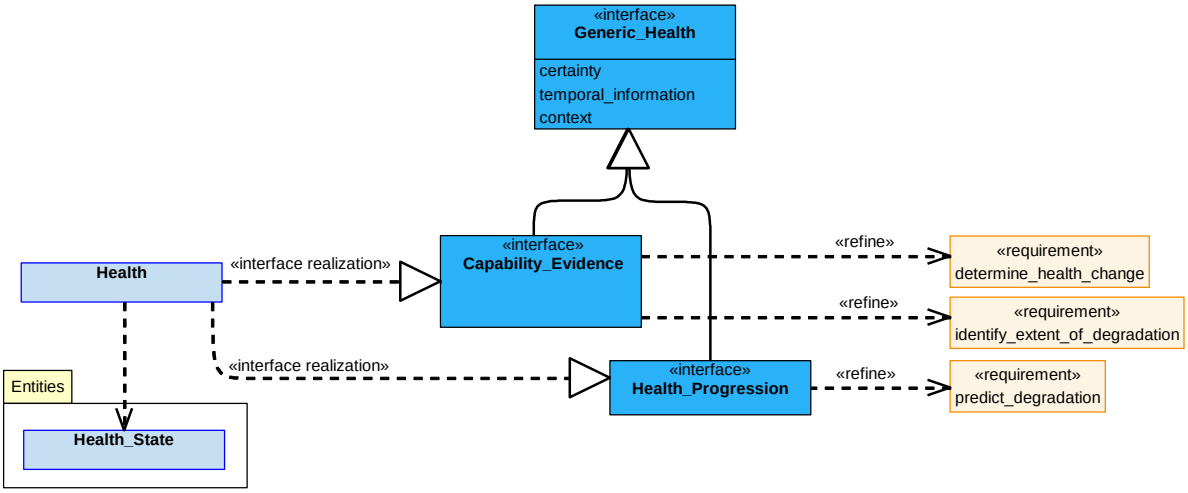

Figure 482: Health Service Definition

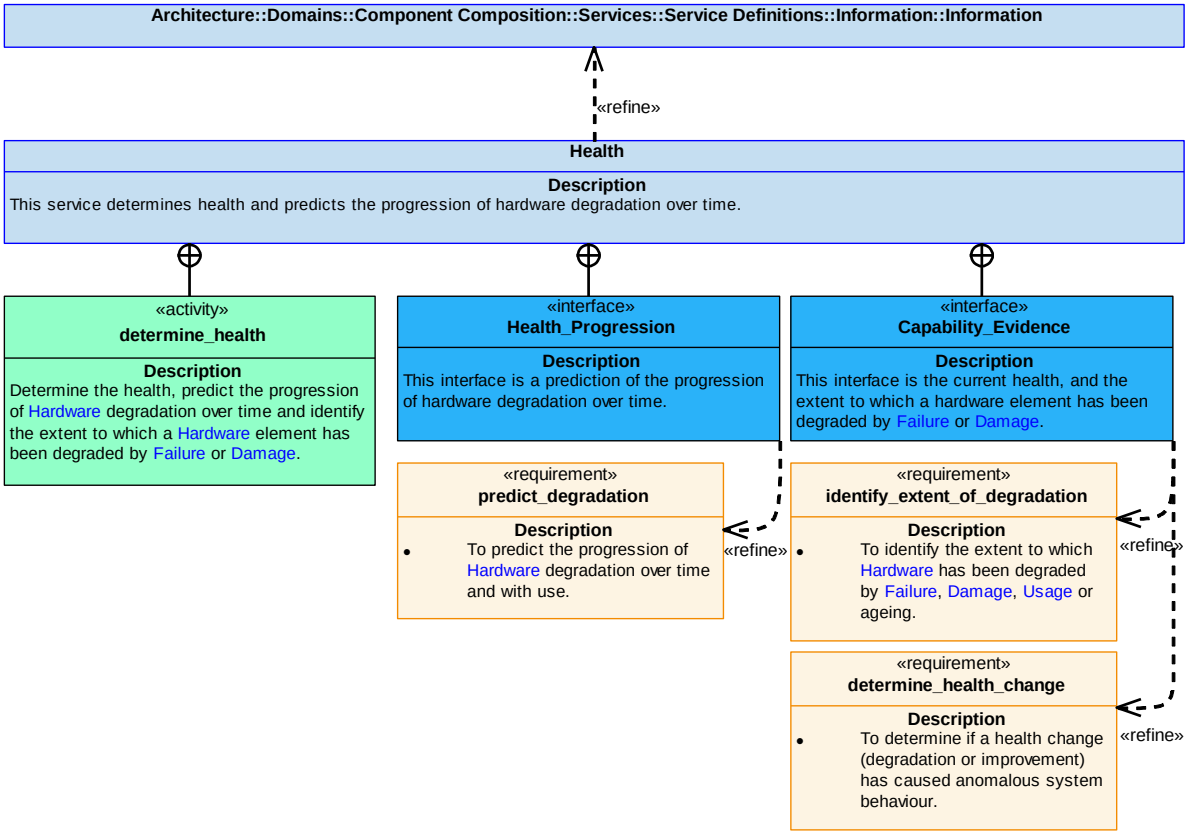

Figure 483: Health Service Policy

Health

This service determines health and predicts the progression of hardware degradation over time.

Interfaces

Capability_Evidence

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.

B.2.23.7.1.2 Usage

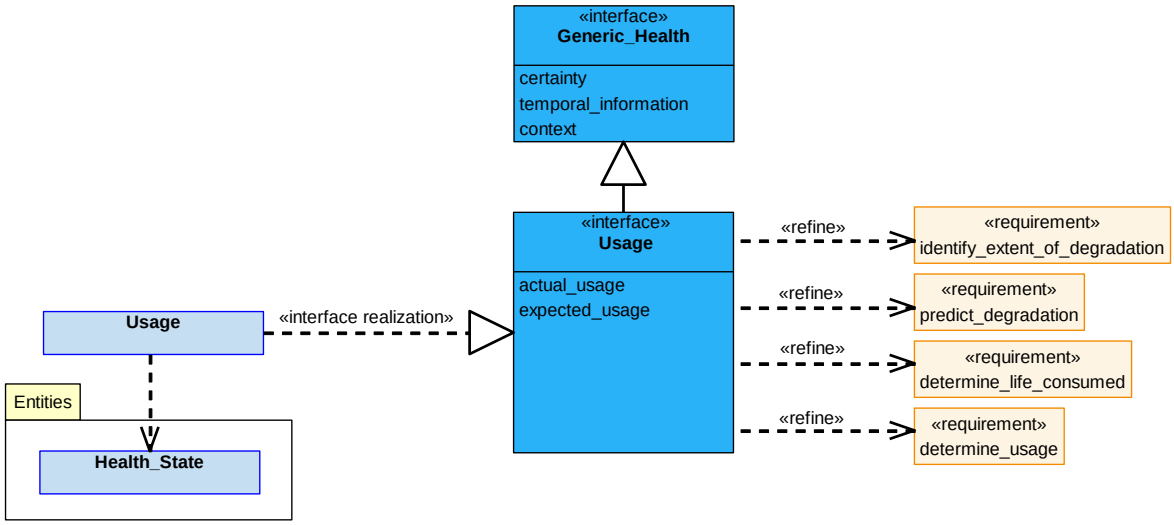

Figure 484: Usage Service Definition

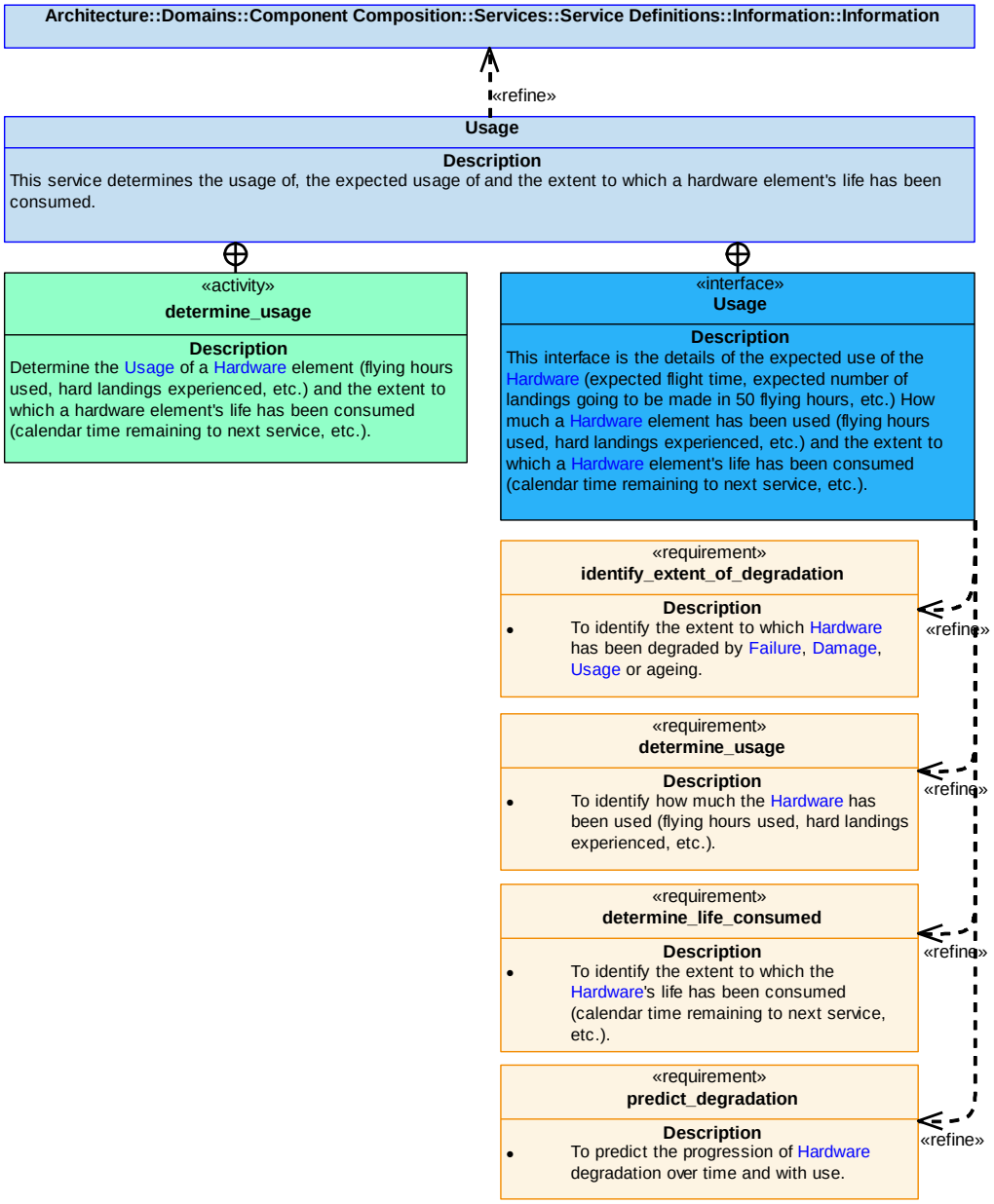

Figure 485: Usage Service Policy

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.).

B.2.23.7.1.3 Anomaly_Evidence

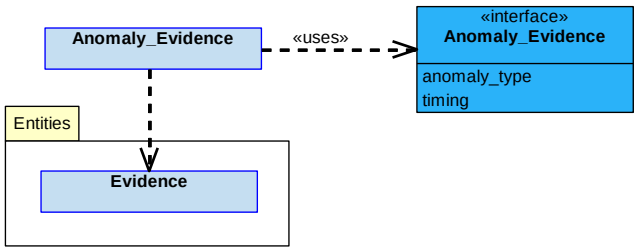

Figure 486: Anomaly_Evidence Service Definition

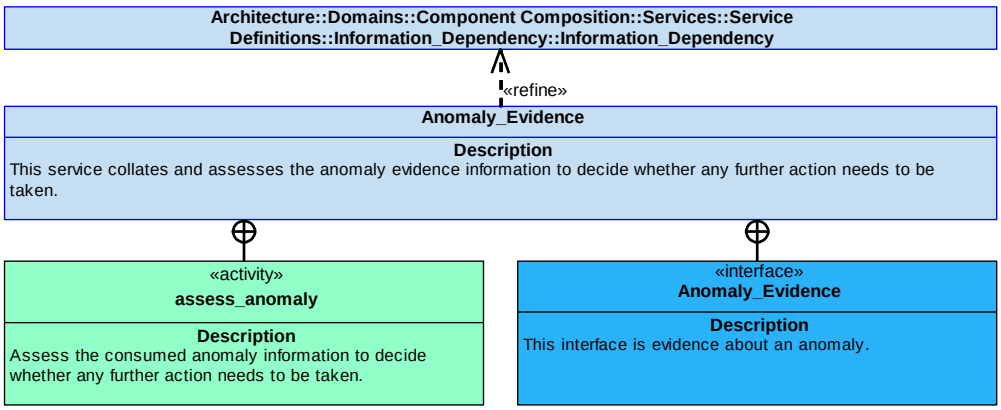

Figure 487: Anomaly_Evidence Service Policy

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.

B.2.23.7.1.4 Health_Evidence

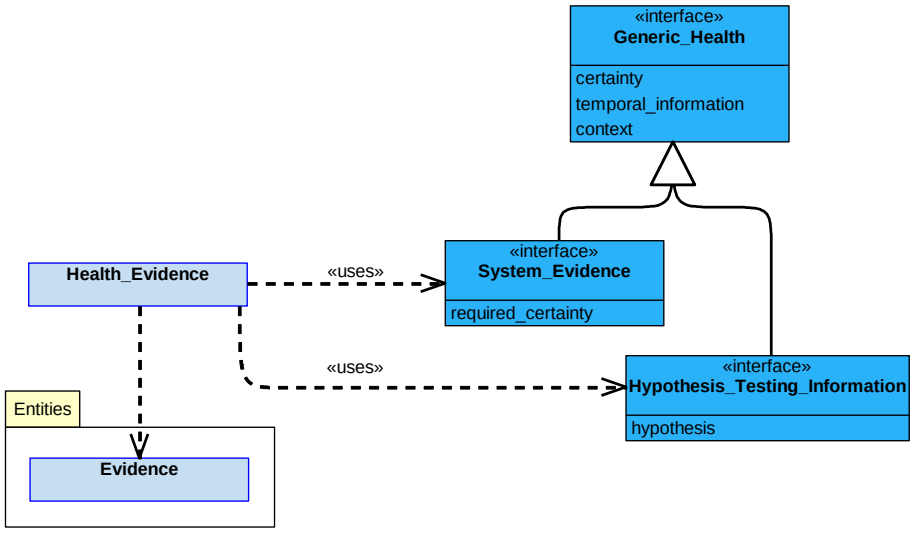

Figure 488: Health_Evidence Service Definition

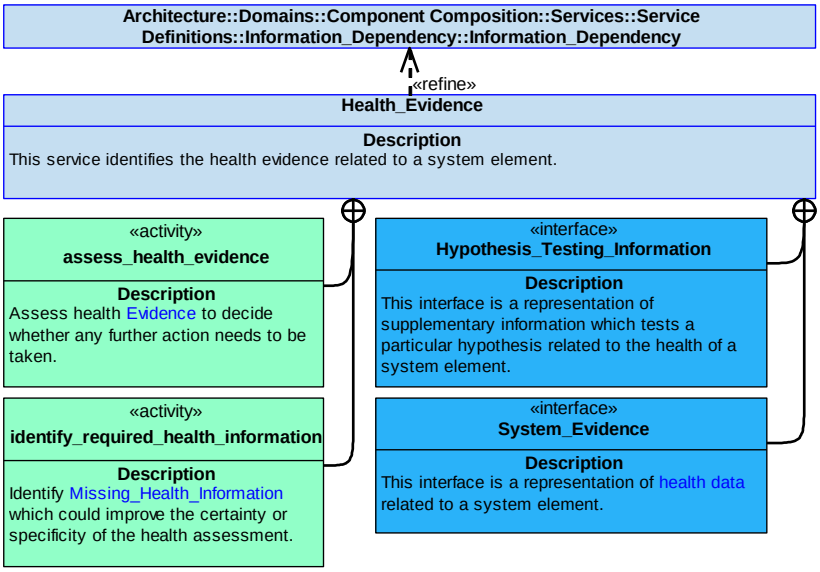

Figure 489: Health_Evidence Service Policy

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, could this element have failed in this 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.

B.2.23.7.1.5 Usage_Evidence

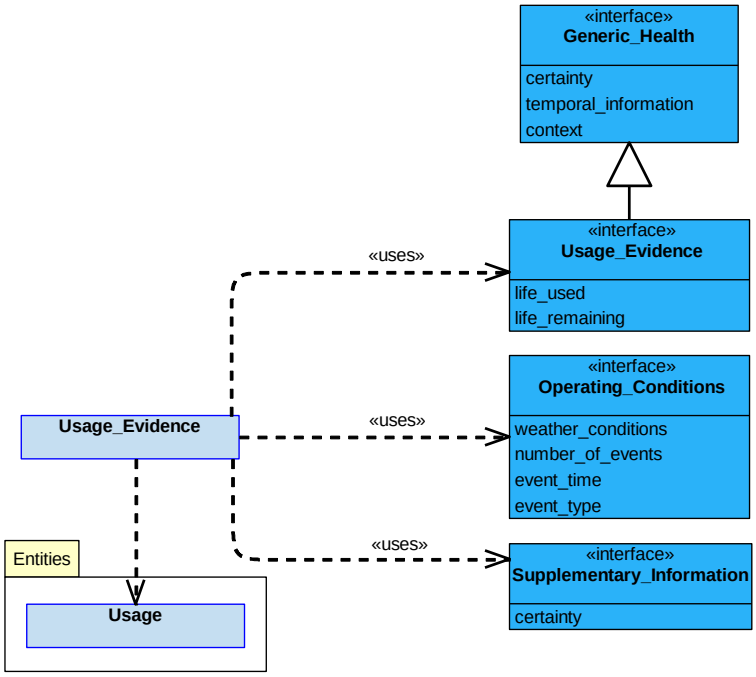

Figure 490: Usage_Evidence Service Definition

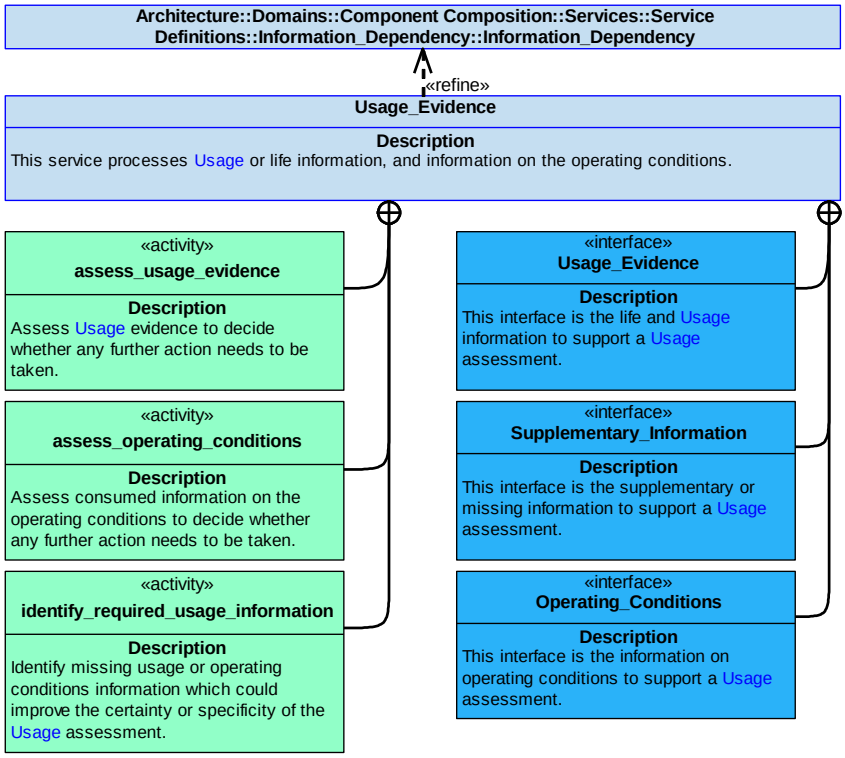

Figure 491: Usage_Evidence Service Policy

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.

B.2.23.7.1.6 System_Configuration

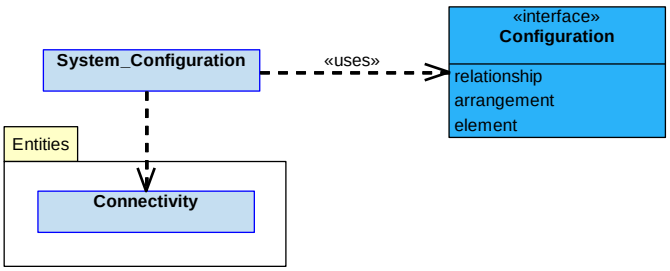

Figure 492: System_Configuration Service Definition

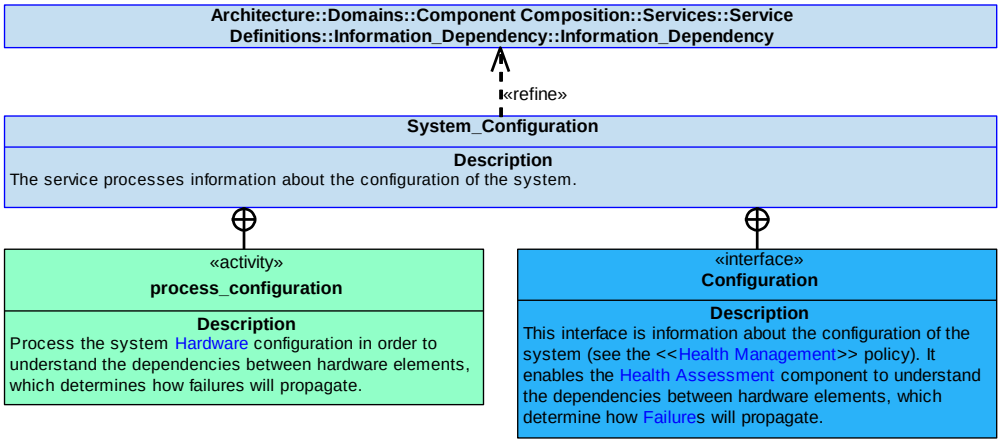

Figure 493: System_Configuration Service Policy

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 policy). 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.

B.2.23.7.2 Data Model

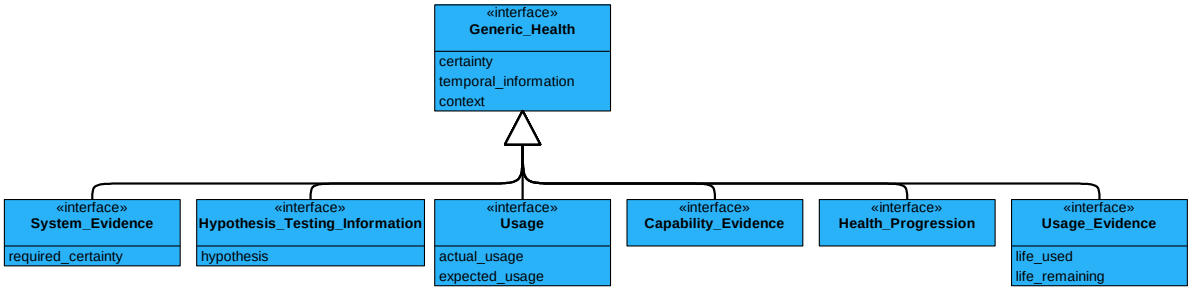

Figure 494: Health Assessment Data Model

The diagram shows a Generic_Health class which individual components will specialise if they are required to undertake health assessment. The generic class allows the components to communicate essential information without reference to specialisations.

Generic_Health

This interface is a generic expression of health assessment that can be specialised to specific components.

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.

B.2.23.7.3 Service Dependencies

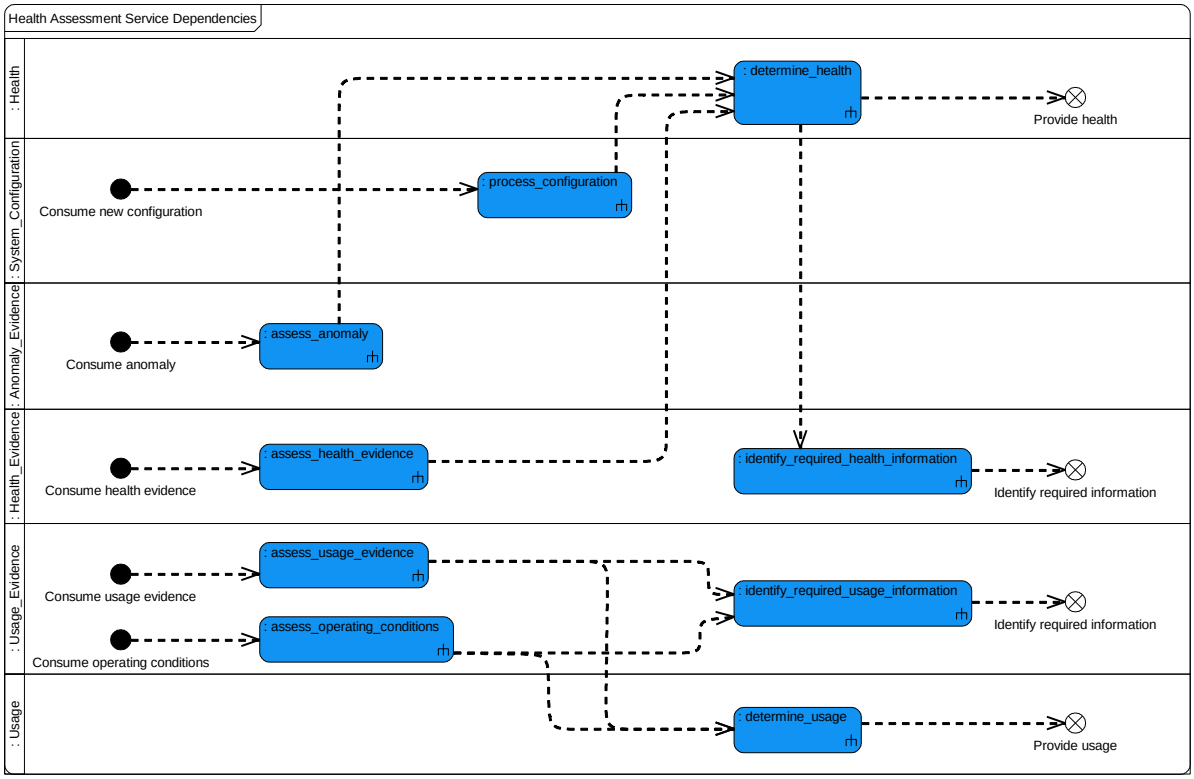

Figure 495: Health Assessment Service Dependencies

B.2.24 HMI Dialogue

B.2.24.1 Role

The role of HMI Dialogue is to manage and curate the information required for interactions between a system and its users.

B.2.24.2 Overview

Control Architecture

HMI Dialogue is a service component as defined in the Control Architecture policy.

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.

B.2.24.3 Service Summary

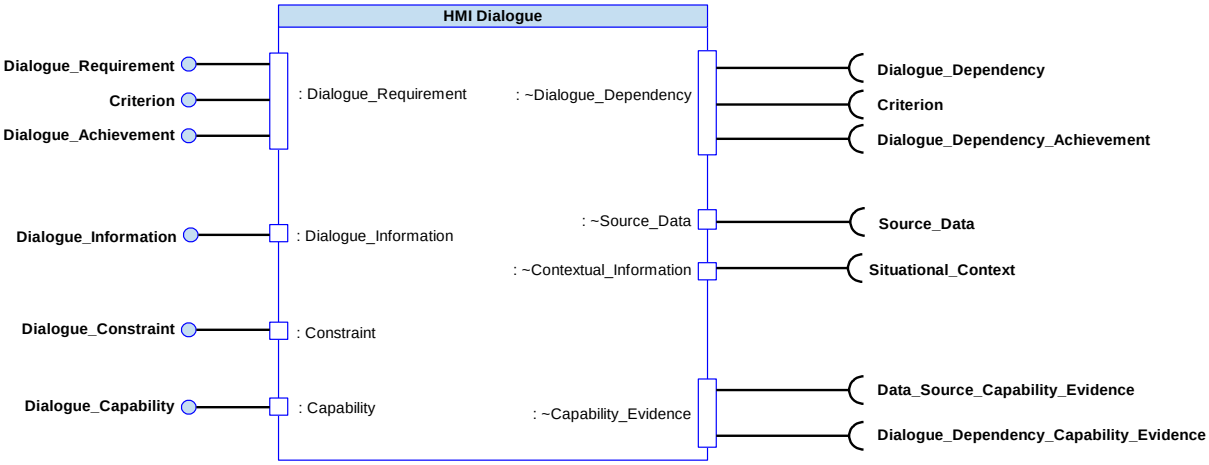

Figure 496: HMI Dialogue Service Summary

B.2.24.4 Responsibilities

assess_dialogue_capability

- To assess the component's Capability to provide a Dialogue_Interaction, taking account of system health and observed anomalies.

predict_capability_progression

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

identify_missing_information

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

capture_participant_relationships

- To capture Participant relationships and associated Comprehension_Rules.

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_information_dependencies

- To capture information dependencies for Dialogue_Interactions, such as a Context change or the condition of Participants.

capture_provided_data_constraints

- To capture Constraints on the provided data that limit the extent of possible dialogue.

determine_comprehension_rules

- To determine Comprehension_Rules for Source_Data in the context of the associated dialogue, within any Constraints.

fulfil_dialogue_requirement

- To derive Presentable_Information in support of dialogue.

identify_interaction_pre-conditions

- To identify Pre-conditions required for a Dialogue_Interaction.

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.

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.

request_participant_interaction

- To request Participants to perform a Dialogue_Interaction which provides additional Source_Data or Context to support a dialogue.

B.2.24.5 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.

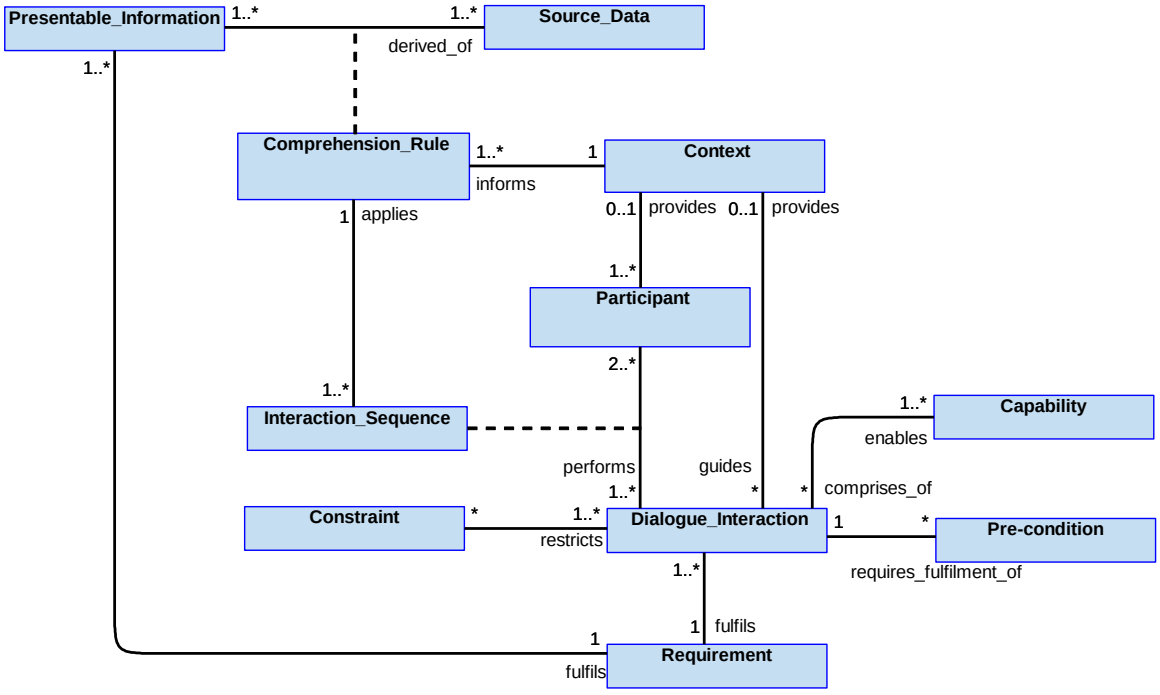

Figure 497: HMI Dialogue Semantics

B.2.24.5.1 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.

B.2.24.6 Design Rationale

B.2.24.6.1 Assumptions

- This component supports user login and role management through managing provision of the login and role assignment processes (see User Management IV).

B.2.24.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining HMI Dialogue:

- Human-Machine Interface - The HMI policy 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 policy.

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.

B.2.24.6.3 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 PRA deployment. Analysis of the specific uses of each instance, by the Exploiting Programme, may justify a less onerous DAL for some instances.

B.2.24.6.4 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 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.

B.2.24.7 Services

B.2.24.7.1 Service Definitions

B.2.24.7.1.1 Dialogue_Requirement

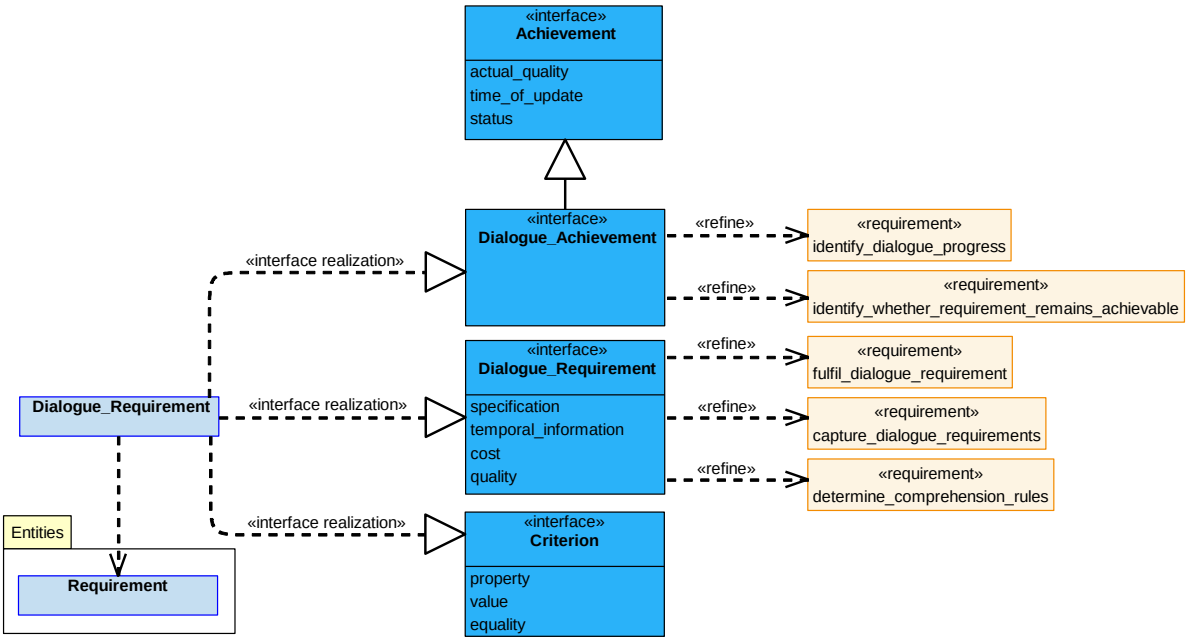

Figure 498: Dialogue_Requirement Service Definition

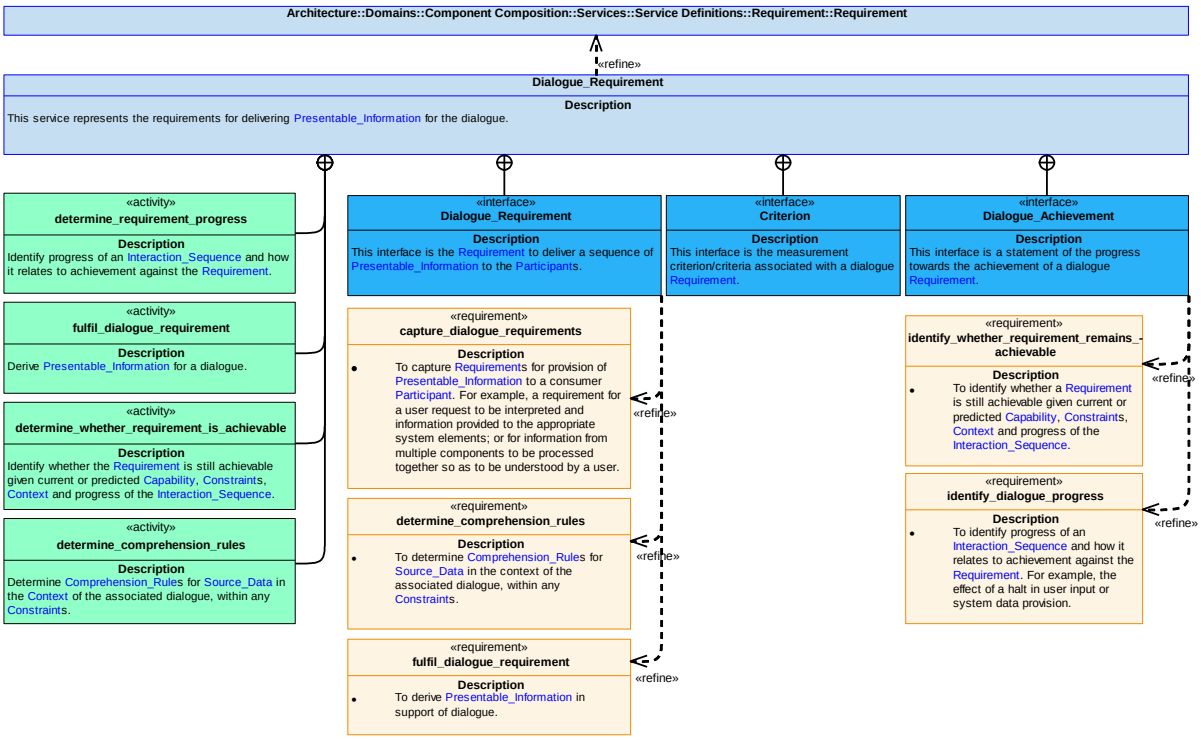

Figure 499: Dialogue_Requirement Service Policy

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.

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.

B.2.24.7.1.2 Dialogue_Dependency

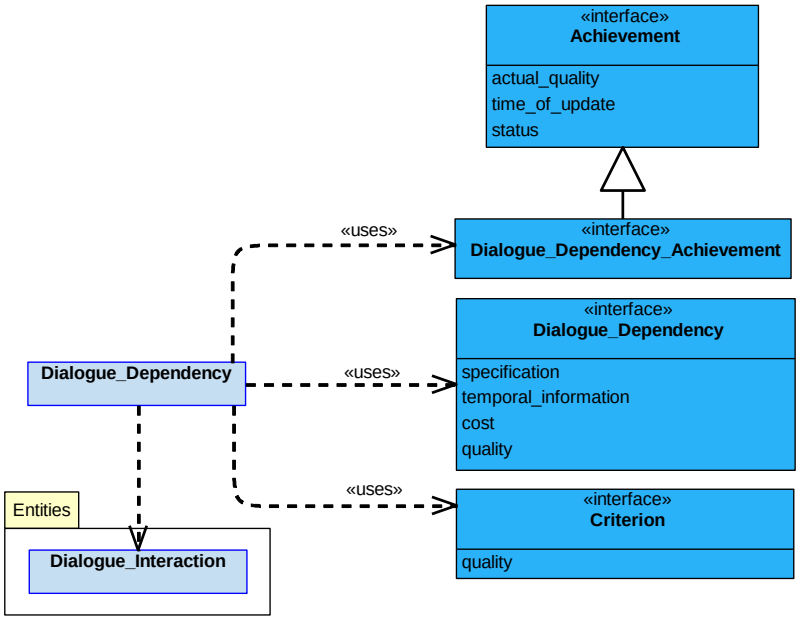

Figure 500: Dialogue_Dependency Service Definition

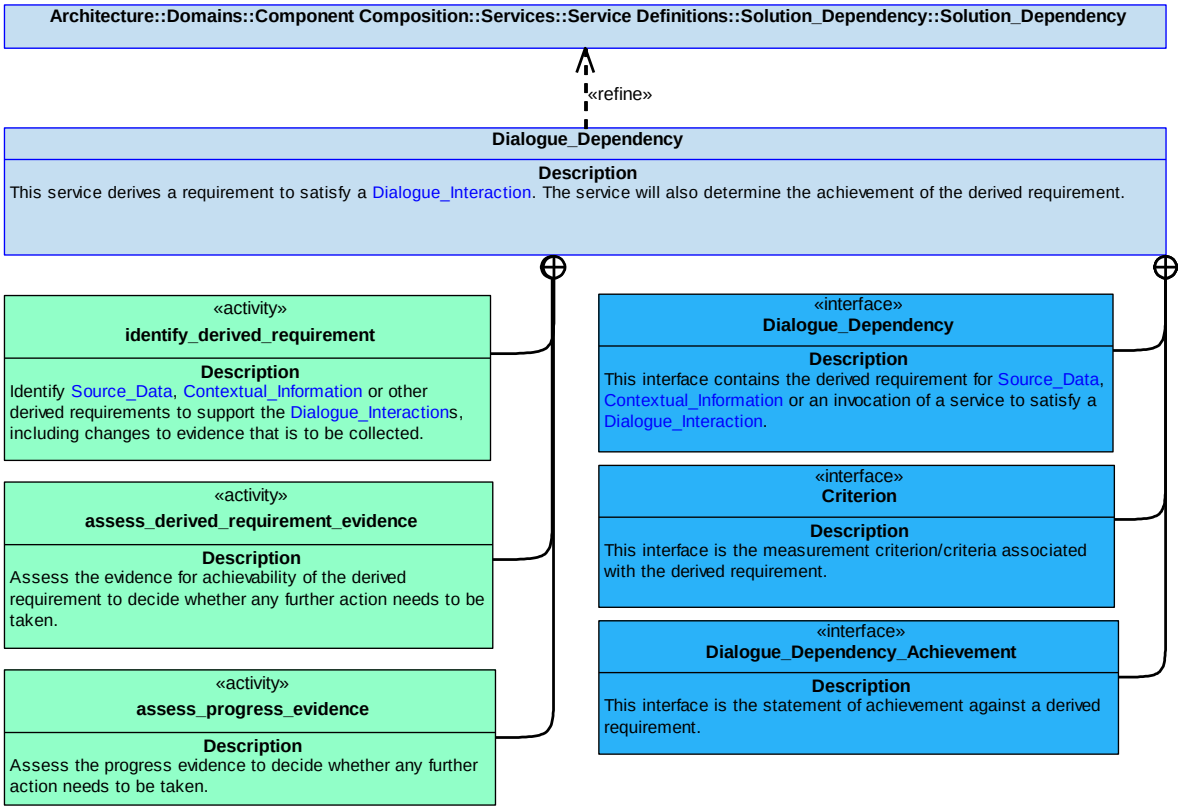

Figure 501: Dialogue_Dependency Service Policy

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 contains the derived requirement for Source_Data, Contextual_Information or an invocation of a service to satisfy a Dialogue_Interaction.

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.
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.

B.2.24.7.1.3 Dialogue_Information

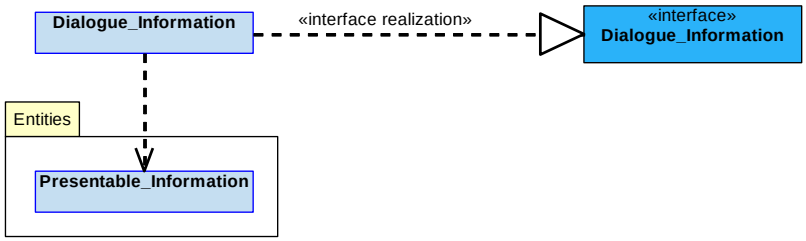

Figure 502: Dialogue_Information Service Definition

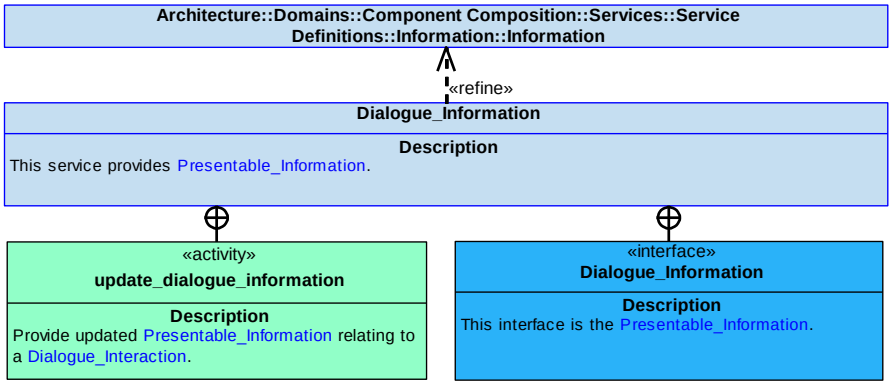

Figure 503: Dialogue_Information Service Policy

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.

B.2.24.7.1.4 Source_Data

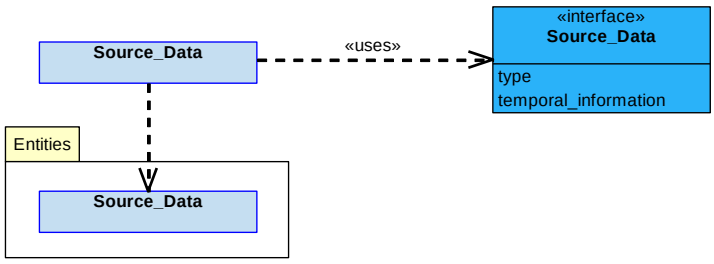

Figure 504: Source_Data Service Definition

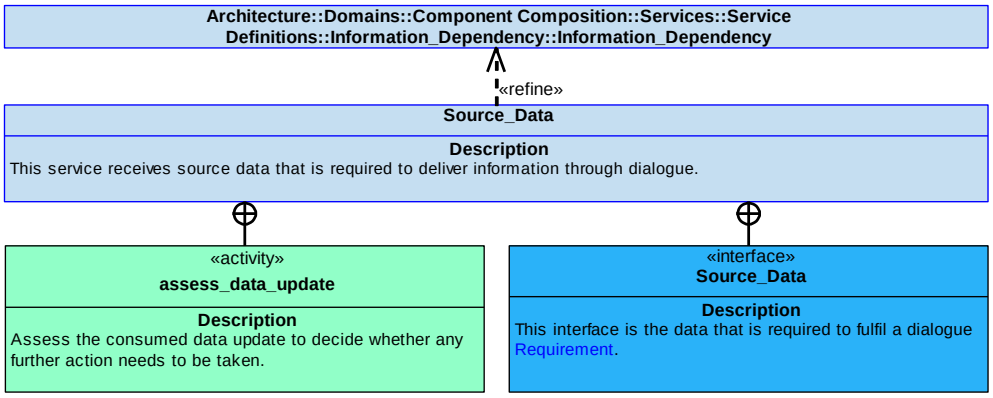

Figure 505: Source_Data Service Policy

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.

B.2.24.7.1.5 Contextual_Information

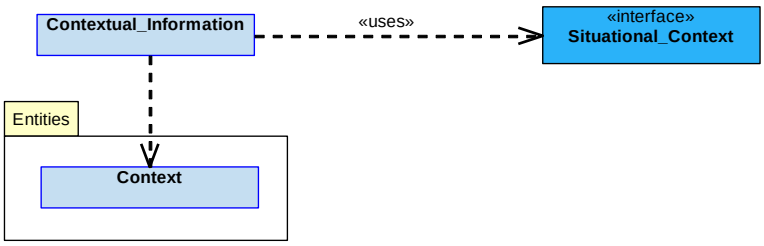

Figure 506: Contextual_Information Service Definition

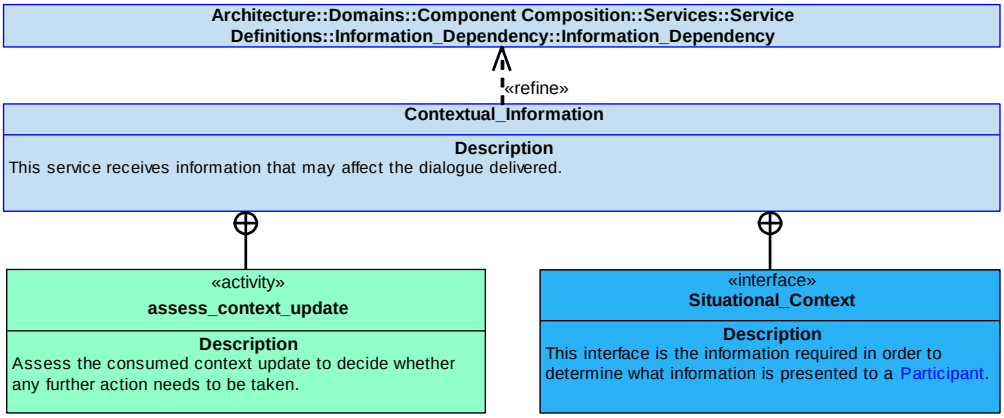

Figure 507: Contextual_Information Service Policy

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.

B.2.24.7.1.6 Constraint

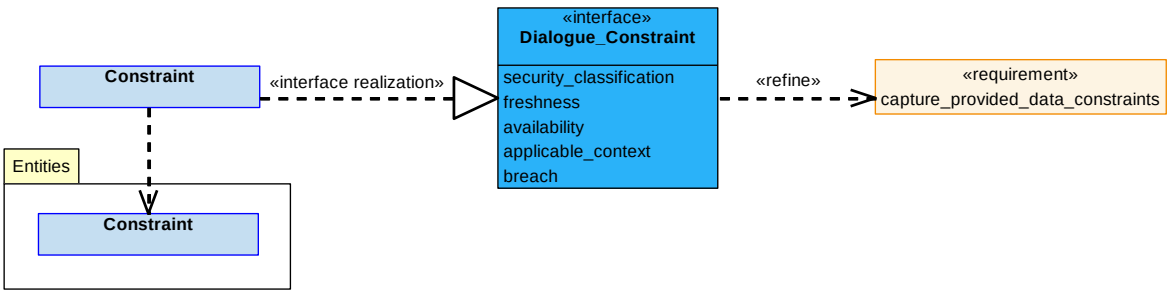

Figure 508: Constraint Service Definition

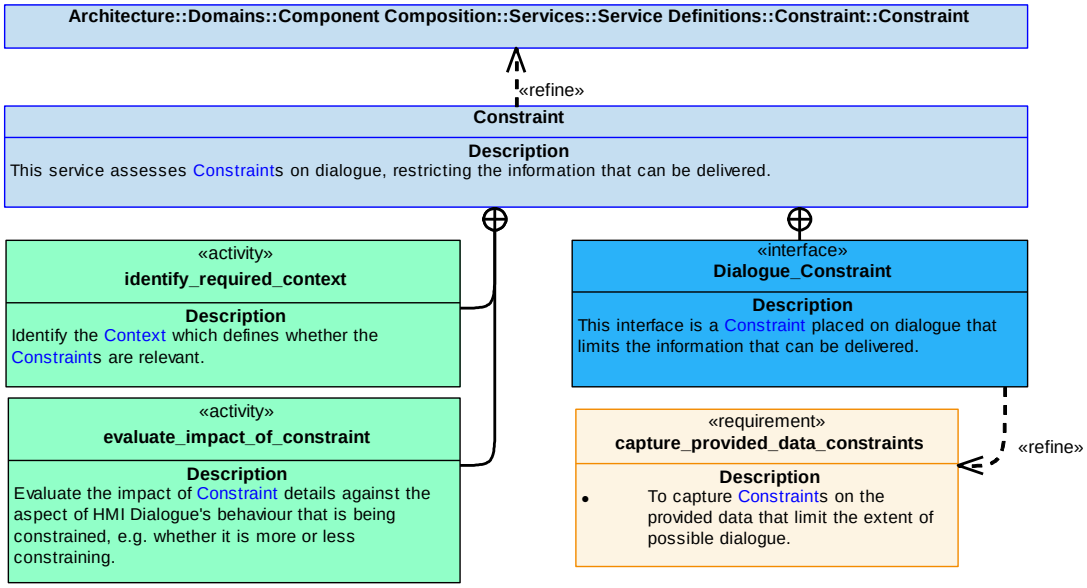

Figure 509: Constraint Service Policy

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.

B.2.24.7.1.7 Capability

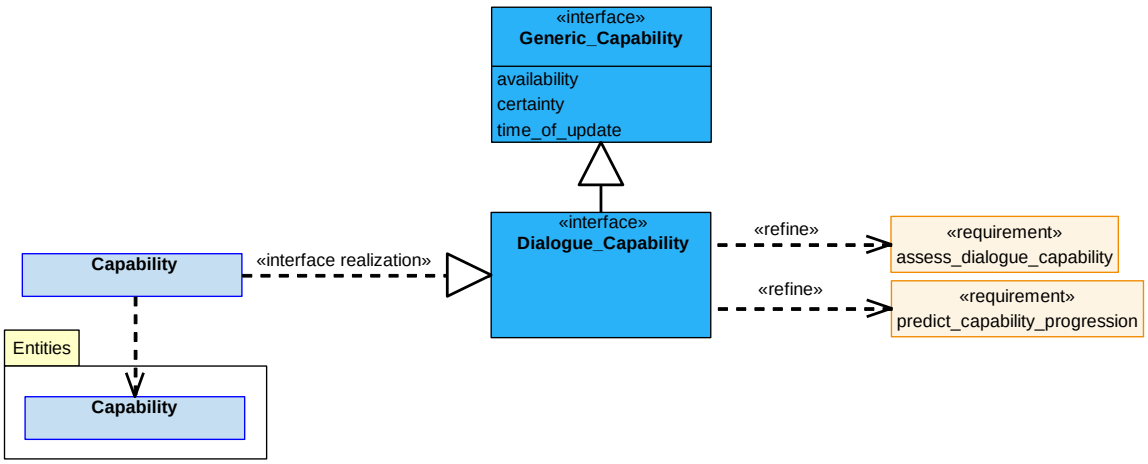

Figure 510: Capability Service Definition

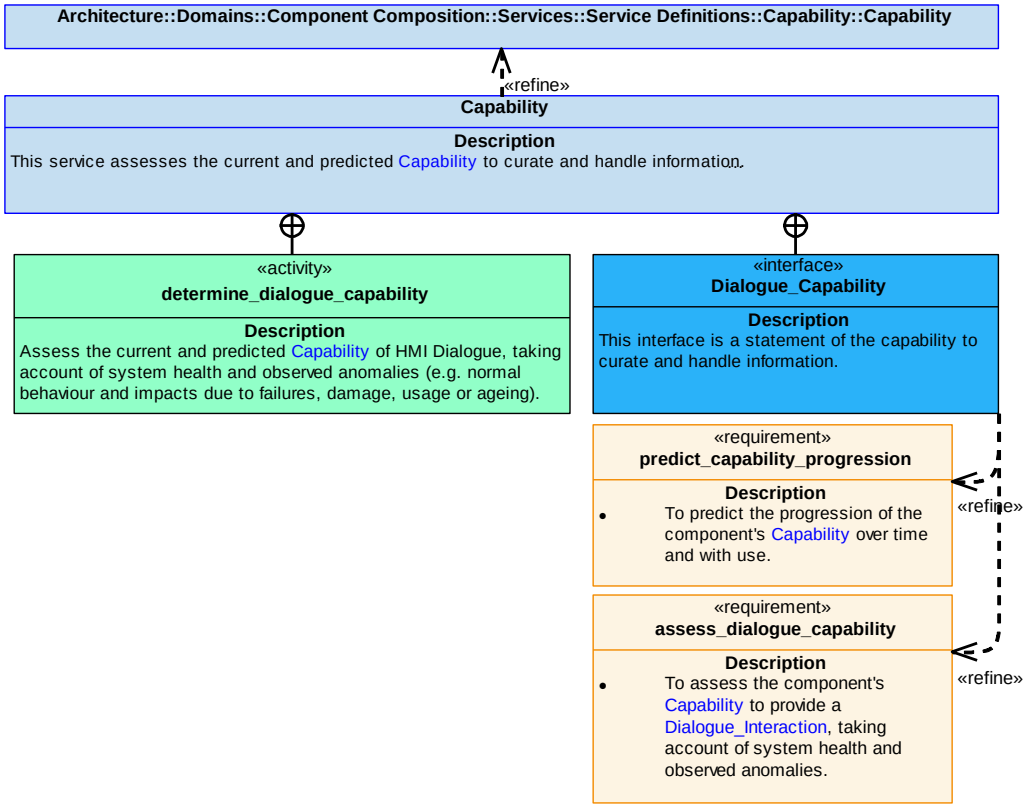

Figure 511: Capability Service Policy

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).

B.2.24.7.1.8 Capability_Evidence

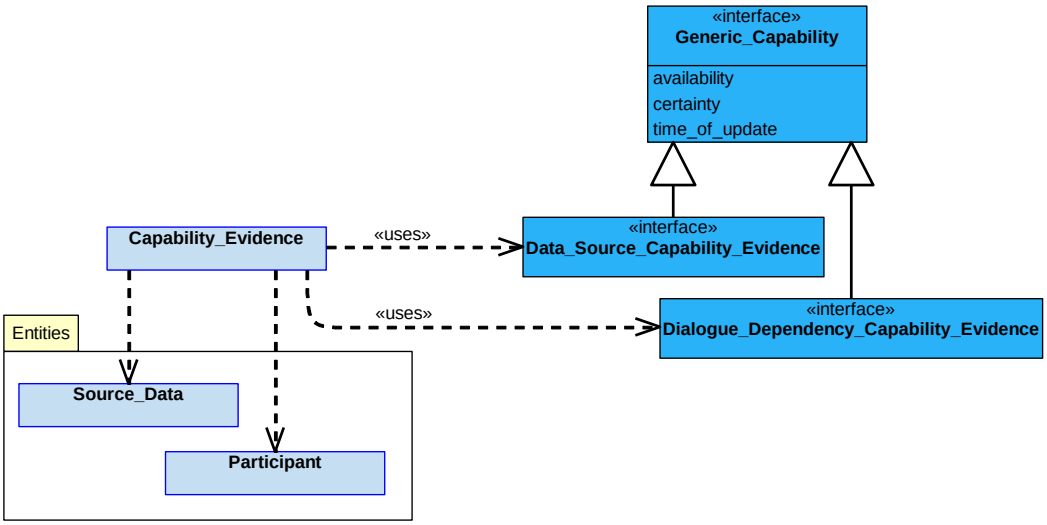

Figure 512: Capability_Evidence Service Definition

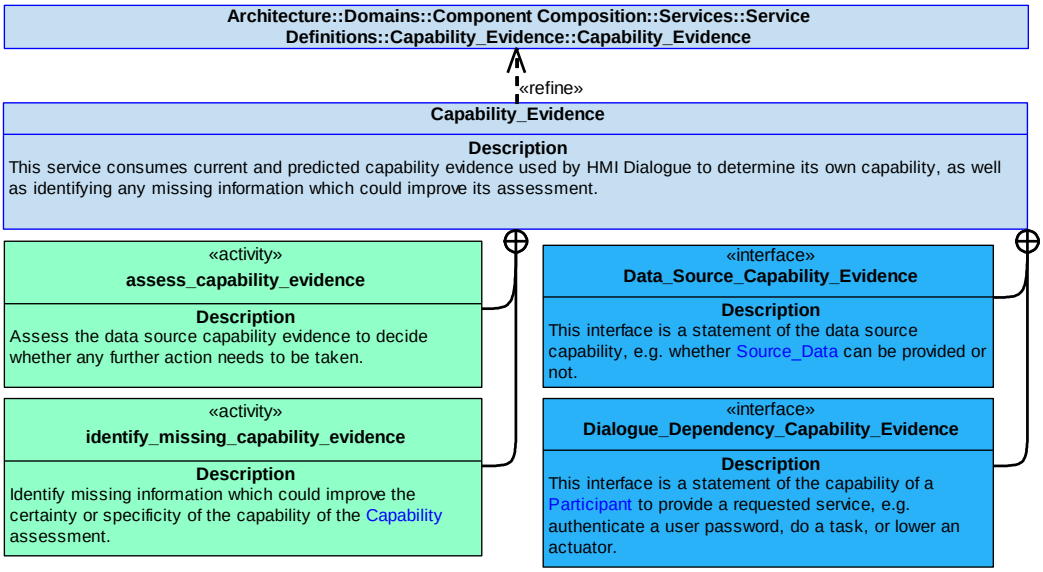

Figure 513: Capability_Evidence Service Policy

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.

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.

B.2.24.7.2 Service Dependencies

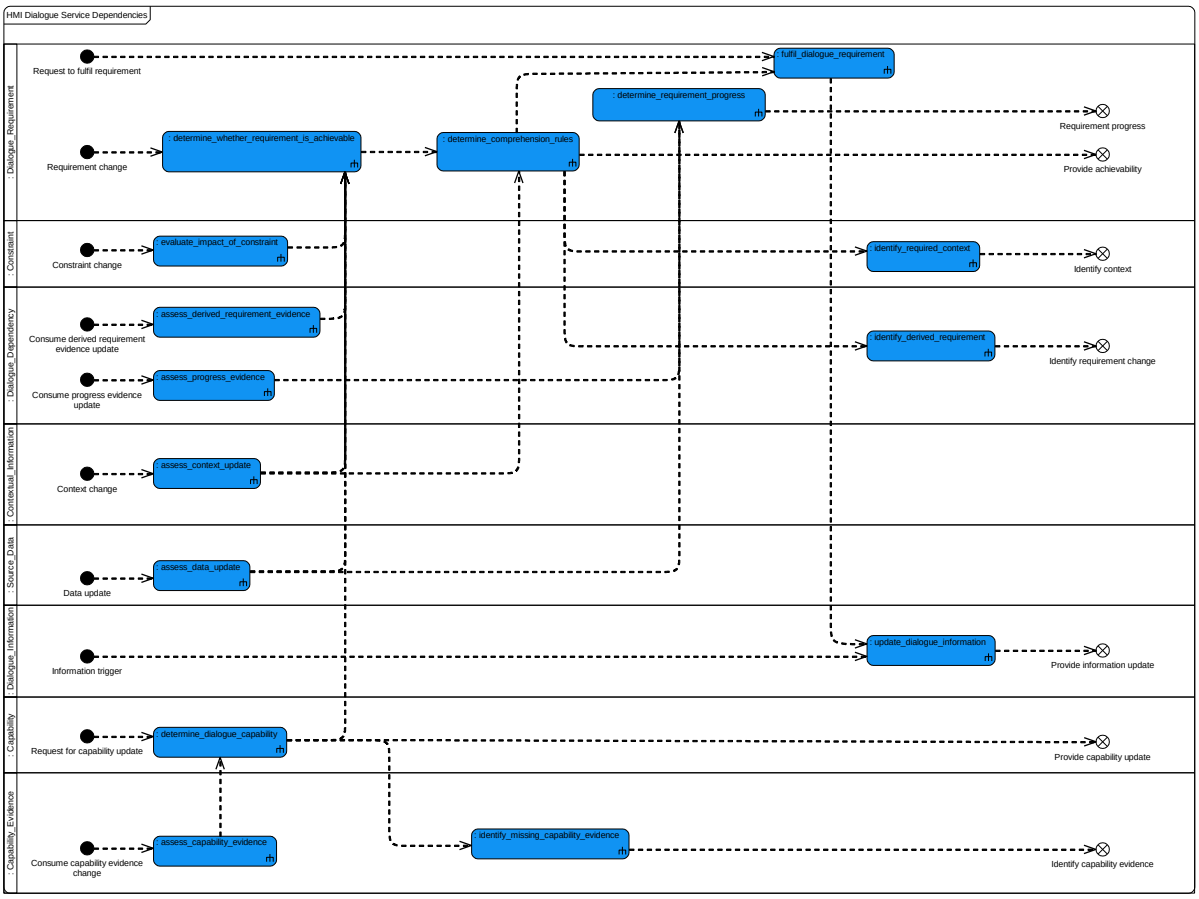

Figure 514: HMI Dialogue Service Dependencies

B.2.25 Human Interaction

B.2.25.1 Role

The role of Human Interaction is to enable communication between humans.

B.2.25.2 Overview

Control Architecture

Human Interaction is an action component as defined in the Control Architecture policy.

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.

B.2.25.3 Service Summary

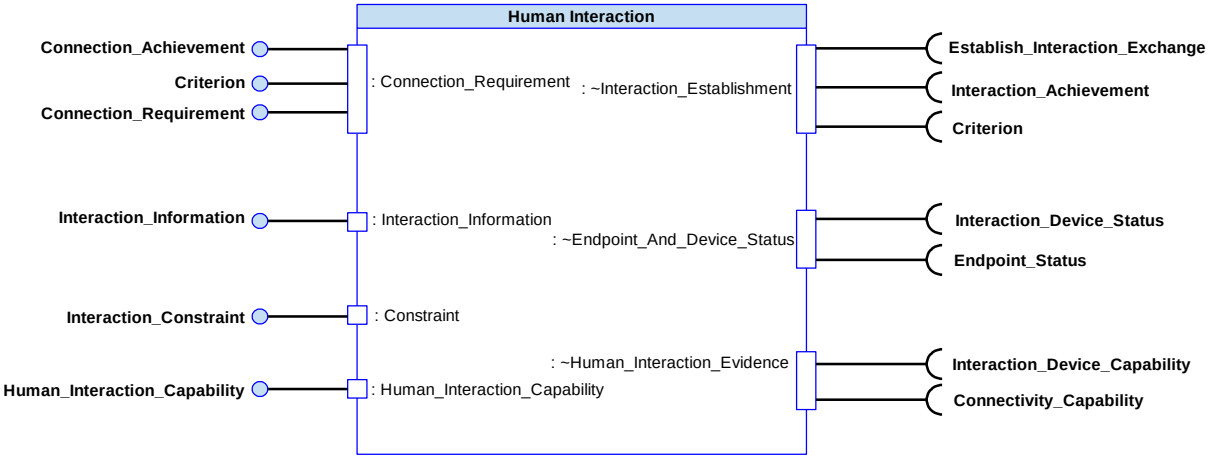

Figure 515: Human Interaction Service Summary

B.2.25.4 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, radio channel).

coordinate_interaction

- To setup, start and end an Interaction.

determine_status_of_interaction

- To determine the status of an Interaction.

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.

identify_contacts

- To identify contacts for possible Interactions (e.g. users, user-aliases and groups of users that can interact).

determine_quality_of_interaction

- To determine the quality of an Interaction against given Measurement_Criterion/criteria.

determine_if_solution_remains_feasible

- To determine if a planned or on-going Interaction_Solution remains feasible given current Constraints and Capability.

B.2.25.5 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.

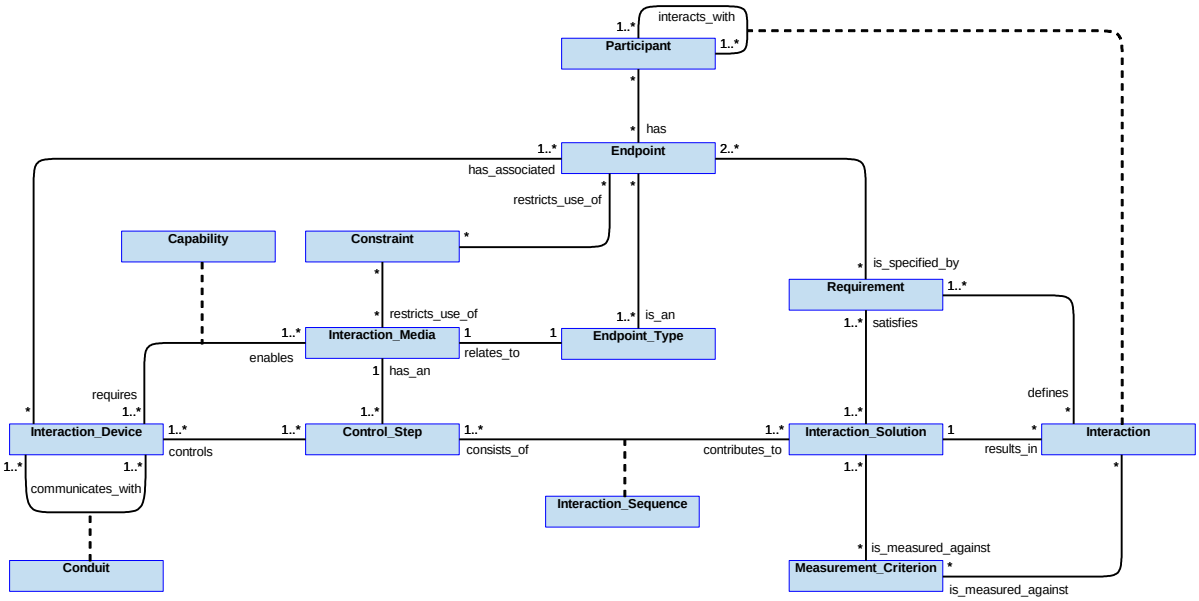

Figure 516: Human Interaction Semantics

B.2.25.5.1 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.

B.2.25.6 Design Rationale

B.2.25.6.1 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.

B.2.25.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Human Interaction:

- Use of Communications - This policy 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

- A directory service defines a contact sufficiently to identify the user, user-alias or group it represents. Full contact information is aggregated from data held in other components through use of counterparts.
- Human Interaction is responsible for the management and monitoring of an interaction.
- Human Interaction facilitates capability such as call groups, call forwarding, call holding, pick-up groups and redirect.
- This component is not a directory service. Full contact information is referenced directly from components that offer user interaction services. In this way data is defined, used and understood only at the point of use and never passed through the component if it has no need to act on the data.

B.2.25.6.3 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.

B.2.25.6.4 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.

B.2.25.7 Services

B.2.25.7.1 Service Definitions

B.2.25.7.1.1 Connection_Requirement

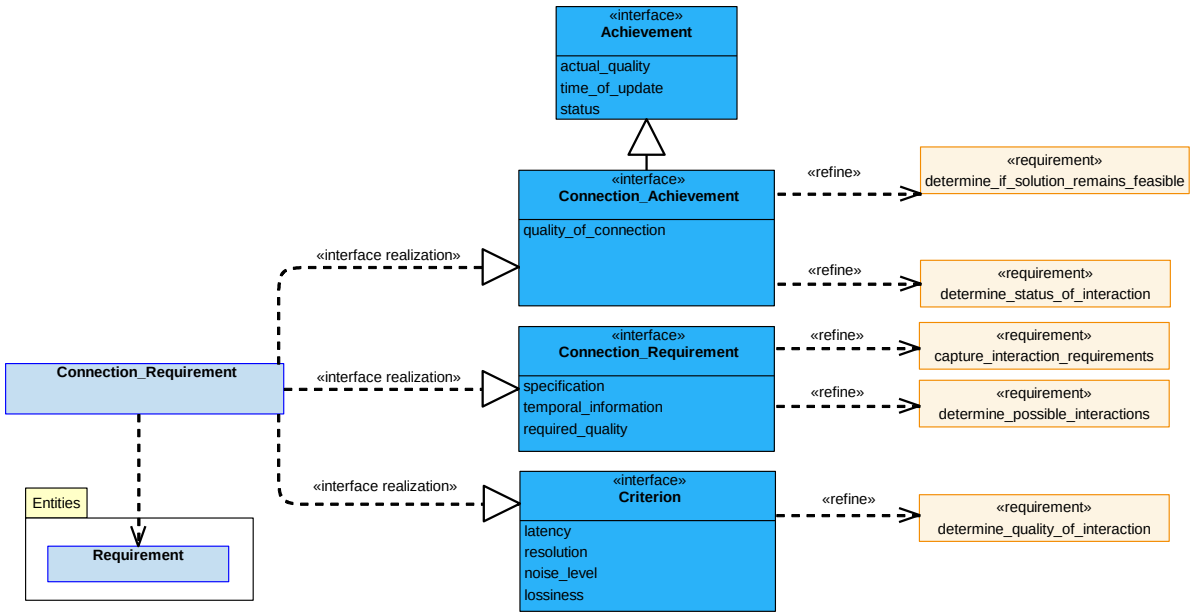

Figure 517: Connection_Requirement Service Definition

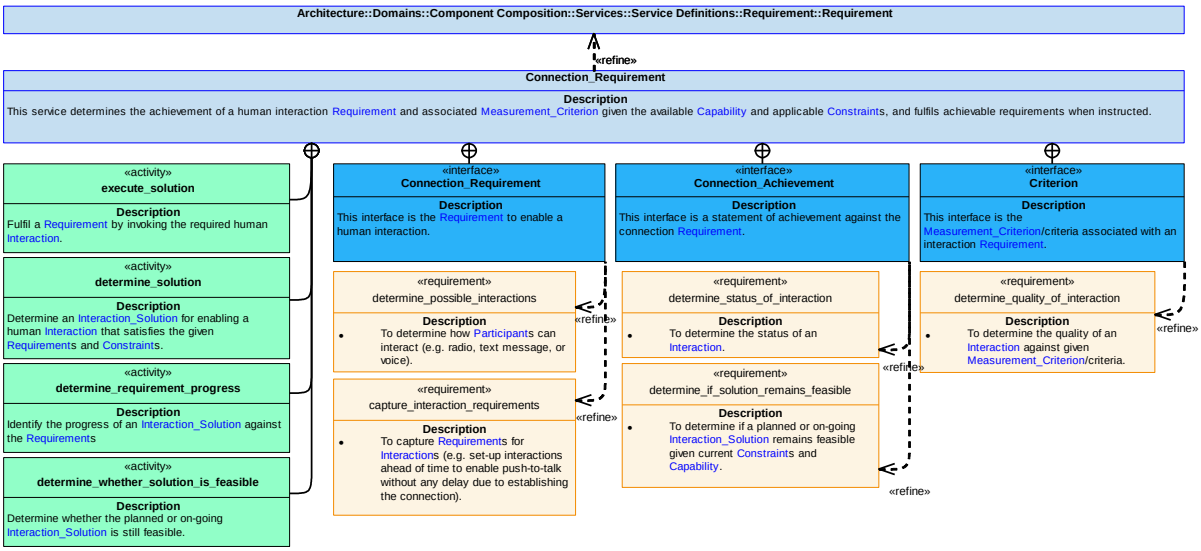

Figure 518: Connection_Requirement Service Policy

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.

B.2.25.7.1.2 Interaction_Establishment

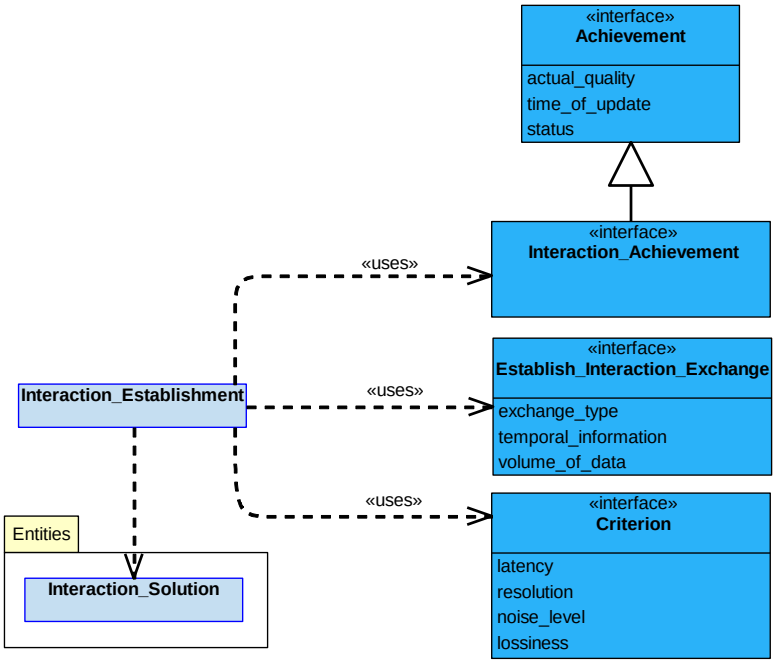

Figure 519: Interaction_Establishment Service Definition

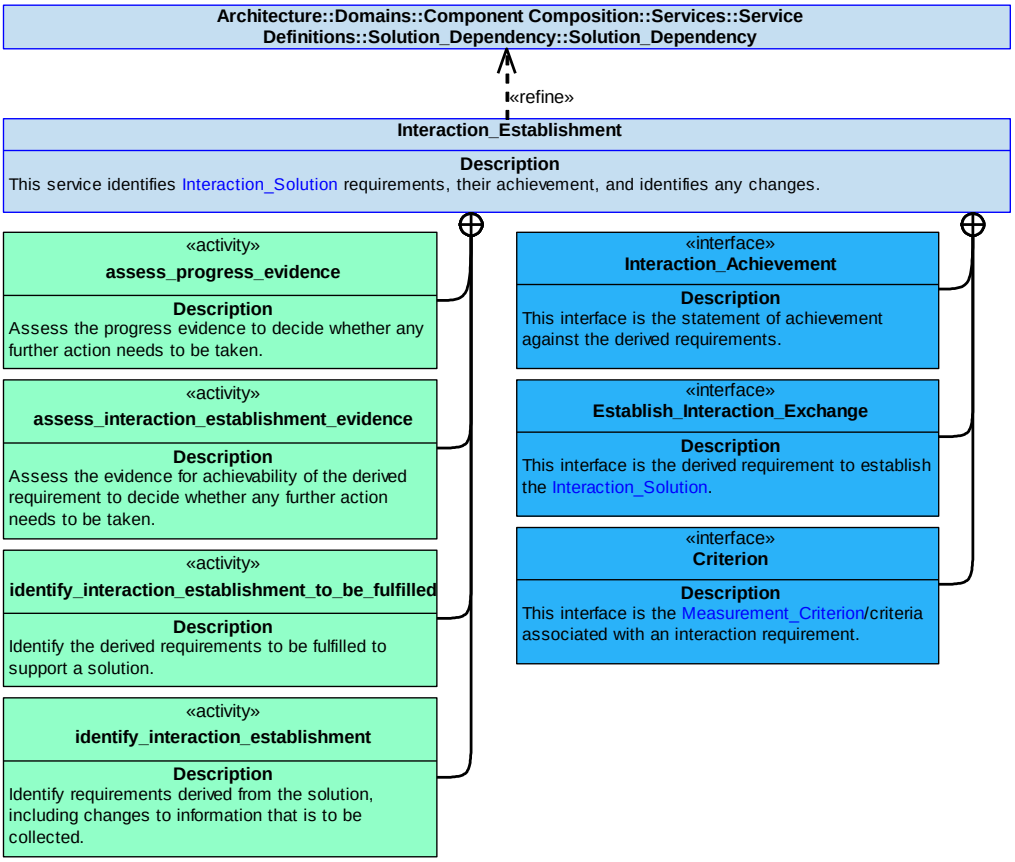

Figure 520: Interaction_Establishment Service Policy

Interaction_Establishment

This service identifies Interaction_Solution requirements, 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.

B.2.25.7.1.3 Interaction_Information

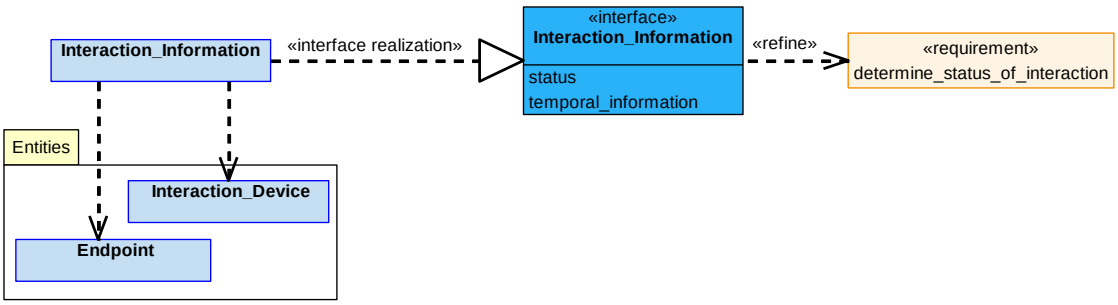

Figure 521: Interaction_Information Service Definition

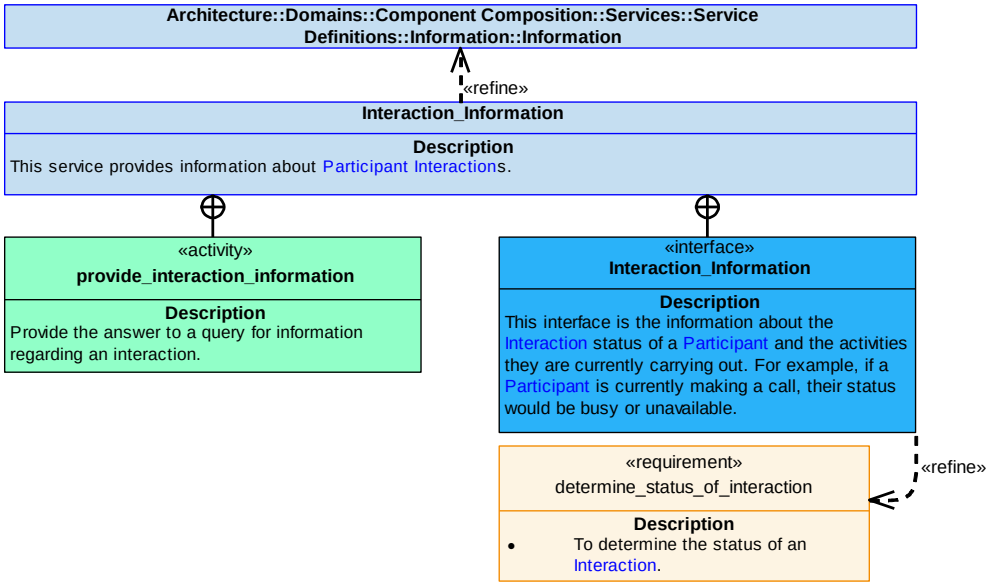

Figure 522: Interaction_Information Service Policy

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.

B.2.25.7.1.4 Endpoint_And_Device_Status

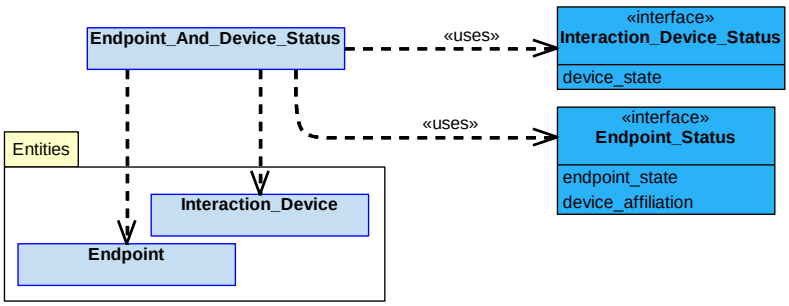

Figure 523: Endpoint_and_Device_Status Service Definition

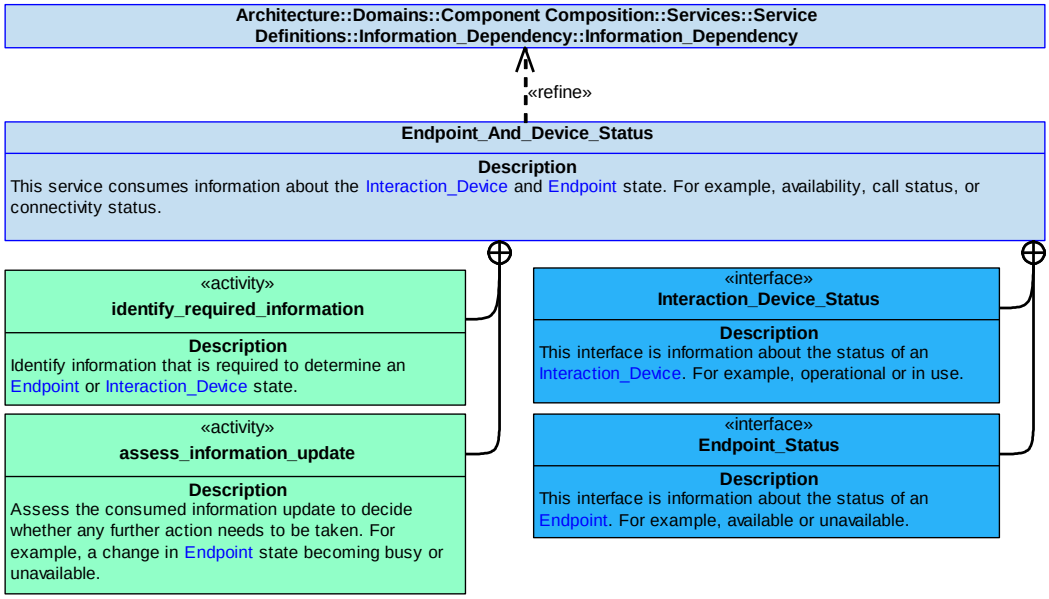

Figure 524: Endpoint_and_Device_Status Service Policy

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.

B.2.25.7.1.5 Constraint

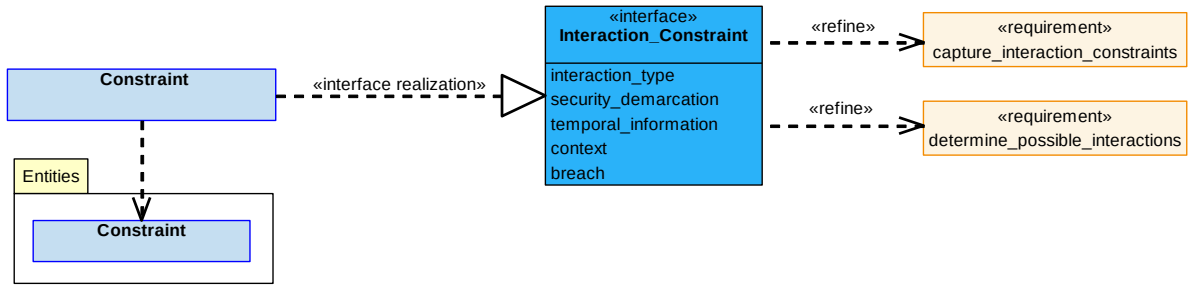

Figure 525: Constraint Service Definition

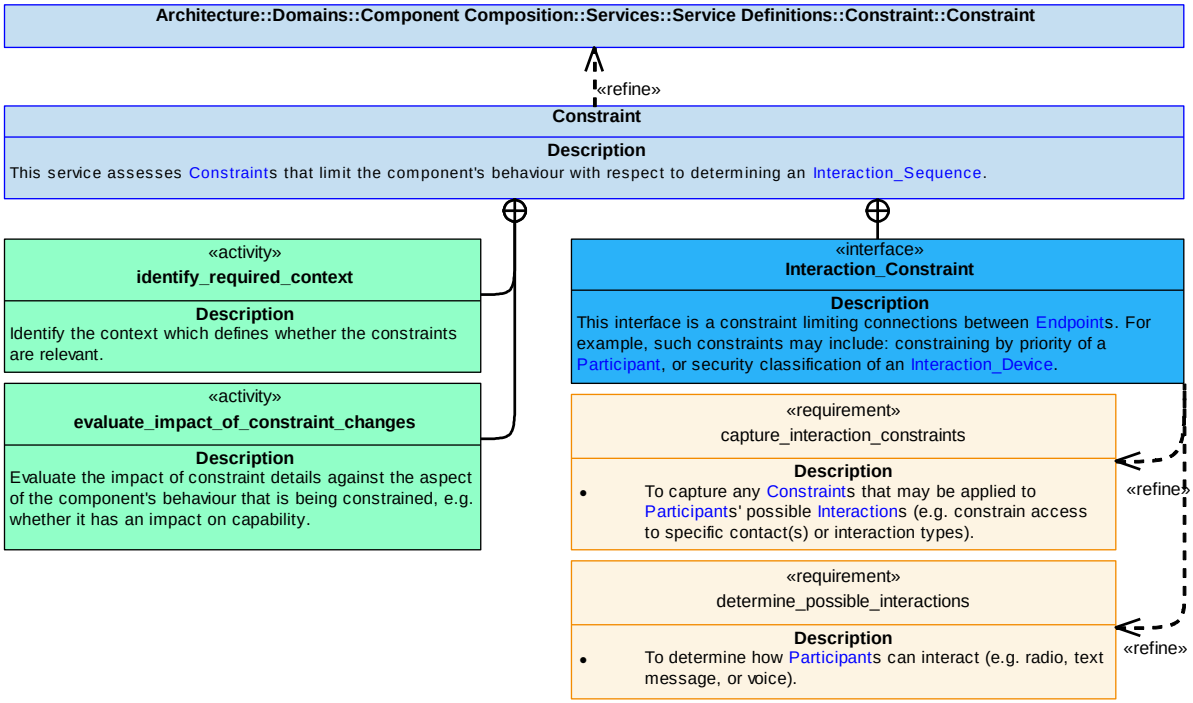

Figure 526: Constraint Service Policy

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.

B.2.25.7.1.6 Human_Interaction_Capability

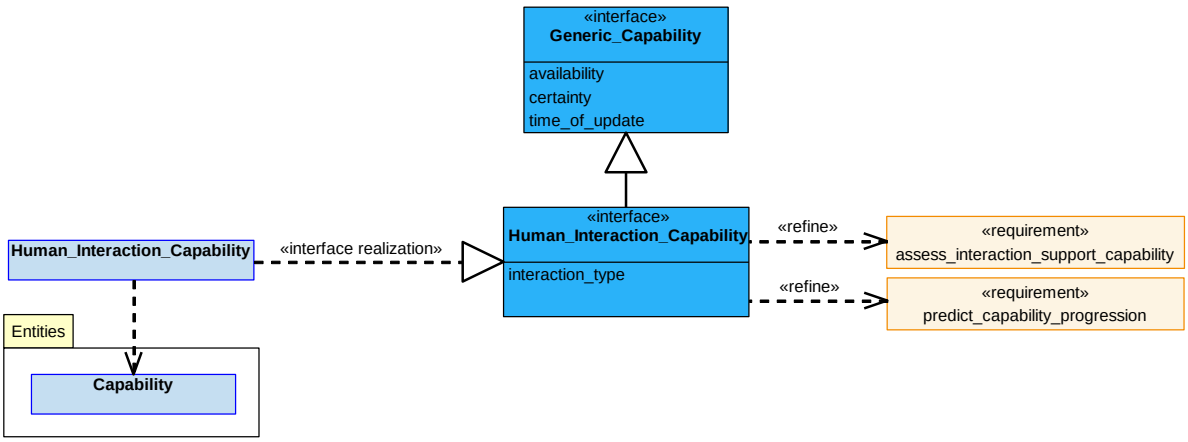

Figure 527: Human_Interaction Capability Service Definition

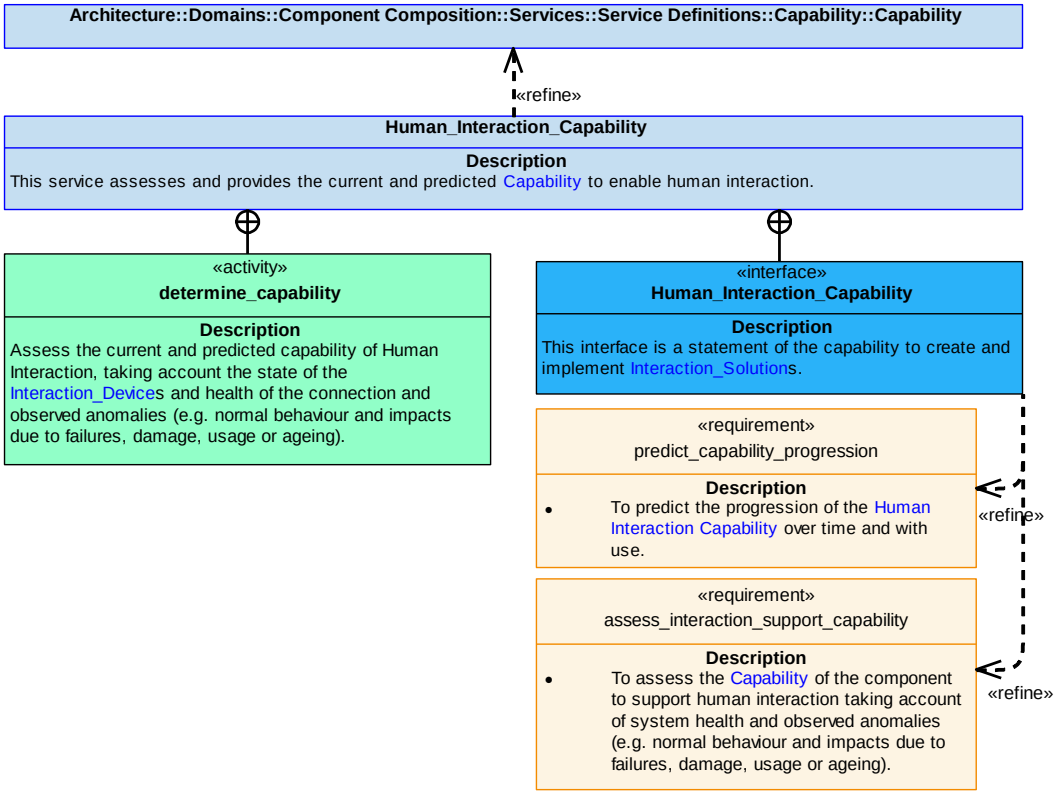

Figure 528: Human_Interaction_Capability Service Policy

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).

B.2.25.7.1.7 Human_Interaction_Evidence

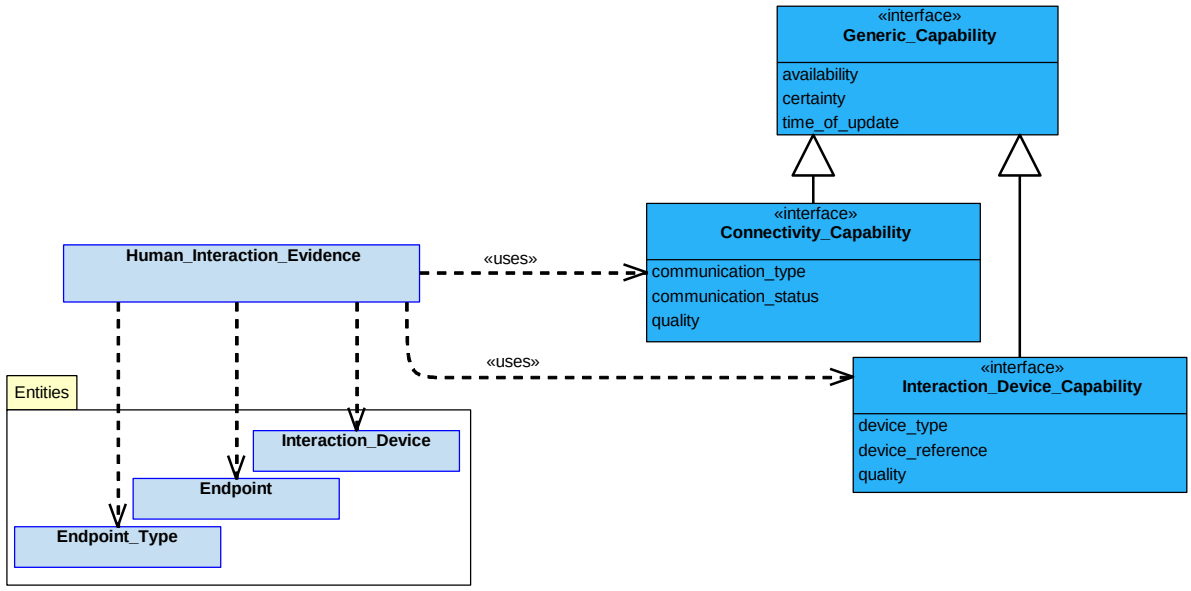

Figure 529: Human_Interaction_Evidence Service Definition

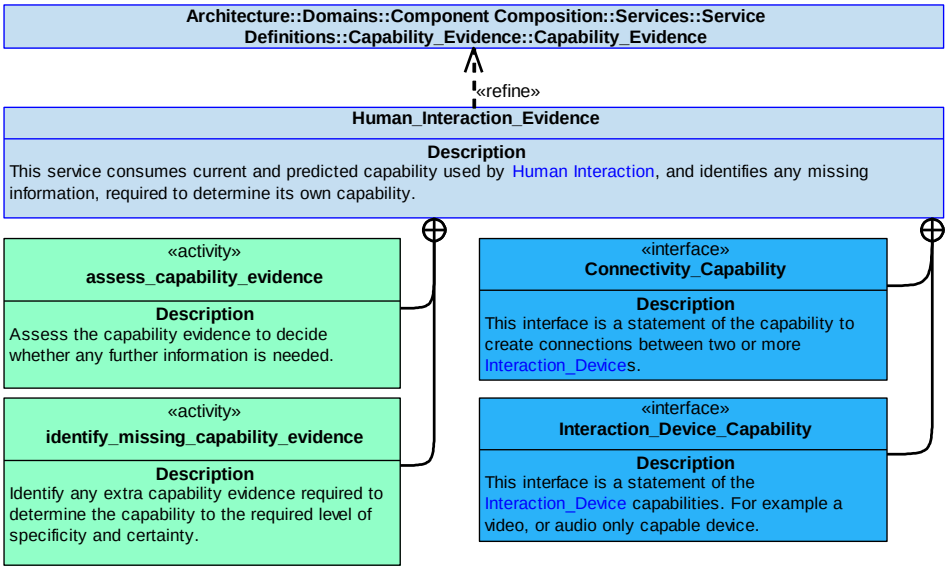

Figure 530: Human_Interaction_Evidence Service Policy

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.

B.2.25.7.2 Service Dependencies

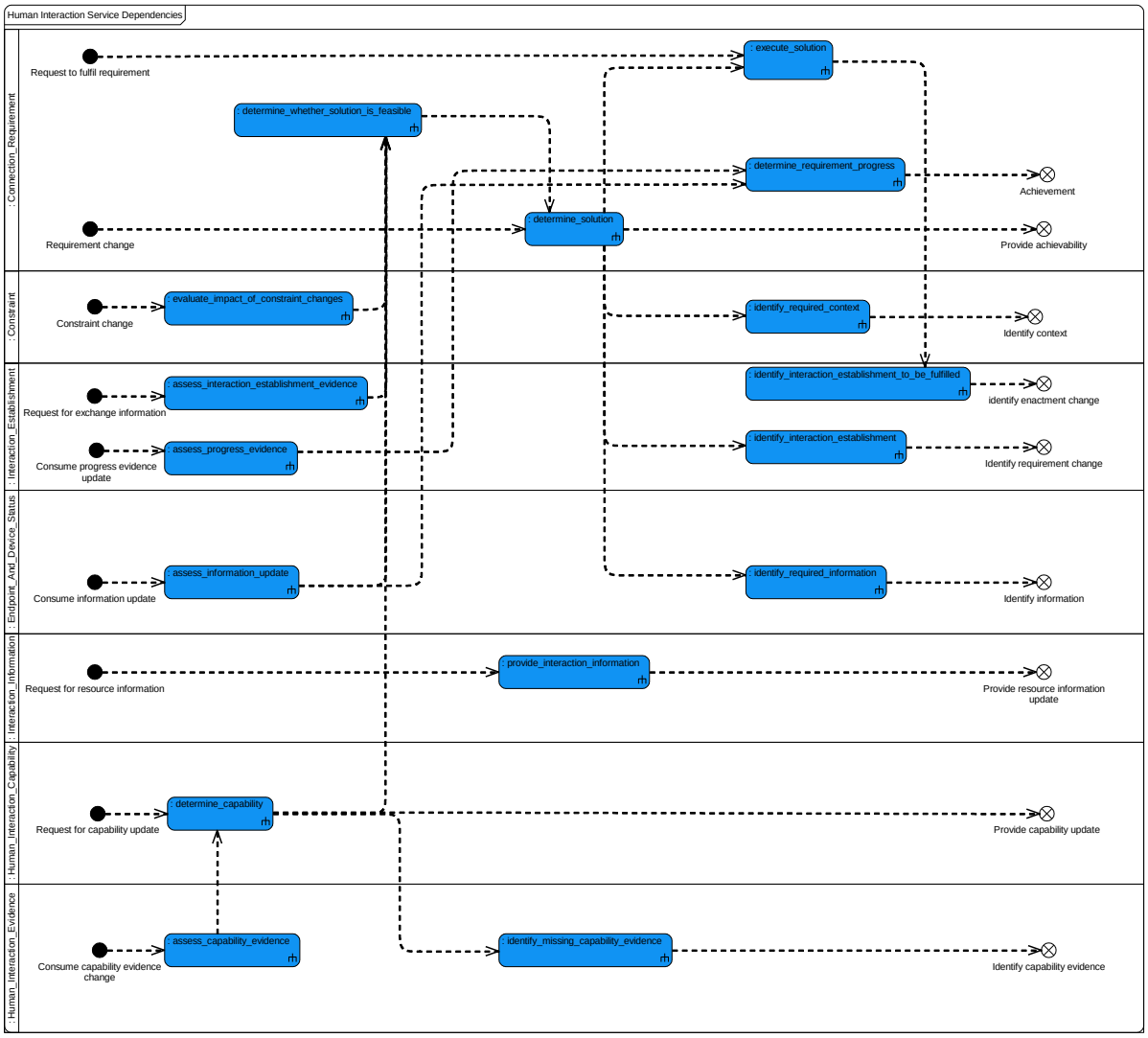

Figure 531: Human Interaction Service Dependencies

B.2.26 Information Brokerage

B.2.26.1 Role

The role of Information Brokerage is to instigate and oversee information exchanges.

B.2.26.2 Overview

Control Architecture

Information Brokerage is an action component as defined in the Control Architecture policy.

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.

B.2.26.3 Service Summary

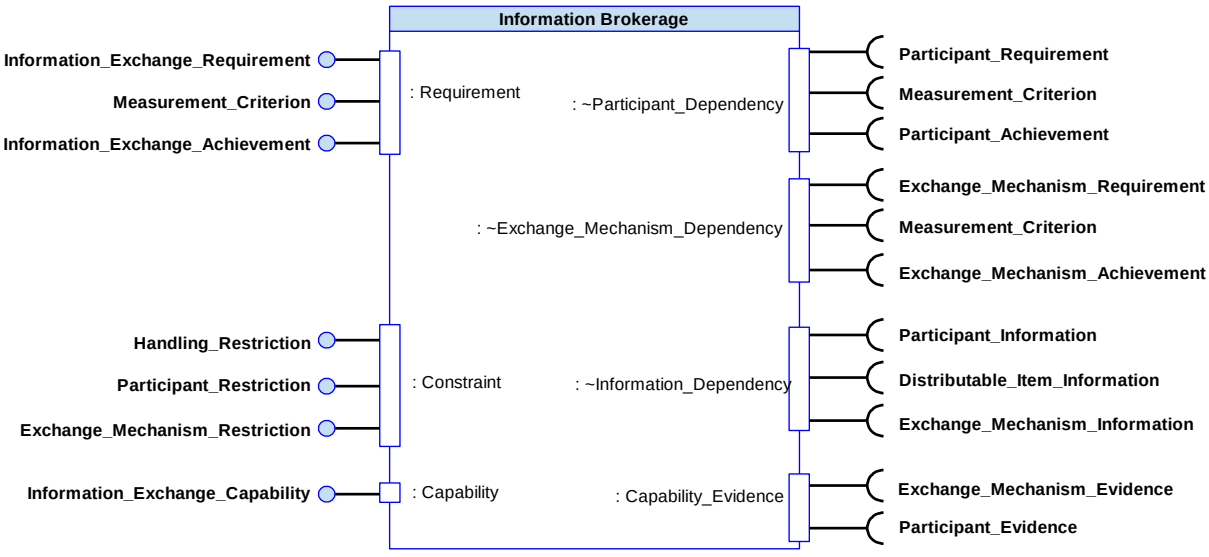

Figure 532: Information Brokerage Service Summary

B.2.26.4 Responsibilities

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).

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_exchange_constraints

- To capture the constraints associated with an Exchange.

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.

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_quality_of_exchange_delivery

- To determine the quality of an information Exchange, measured against given requirements and measurement criteria.

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.

capture_exchange_mechanism_of_participants

- To capture the information Exchange_Mechanisms of Participants.

capture_measurement_criteria_for_exchange

- To capture the Delivery_Characteristic required for an information exchange.

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.

B.2.26.5 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.

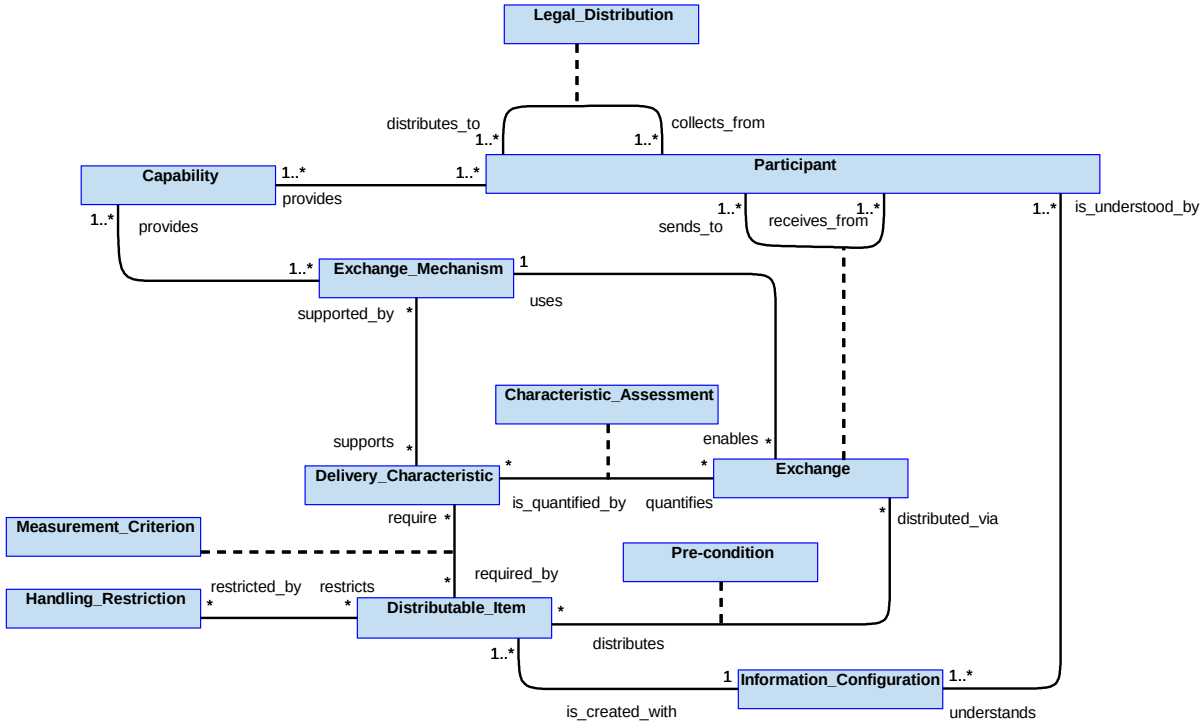

Figure 533: Information Brokerage Semantics

B.2.26.5.1 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.

B.2.26.6 Design Rationale

B.2.26.6.1 Assumptions

- There are no specific assumptions associated with this component.

B.2.26.6.2 Design Considerations

Directly Applicable Policies

These policies 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 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 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.

B.2.26.6.3 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.

B.2.26.6.4 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.

The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about security of data exchange.

B.2.26.7 Services

B.2.26.7.1 Service Definitions

B.2.26.7.1.1 Requirement

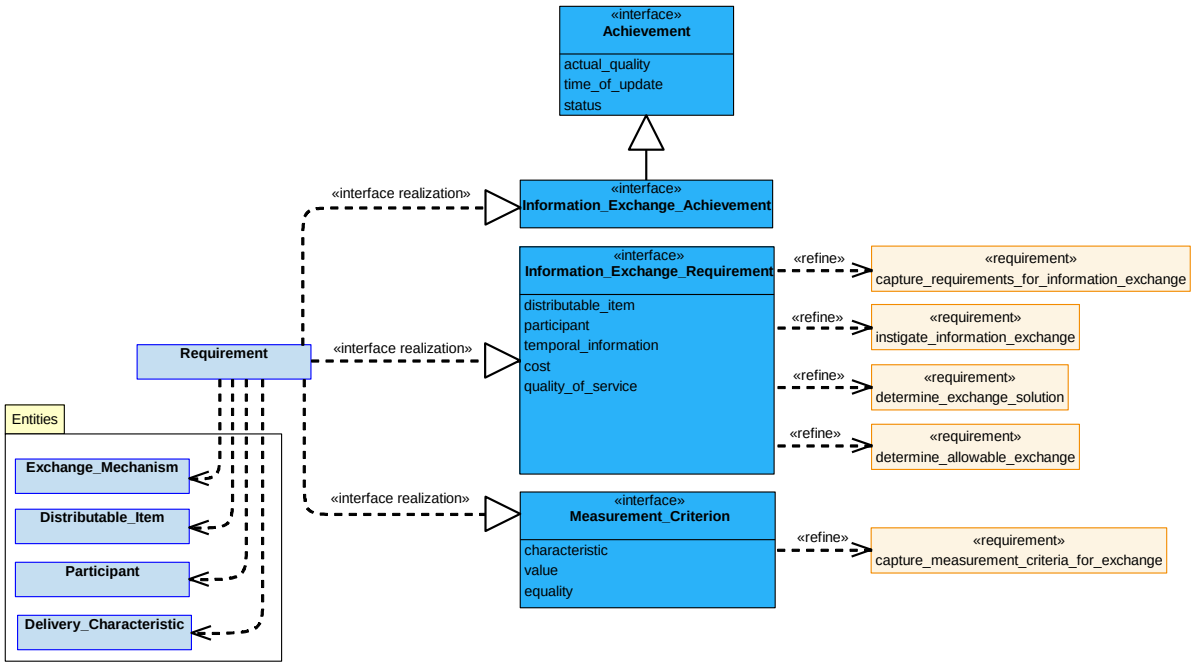

Figure 534: Requirement Service Definition

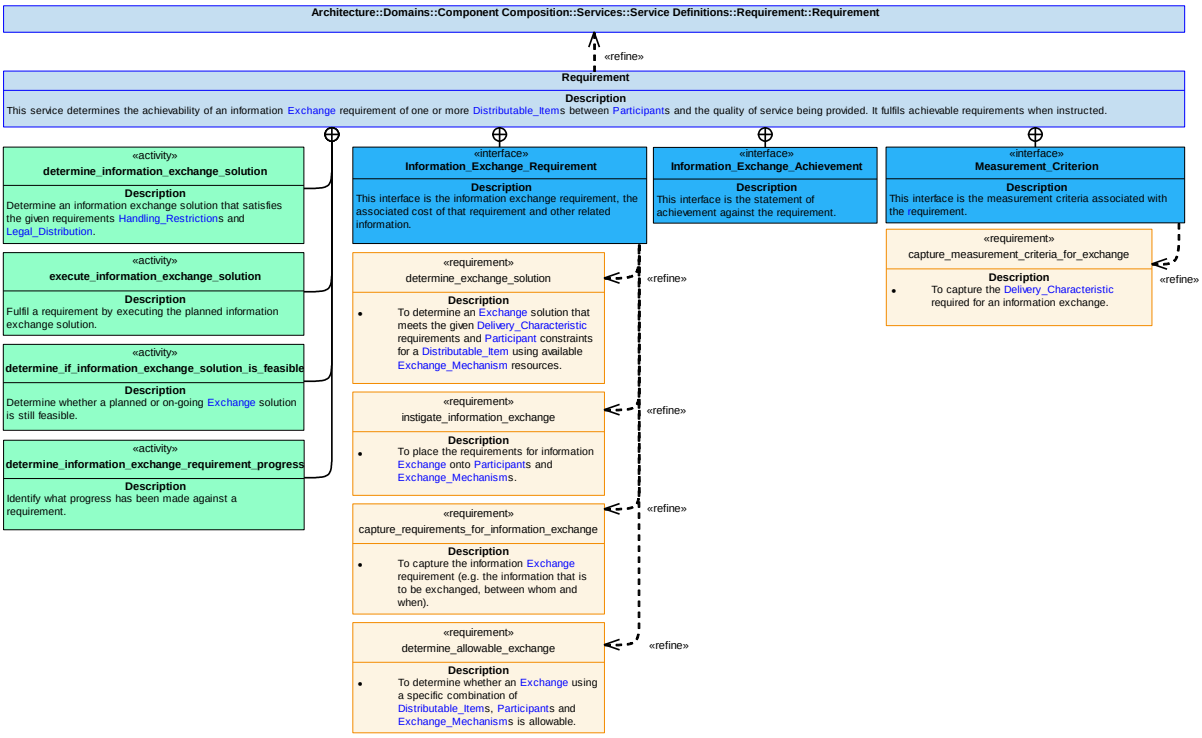

Figure 535: Requirement Service Policy

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.

B.2.26.7.1.2 Participant_Dependency

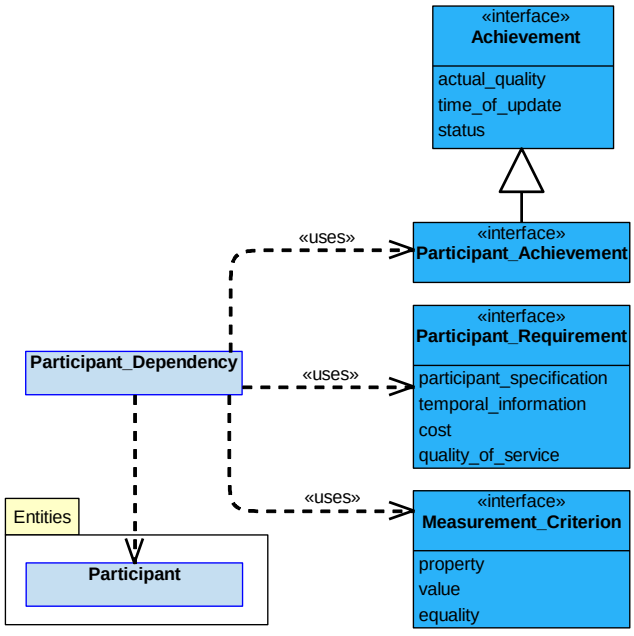

Figure 536: Participant_Dependency Service Definition

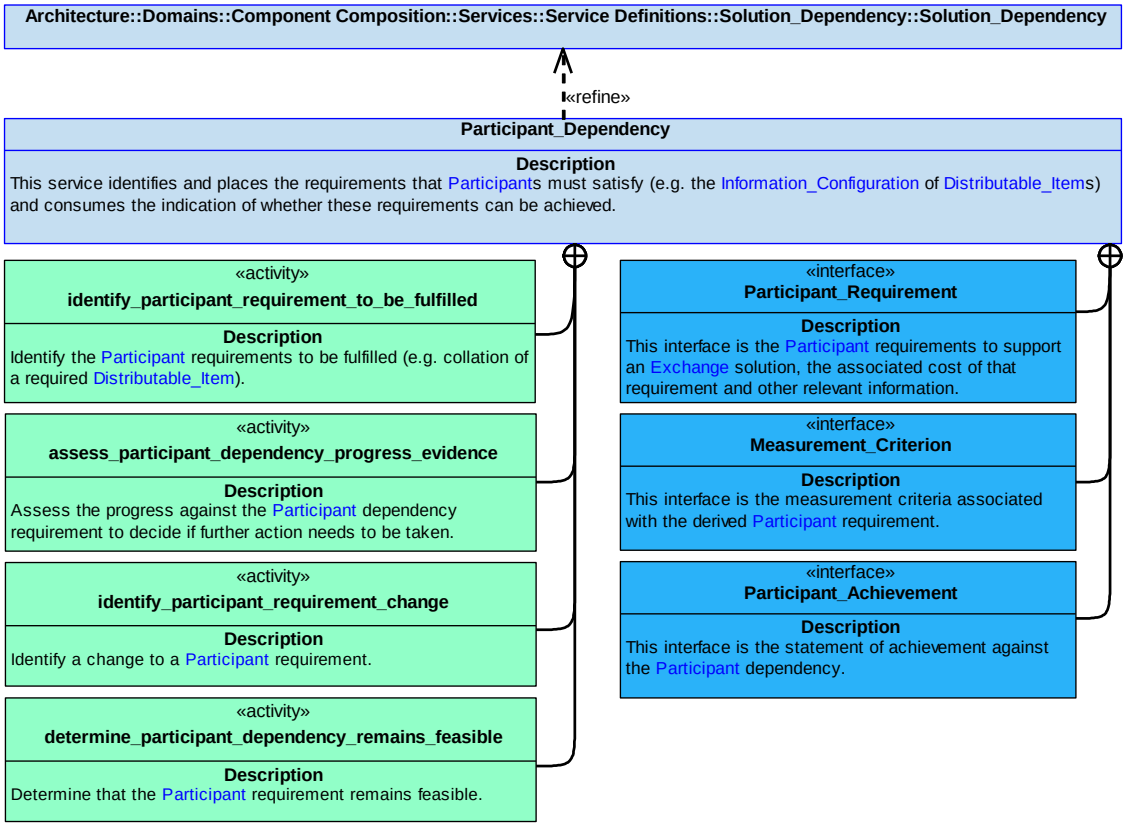

Figure 537: Participant_Dependency Service Policy

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.

B.2.26.7.1.3 Exchange_Mechanism_Dependency

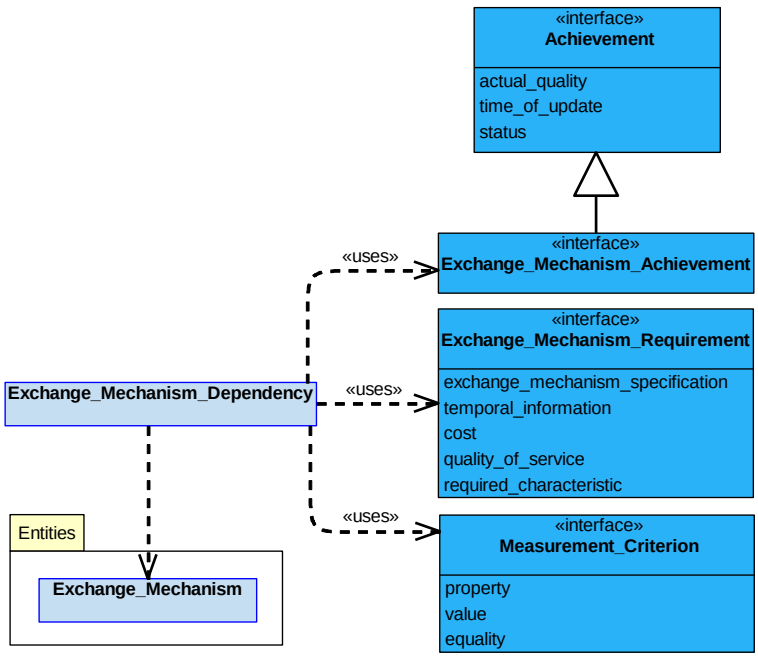

Figure 538: Exchange_Mechanism_Dependency Service Definition

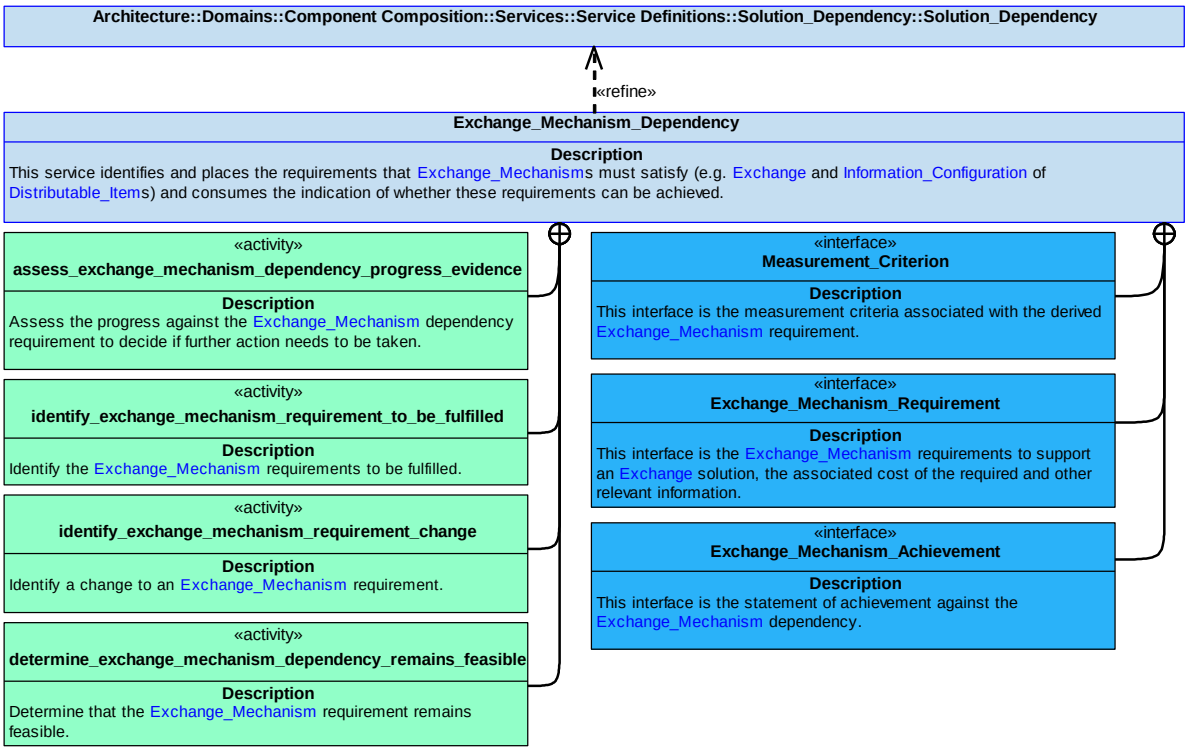

Figure 539: Exchange_Mechanism_Dependency Service Policy

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.

B.2.26.7.1.4 Information_Dependency

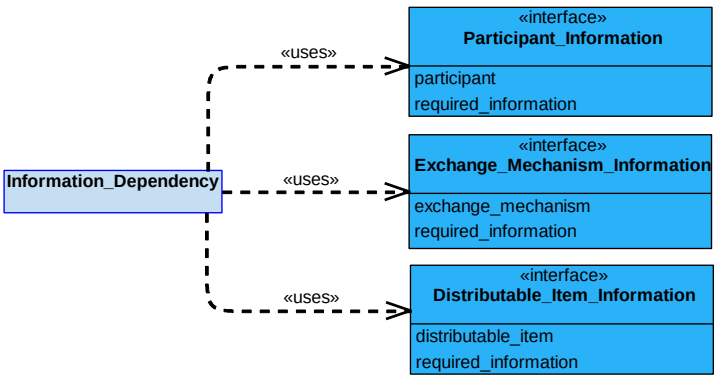

Figure 540: Information_Dependency Service Definition

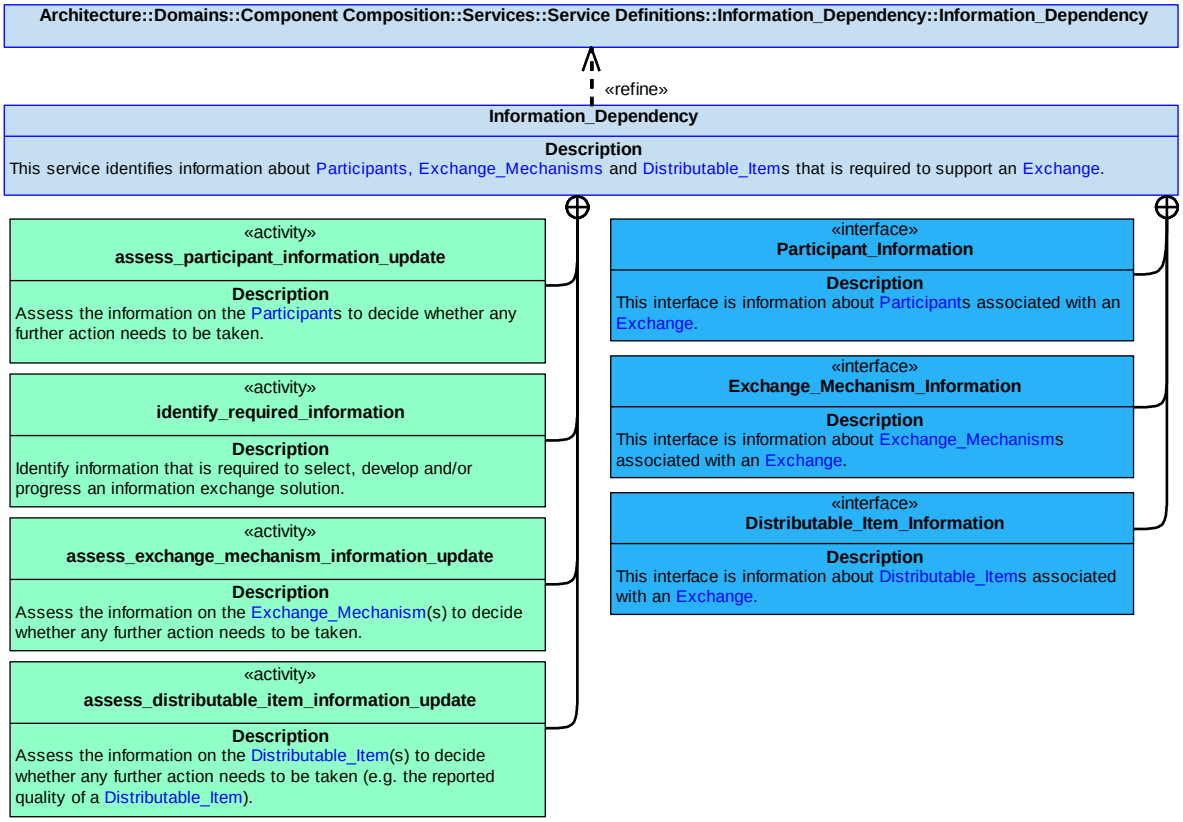

Figure 541: Information_Dependency Service Policy

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).

B.2.26.7.1.5 Constraint

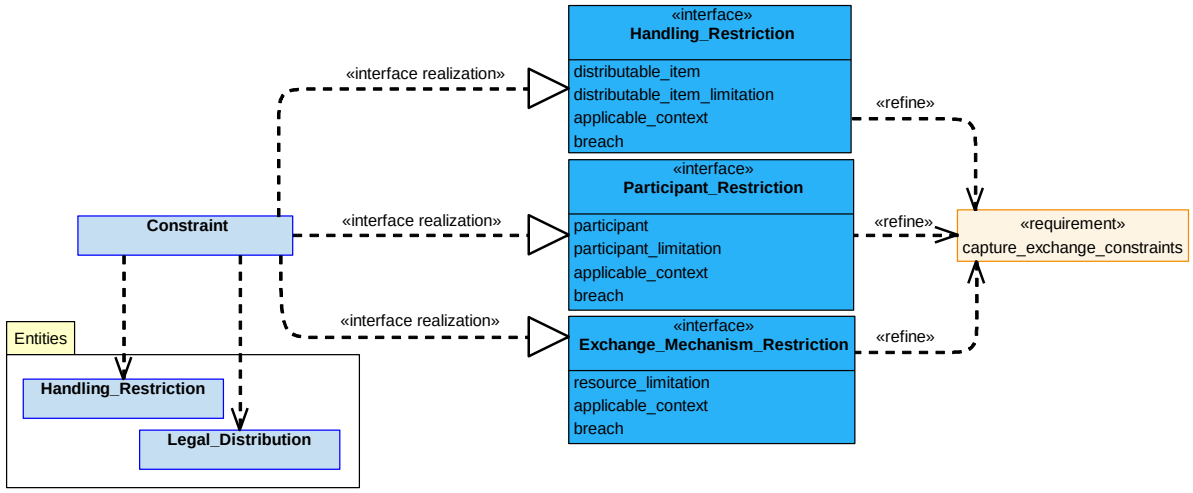

Figure 542: Constraint Service Definition

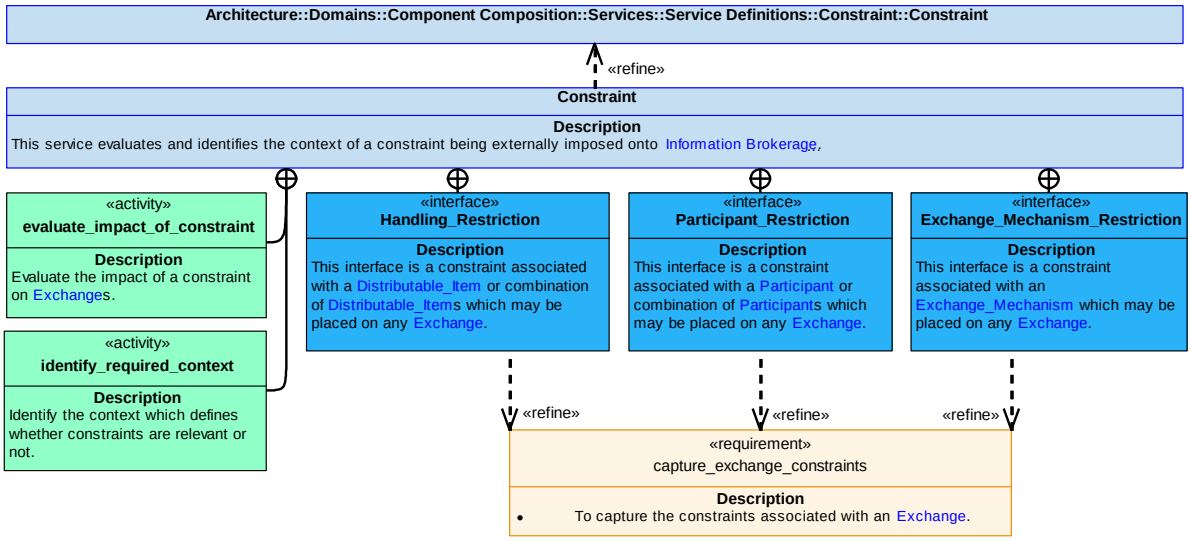

Figure 543: Constraint Service Policy

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.

B.2.26.7.1.6 Capability

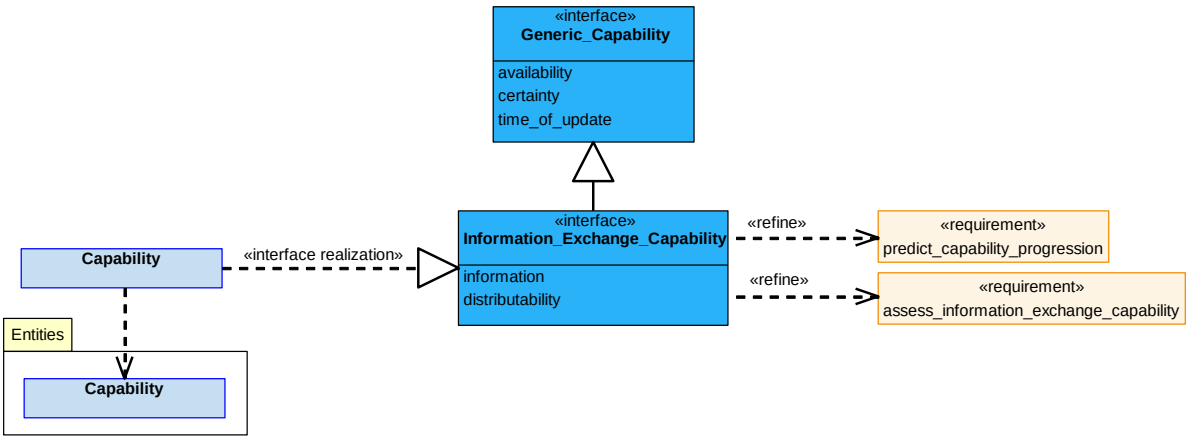

Figure 544: Capability Service Definition

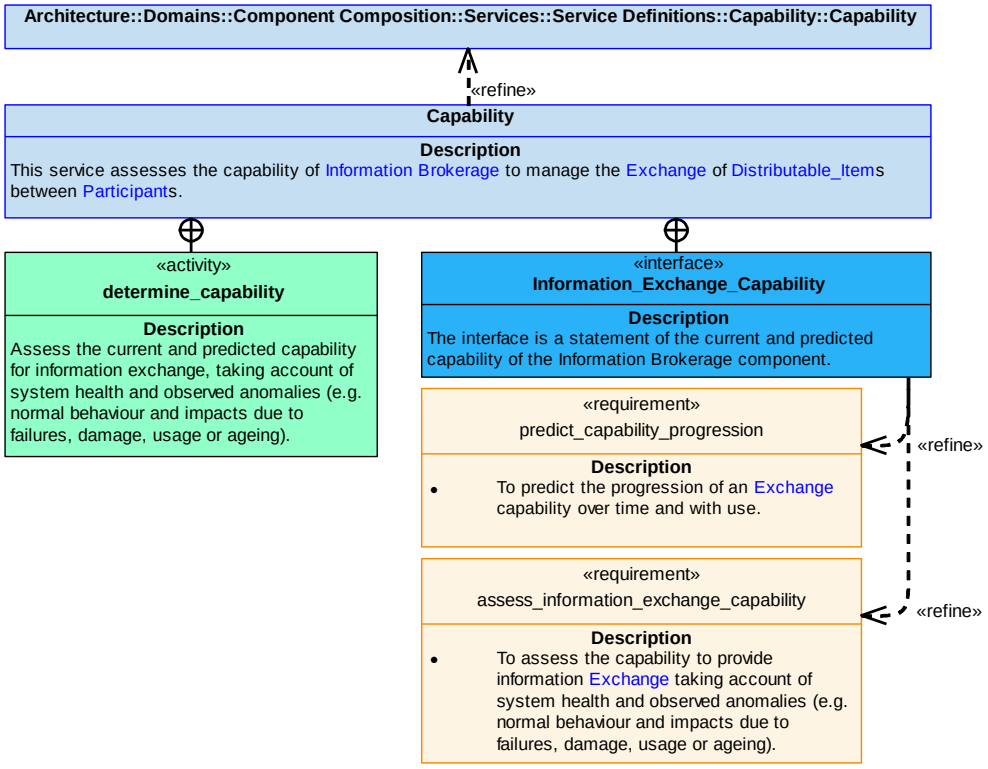

Figure 545: Capability Service Policy

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).

B.2.26.7.1.7 Capability_Evidence

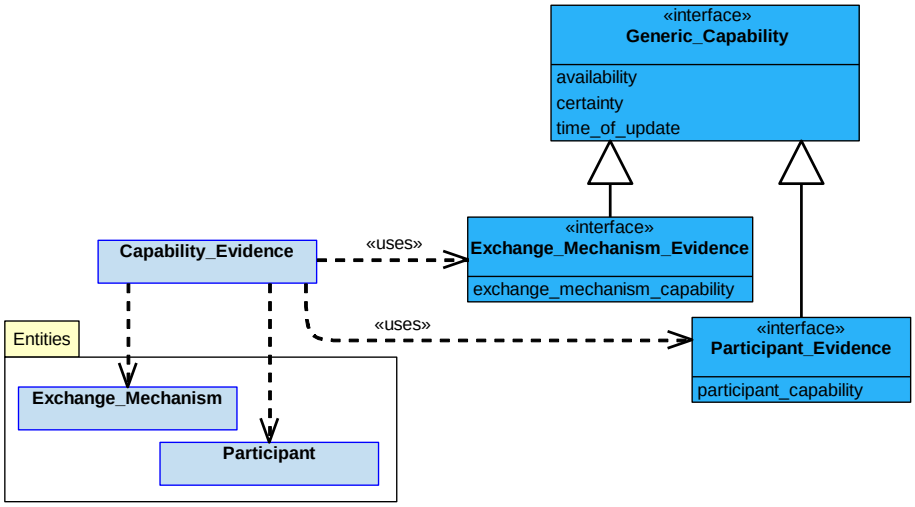

Figure 546: Capability_Evidence Service Definition

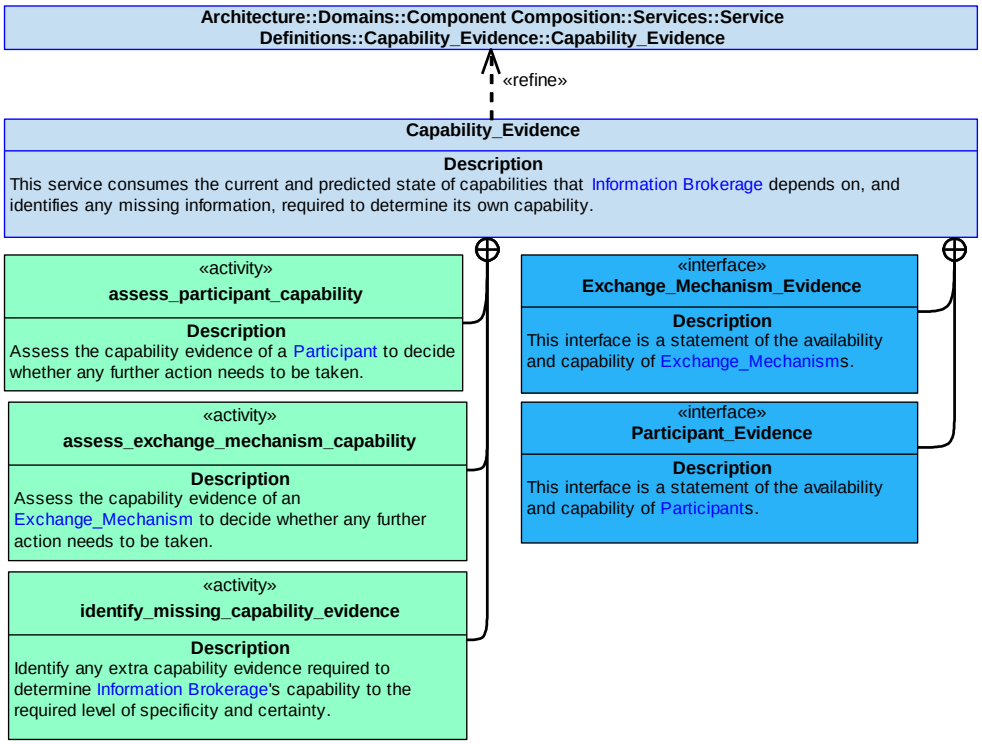

Figure 547: Capability_Evidence Service Policy

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.

B.2.26.7.2 Service Dependencies

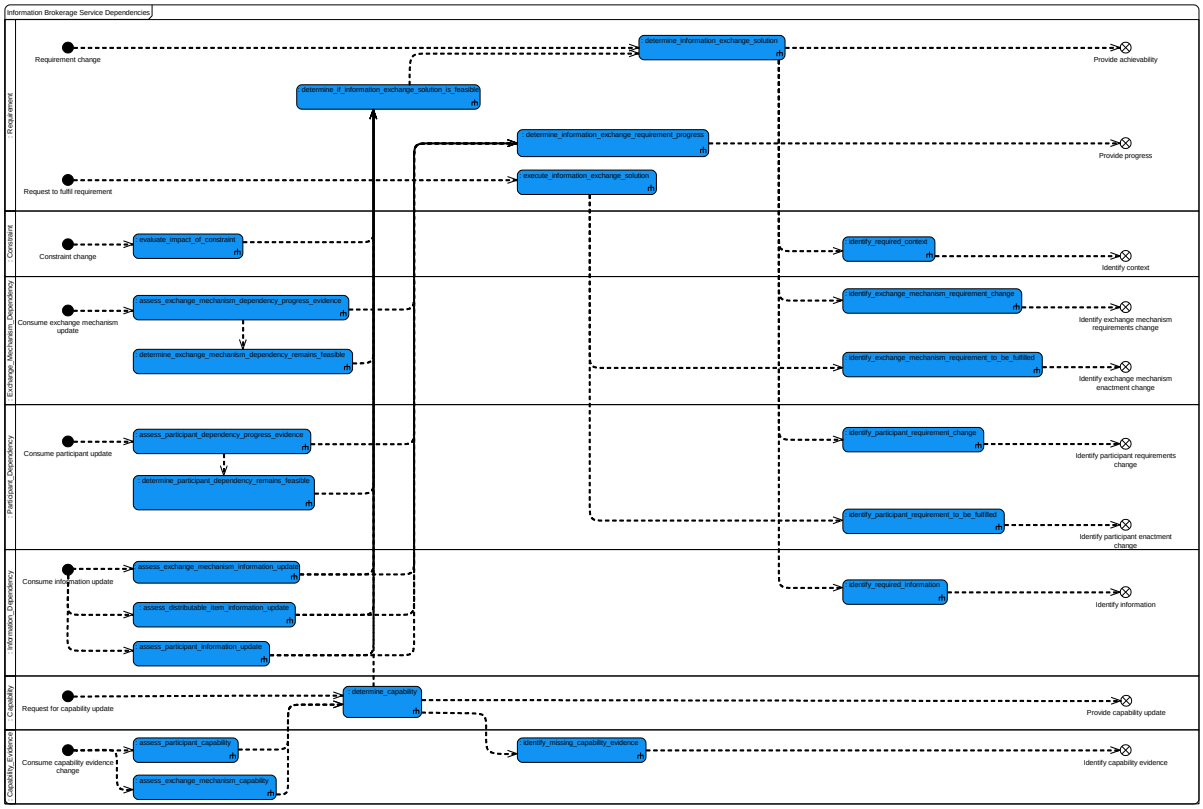

Figure 548: Information Brokerage Service Dependencies

B.2.27 Information Presentation

B.2.27.1 Role

The role of Information Presentation is to represent the conveyance and perception of information between the system and an HMI user.

B.2.27.2 Overview

Control Architecture

Information Presentation is a service component as defined in the Control Architecture policy.

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.

B.2.27.3 Service Summary

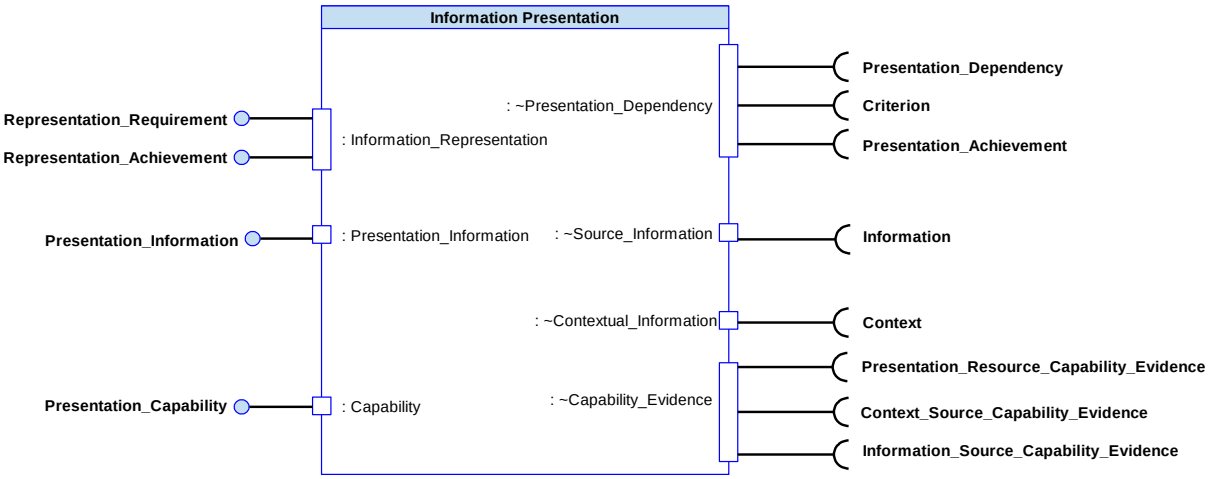

Figure 549: Information Presentation Service Summary

B.2.27.4 Responsibilities

capture_conveyance_requirement

- To capture Requirements for Presentation_Interactions. This can be for information coming into the system or from the system.

fulfil_requirement

- To convey the 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.

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.

B.2.27.5 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.

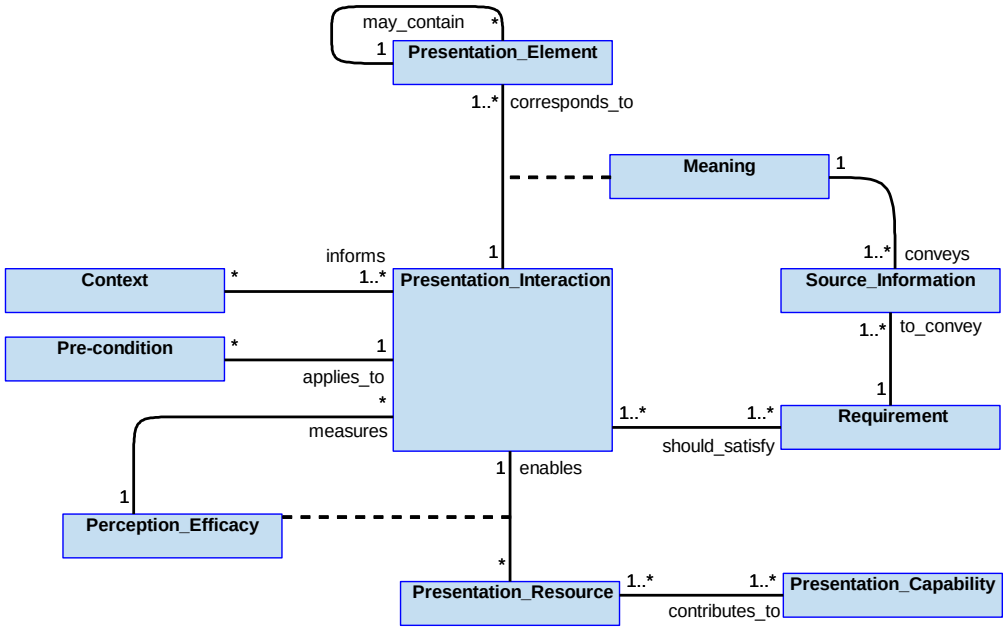

Figure 550: Information Presentation Semantics

B.2.27.5.1 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 lat/long of a waypoint.

B.2.27.6 Design Rationale

B.2.27.6.1 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.

B.2.27.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Information Presentation:

- Human-Machine Interface - The HMI policy 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, depending on the requirements of the exploitation.
- 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. [20], 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.

B.2.27.6.3 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 PRA deployment. Analysis of the specific uses of each instance, by the Exploiting Programme, may justify a less onerous DAL for some instances.

B.2.27.6.4 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 performed, etc.

B.2.27.7 Services

B.2.27.7.1 Service Definitions

B.2.27.7.1.1 Information_Representation

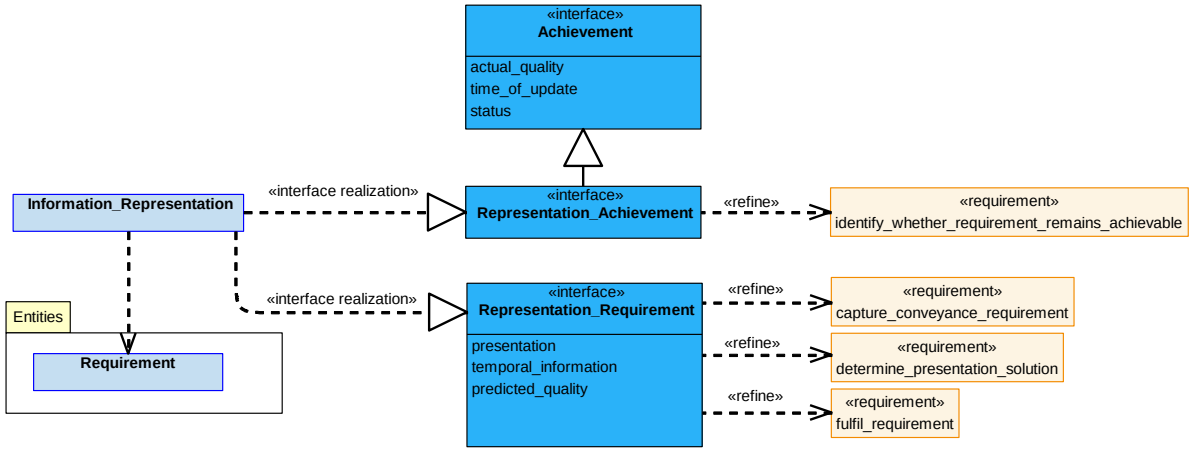

Figure 551: Information_Representation Service Definition

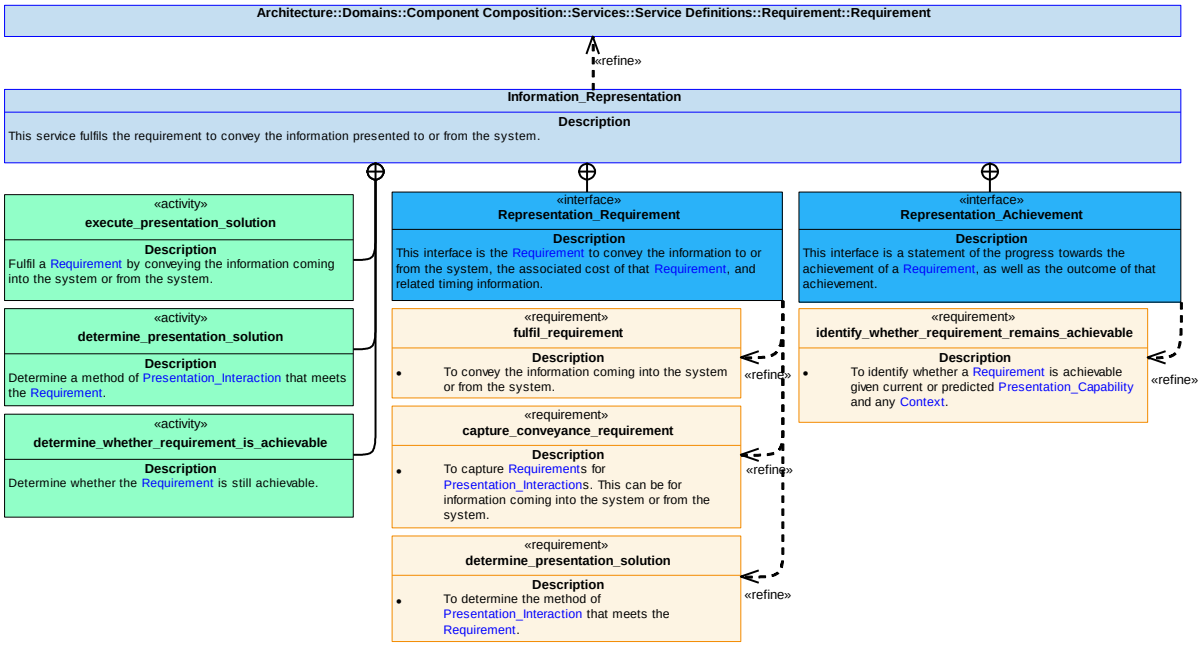

Figure 552: Information_Representation Service Policy

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.

B.2.27.7.1.2 Presentation_Dependency

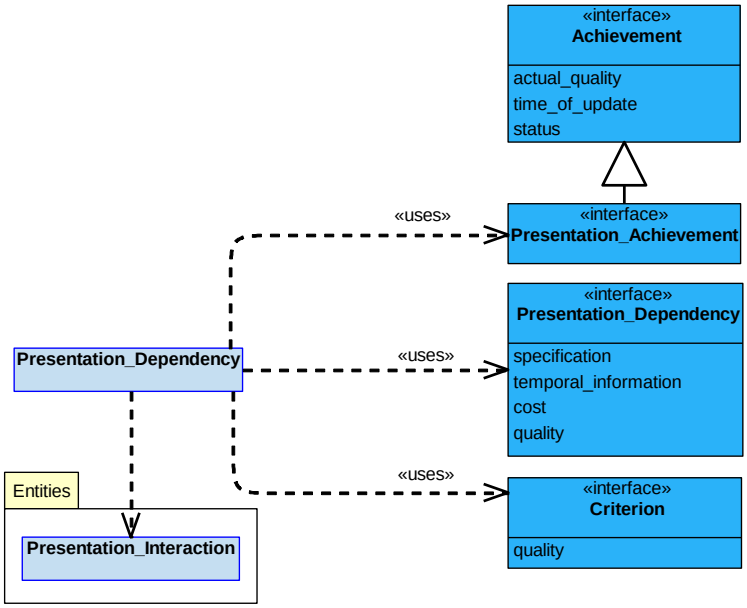

Figure 553: Presentation_Dependency Service Definition

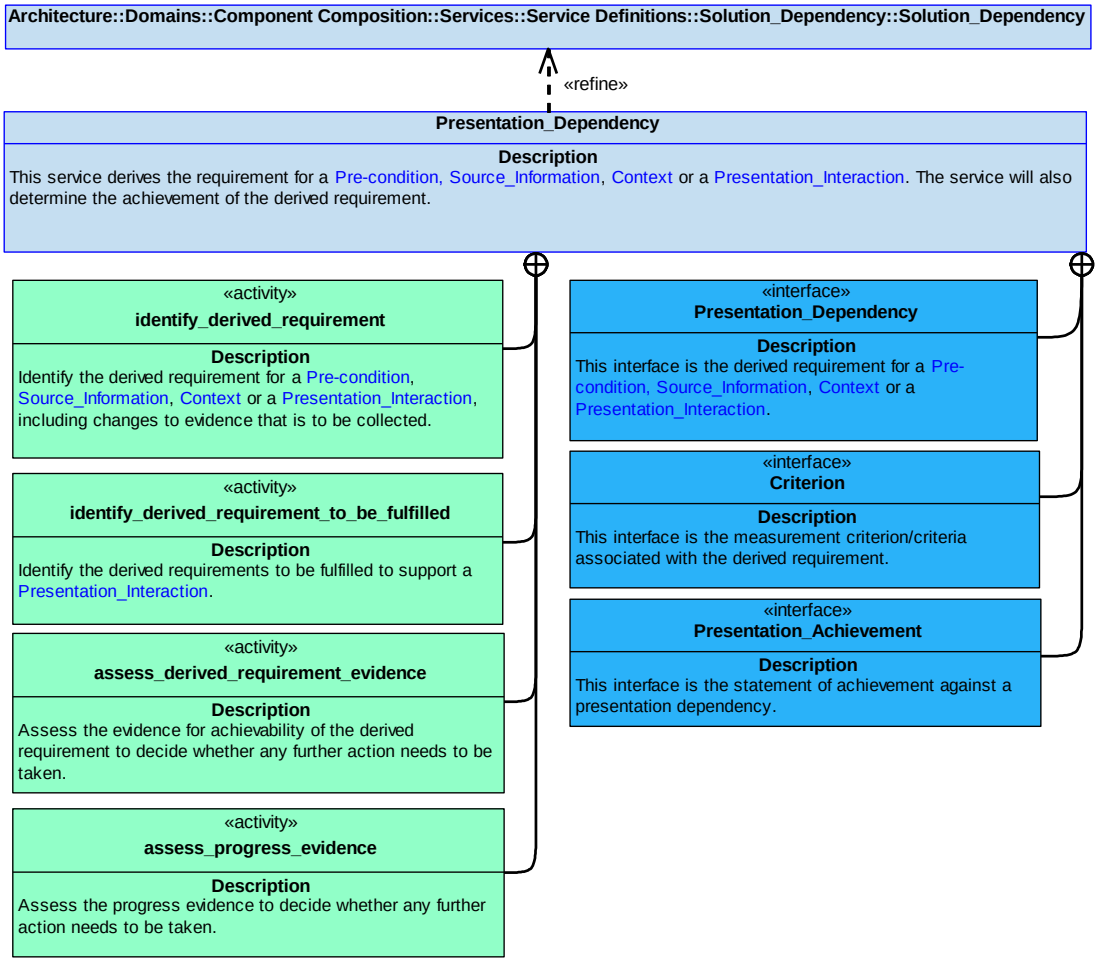

Figure 554: Presentation_Dependency Service Policy

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, 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.

B.2.27.7.1.3 Presentation_Information

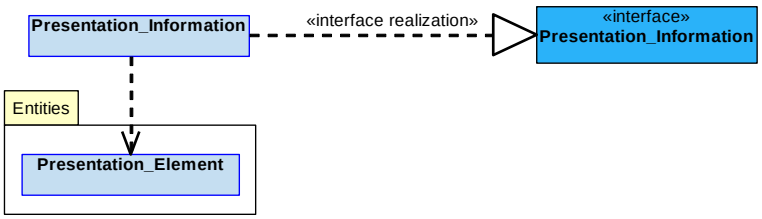

Figure 555: Presentation_Information Service Definition

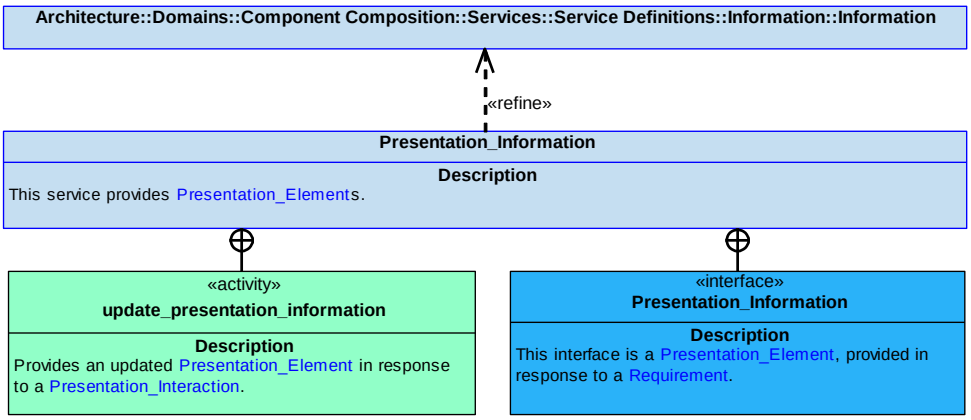

Figure 556: Presentation_Information Service Policy

Presentation_Information

This service provides Presentation_Elements.

Interface

Presentation_Information

This interface is a Presentation_Element, provided in response to a Requirement.

Activity

update_presentation_information

Provides an updated Presentation_Element in response to a Presentation_Interaction.

B.2.27.7.1.4 Source_Information

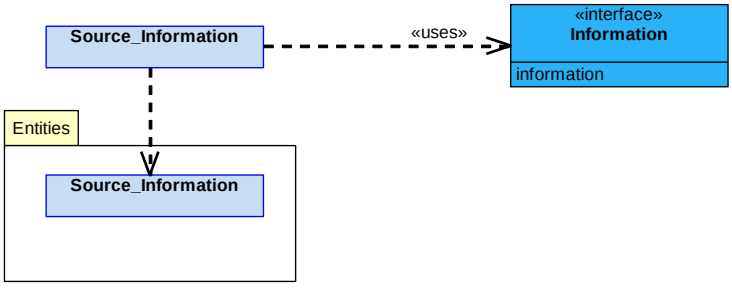

Figure 557: Source_Information Service Definition

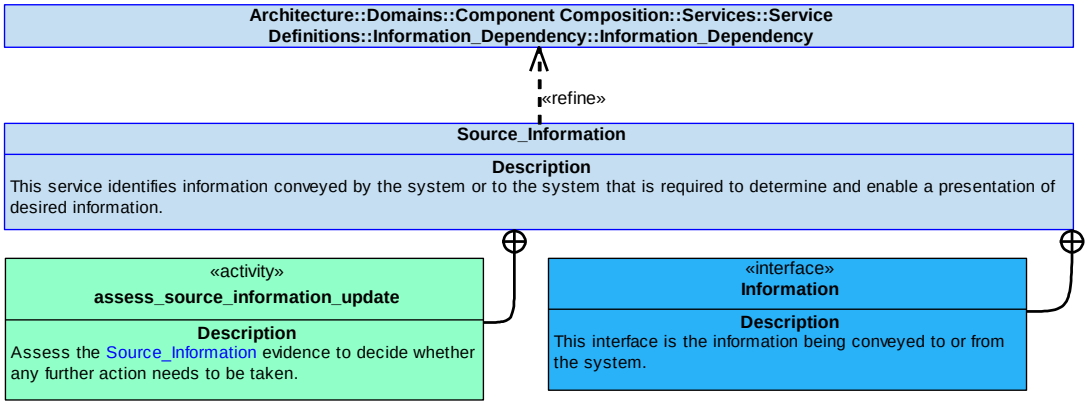

Figure 558: Source_Information Service Policy

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.

B.2.27.7.1.5 Contextual_Information

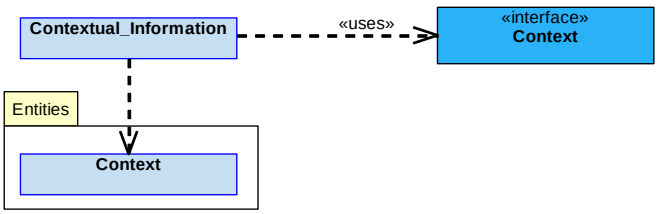

Figure 559: Contextual_Information Service Definition

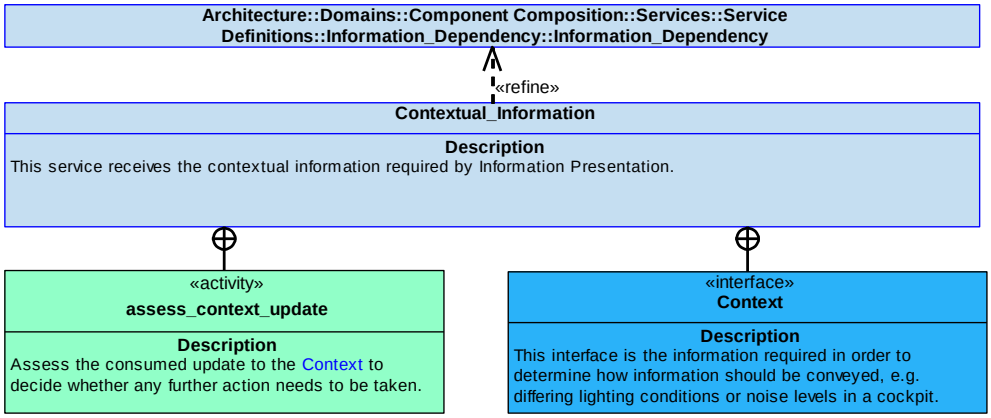

Figure 560: Contextual_Information Service Policy

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.

B.2.27.7.1.6 Capability

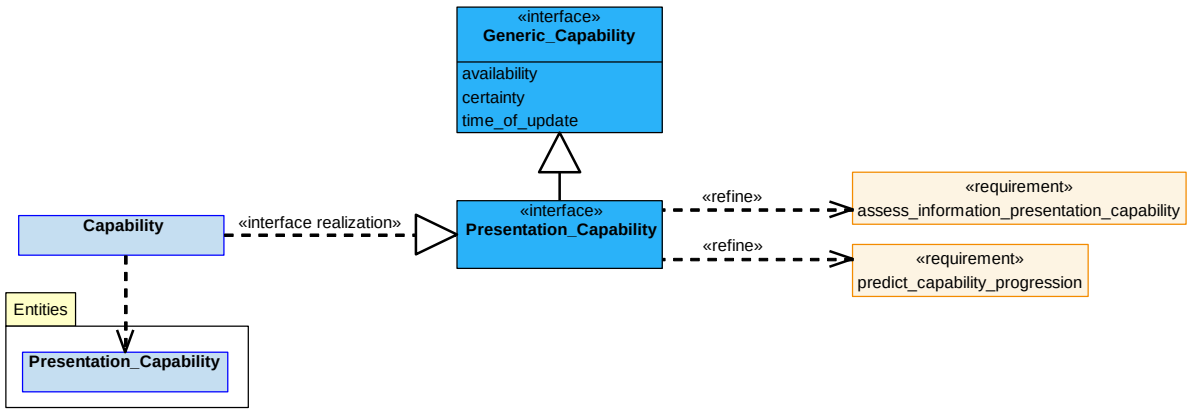

Figure 561: Capability Service Definition

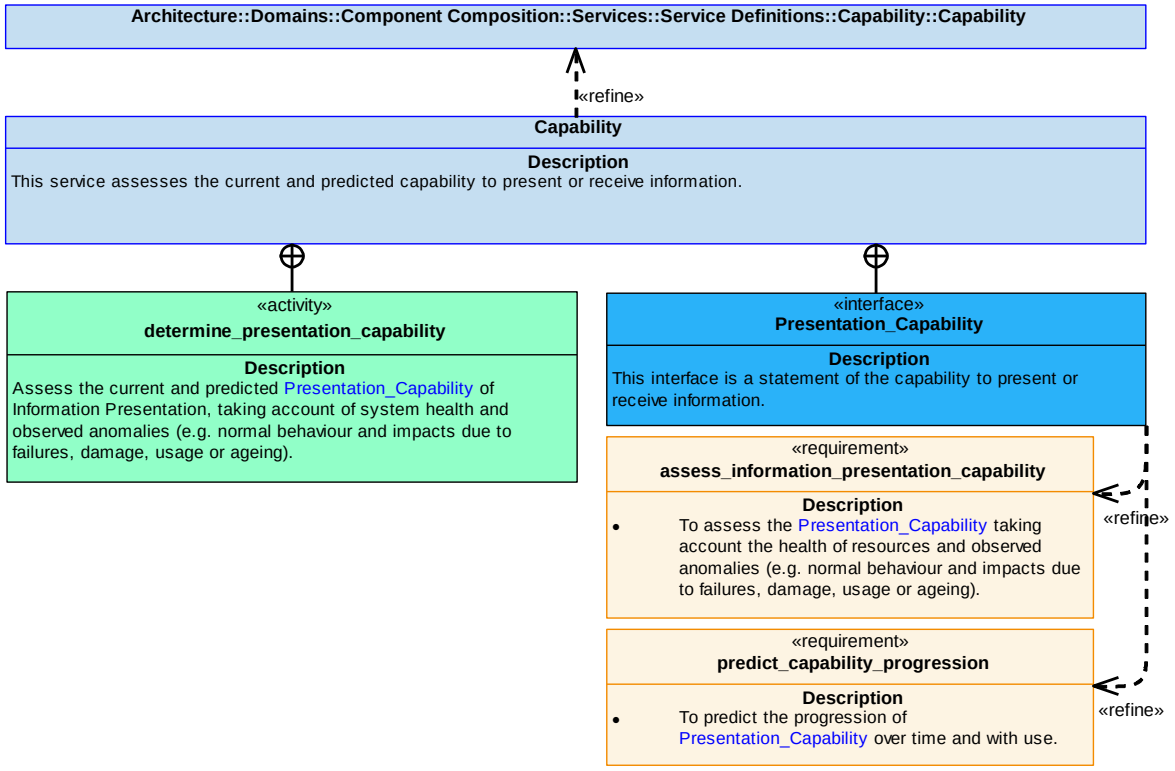

Figure 562: Capability Service Policy

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).

B.2.27.7.1.7 Capability_Evidence

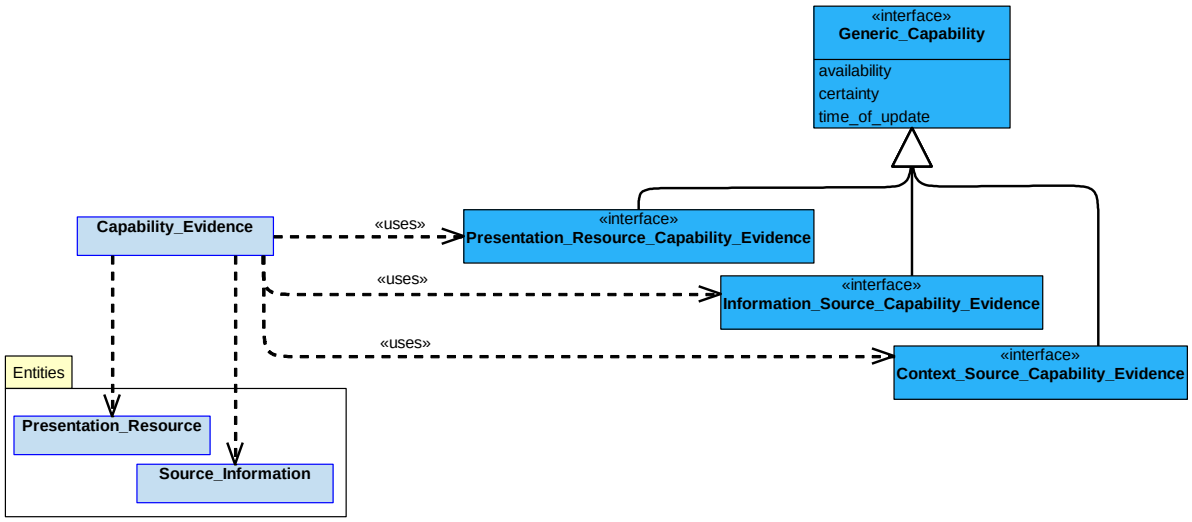

Figure 563: Capability_Evidence Service Definition

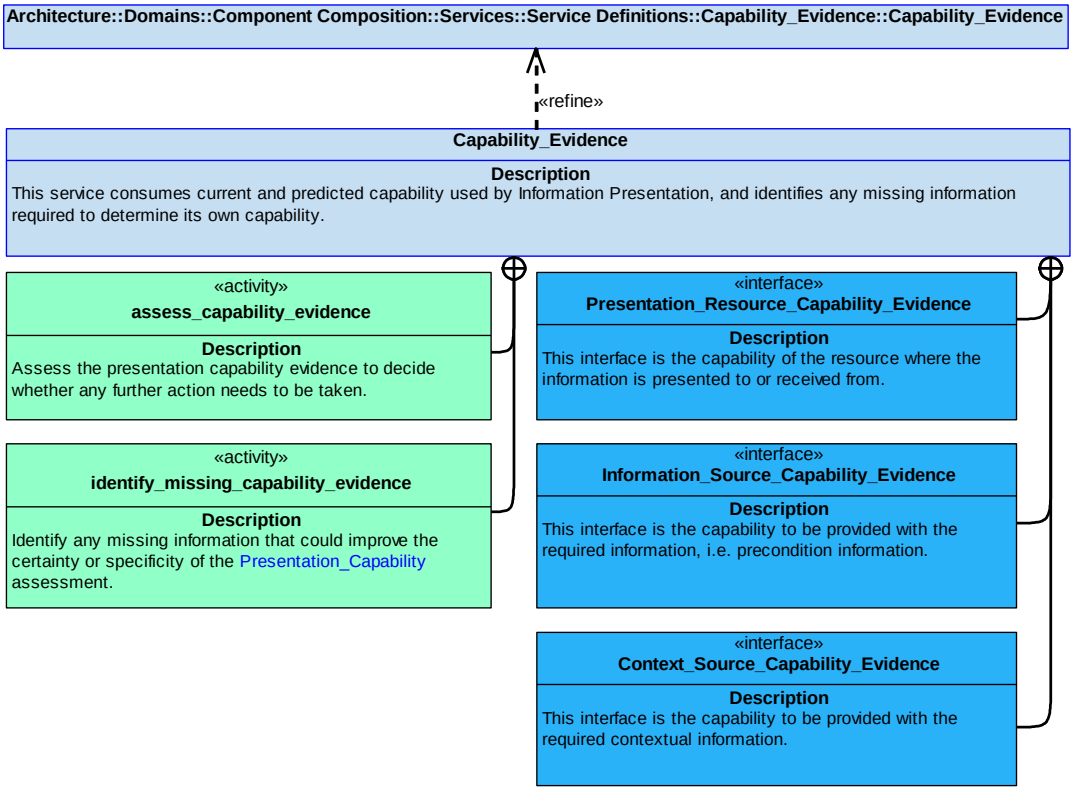

Figure 564: Capability_Evidence Service Policy

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.

B.2.27.7.2 Service Dependencies

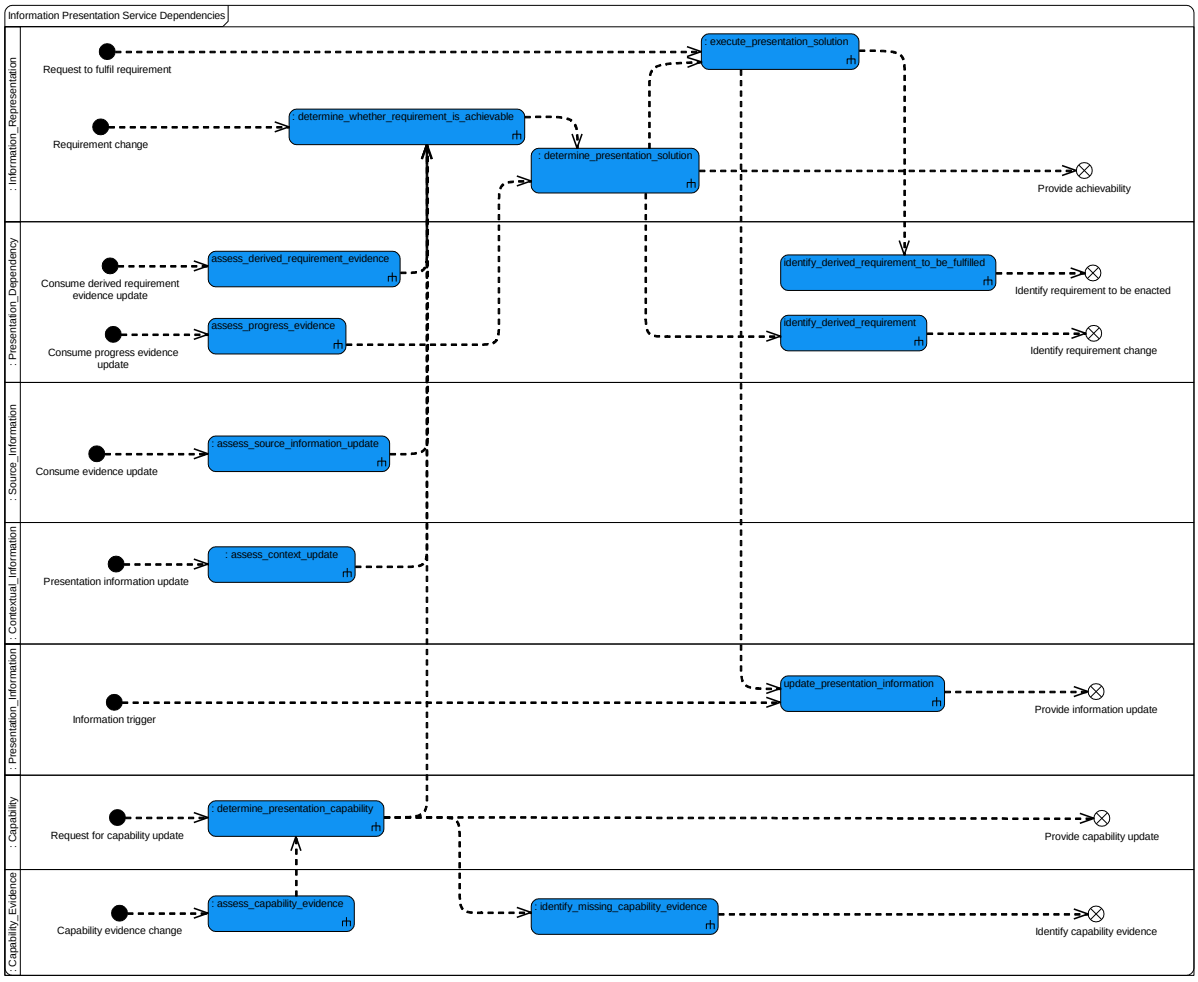

Figure 565: Information Presentation Service Dependencies

B.2.28 Interlocks

B.2.28.1 Role

The role of Interlocks is to enable and disable functions by checking when defined condition values have been met.

B.2.28.2 Overview

Control Architecture

Interlocks is a resource component as defined in the Control Architecture policy.

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.

B.2.28.3 Service Summary

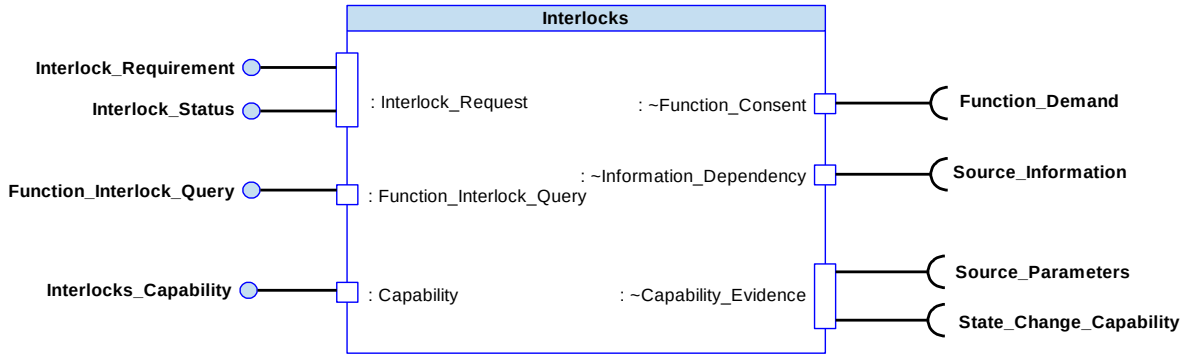

Figure 566: Interlocks Service Summary

B.2.28.4 Responsibilities

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.

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.

control_interlock

- To control Function_Interlocks to enable or disable functions.

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).

identify_current_states

- To maintain a view of the current states of Function_Interlocks.

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.

determine_if_conditional_check_remains_feasible

- To determine if a planned or on-going Conditional_Check remains feasible.

B.2.28.5 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.

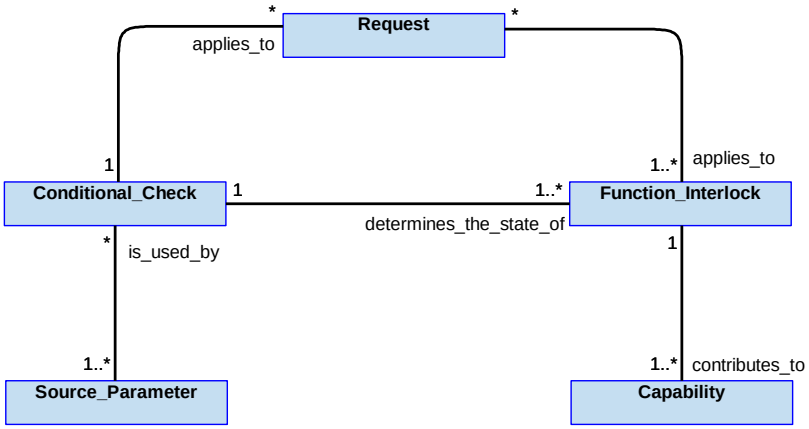

Figure 567: Interlocks Semantics

B.2.28.5.1 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.

B.2.28.6 Design Rationale

B.2.28.6.1 Assumptions

None.

B.2.28.6.2 Design Considerations

Directly Applicable Policies

These policies 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).

B.2.28.6.3 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.
- 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.

B.2.28.6.4 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.

B.2.28.7 Services

B.2.28.7.1 Service Definitions

B.2.28.7.1.1 Interlock_Request

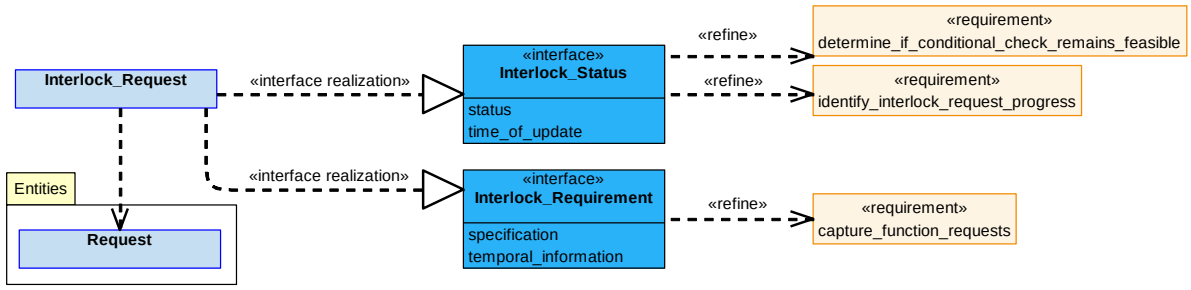

Figure 568: Interlock_Request Service Definition

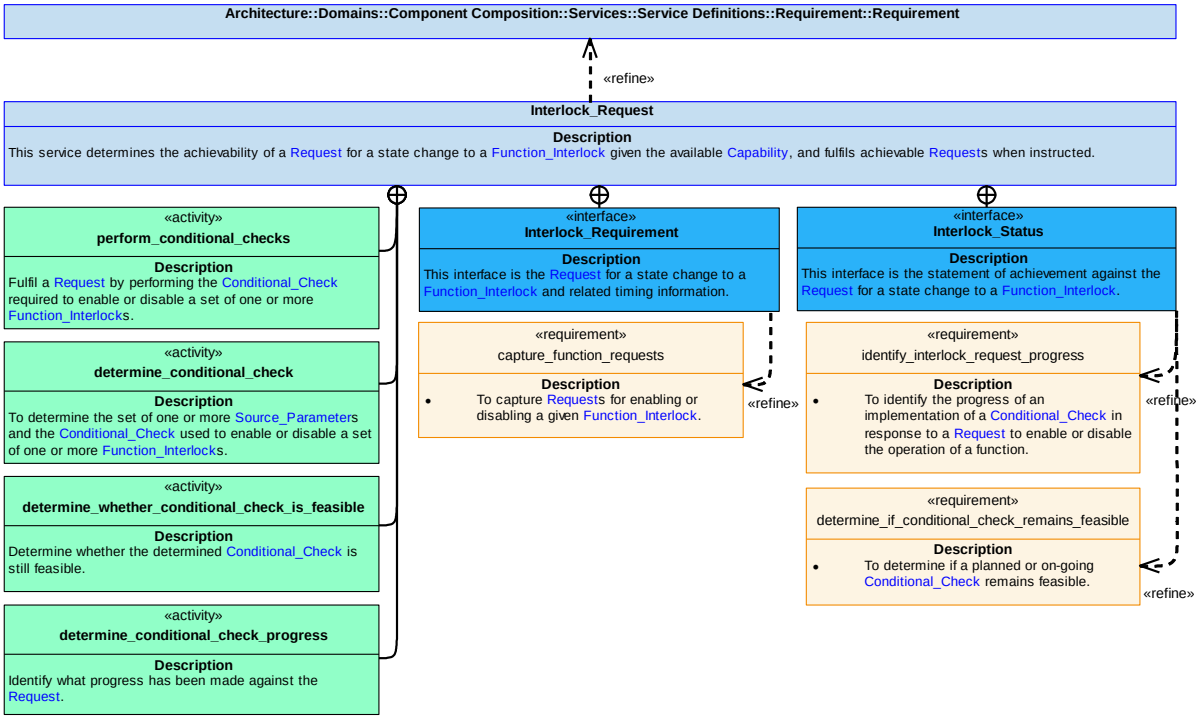

Figure 569: Interlock_Request Service Policy

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.

B.2.28.7.1.2 Function_Consent

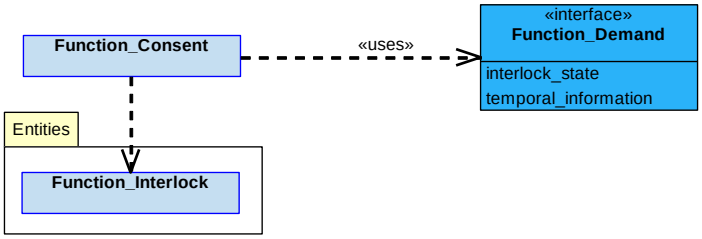

Figure 570: Function_Consent Service Definition

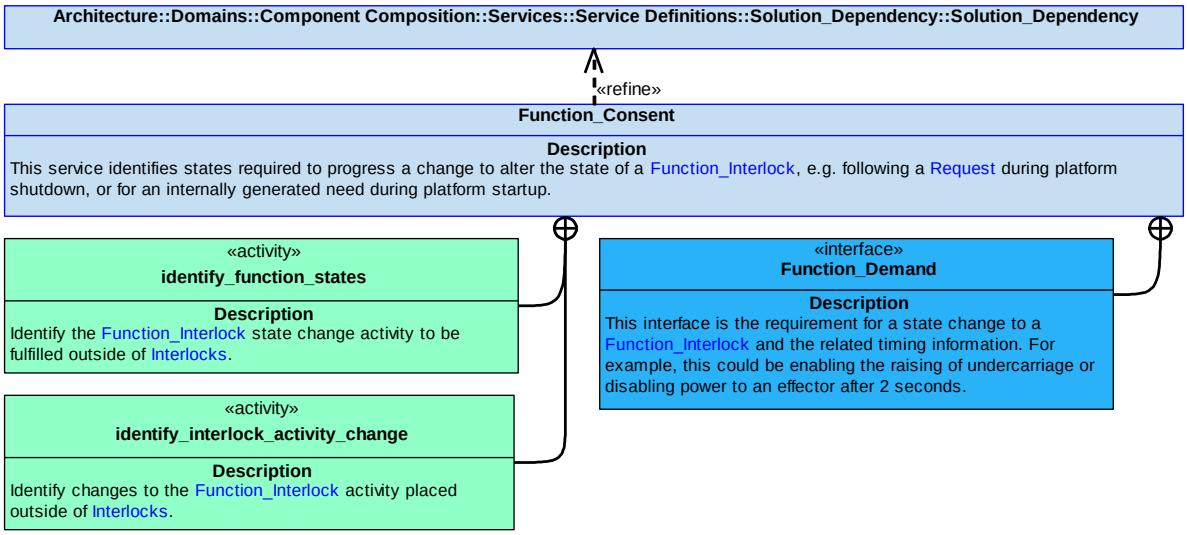

Figure 571: Function_Consent Service Policy

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 startup.

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.

B.2.28.7.1.3 Function_Interlock_Query

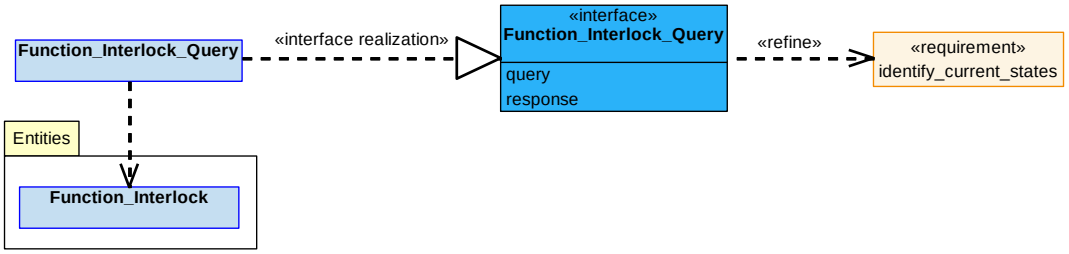

Figure 572: Function_Interlock_Query Service Definition

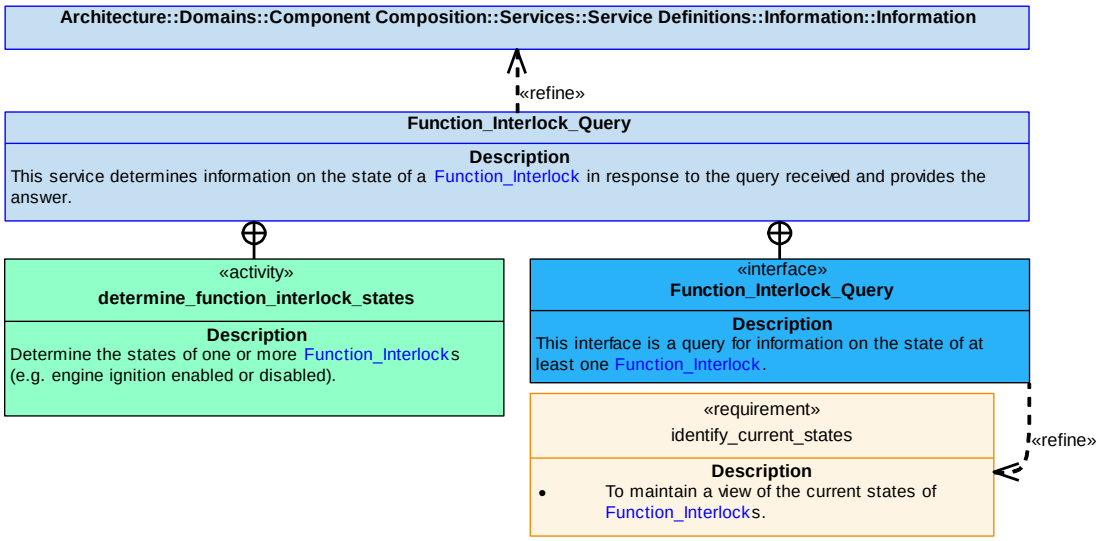

Figure 573: Function_Interlock_Query Service Policy

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).

B.2.28.7.1.4 Information_Dependency

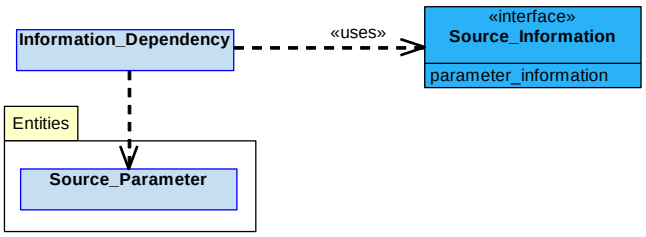

Figure 574: Information_Dependency Service Definition

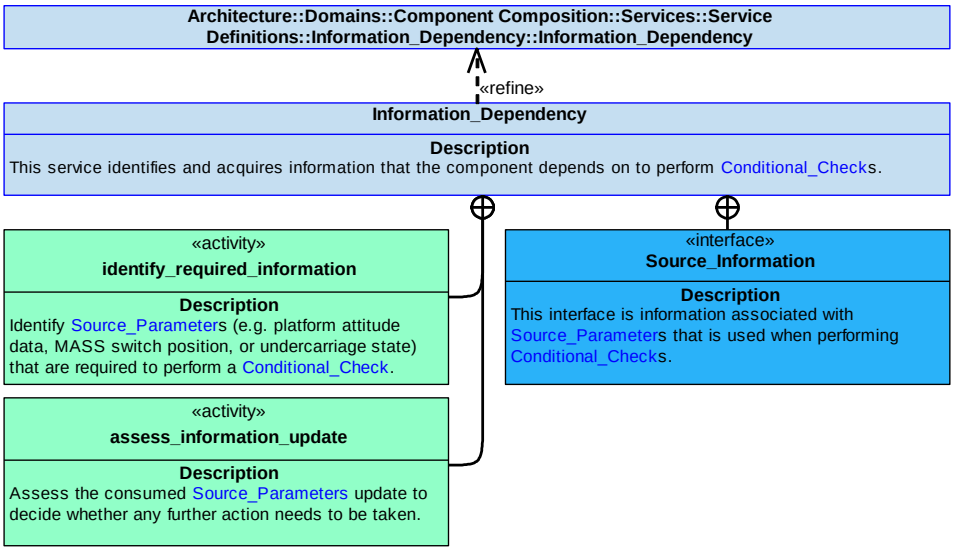

Figure 575: Information_Dependency Service Policy

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.

B.2.28.7.1.5 Capability

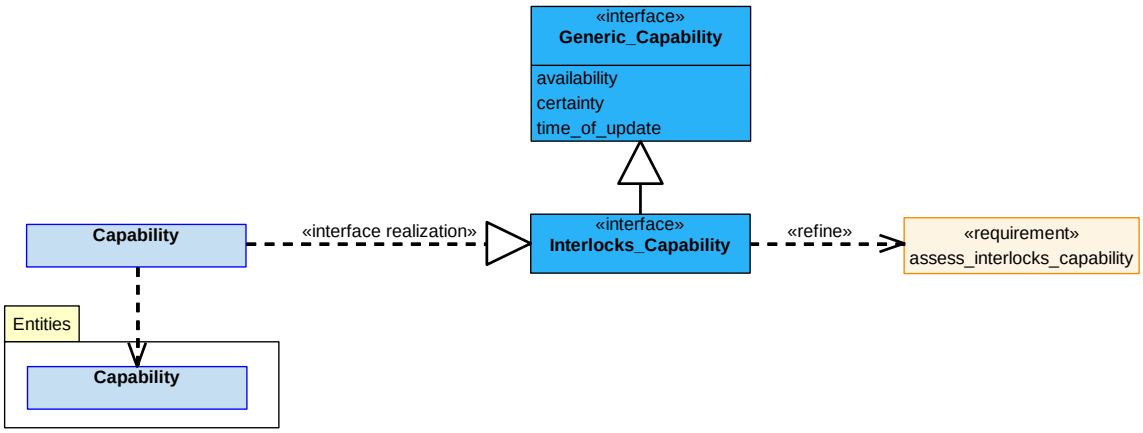

Figure 576: Capability Service Definition

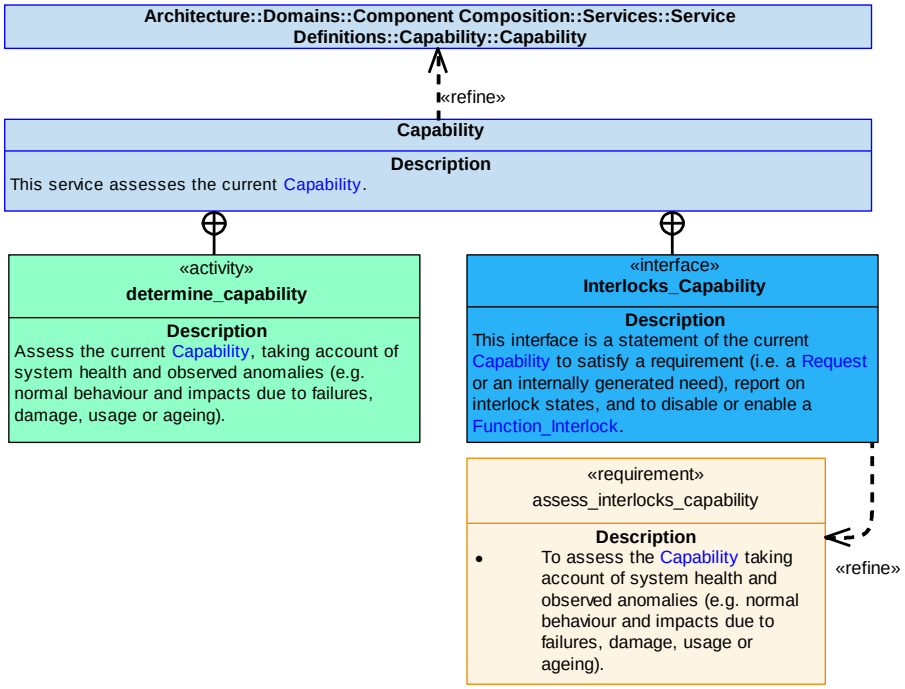

Figure 577: Capability Service Policy

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).

B.2.28.7.1.6 Capability_Evidence

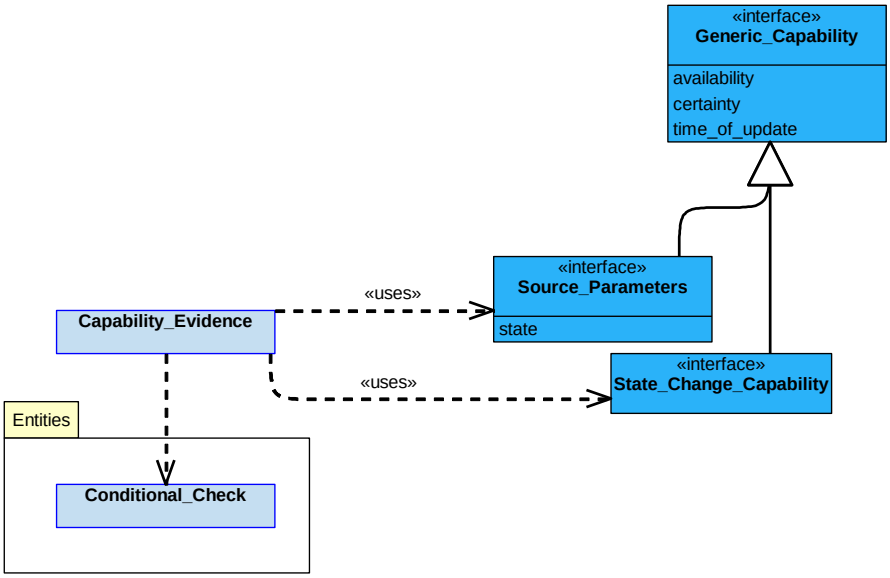

Figure 578: Capability_Evidence Service Definition

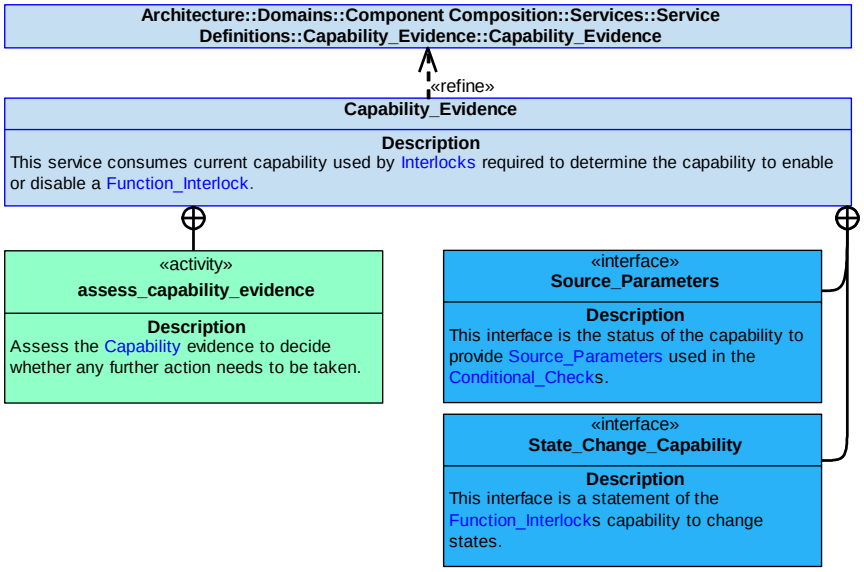

Figure 579: Capability_Evidence Service Policy

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.

B.2.28.7.2 Service Dependencies

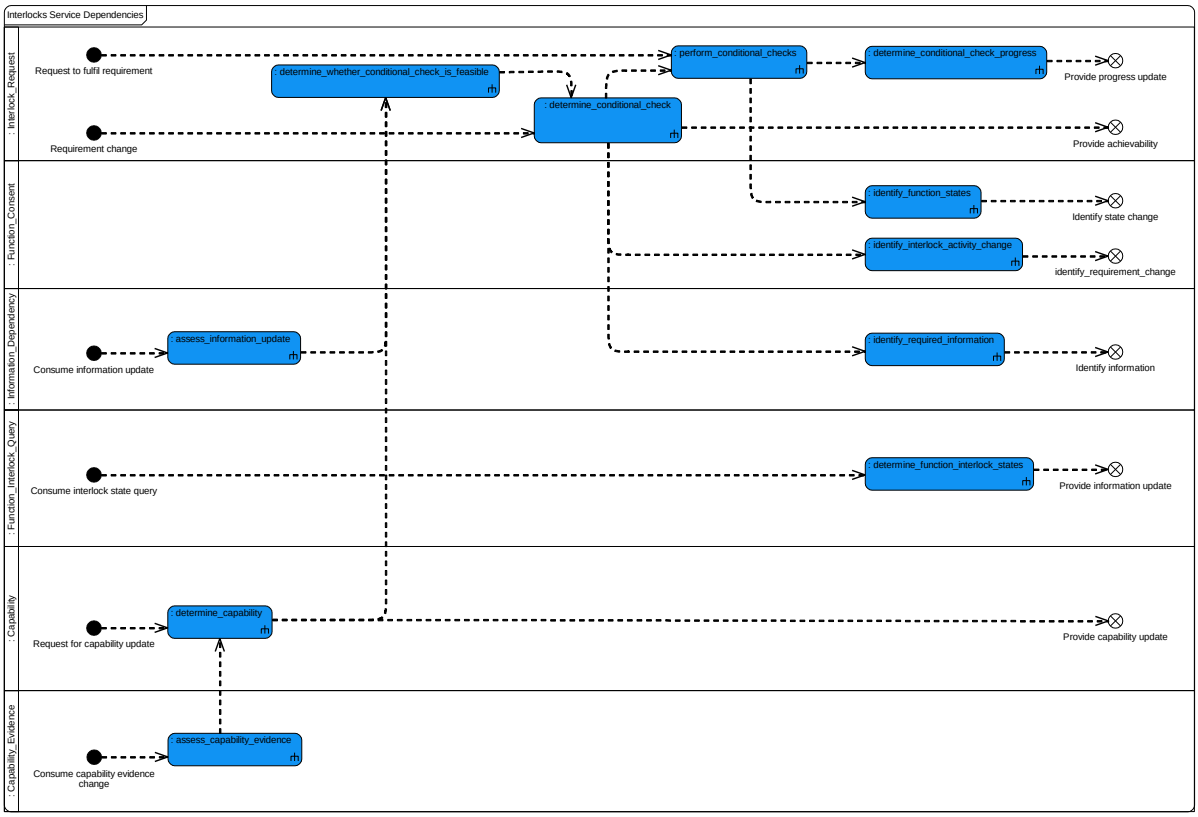

Figure 580: Interlocks Service Dependencies

B.2.29 Inventory

B.2.29.1 Role

The role of Inventory is to determine and verify the system inventory.

B.2.29.2 Overview

Control Architecture

Inventory is a service component as defined in the Control Architecture policy.

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.

B.2.29.3 Service Summary

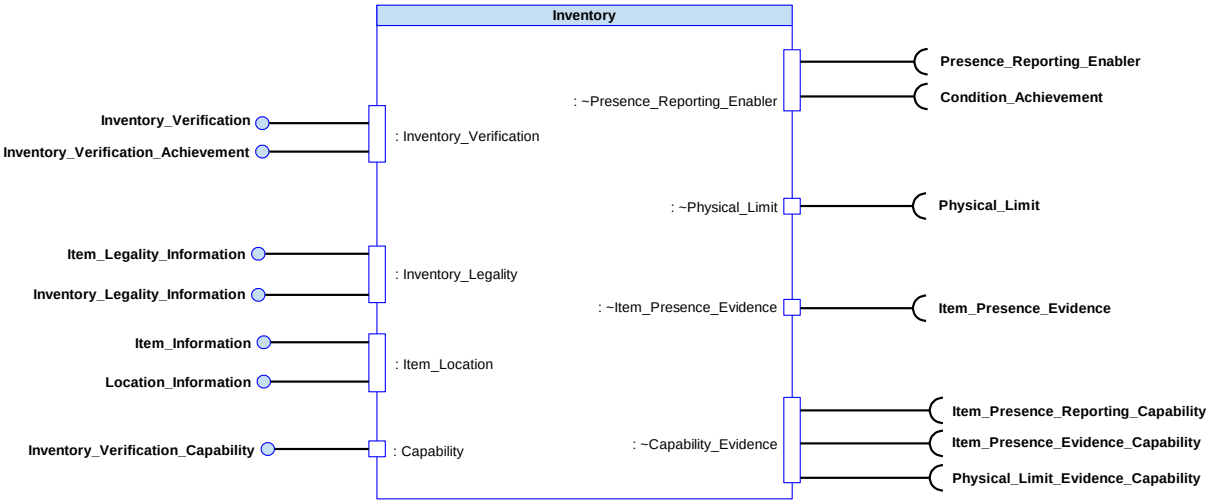

Figure 581: Inventory Service Summary

B.2.29.4 Responsibilities

verify_inventory

- To verify the current Inventory against the planned Inventory and to verify that an Inventory is a Legal_Inventory.

determine_inventory

- To determine the presence and identity of physical Items within the system Inventory.

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).

predict_inventory_verification_capability_progression

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

identify_progress_of_inventory_verification

- To identify the progress of Inventory determination and verification against the requirement.

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.

identify_missing_information

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

B.2.29.5 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.

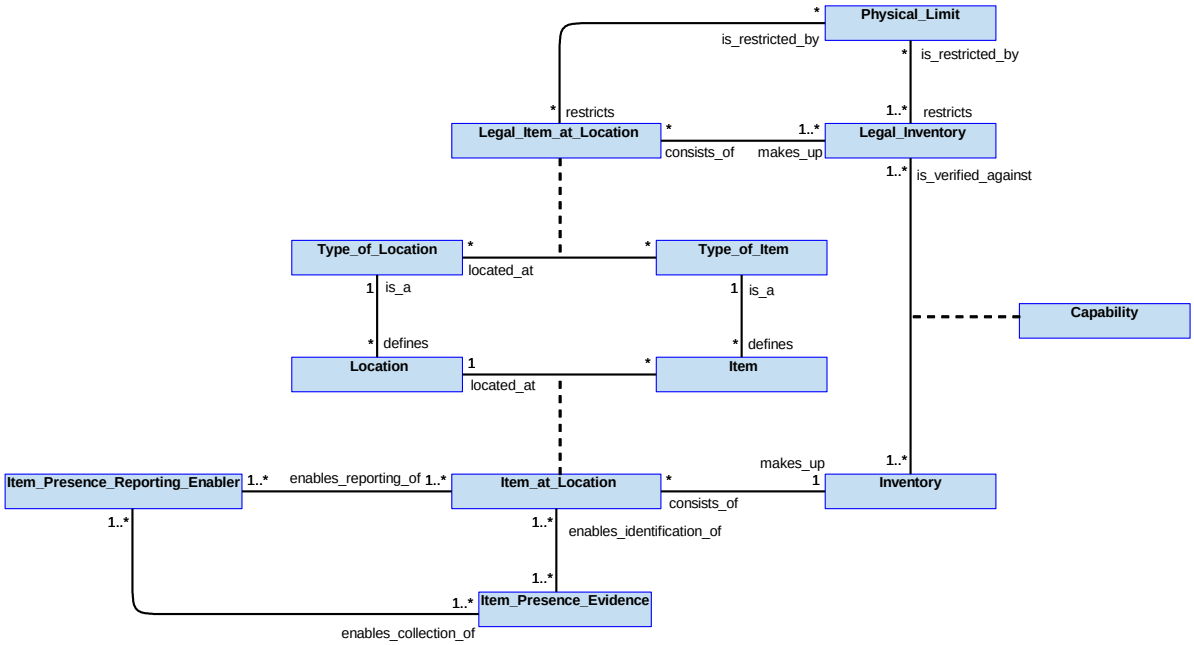

Figure 582: Inventory Semantics

B.2.29.5.1 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 500lb 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.

B.2.29.6 Design Rationale

B.2.29.6.1 Assumptions

None.

B.2.29.6.2 Design Considerations

Directly Applicable Policies

These policies 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 deployment time data in accordance with the Data Driving policy.
- 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 policy.

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.

B.2.29.6.3 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.

B.2.29.6.4 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.

B.2.29.7 Services

B.2.29.7.1 Service Definitions

B.2.29.7.1.1 Inventory_Verification

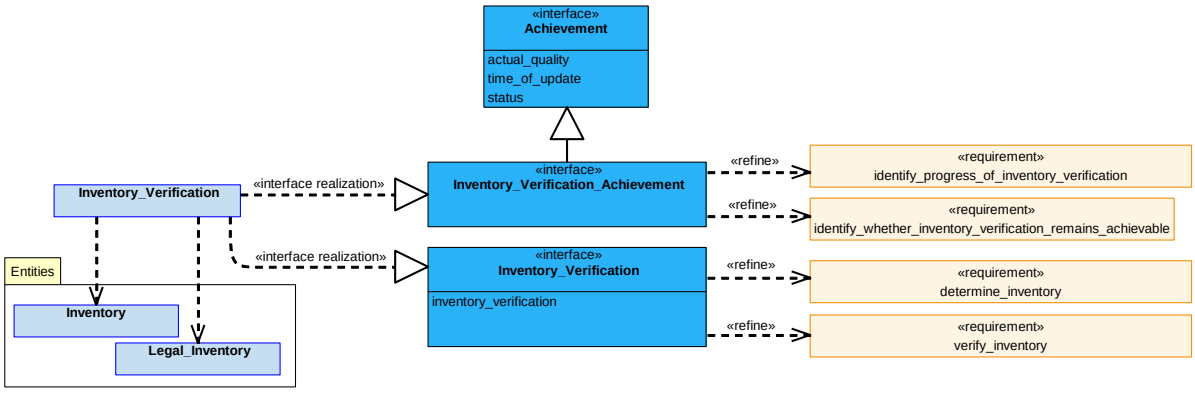

Figure 583: Inventory_Verification Service Definition

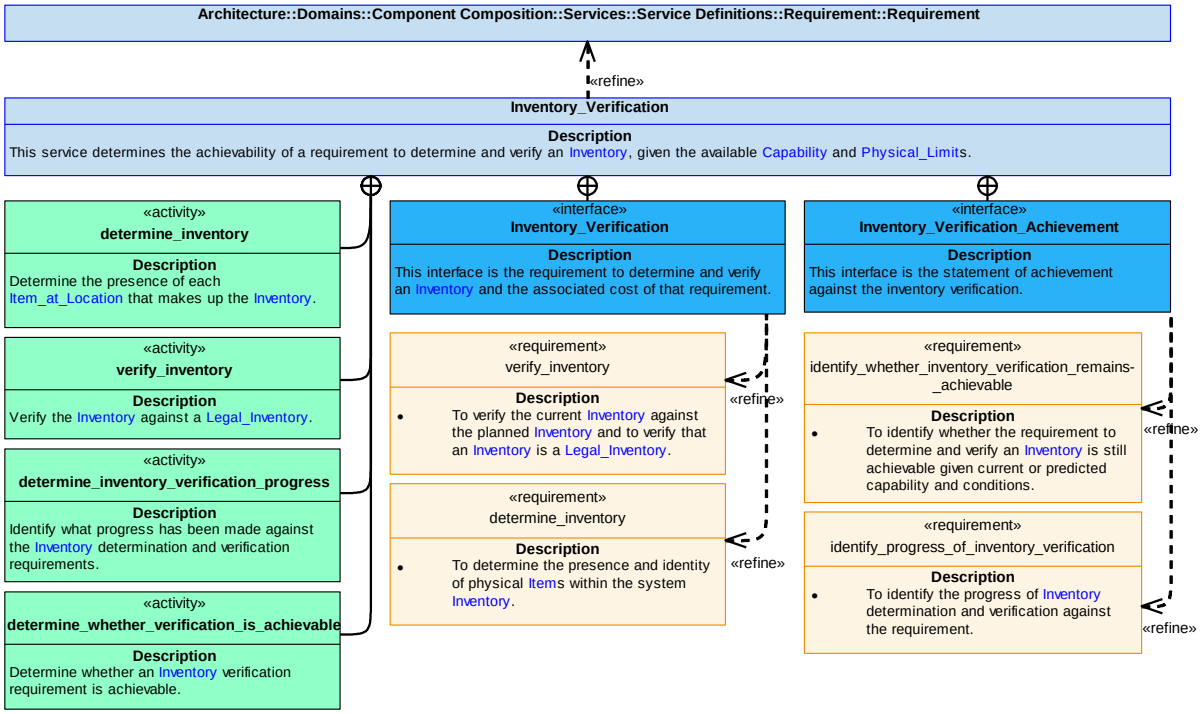

Figure 584: Inventory_Verification Service Policy

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.

B.2.29.7.1.2 Presence_Reporting_Enabler

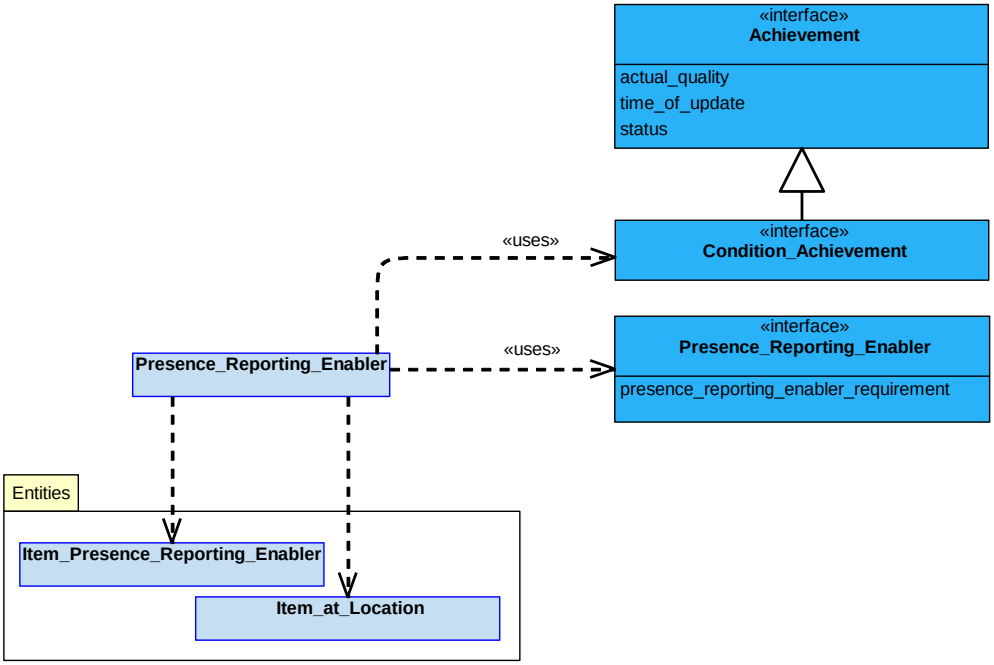

Figure 585: Presence_Reporting_Enabler Service Definition

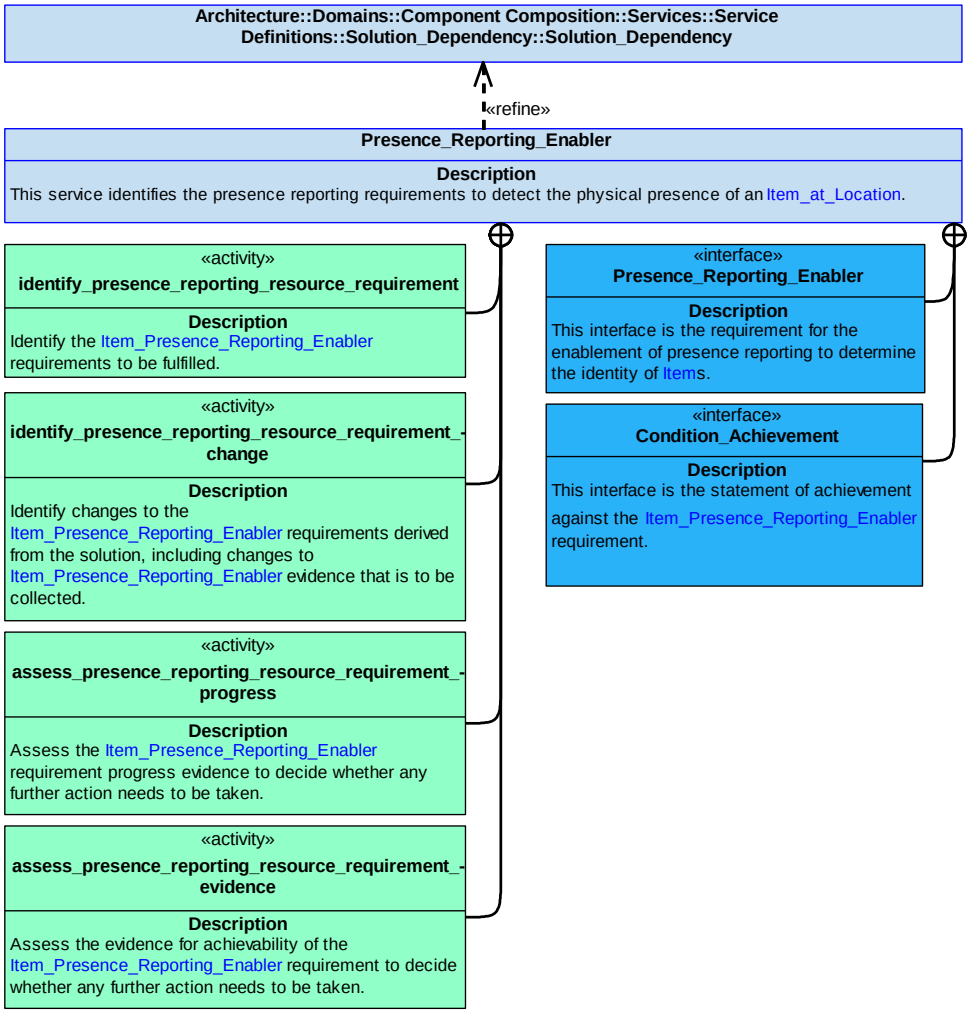

Figure 586: Presence_Reporting_Enabler Service Policy

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.

B.2.29.7.1.3 Physical_Limit

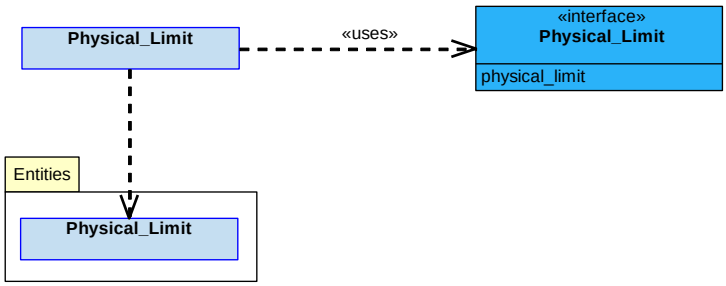

Figure 587: Physical_Limit Service Definition

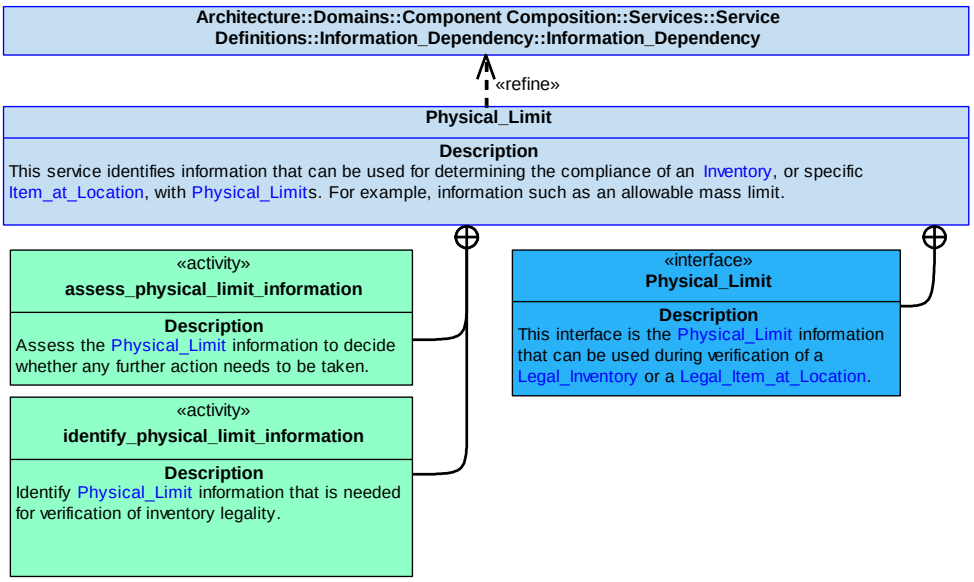

Figure 588: Physical_Limit Service Policy

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.

Interface

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.

Attribute

physical_limit The Physical_Limit information, e.g. the mass limit for a position associated with a Location.

Activities

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.

B.2.29.7.1.4 Inventory_Legality

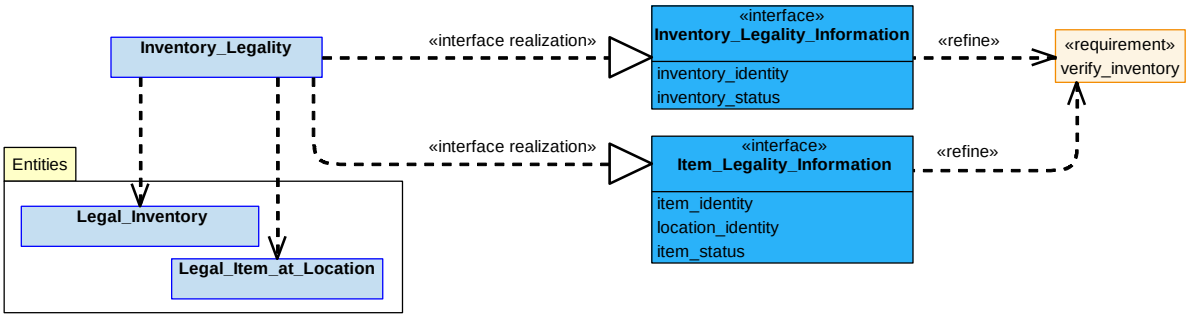

Figure 589: Inventory_Legality Service Definition

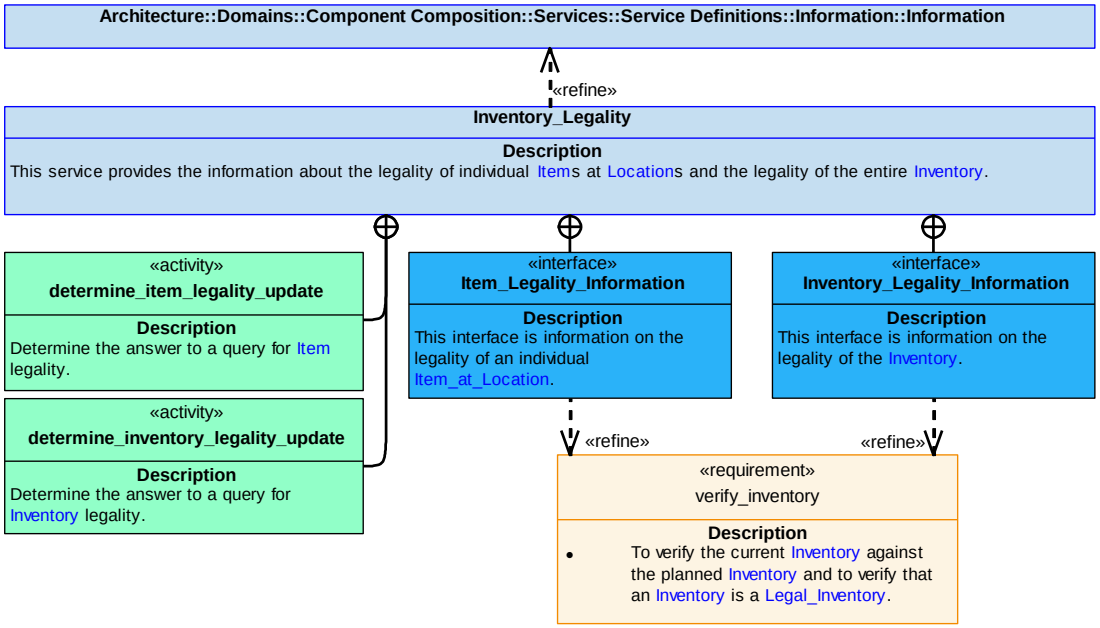

Figure 590: Inventory_Legality Service Policy

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.

B.2.29.7.1.5 Item_Location

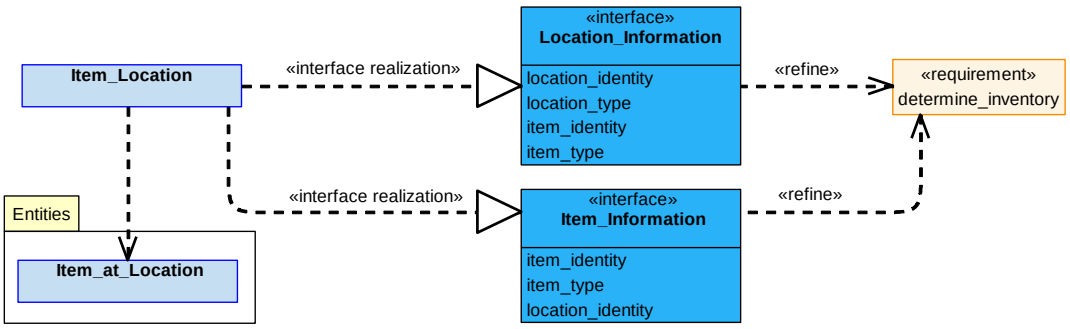

Figure 591: Item_Location Service Definition

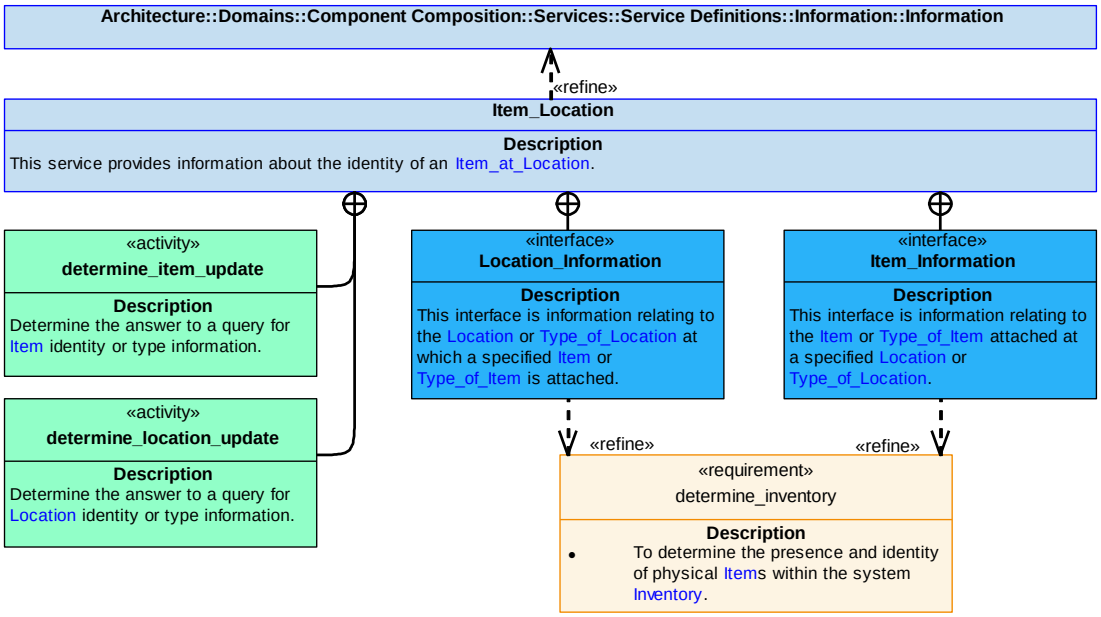

Figure 592: Item_Location Service Policy

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.

B.2.29.7.1.6 Item_Presence_Evidence

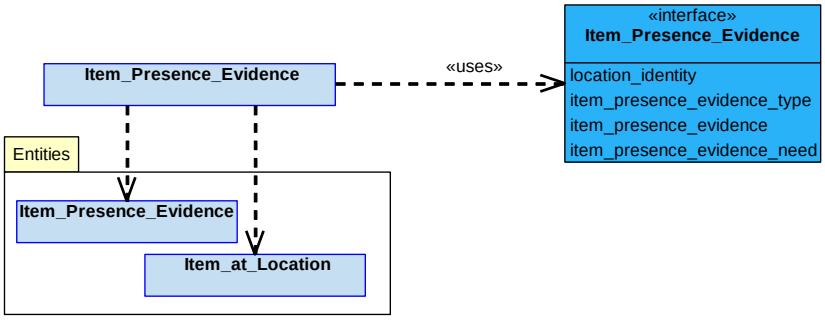

Figure 593: Item_Presence_Evidence Service Definition

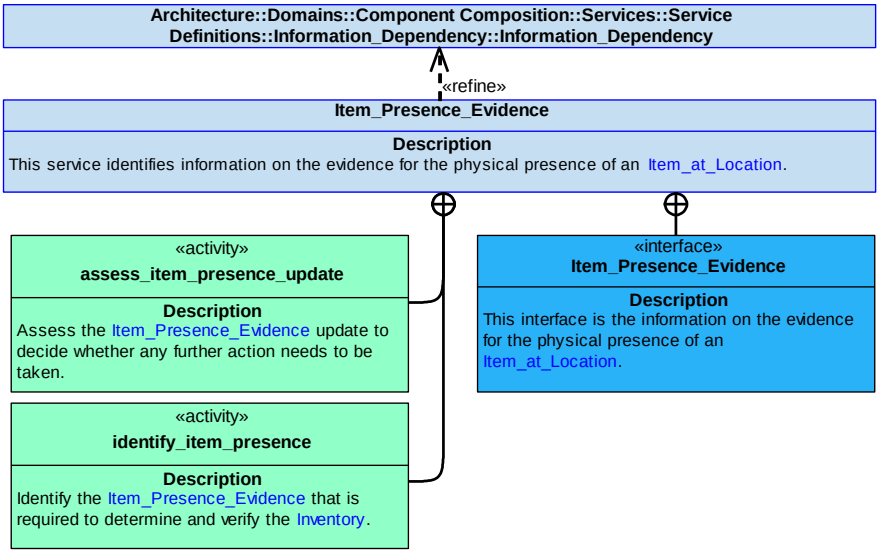

Figure 594: Item_Presence_Evidence Service Policy

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.

B.2.29.7.1.7 Capability

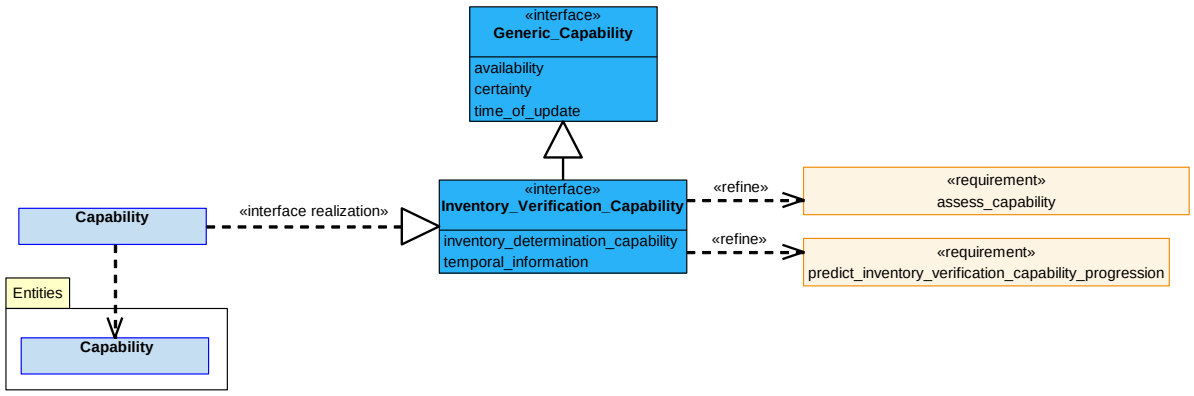

Figure 595: Capability Service Definition

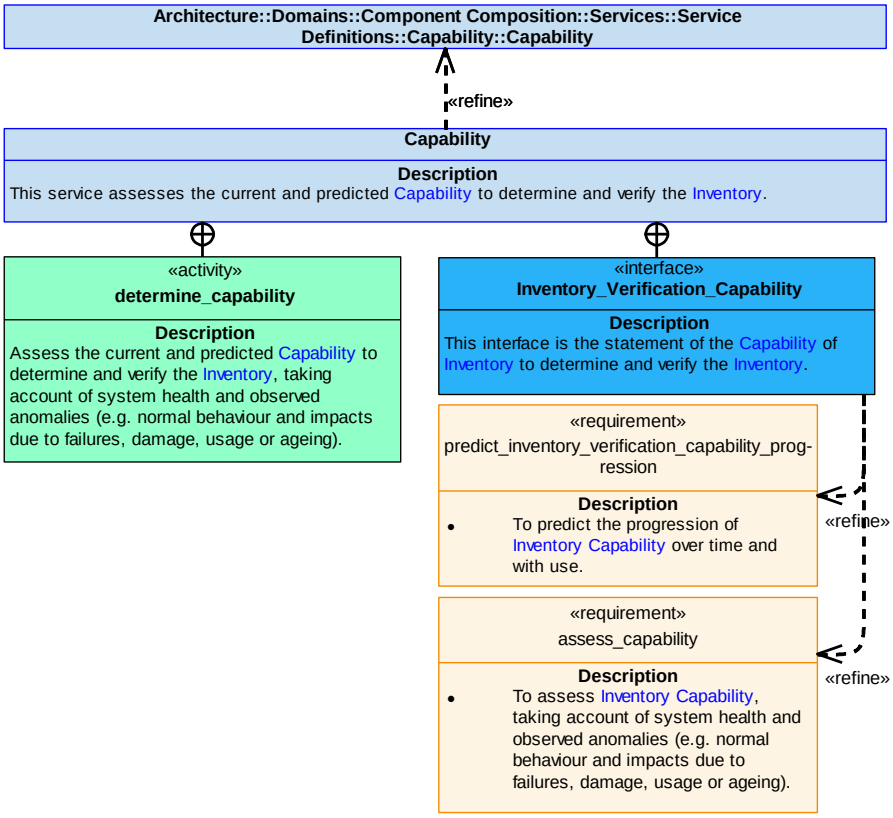

Figure 596: Capability Service Policy

Capability

This service assesses the current and predicted Capability to determine and verify the Inventory.

Interface

Inventory_Verification_Capability

This interface is the statement of the Capability of Inventory to determine and verify the Inventory.

Attributes

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).

B.2.29.7.1.8 Capability_Evidence

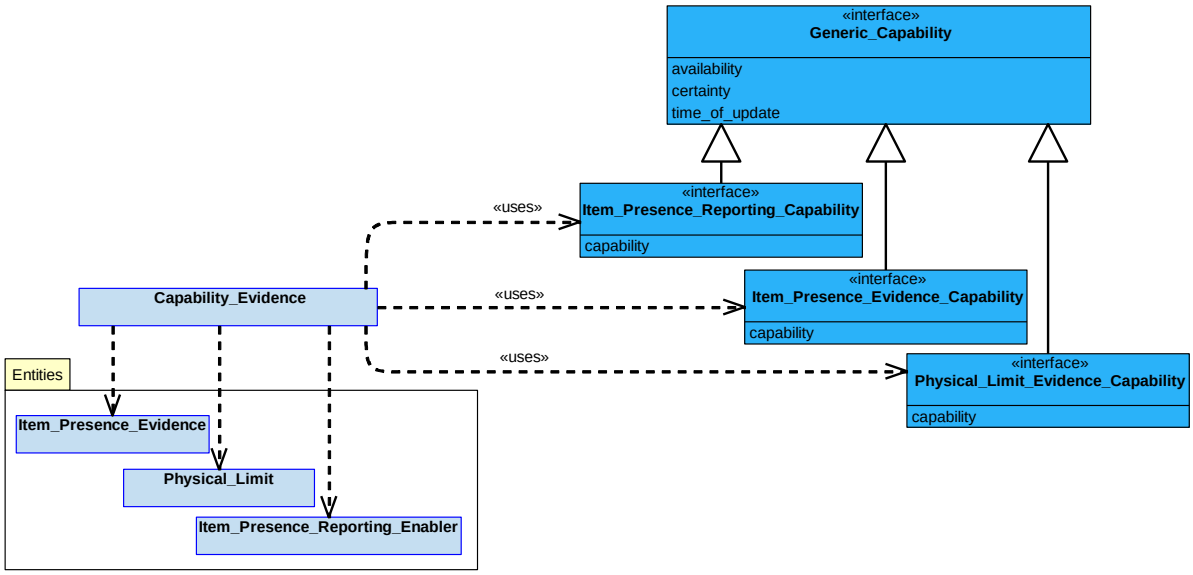

Figure 597: Capability_Evidence Service Definition

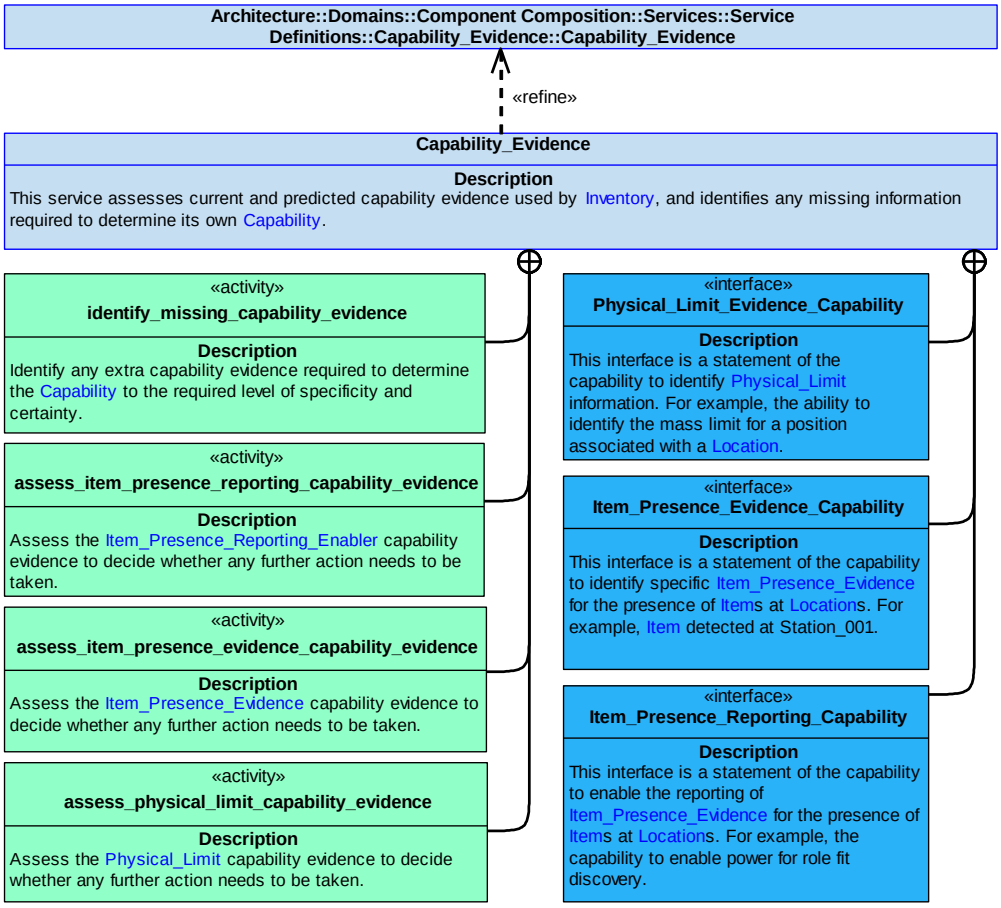

Figure 598: Capability_Evidence Service Policy

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.

B.2.29.7.2 Service Dependencies

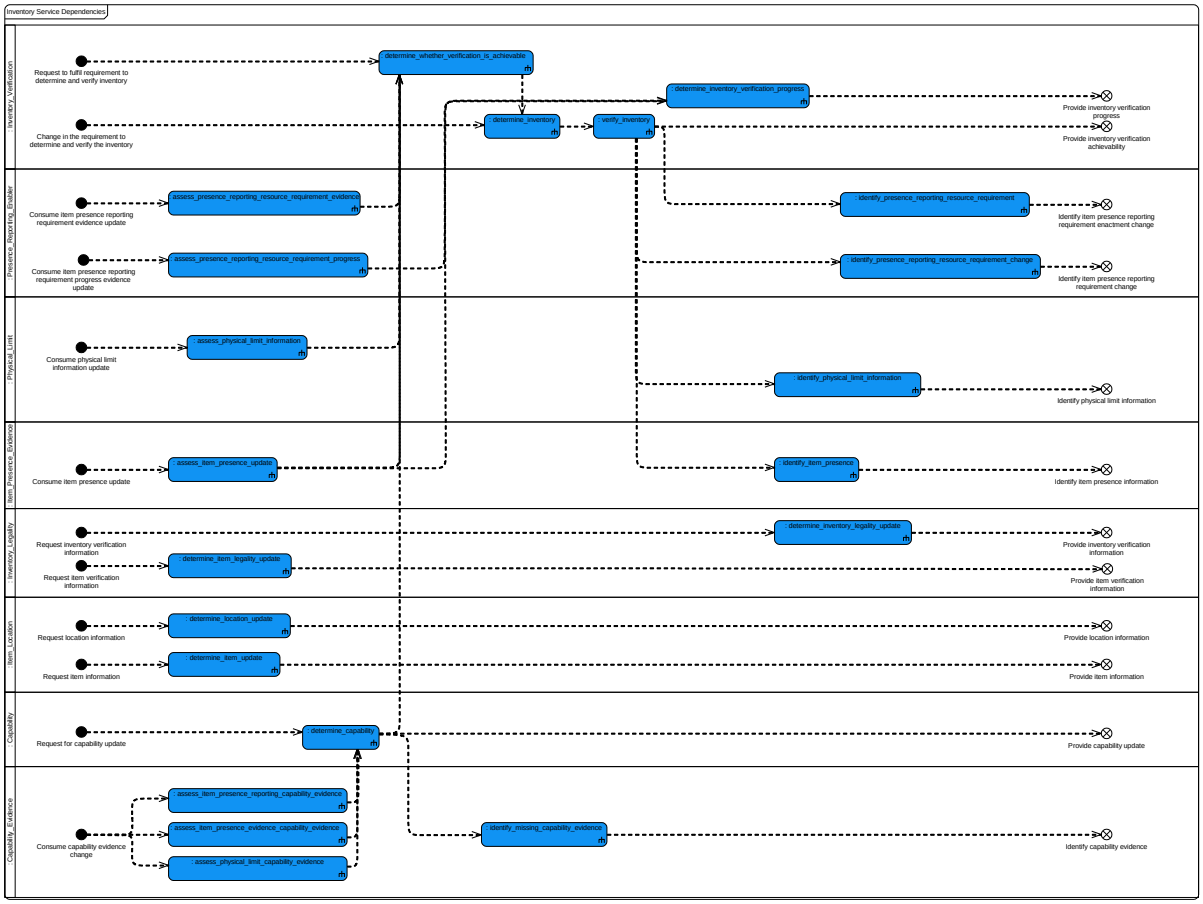

Figure 599: Inventory Service Dependencies

B.2.30 Jettison

B.2.30.1 Role

The role of Jettison is to coordinate the jettison of physical items from the platform.

B.2.30.2 Overview

Control Architecture

Jettison is an action component as defined in the Control Architecture policy.

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.

B.2.30.3 Service Summary

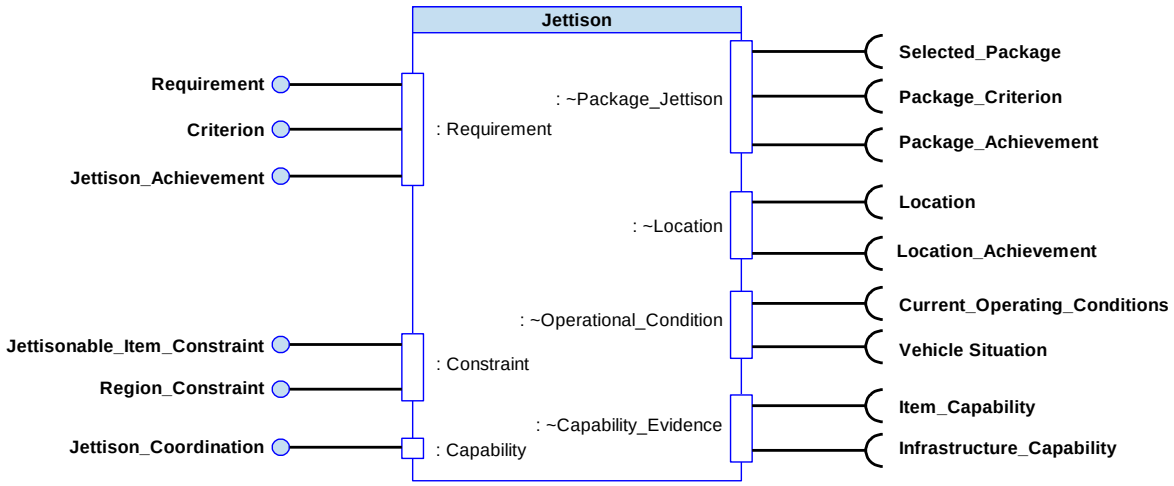

Figure 600: Jettison Service Summary

B.2.30.4 Responsibilities

capture_jettison_requirements

- To capture provided Requirements (e.g. mass to be removed) for jettison activities.

capture_jettison_constraints

- To capture provided Constraints (e.g. stores that are not to be jettisoned) for jettison activities.

capture_jettison_locations

- To capture locations in which jettison is allowable.

capture_measurement_criteria_for_jettison_actions

- To capture provided Measurement_Criterion/criteria for jettison activities.

determine_predicted_quality_of_jettison_deliverables

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

determine_jettison_solution

- To determine a Jettison_Solution that meets the given Requirements and Constraints for jettison using available Jettison_Resources.

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.

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).

predict_capability_progression

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

identify_missing_information

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

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current or predicted Jettison_Capability and Constraints.

determine_actual_quality_of_jettison_deliverables

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

B.2.30.5 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.

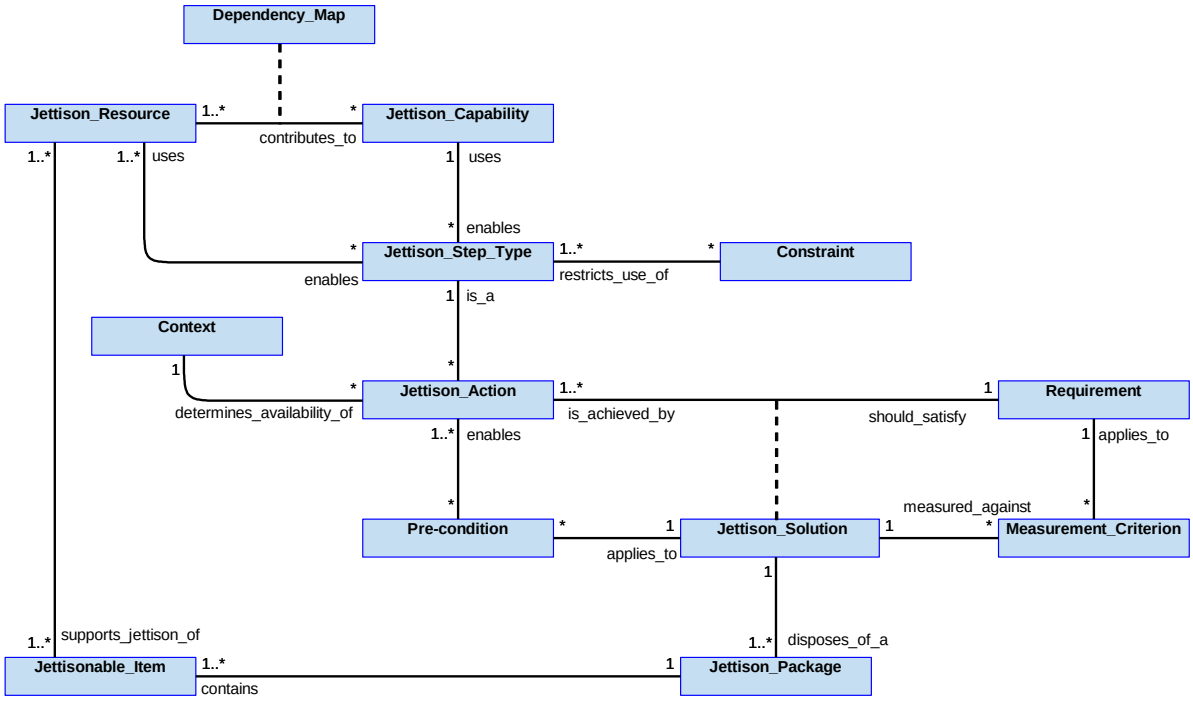

Figure 601: Jettison Semantics

B.2.30.5.1 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 activity 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

A selection of one or more objects consisting of zero or more items and zero or more fluid measures, to be jettisoned together.

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 activity 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.

B.2.30.6 Design Rationale

B.2.30.6.1 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.

B.2.30.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.30.6.3 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.

B.2.30.6.4 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).

B.2.30.7 Services

B.2.30.7.1 Service Definitions

B.2.30.7.1.1 Requirement

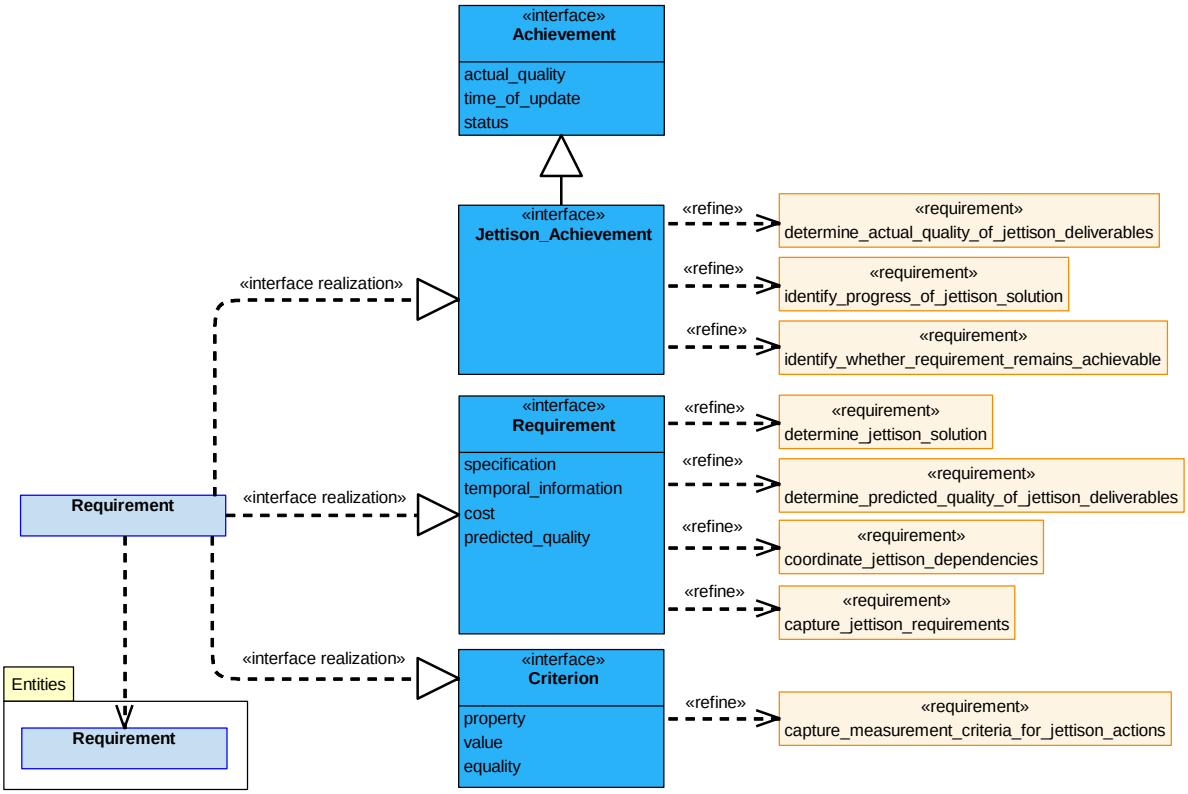

Figure 602: Requirement Service Definition

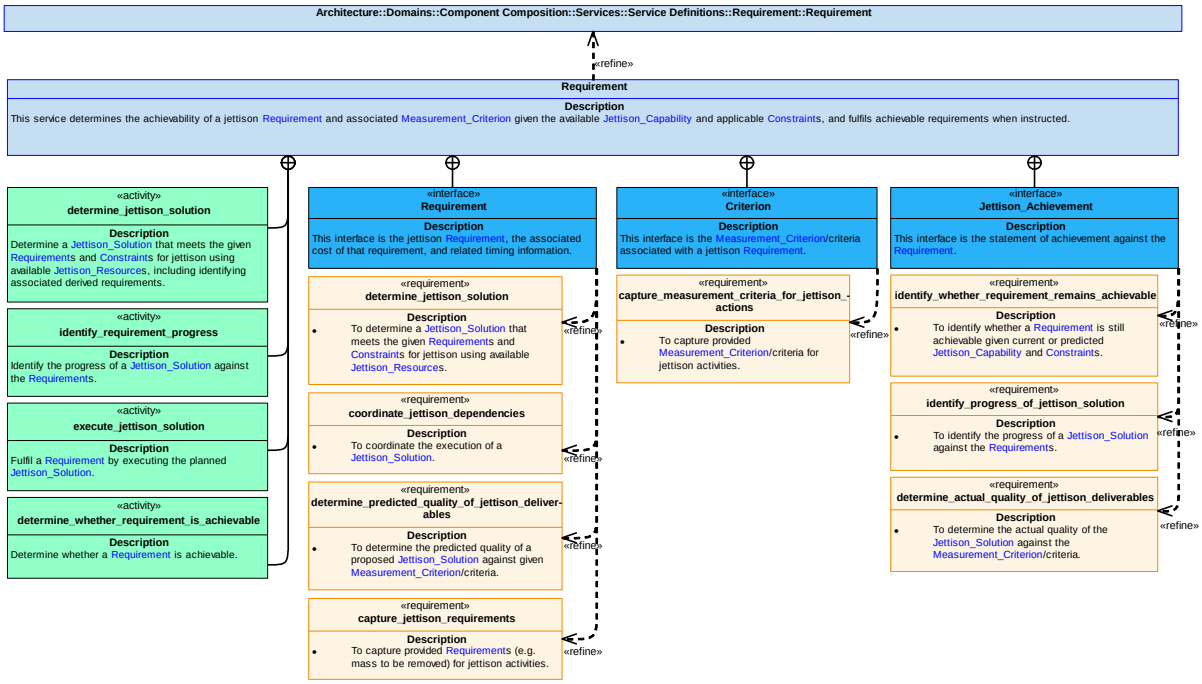

Figure 603: Requirement Service Policy

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.

B.2.30.7.1.2 Package_Jettison

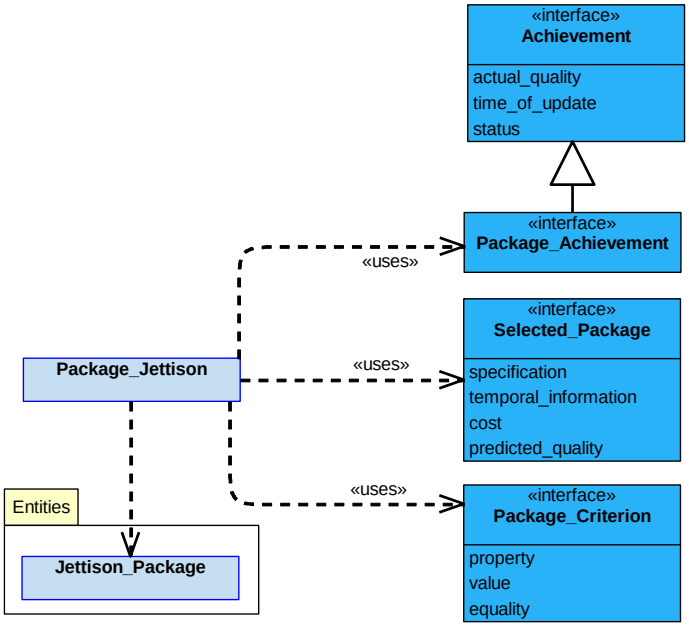

Figure 604: Package_Jettison Service Definition

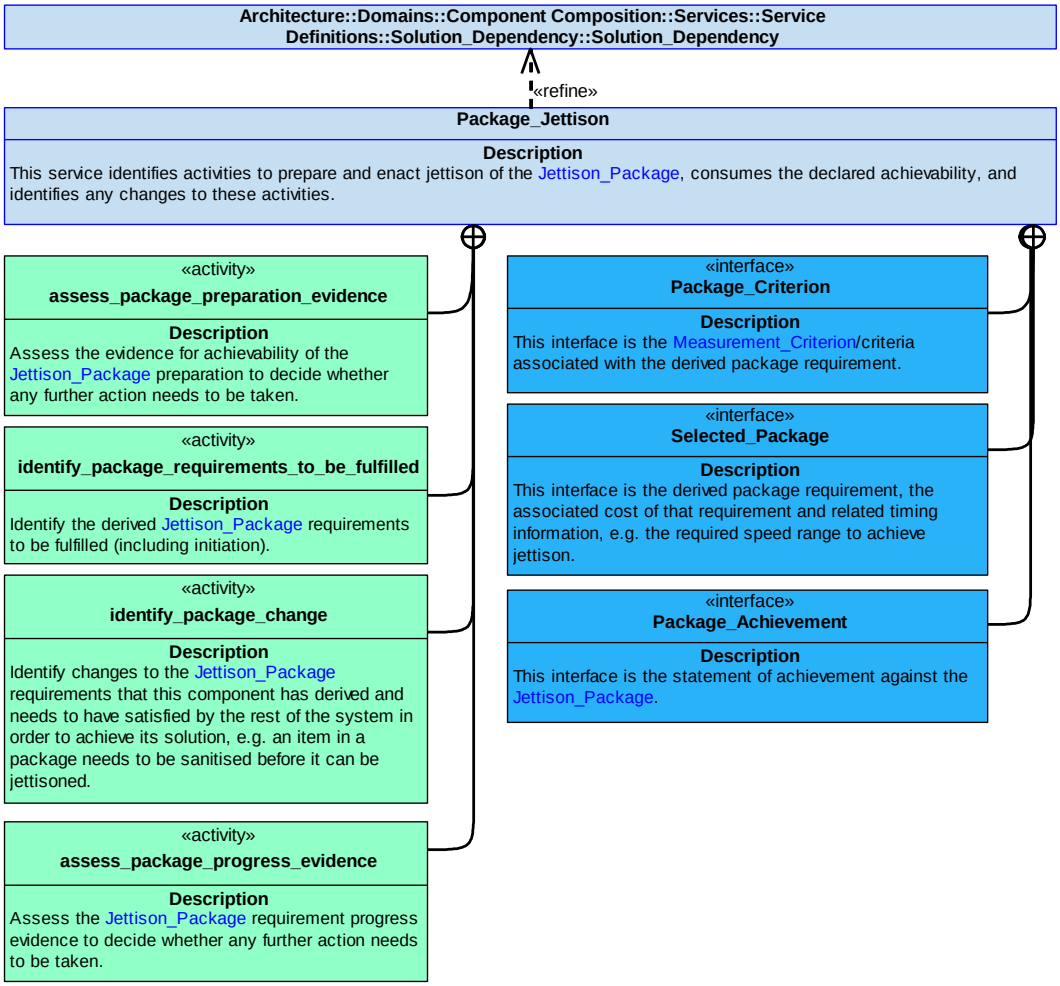

Figure 605: Package_Jettison Service Policy

Package_Jettison

This service identifies activities to prepare and enact jettison of the Jettison_Package, consumes the declared achievability, and identifies any changes to these activities.

Interfaces

Selected_Package

This interface is the derived package requirement, the associated cost of that requirement and related timing information, e.g. the required speed range to achieve jettison.

Attributes

specification	The definition of the derived package 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 solution, e.g. resources used or time taken.
predicted_quality	How well the planned solution is predicted to satisfy the derived requirement.

Package_Criterion

This interface is the Measurement_Criterion/criteria associated with the derived package requirement.

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.

Package_Achievement

This interface is the statement of achievement against the Jettison_Package.

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.

B.2.30.7.1.3 Location

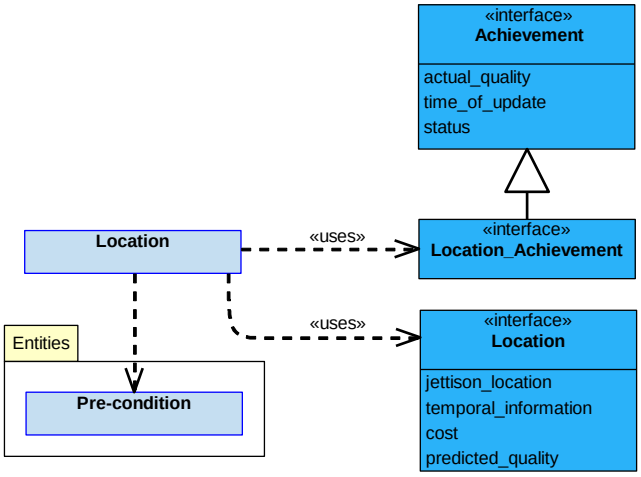

Figure 606: Location Service Definition

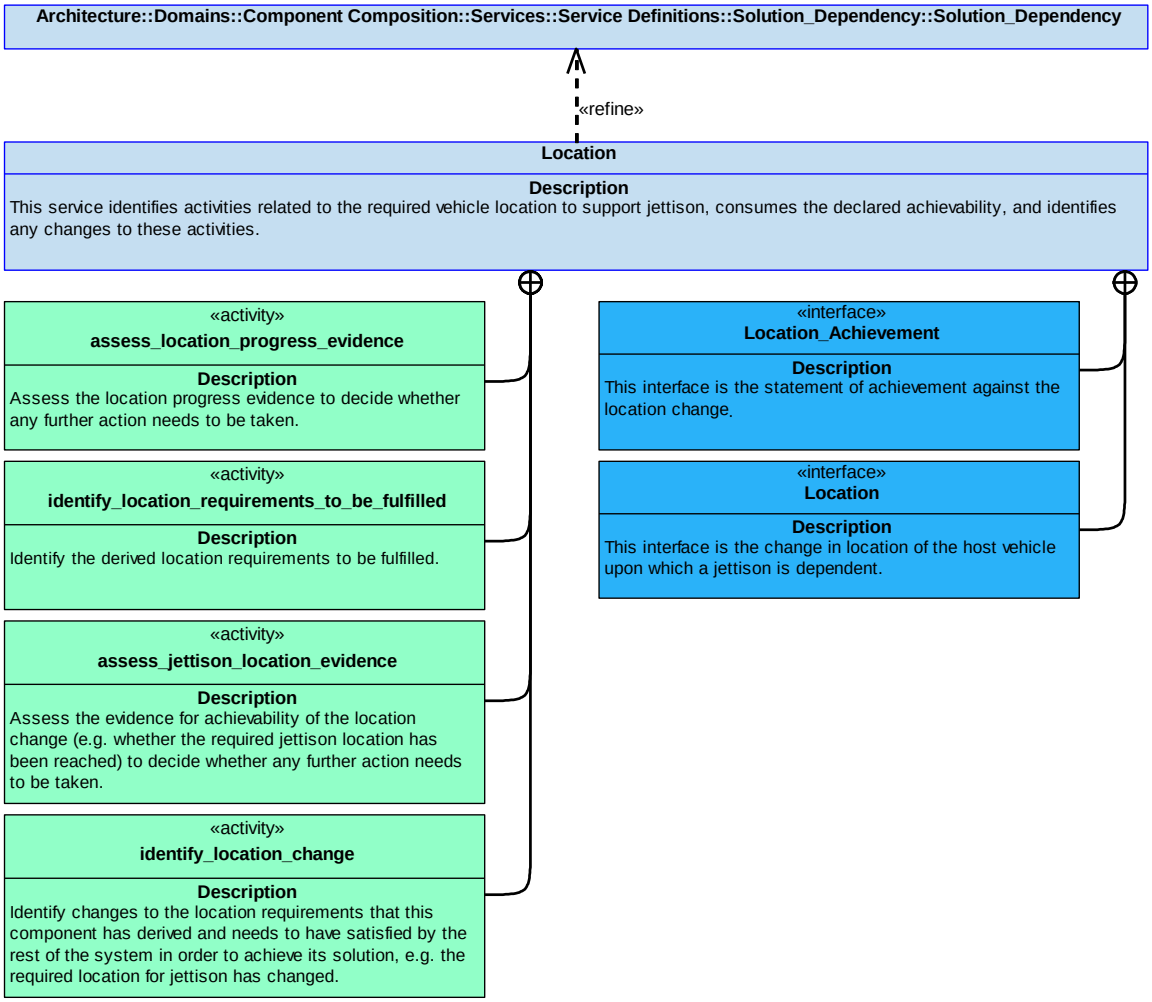

Figure 607: Location Service Policy

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.

B.2.30.7.1.4 Operational_Condition

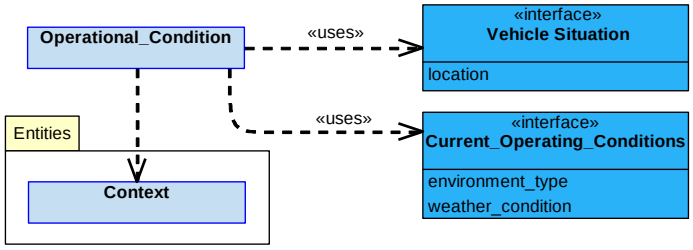

Figure 608: Operational_Condition Service Definition

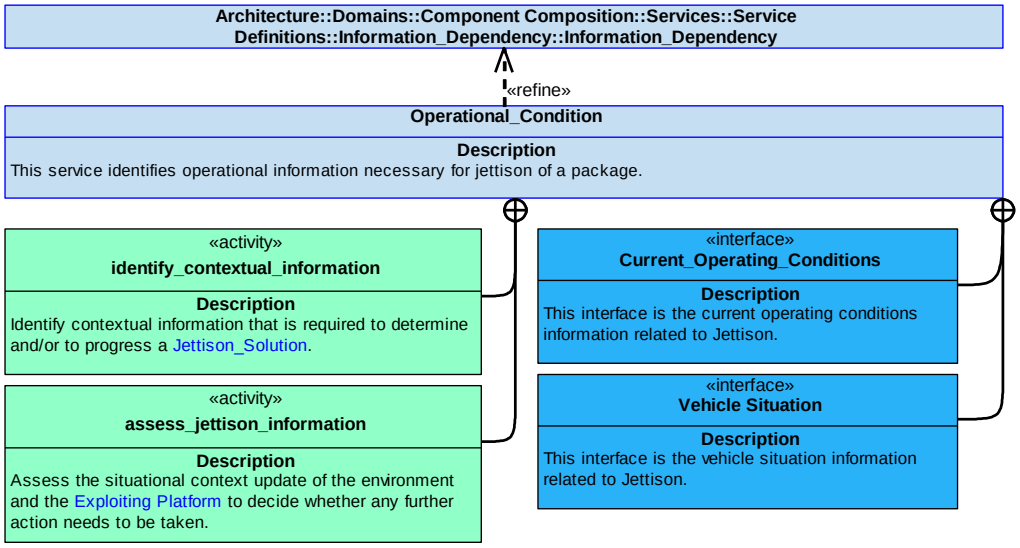

Figure 609: Operational_Condition Service Policy

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.

B.2.30.7.1.5 Constraint

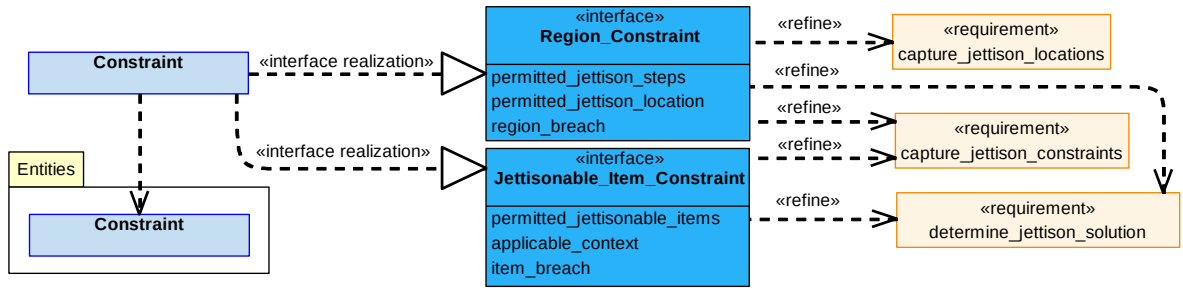

Figure 610: Constraint Service Definition

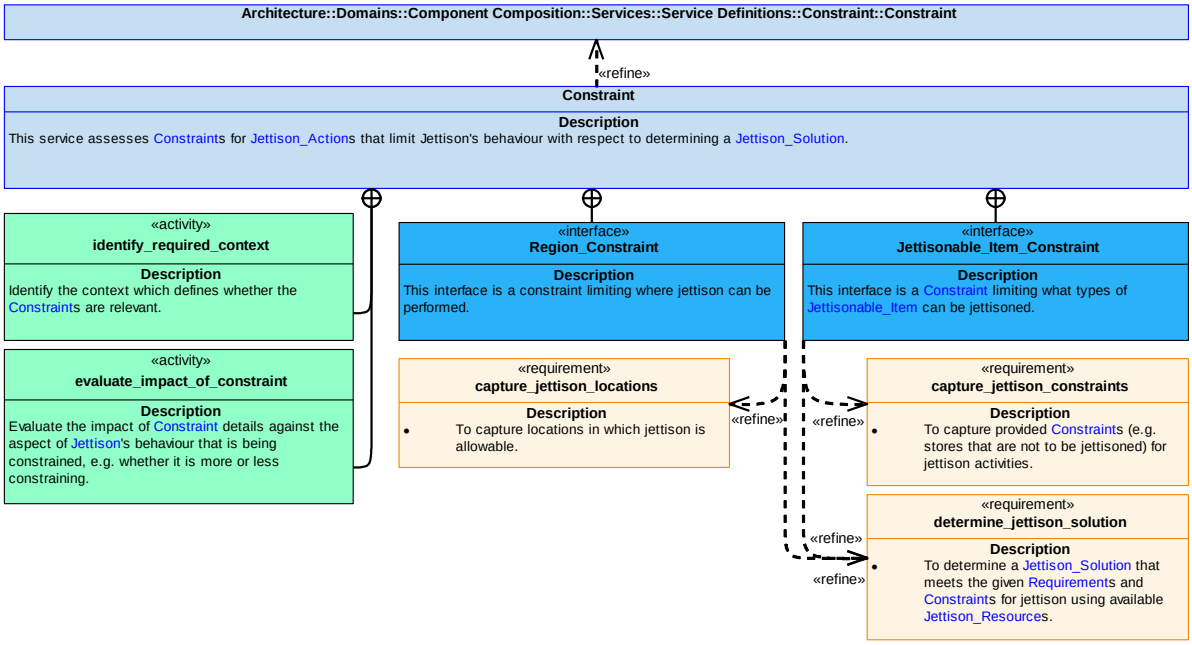

Figure 611: Constraint Service Policy

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_steps	The Jettison_Step_Types that are permitted.
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.

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.

B.2.30.7.1.6 Capability

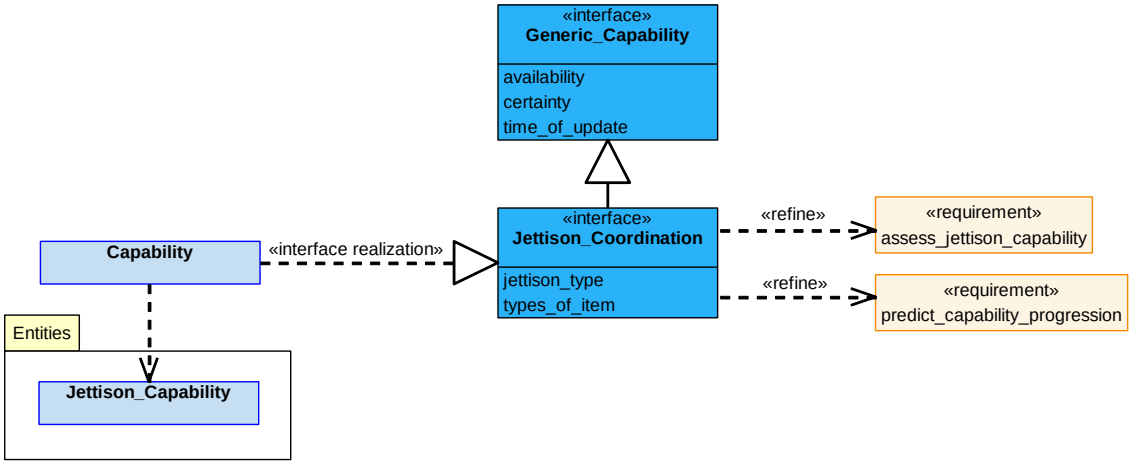

Figure 612: Capability Service Definition

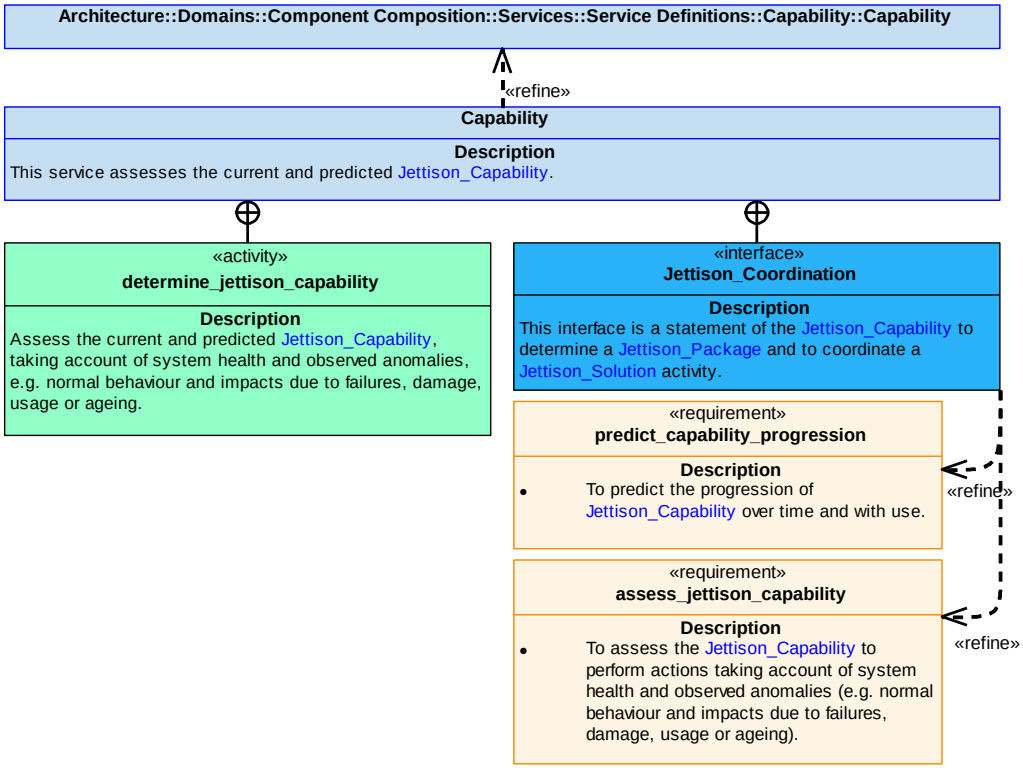

Figure 613: Capability Service Policy

Capability

This service assesses the current and predicted Jettison_Capability.

Interface

Jettison_Coordination

This interface is a statement of the Jettison_Capability to determine a Jettison_Package and to coordinate a Jettison_Solution activity.

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.

B.2.30.7.1.7 Capability_Evidence

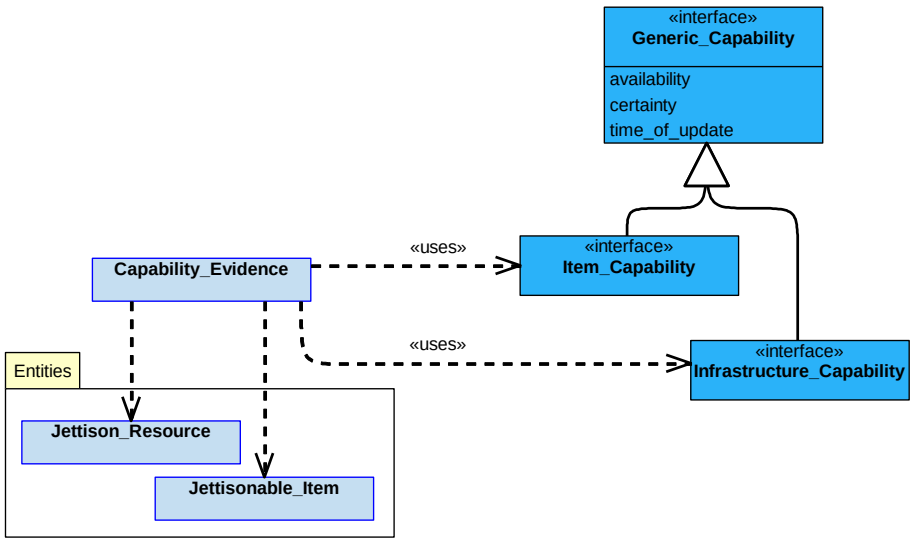

Figure 614: Capability_Evidence Service Definition

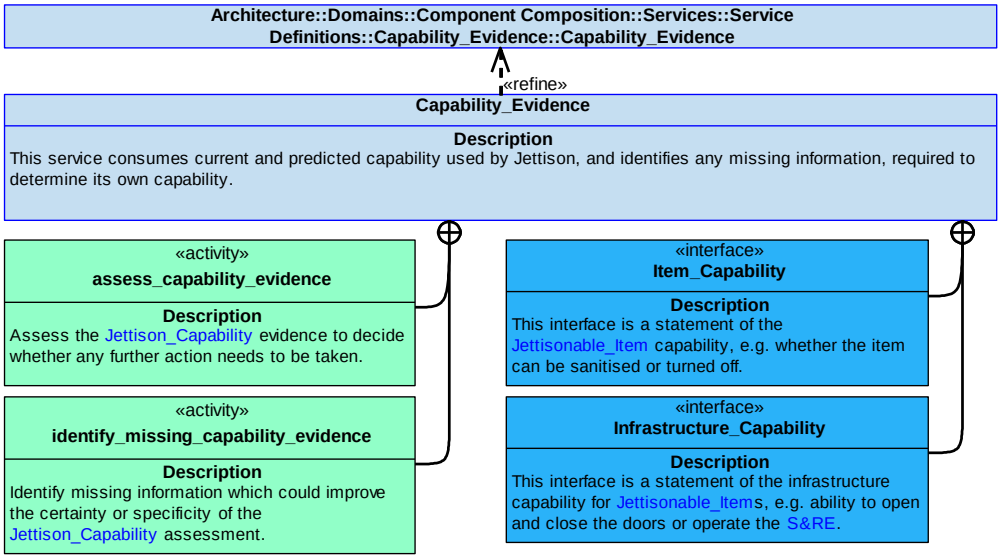

Figure 615: Capability_Evidence Service Policy

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.

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.

B.2.30.7.2 Service Dependencies

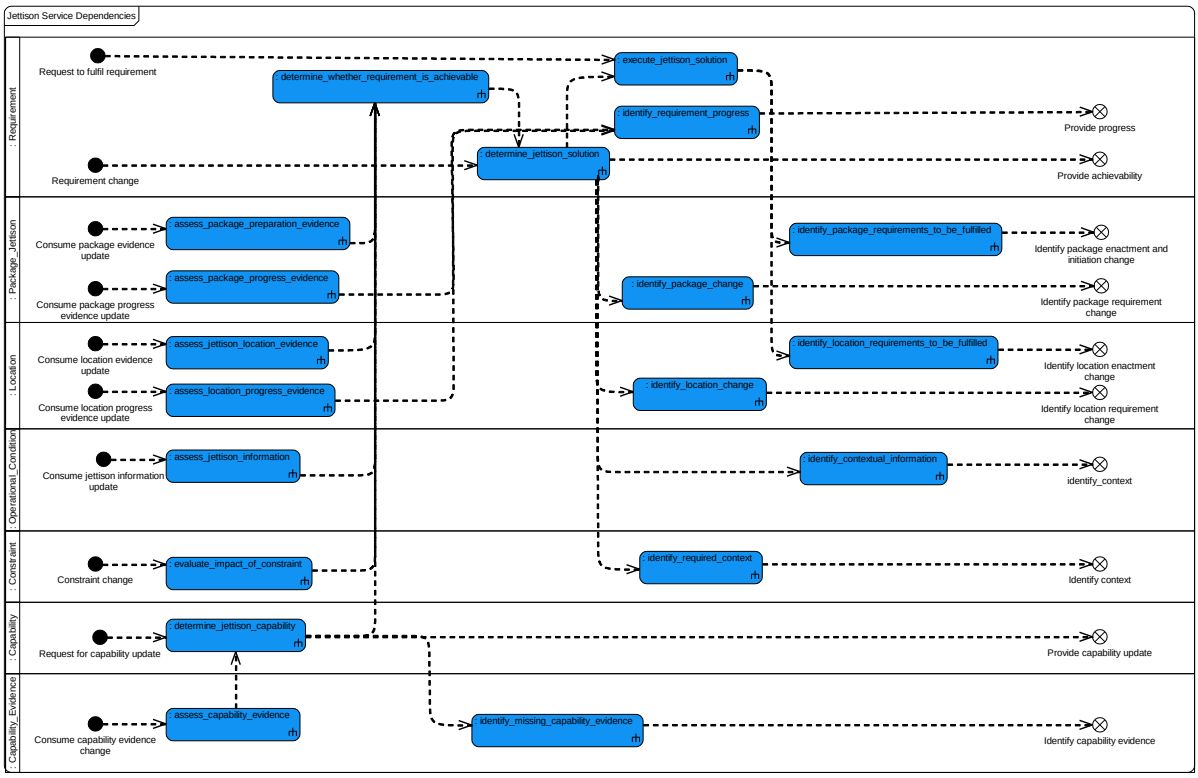

Figure 616: Jettison Service Dependencies

B.2.31 Lights

B.2.31.1 Role

The role of Lights is to control the lighting on an Exploiting Platform.

B.2.31.2 Overview

Control Architecture

Lights is an action component as defined in the Control Architecture policy.

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.

B.2.31.3 Service Summary

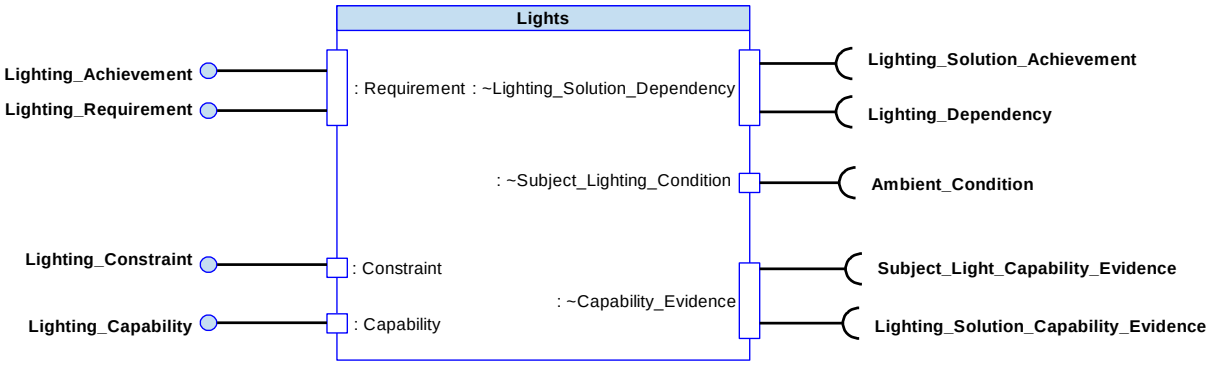

Figure 617: Lights Service Summary

B.2.31.4 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.

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_lighting_solution_in_progress_remains_feasible

- To identify whether a Lighting_Solution in progress remains feasible given current resources.

predict_capability_progression

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

identify_missing_information

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

B.2.31.5 Subject Matter Semantics

The subject matter of Lights is the resources that can be used to provide illumination.

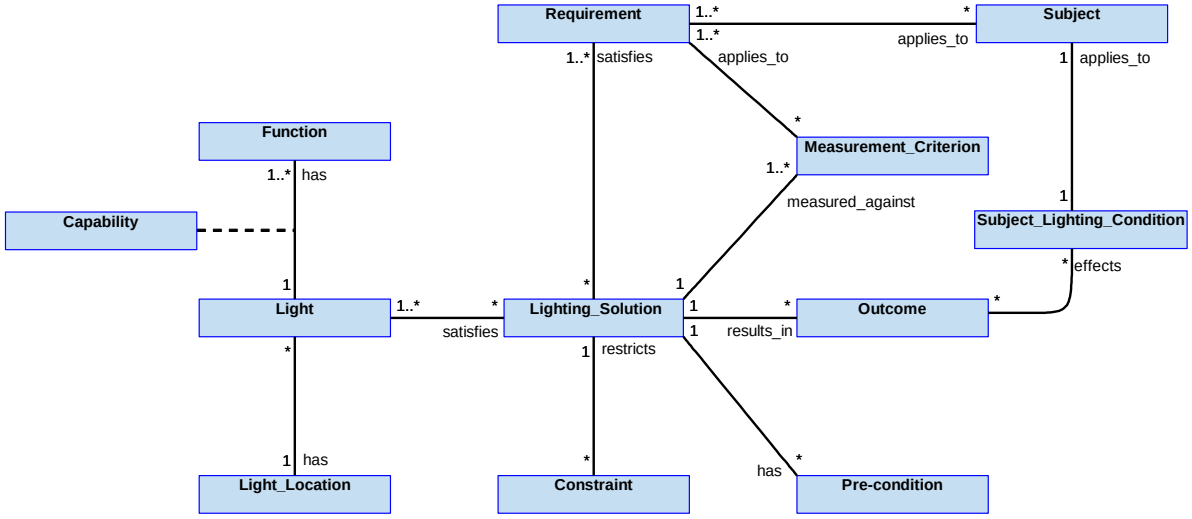

Figure 618: Lights Semantics

B.2.31.5.1 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.

B.2.31.6 Design Rationale

B.2.31.6.1 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.

B.2.31.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Lights:

- Data Driving - To facilitate different pre-defined configurations between different operators.

B.2.31.6.3 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.

B.2.31.6.4 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.

B.2.31.7 Services

B.2.31.7.1 Service Definitions

B.2.31.7.1.1 Requirement

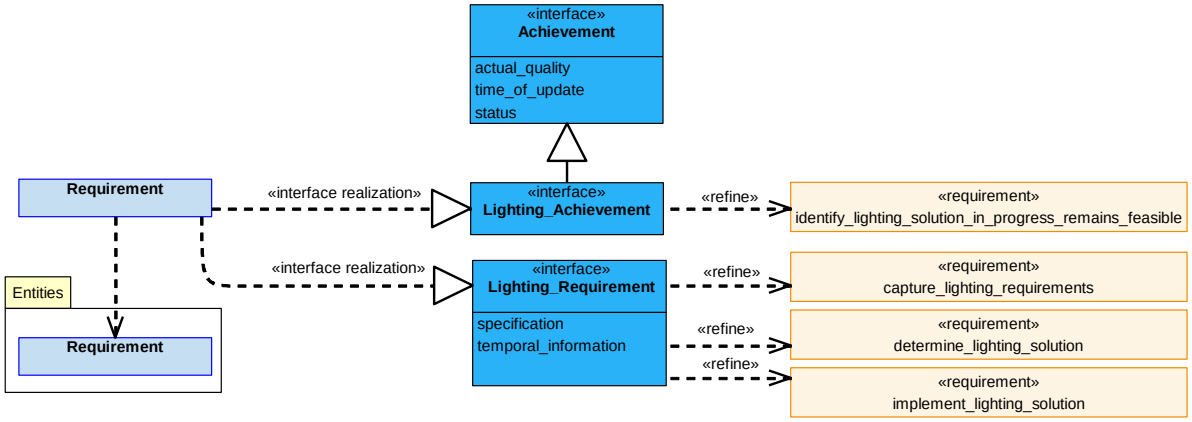

Figure 619: Requirement Service Definition

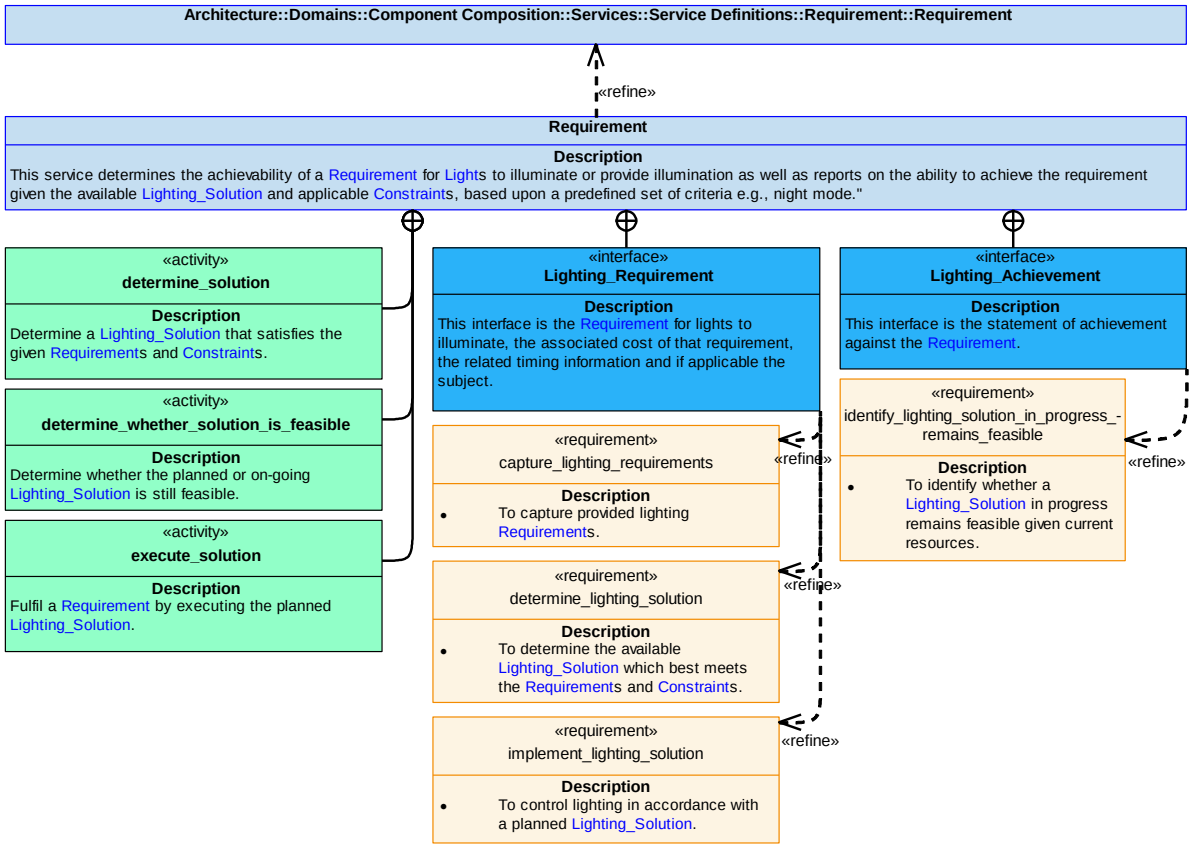

Figure 620: Requirement Service Policy

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.

B.2.31.7.1.2 Lighting_Solution_Dependency

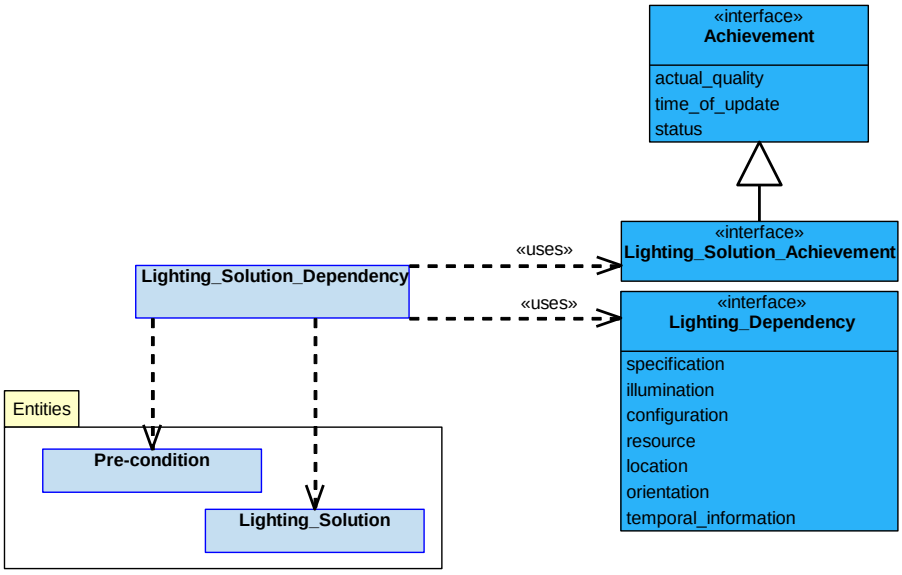

Figure 621: Lighting_Solution_Dependency Service Definition

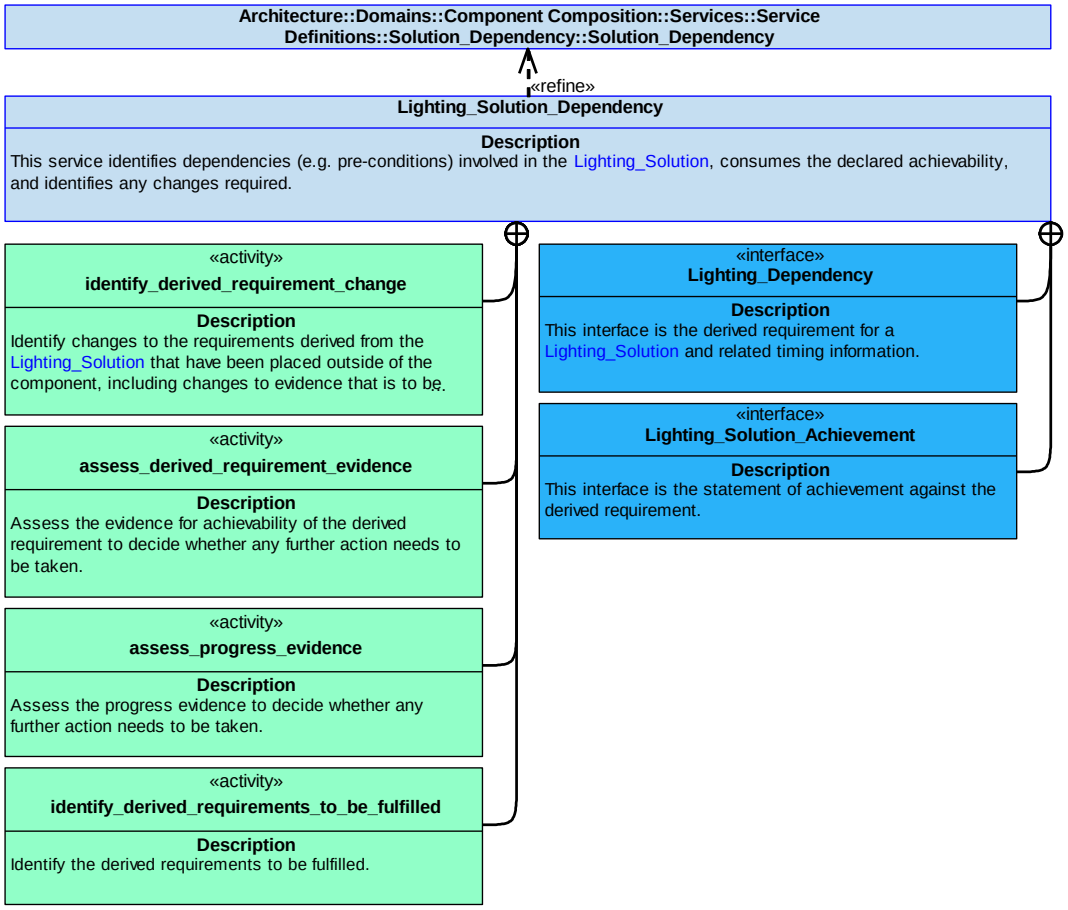

Figure 622: Lighting_Solution_Dependency Service Policy

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.

B.2.31.7.1.3 Subject_Lighting_Condition

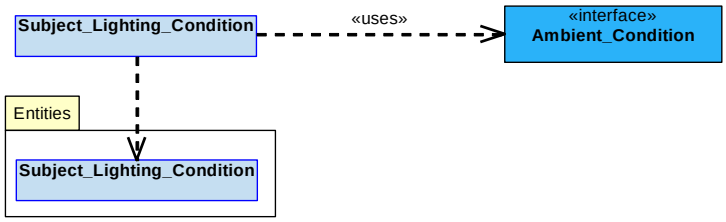

Figure 623: Subject_Lighting_Condition Service Diagram

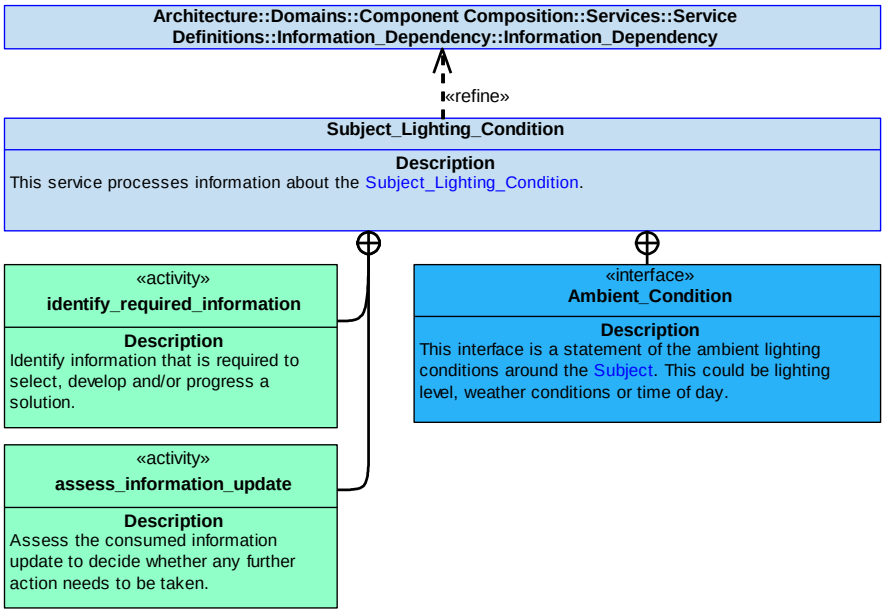

Figure 624: Subject_Lighting_Condition Service Policy

Subject_Lighting_Condition

This service processes information about the Subject_Lighting_Condition.

Interface

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.

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 solution.

B.2.31.7.1.4 Constraint

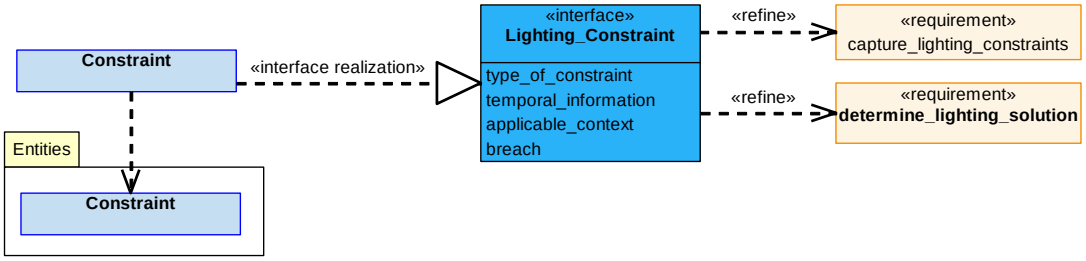

Figure 625: Constraint Service Definition

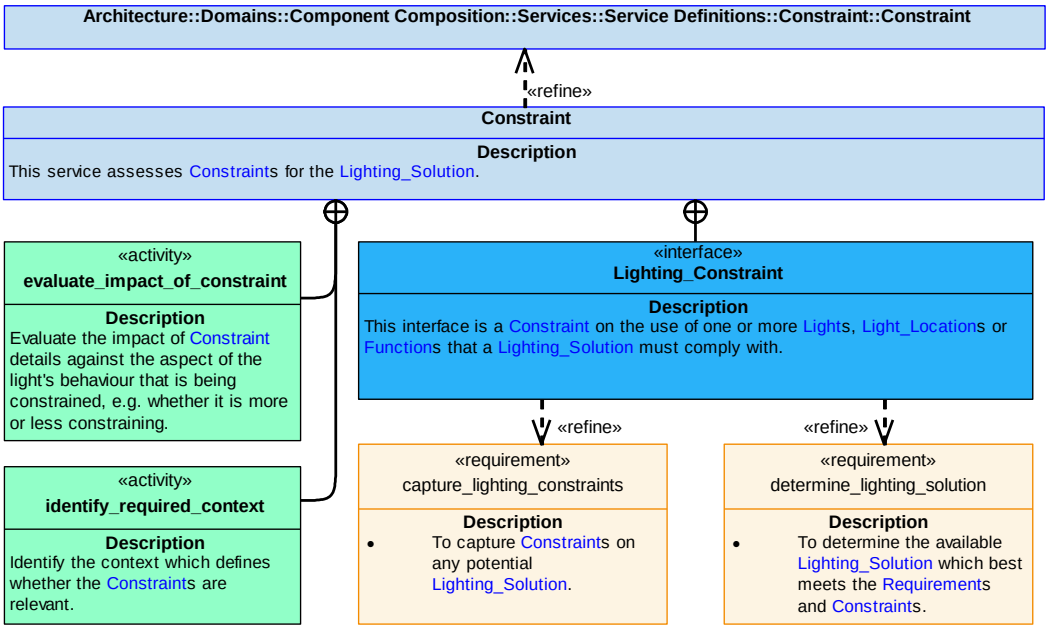

Figure 626: Constraint Service Policy

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.

B.2.31.7.1.5 Capability

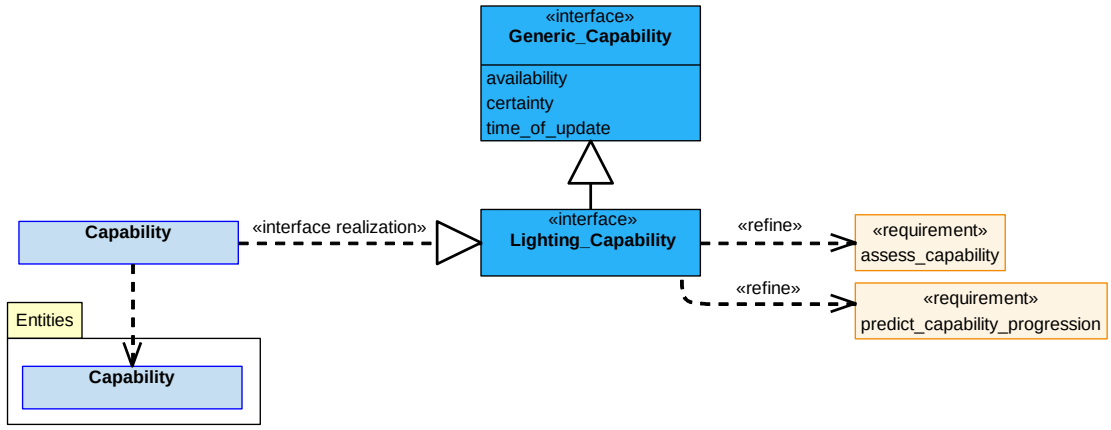

Figure 627: Capability Service Definition

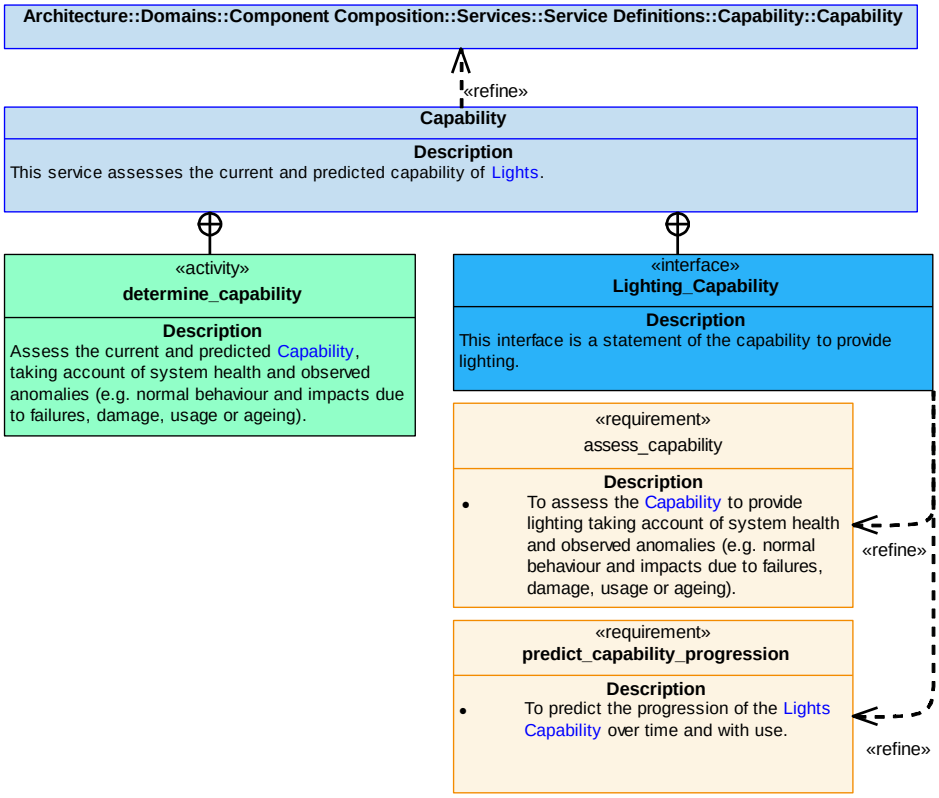

Figure 628: Capability Service Policy

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).

B.2.31.7.1.6 Capability_Evidence

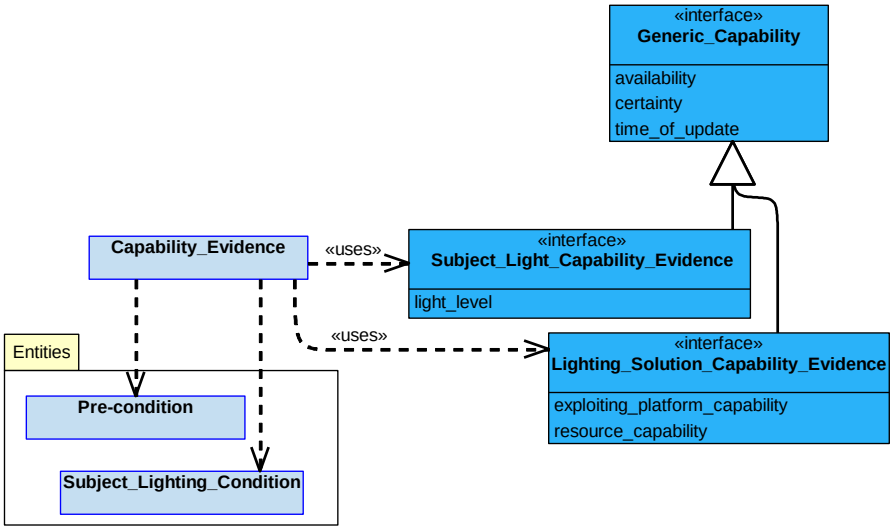

Figure 629: Capability Evidence Service Definition

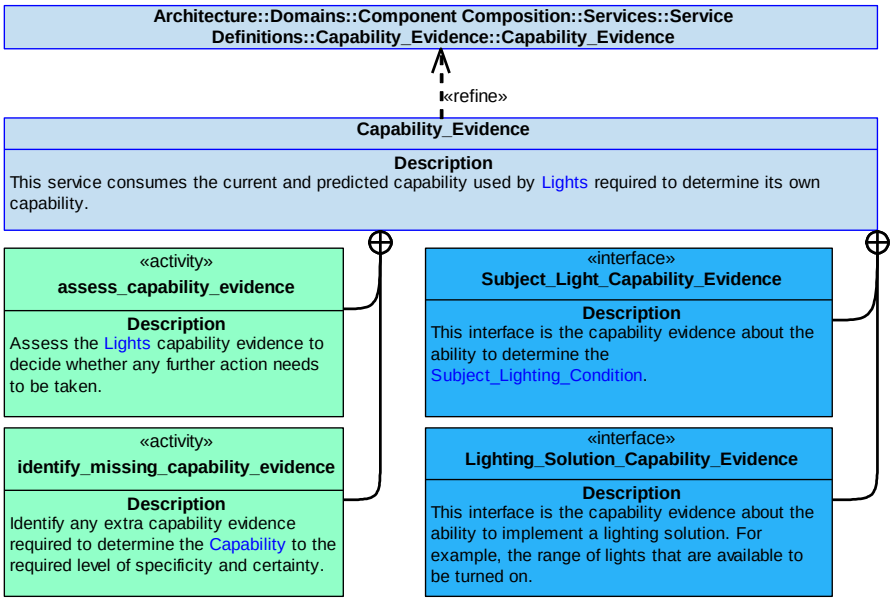

Figure 630: Capability Evidence Service Policy

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.

B.2.31.7.2 Service Dependencies

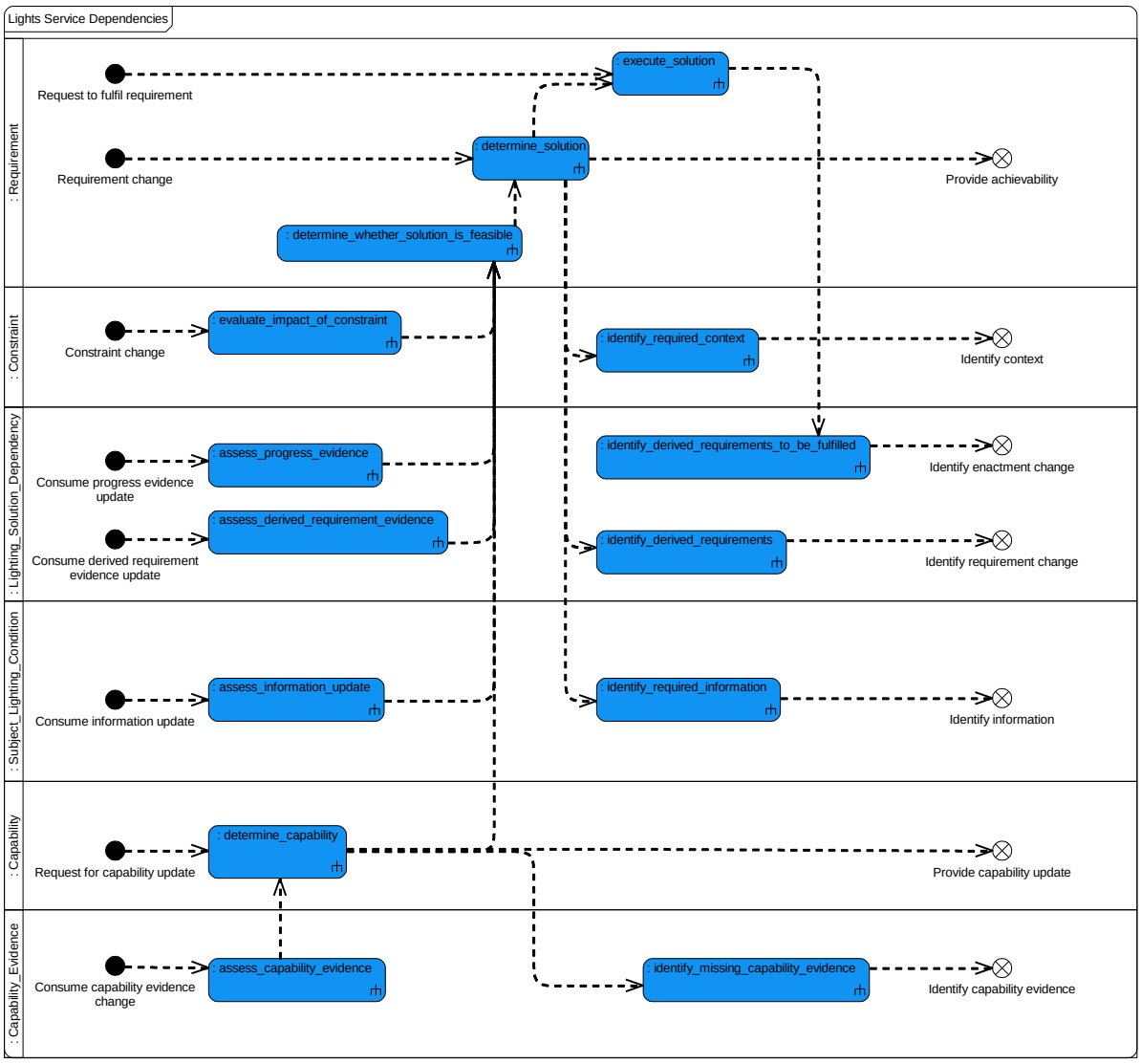

Figure 631: Lights Service Dependencies

B.2.32 Location and Orientation

B.2.32.1 Role

The role of Location and Orientation is to determine the location and spatial orientation including any derivatives of a platform.

B.2.32.2 Overview

Control Architecture

Location and Orientation is a service component as defined in the Control Architecture policy.

Standard Pattern of Use

A requirement is placed on Location and Orientation to provide an estimate of the Location and/or Orientation of a Platform. Source_Parameters from one or more Sources are used to determine the Location and/or Orientation, defined against a particular Reference_Frame.

Examples of Use

Location and Orientation will be used where:

- The determination of Location and/or Orientation is required in order to safely and efficiently control some aspect of the Platform's operation.

B.2.32.3 Service Summary

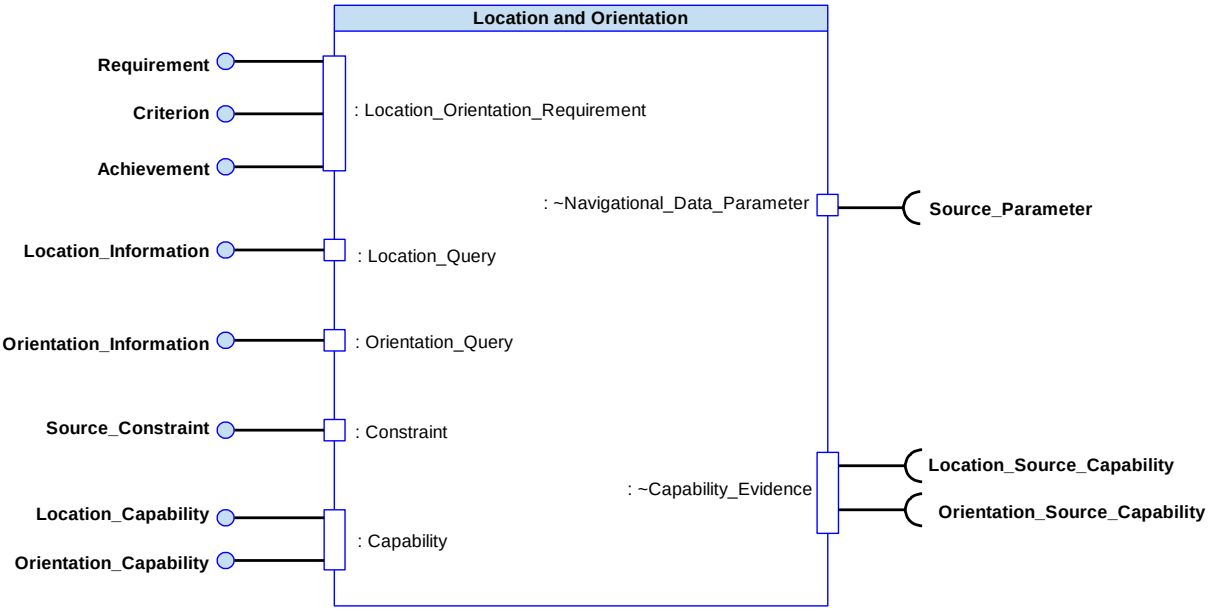

Figure 632: Location and Orientation Service Summary

B.2.32.4 Responsibilities

capture_reference_frame

- To capture given Reference_Frames. This includes absolute and relative Reference_Frames.

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_location_and_orientation_quality

- To determine the accuracy and precision of a Location parameter or Orientation parameter.

determine_derivatives

- To determine the derivatives of Location and/or Orientation relative to the Reference_Frame.

assess_location_and_orientation_capability

- To identify the system's Capability to determine the Location and/or Orientation of a Platform, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

capture_constraints

- To capture Constraints on the use of Sources.

capture_source_parameters

- To capture given Source_Parameters from available Sources.

capture_location_and_orientation_requirements

- To capture the Requirements for determining Location and Orientation.

predict_capability_progression

- To predict the progression of the Location and Orientation Capability over time and with use.

identify_missing_information

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

determine_if_requirement_is_achievable

- To determine if a Requirement is achievable given current Capability and Constraints.

B.2.32.5 Subject Matter Semantics

The subject matter of Location and Orientation is the Location and Orientation of a Platform.

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.

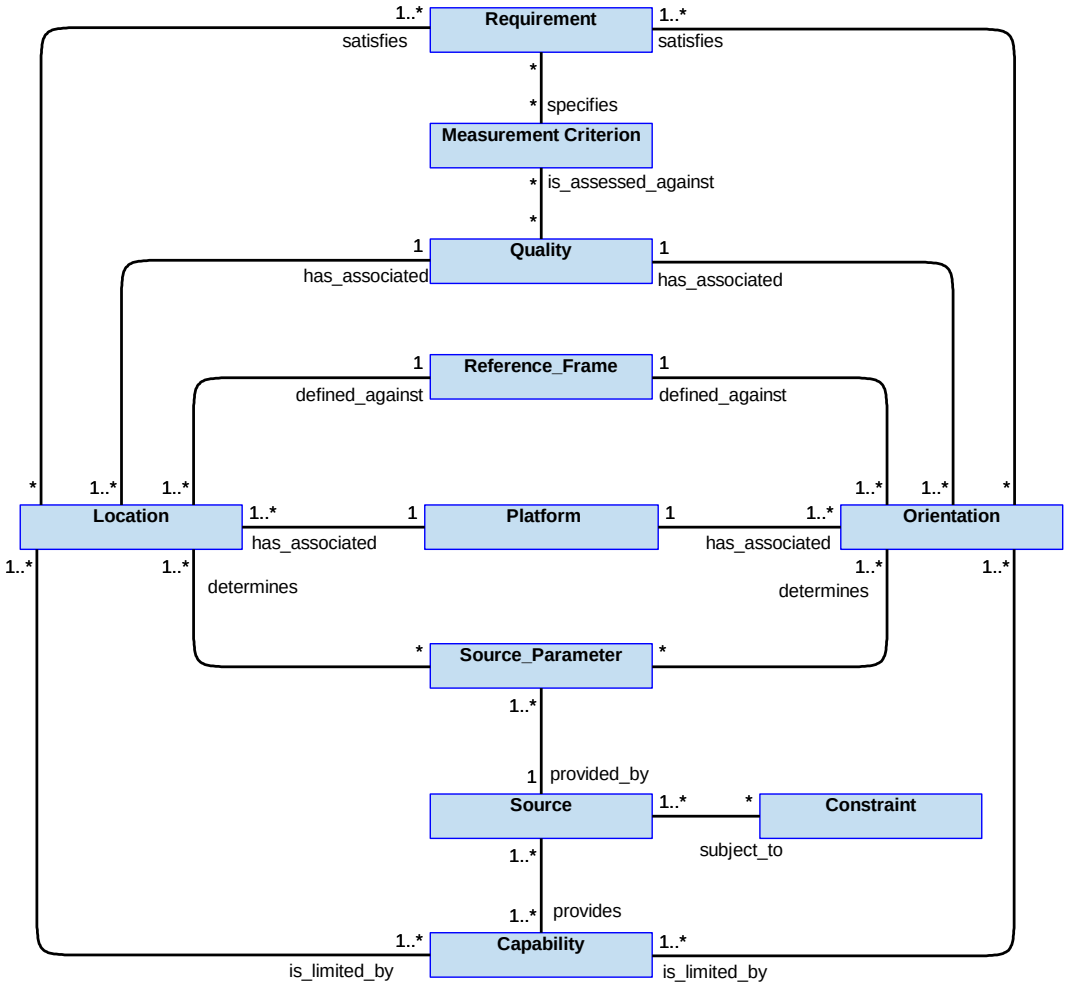

Figure 633: Location and Orientation Semantics

B.2.32.5.1 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 or Orientation, 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, including derivatives, of a Platform defined against a Reference_Frame. This could be expressed as latitude, longitude, altitude, etc. Any of these values could have derivatives such as rates, accelerations, etc.

Orientation

An estimate of the orientation, including derivatives, of a Platform defined against a Reference_Frame. This could include attitude (pitch, roll and yaw). Any of these values could have derivatives such as rates, accelerations, etc.

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 Location solution or an Orientation solution.

B.2.32.6 Design Rationale

B.2.32.6.1 Assumptions

None.

B.2.32.6.2 Design Considerations

Directly Applicable Policies

These policies 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 are designed to be data driven at build (i.e. development) time. This allows the component to be configurable and flexible to cater for any set of required information and allows reusability between multiple Exploiting Platforms. There is no consideration of the introduction of additional output parameters being required during operation.

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.

B.2.32.6.3 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.

B.2.32.6.4 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.

B.2.32.7 Services

B.2.32.7.1 Service Definitions

B.2.32.7.1.1 Location_Orientation_Requirement

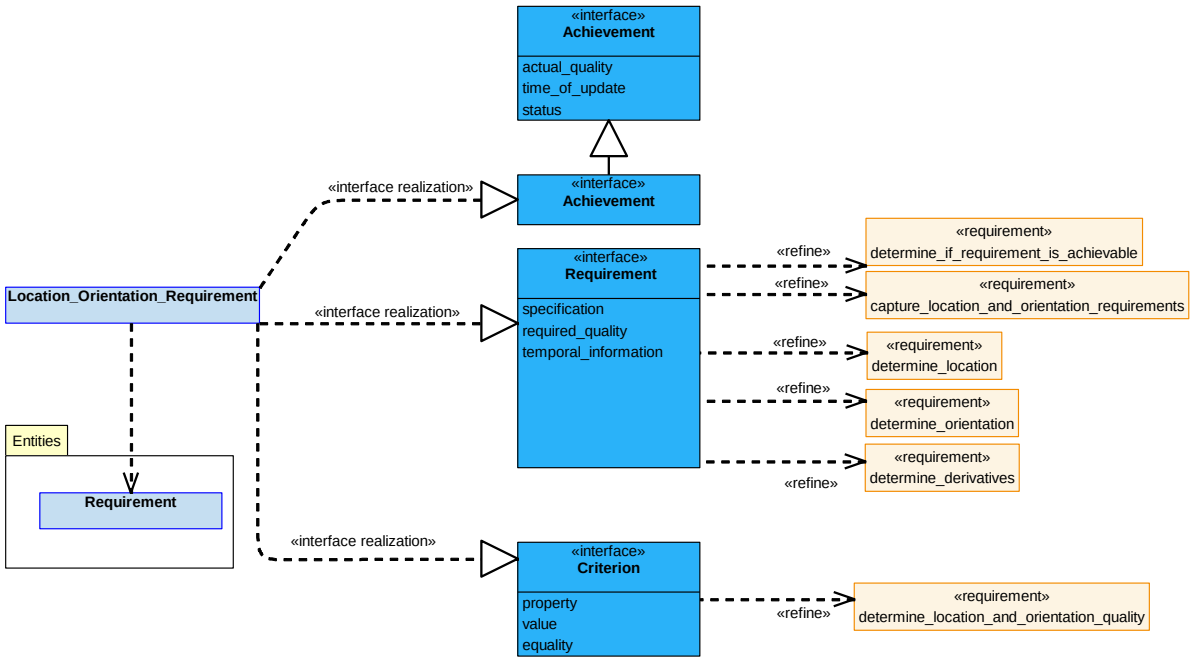

Figure 634: Location_Orientation_Requirement Service Definition

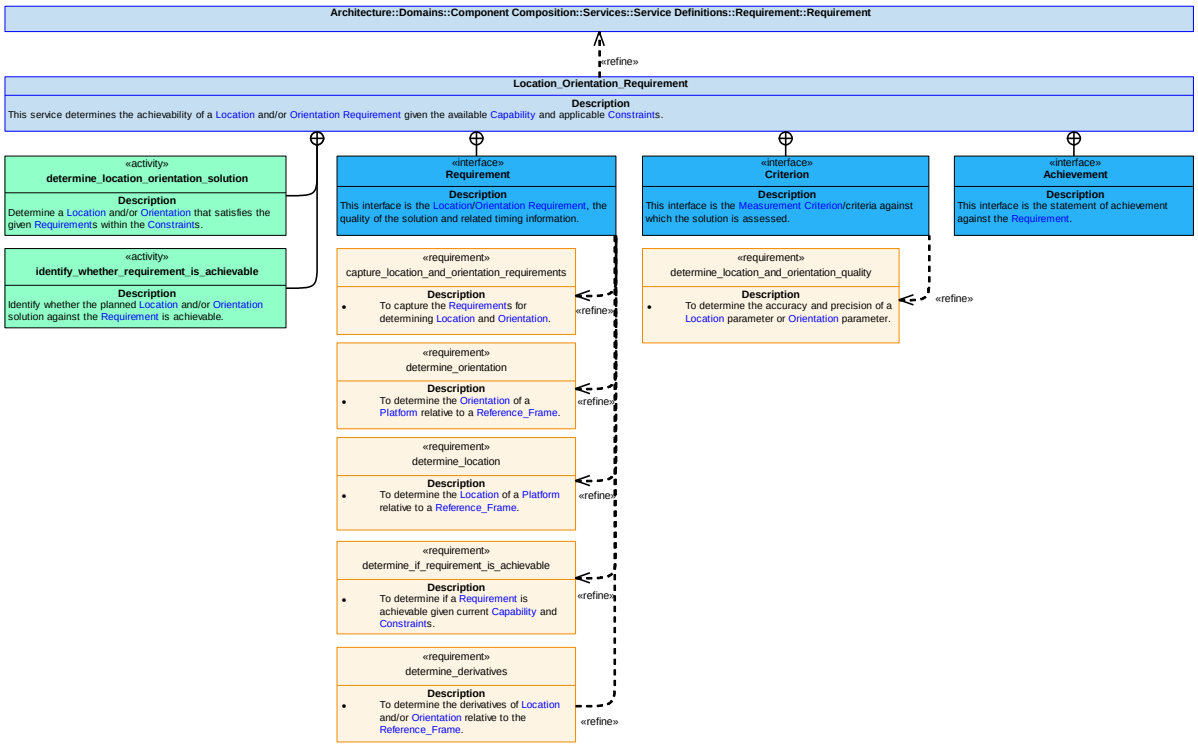

Figure 635: Location_Orientation_Requirement Service Policy

Location_Orientation_Requirement

This service determines the achievability of a Location and/or Orientation Requirement given the available Capability and applicable Constraints.

Interfaces

Requirement

This interface is the Location/Orientation Requirement, the 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_location_orientation_solution

Determine a Location and/or Orientation that satisfies the given Requirements within the Constraints.

identify_whether_requirement_is_achievable

Identify whether the planned Location and/or Orientation solution against the Requirement is achievable.

B.2.32.7.1.2 Location_Query

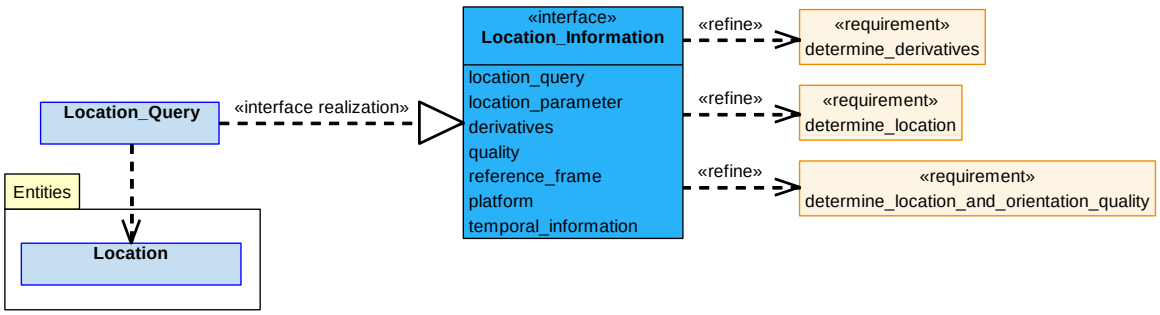

Figure 636: Location_Query Service Definition

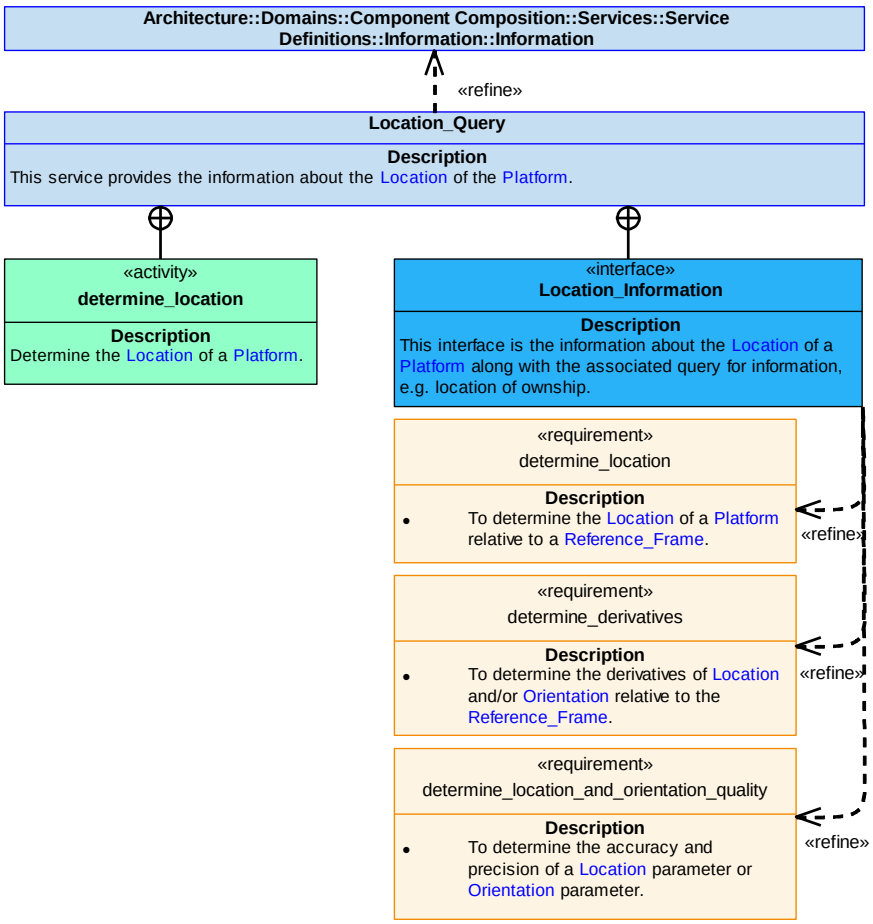

Figure 637: Location_Query Service Policy

Location_Query

This service provides the information about the Location of the Platform.

Interface

Location_Information

This interface is the information about the Location of a Platform along with the associated query for information, e.g. location of ownship.

Attributes

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, altitude expressed as geodetic coordinates.
derivatives	The derivatives associated with the Location, e.g. velocity and acceleration.
quality	The Quality of the Location information.
reference_frame	The abstract coordinate system that defines a set of physical reference points.
platform	The Platform for which the Location information is being provided, e.g. ownship.
temporal_information	Timing information, such as when the Location information was provided.

Activity

determine_location

Determine the Location of a Platform.

B.2.32.7.1.3 Orientation_Query

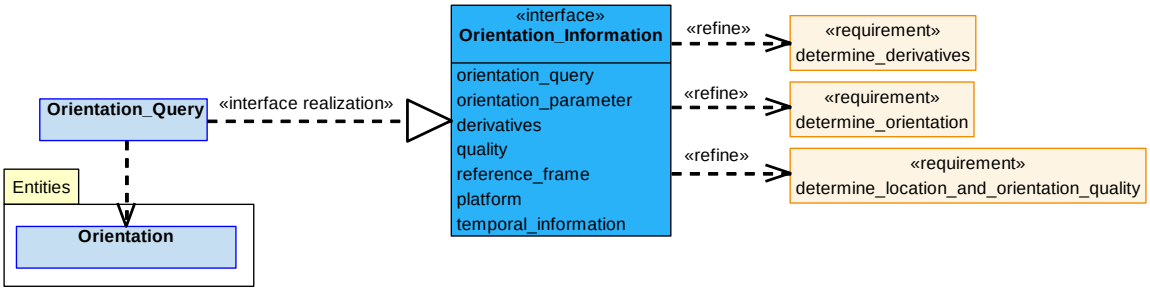

Figure 638: Orientation_Query Service Definition

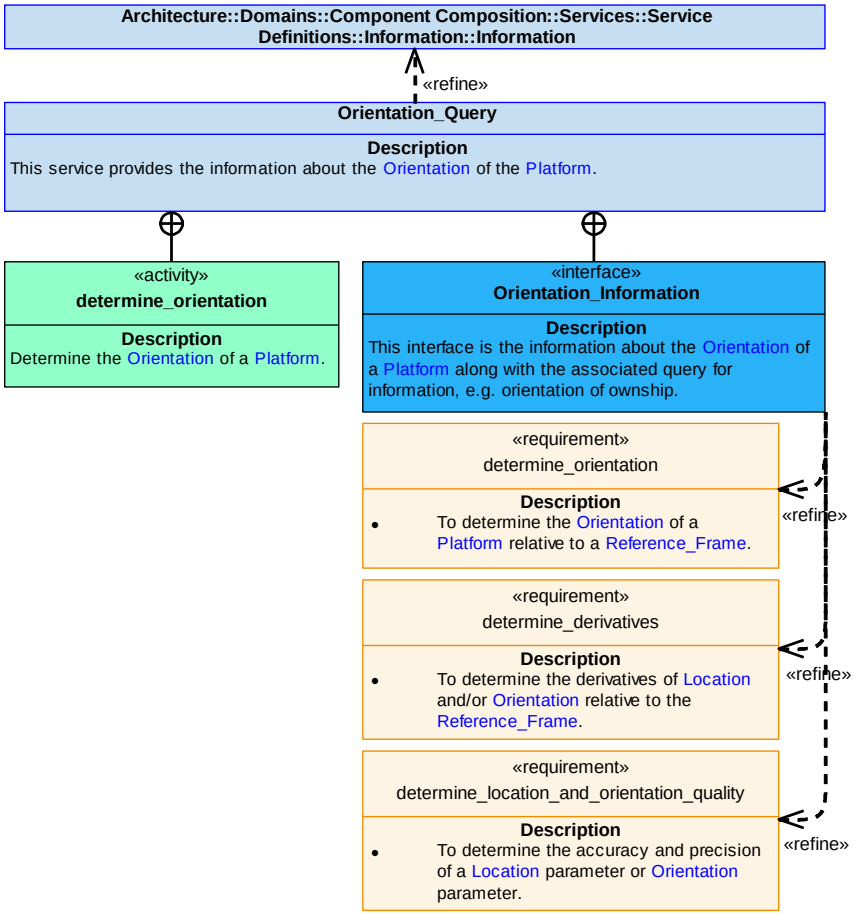

Figure 639: Orientation_Query Service Policy

Orientation_Query

This service provides the information about the Orientation of the Platform.

Interface

Orientation_Information

This interface is the information about the 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.
derivatives	The derivatives associated with the Orientation, e.g. angular velocity and angular acceleration.
quality	The Quality of the Orientation information.
reference_frame	The abstract coordinate system that defines a set of physical reference points.
platform	The Platform for which the Orientation information is being provided, e.g. ownship.
temporal_information	Timing information, such as when the Orientation information was provided.

Activity

determine_orientation

Determine the Orientation of a Platform.

B.2.32.7.1.4 Navigational_Data_Parameter

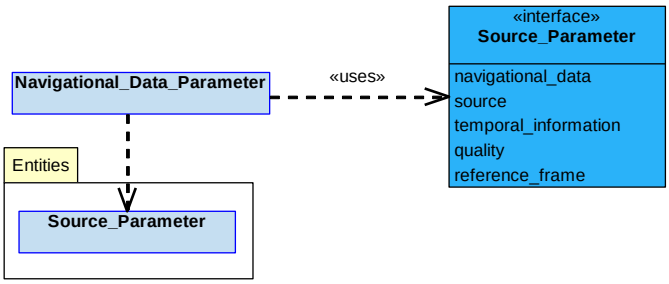

Figure 640: Navigational_Data_Parameter Service Definition

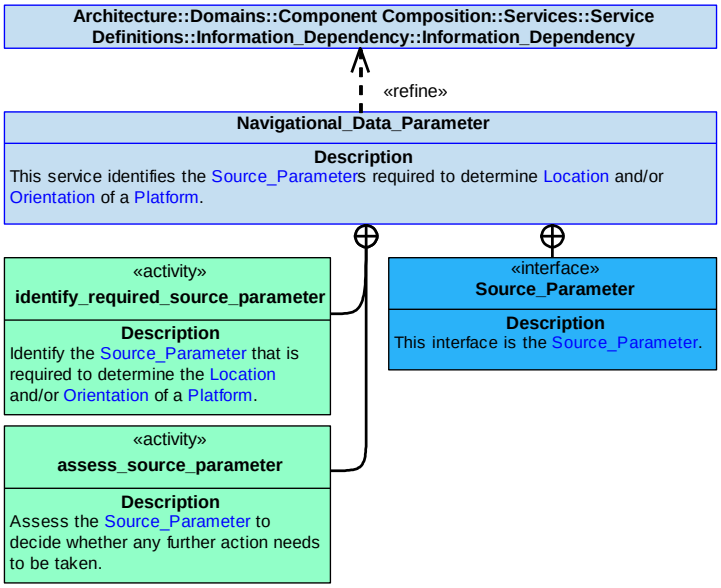

Figure 641: Navigational_Data_Parameter Service Policy

Navigational_Data_Parameter

This service identifies the Source_Parameters required to determine Location and/or Orientation of a Platform.

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.

B.2.32.7.1.5 Constraint

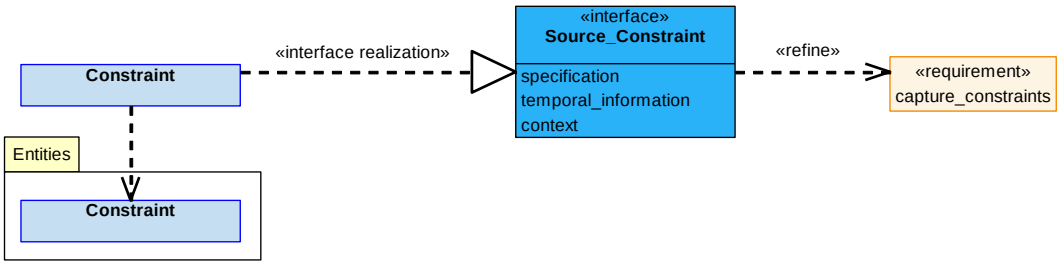

Figure 642: Constraint Service Definition

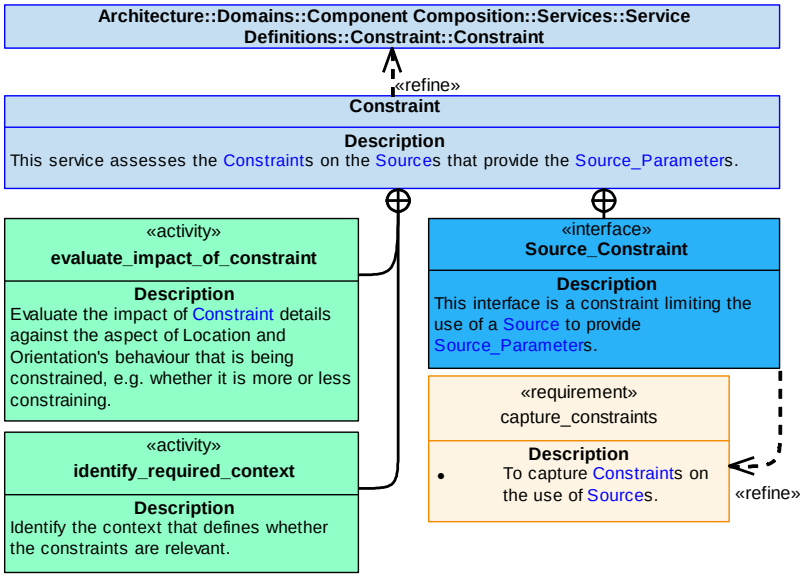

Figure 643: Constraint Service Policy

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. due to accuracy of Sources in polar 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.

B.2.32.7.1.6 Capability

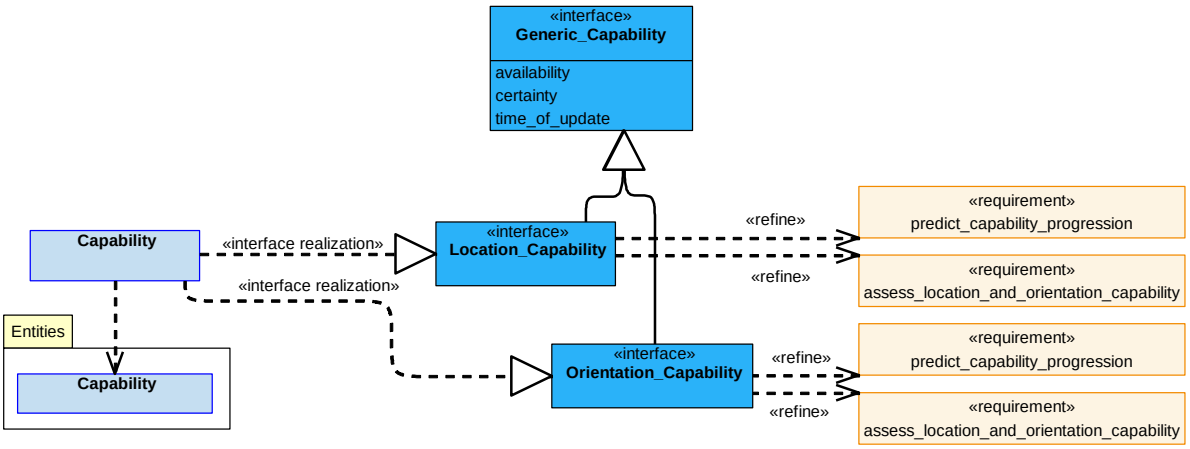

Figure 644: Capability Service Definition

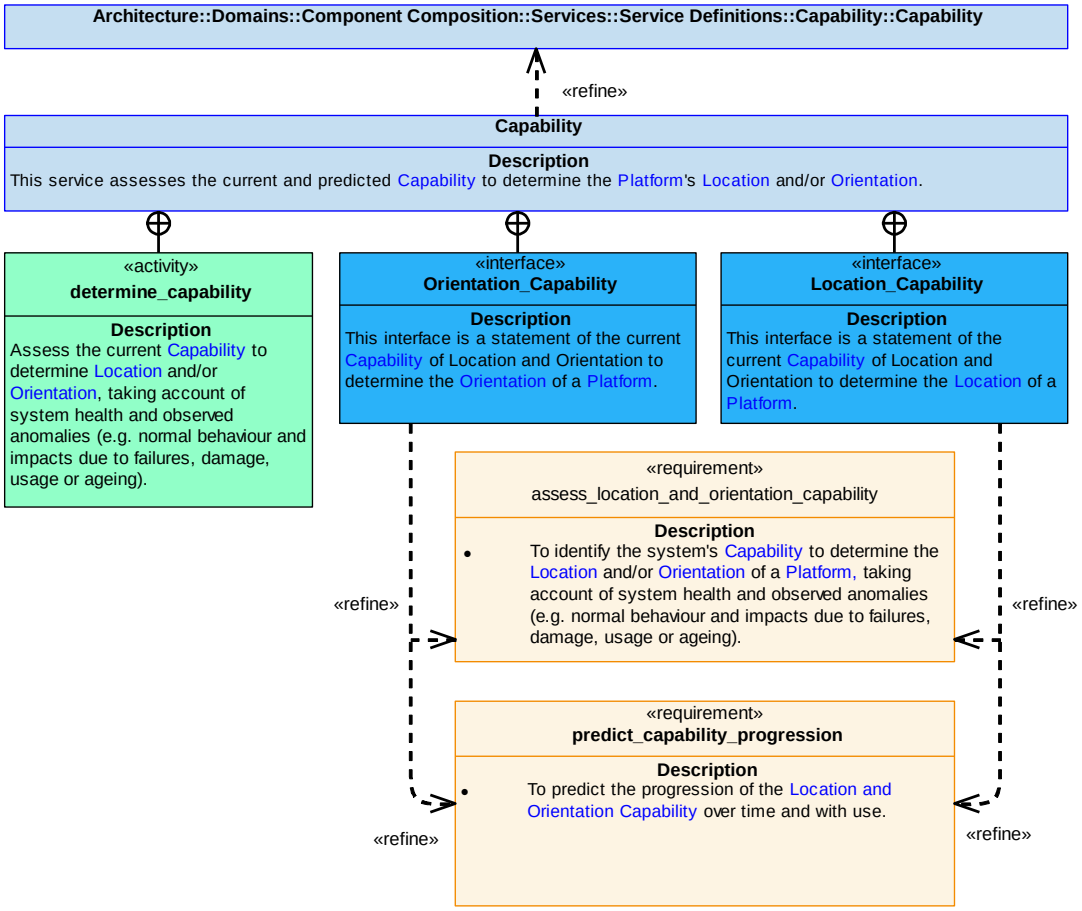

Figure 645: Capability Service Policy

Capability

This service assesses the current and predicted Capability to determine the Platform's Location and/or Orientation.

Interfaces

Location_Capability

This interface is a statement of the current Capability of Location and Orientation to determine the Location of a Platform.

Orientation_Capability

This interface is a statement of the current Capability of Location and Orientation to determine the Orientation of a Platform.

Activity

determine_capability

Assess the current Capability to determine Location and/or Orientation, taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

B.2.32.7.1.7 Capability_Evidence

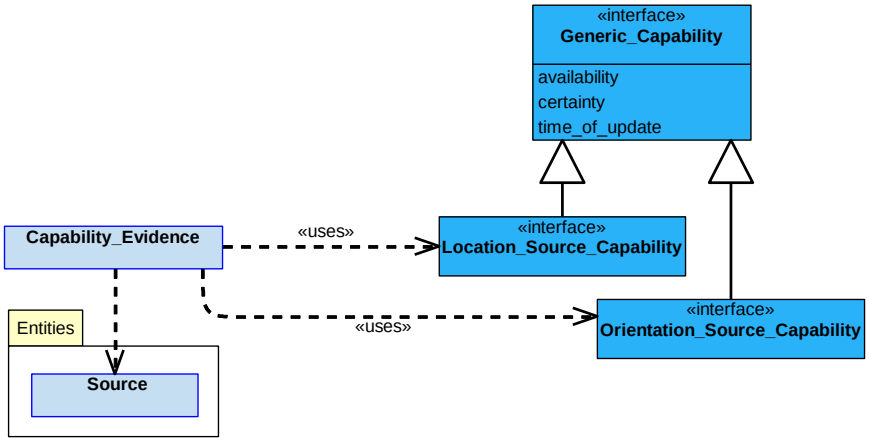

Figure 646: Capability_Evidence Service Definition

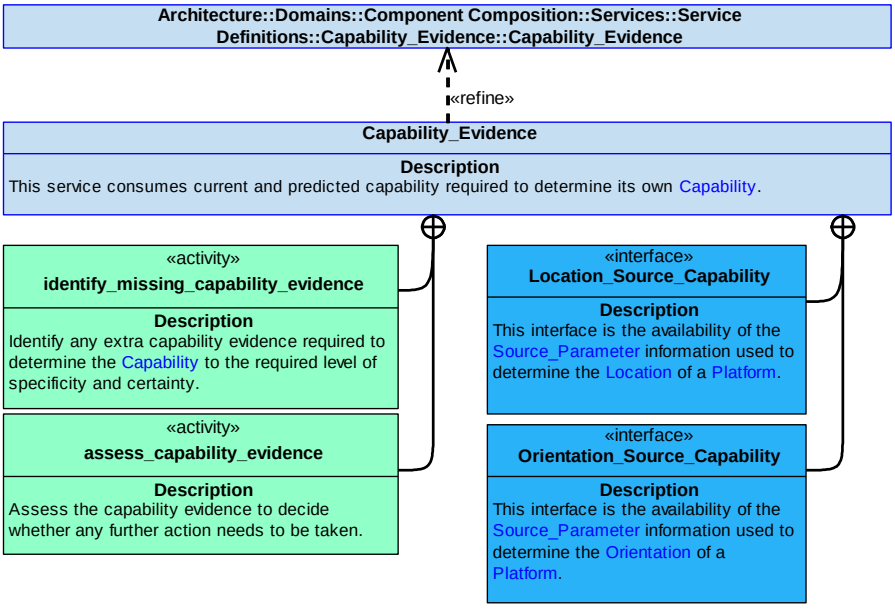

Figure 647: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability required to determine its own Capability.

Interfaces

Location_Source_Capability

This interface is the availability of the Source_Parameter information used to determine the Location of a Platform.

Orientation_Source_Capability

This interface is the availability of the Source_Parameter information used to determine the Orientation of a Platform.

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.

B.2.32.7.2 Service Dependencies

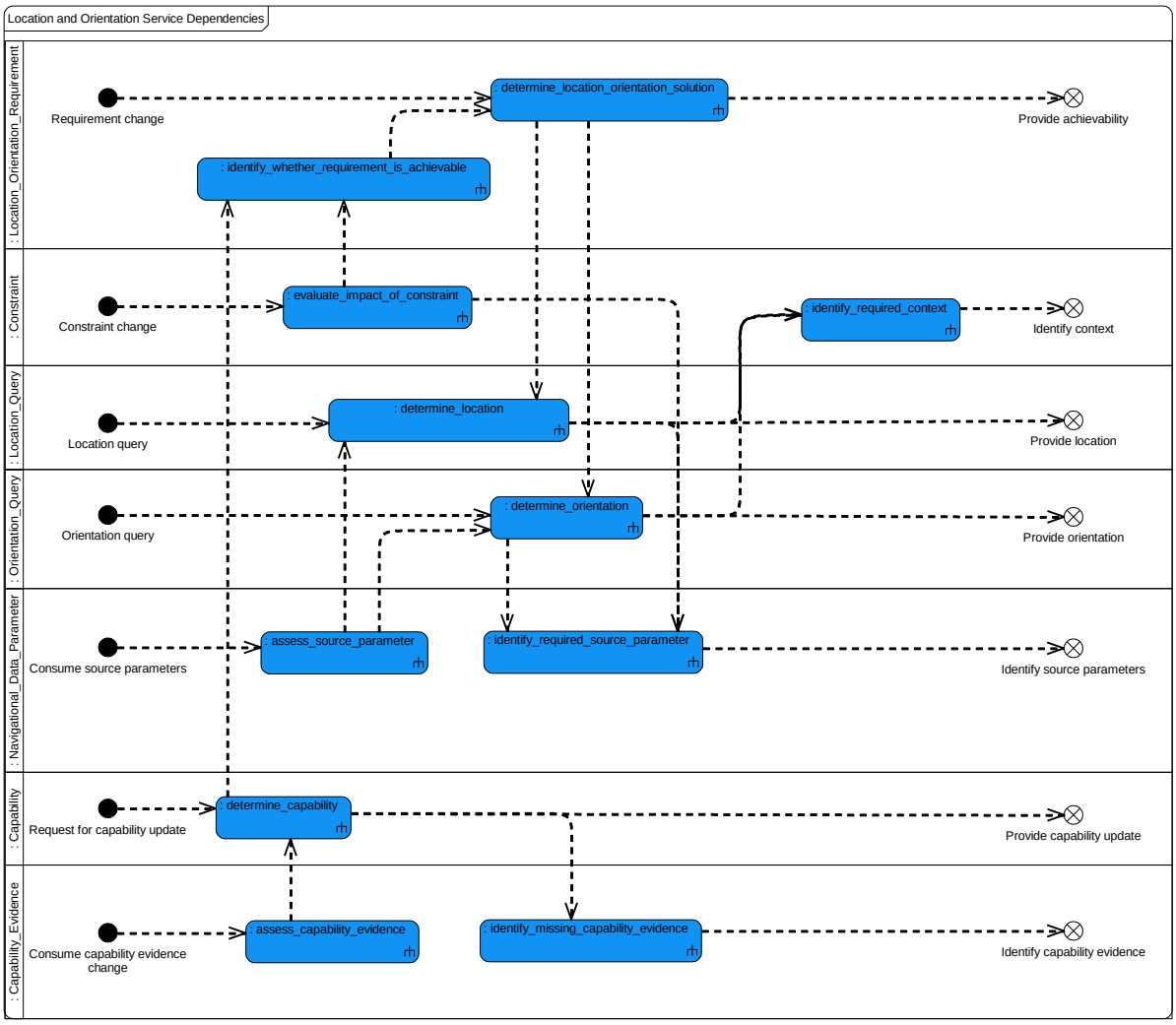

Figure 648: Location and Orientation Service Dependencies

B.2.33 Mass and Balance

B.2.33.1 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.

B.2.33.2 Overview

Control Architecture

Mass and Balance is a service component as defined in the Control Architecture policy.

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.

B.2.33.3 Service Summary

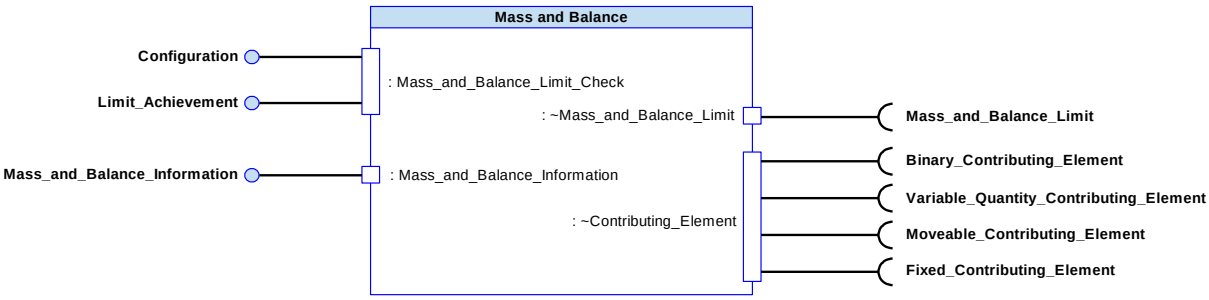

Figure 649: Mass and Balance Service Summary

B.2.33.4 Responsibilities

capture_mass_and_balance_limit

- To capture the Mass_and_Balance_Limits.

capture_inertia_and_balance_requirements

- To capture requirements for maintaining the Moment_of_Inertia and Centre_of_Mass for a given 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.

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.

determine_total_mass

- To determine the Total_Mass of Contributing_Elements within Configurations.

determine_moment_of_inertia

- To determine the Moment_of_Inertia of Contributing_Elements within Configurations.

determine_centre_of_mass

- To determine the Centre_of_Mass 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.

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.

determine_changes_conform_to_limits

- To determine whether Configuration changes (e.g. a planned stores release package) conform to Mass_and_Balance_Limits.

B.2.33.5 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.

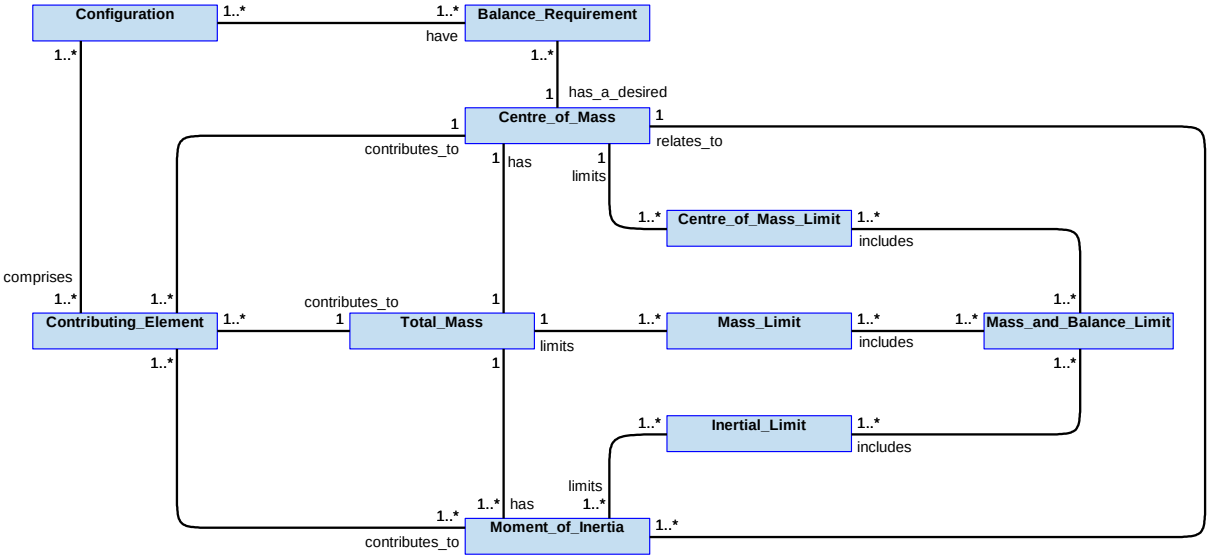

Figure 650: Mass and Balance Semantics

B.2.33.5.1 Entities

Balance_Requirement

A requirement for values of Centre_of_Mass placed upon a Configuration of Contributing_Elements.

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.

B.2.33.6 Design Rationale

B.2.33.6.1 Assumptions

- The component will have knowledge of Contributing_Elements and how they affect Total_Mass, Moment_of_Inertia and Centre_of_Mass.

B.2.33.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

Note: The capability of Mass and Balance does not change based on availability of supporting data. If this data is unavailable, Mass and Balance will still perform its role, using assumptions where necessary. Hence Mass and Balance does not provide an evolving view of its capabilities.

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.

B.2.33.6.3 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.

B.2.33.6.4 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.

B.2.33.7 Services

B.2.33.7.1 Service Definitions

B.2.33.7.1.1 Mass_and_Balance_Limit_Check

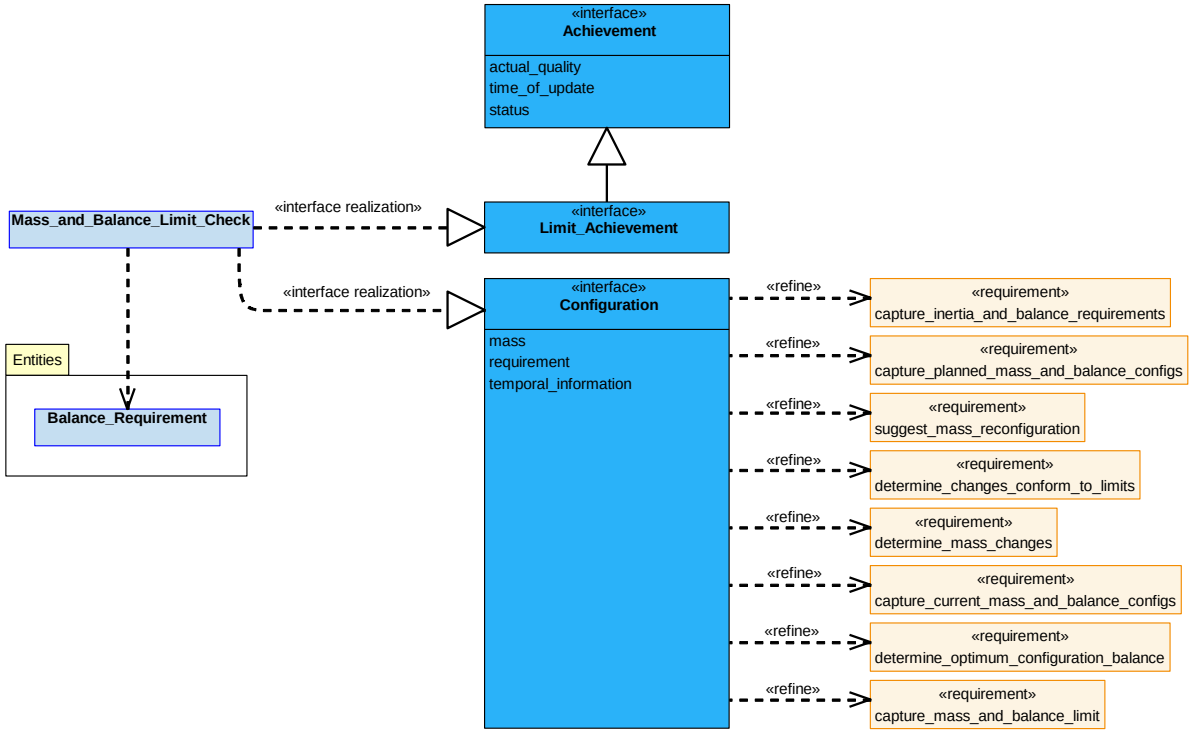

Figure 651: Mass_and_Balance_Limit_Check Service Definition

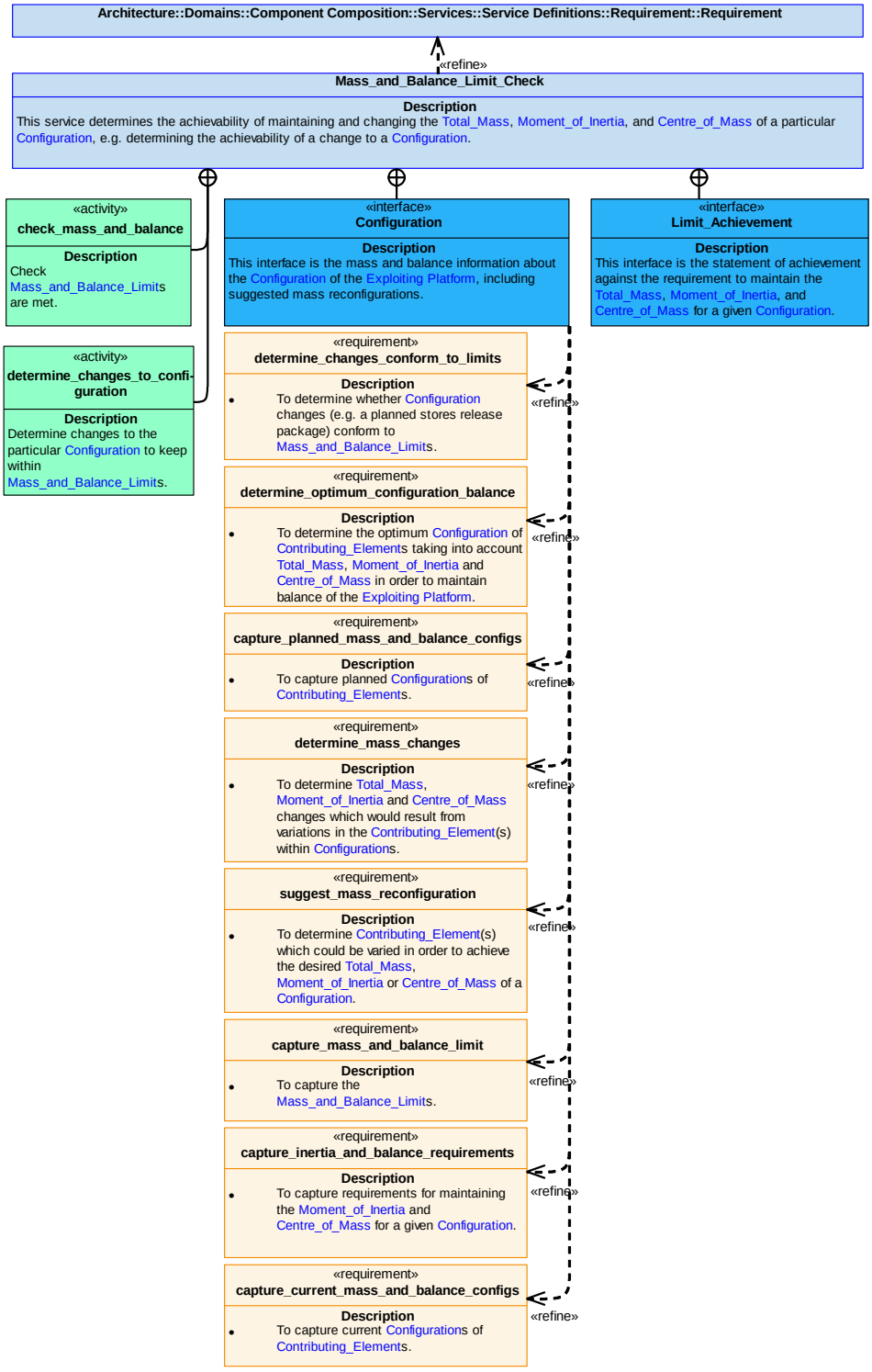

Figure 652: Mass_and_Balance_Limit_Check Service Policy

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.

B.2.33.7.1.2 Mass_and_Balance_Limit

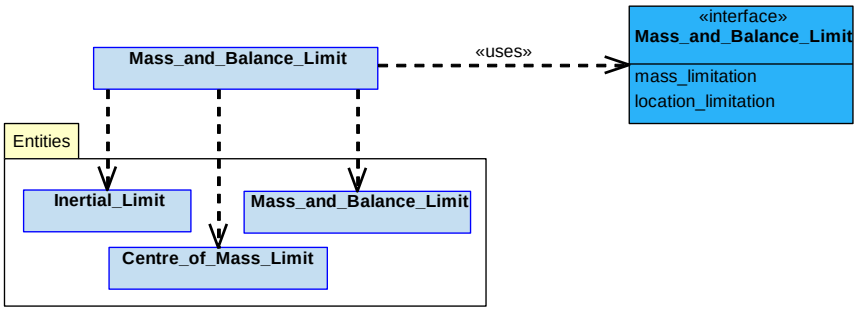

Figure 653: Mass_and_Balance_Limit Service Definition

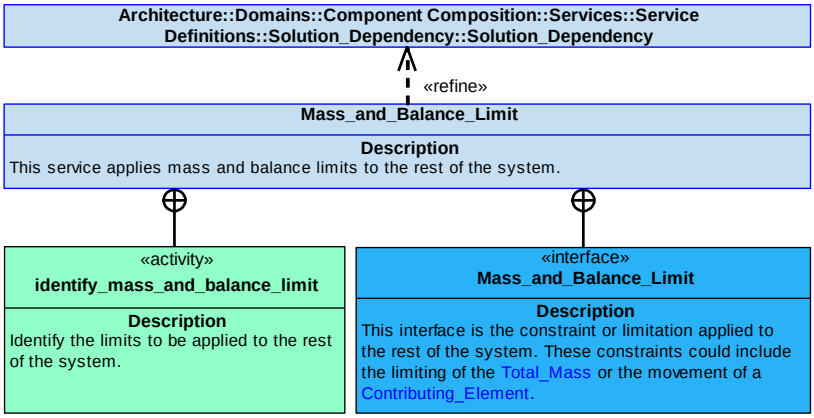

Figure 654: Mass_and_Balance_Limit Service Policy

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.

Activity

identify_mass_and_balance_limit

Identify the limits to be applied to the rest of the system.

B.2.33.7.1.3 Mass_and_Balance_Information

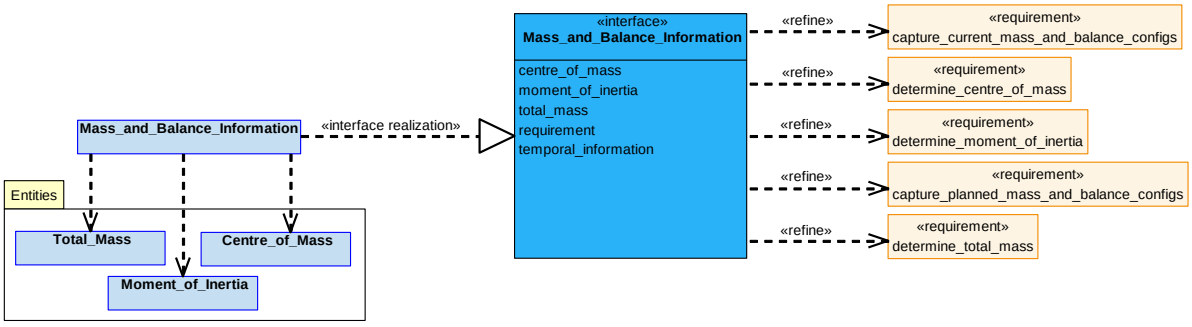

Figure 655: Mass_and_Balance_Information Service Definition

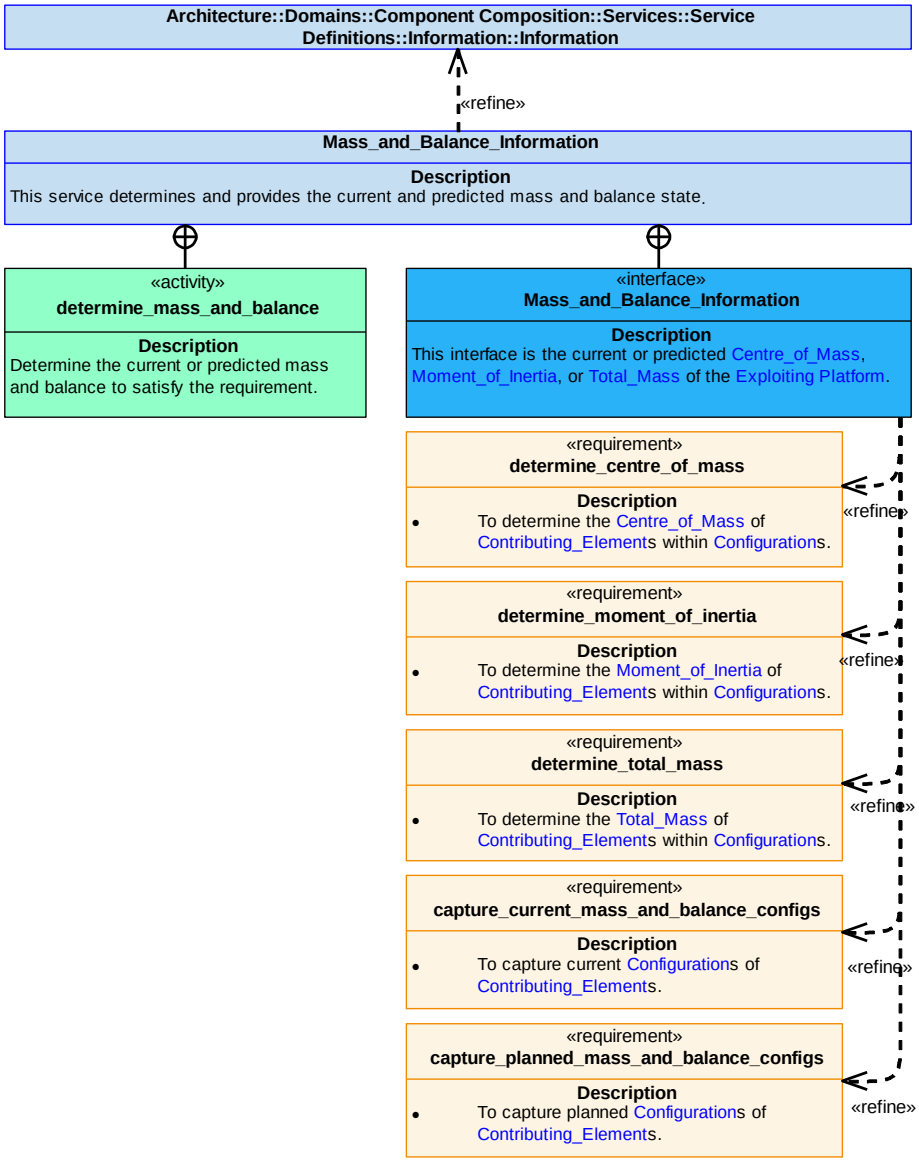

Figure 656: Mass_and_Balance_Information Service Policy

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.

B.2.33.7.1.4 Contributing_Element

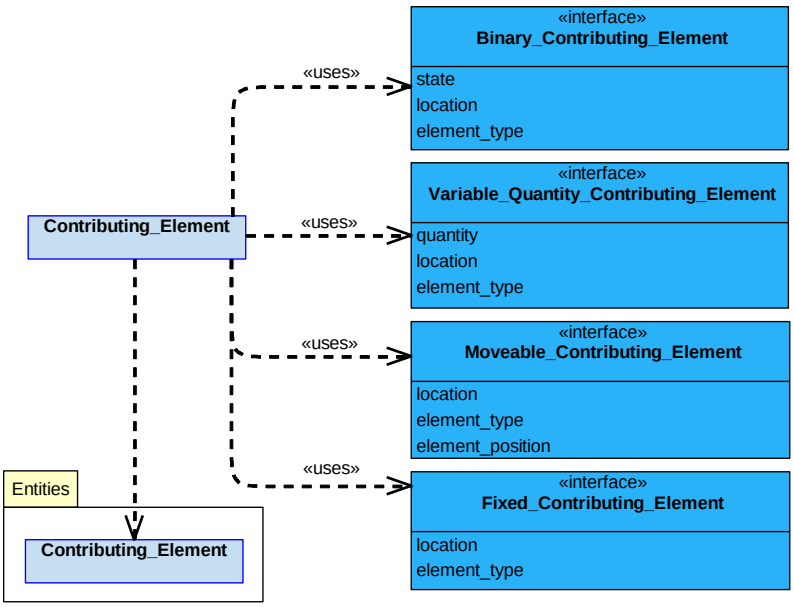

Figure 657: Contributing_Element Service Definition

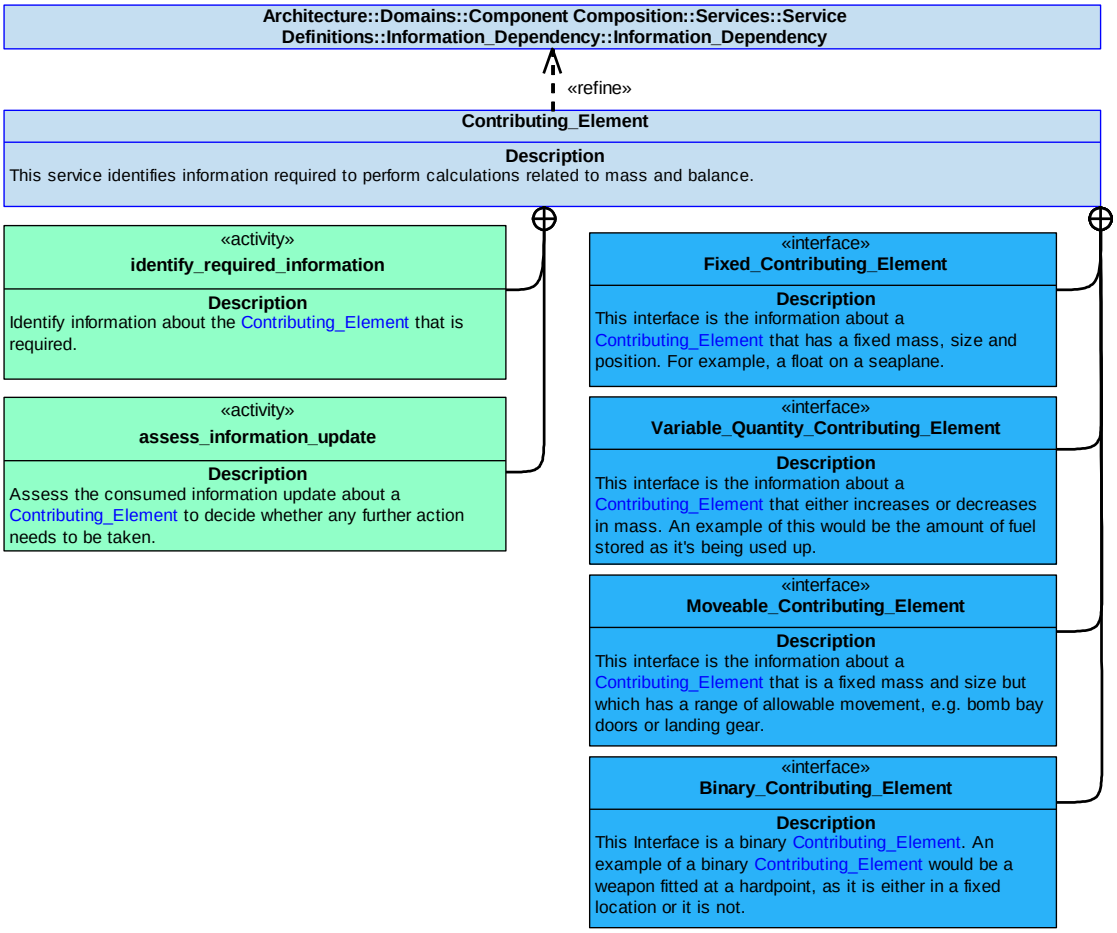

Figure 658: Contributing_Element Service Policy

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.

B.2.33.7.2 Service Dependencies

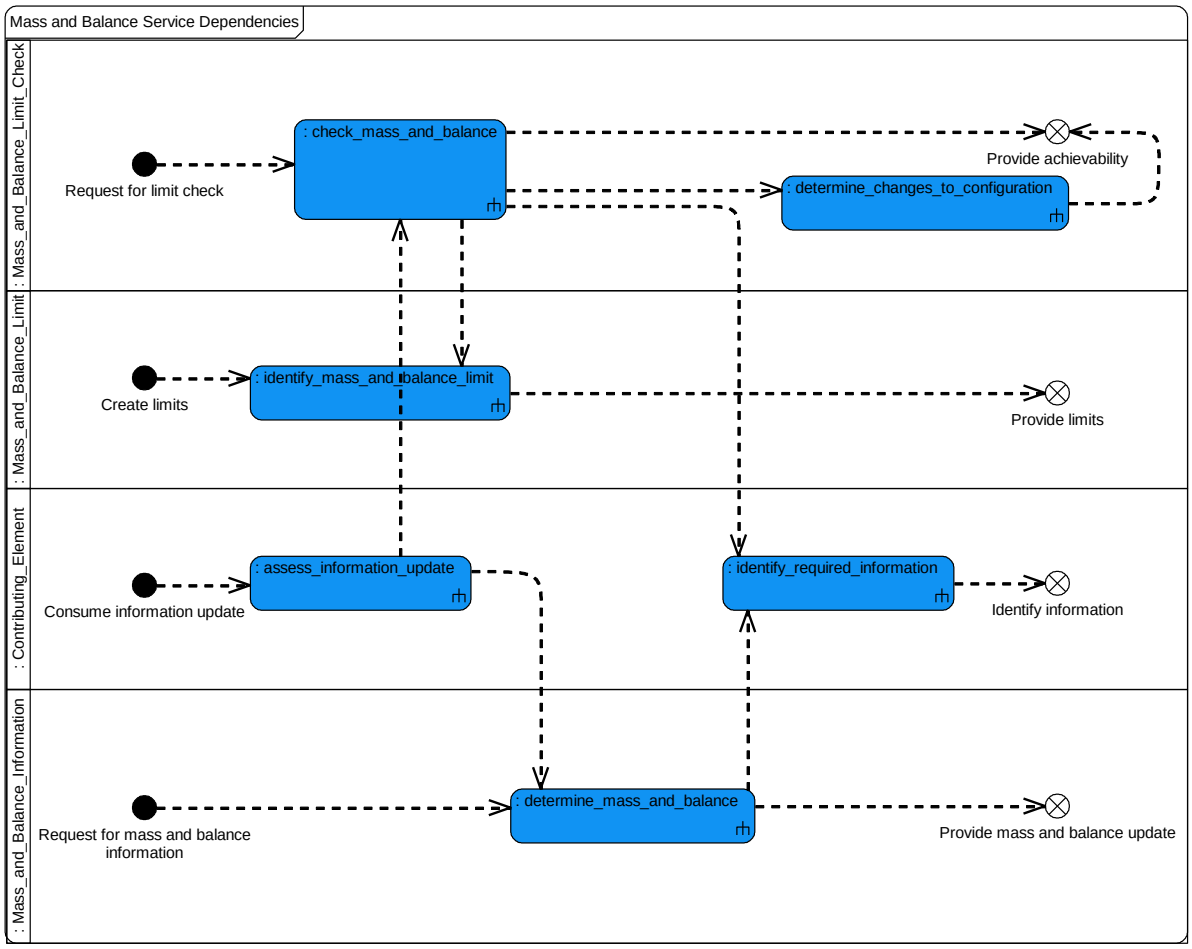

Figure 659: Mass and Balance Service Dependencies

B.2.34 Mechanical Positioning

B.2.34.1 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.

B.2.34.2 Overview

Control Architecture

Mechanical Positioning is a resource component as defined in the Control Architecture policy.

Standard Pattern of Use

When there is a Requirement for movement of a Physical_Element on the Exploiting Platform, Mechanical Positioning will command its associated Effector(s) (e.g. a hydraulic actuator) 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

B.2.34.3 Service Summary

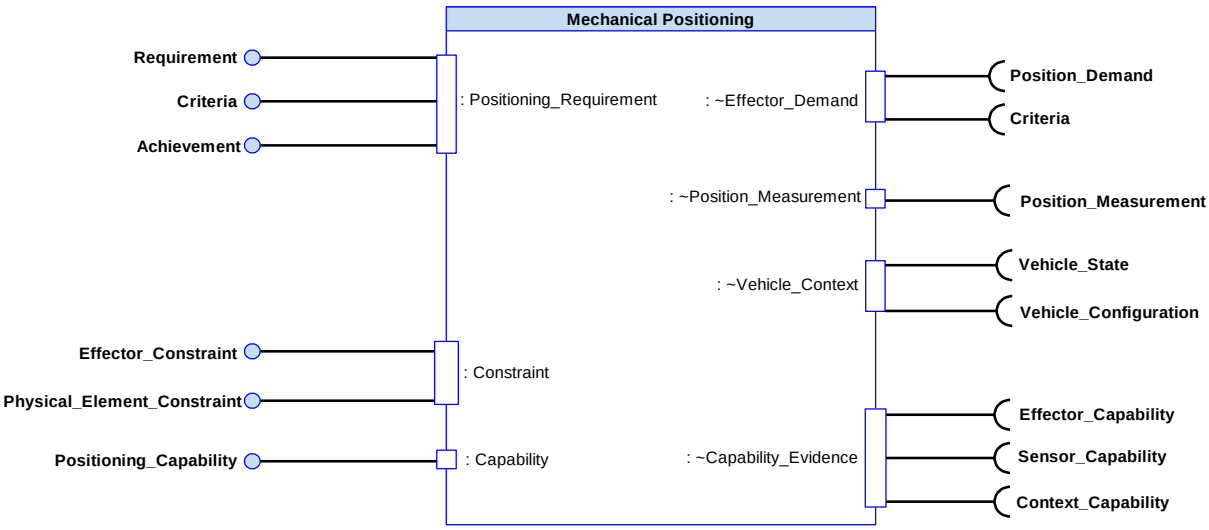

Figure 660: Mechanical Positioning Service Summary

B.2.34.4 Responsibilities

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).

capture_positioning_constraints

- To capture provided Physical_Element_Constraints for a Physical_Element.

capture_positioning_requirements

- To capture provided positioning Requirements for a Physical_Element.

control_position

- To control the position of a Physical_Element by commanding Effectors.

determine_required_position_solution

- To determine the Effector positions to achieve the required position of a Physical_Element.

predict_movement_capability_progression

- To predict the progression of the Movement_Capability of a Physical_Element over time and with use.

capture_effector_constraints

- To capture provided Effector_Constraints for an Effector.

capture_measurement_criteria

- To capture given Measurement_Criterion.

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.

identify_positioning_progress

- To identify the progress of a Positioning_Solution against a Requirement.

identify_missing_capability_information

- To identify missing information which could improve the certainty or specificity of Movement_Capability determination.

B.2.34.5 Subject Matter Semantics

The subject matter of Mechanical Positioning is moveable Physical_Elements and their positions.

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.

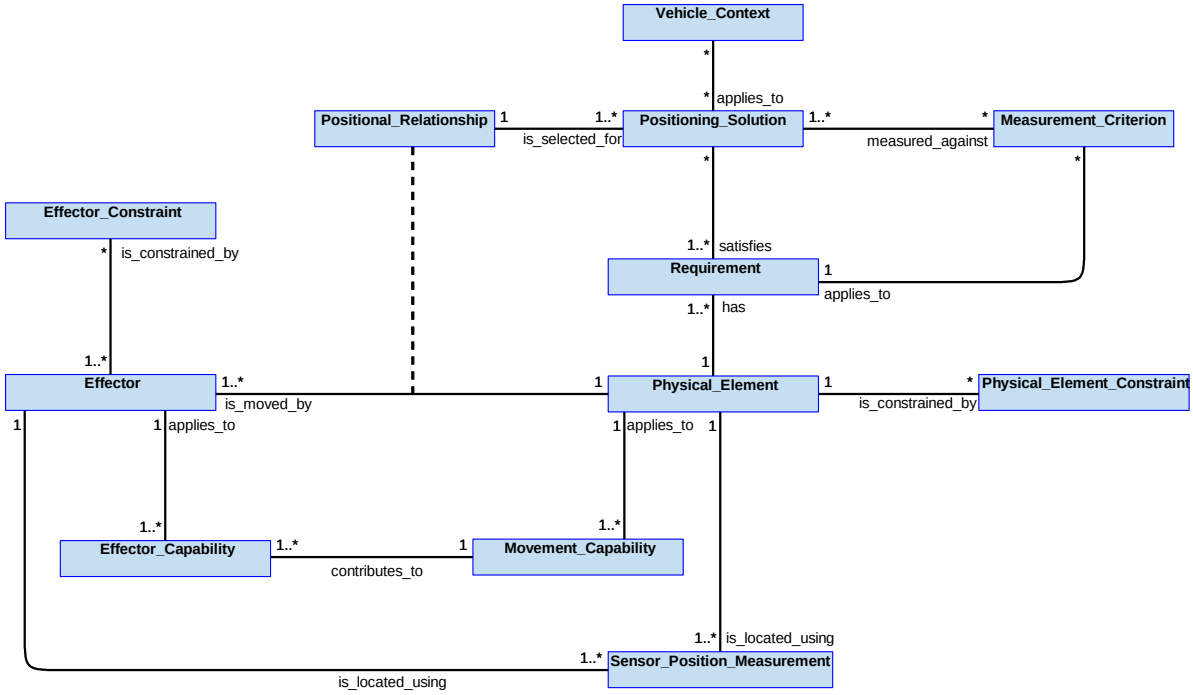

Figure 661: Mechanical Positioning Semantics

B.2.34.5.1 Entities

Effector

An item used to attain a desired change in the position of an associated Physical_Element, e.g. a control surface actuator is an effector.

Effector_Constraint

A constraint 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 constraint on the way that a Physical_Element may be moved or positioned.

Positional_Relationship

The relationship between the position 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 that a Physical_Element needs to attain a defined position.

Movement_Capability

The ability of the component to implement a positioning Requirement for a Physical_Element.

Sensor_Position_Measurement

The determined values about the position of an Effector or a Physical_Element.

Effector_Capability

The capability of an Effector to enact a Physical_Element position change.

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.

B.2.34.6 Design Rationale

B.2.34.6.1 Assumptions

- An instance of the Mechanical Positioning component may control several related Effectors (e.g. a number of actuators attached to a control surface), but an Effector can only be under control of a single Mechanical Positioning instance.
- Any safety related limits for an Effector's position or use have been agreed outside this component.
- This component does not have details of any Exploiting Platform's performance data associated with its Effector movements.

B.2.34.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Mechanical Positioning:

- Resource Management - Each instantiation of Mechanical Positioning will represent the 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 deployment 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.

B.2.34.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- The uses of this component include controlling effectors that move 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.

B.2.34.6.4 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.

B.2.34.7 Services

B.2.34.7.1 Service Definitions

B.2.34.7.1.1 Positioning_Requirement

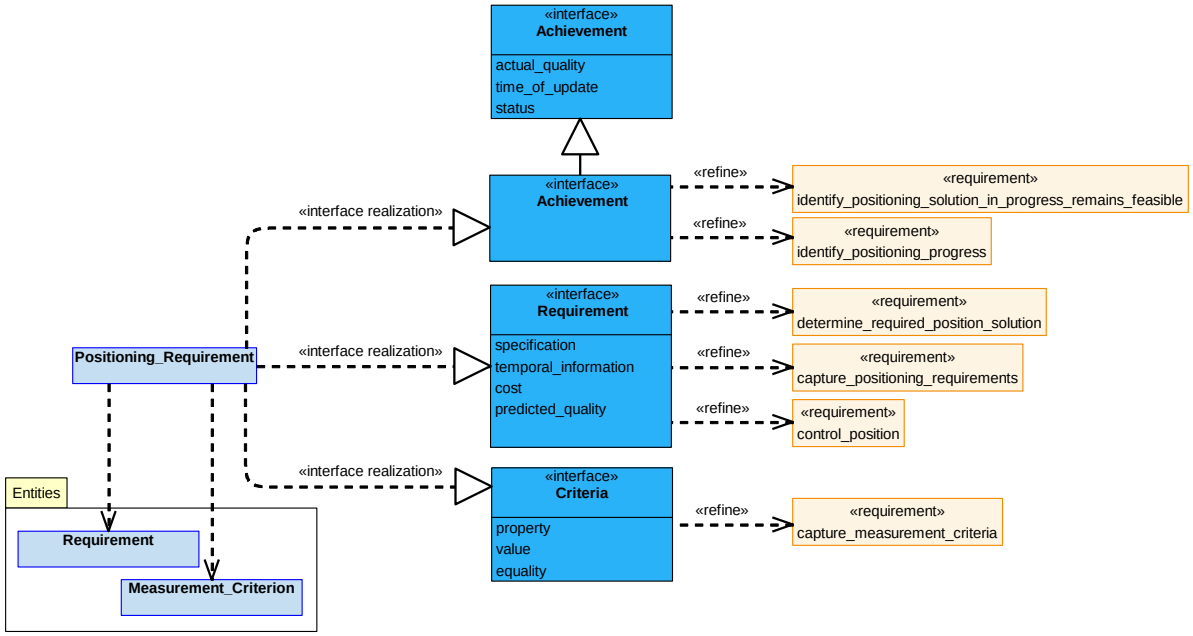

Figure 662: Positioning_Requirement Service Definition

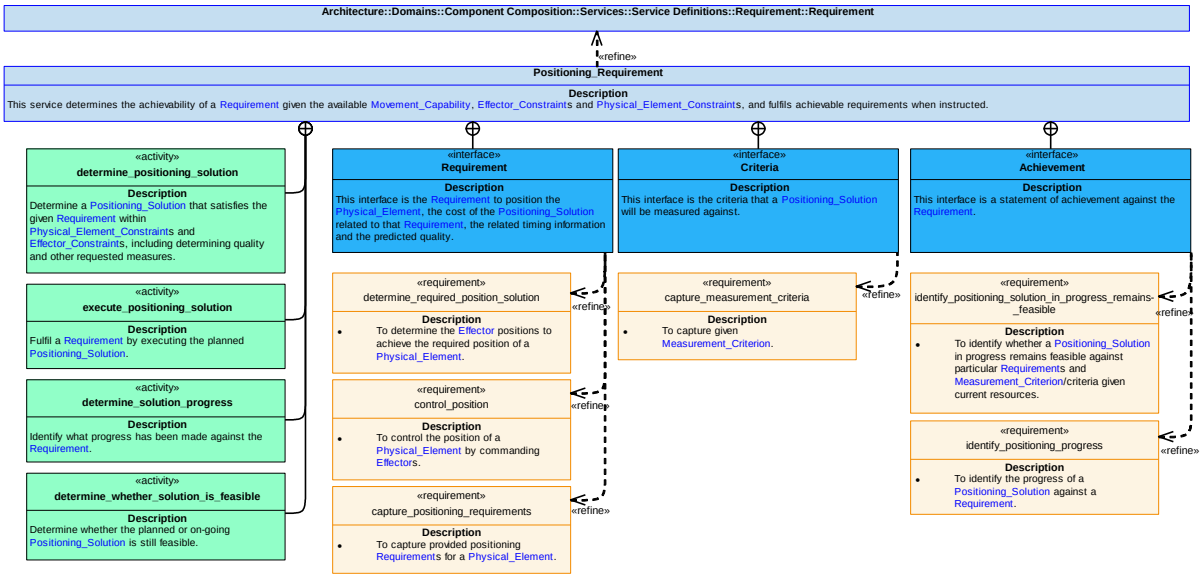

Figure 663: Positioning_Requirement Service Policy

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.

B.2.34.7.1.2 Effector_Demand

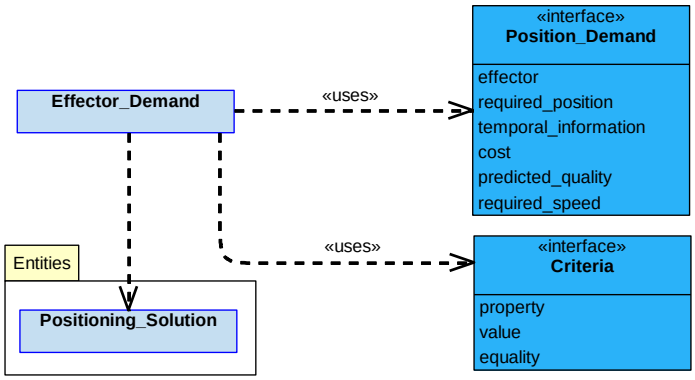

Figure 664: Effector_Demand Service Definition

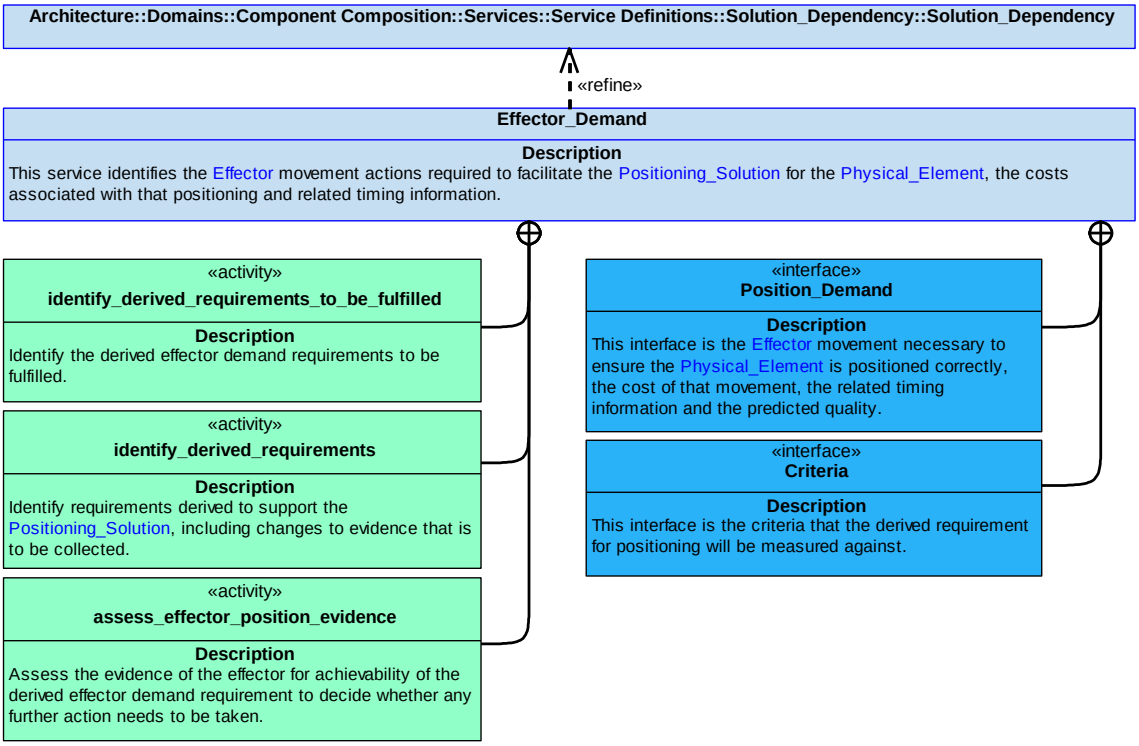

Figure 665: Effector_Demand Service Policy

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

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).
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.
required_speed	The speed the Effector needs to be moved at (e.g. extension at 10mm per second).

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 required position of the Effector to achieve the desired Physical_Element position.
value	The measured value of the property, e.g. mm from a reference point.
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.

B.2.34.7.1.3 Position_Measurement

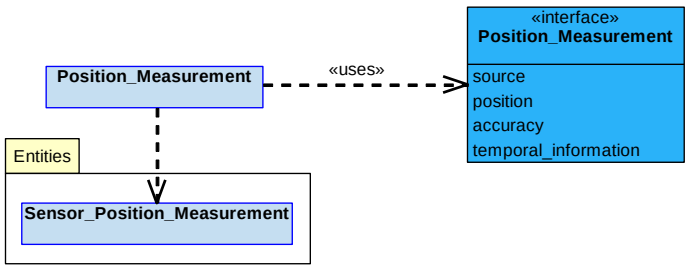

Figure 666: Position_Measurement Service Definition

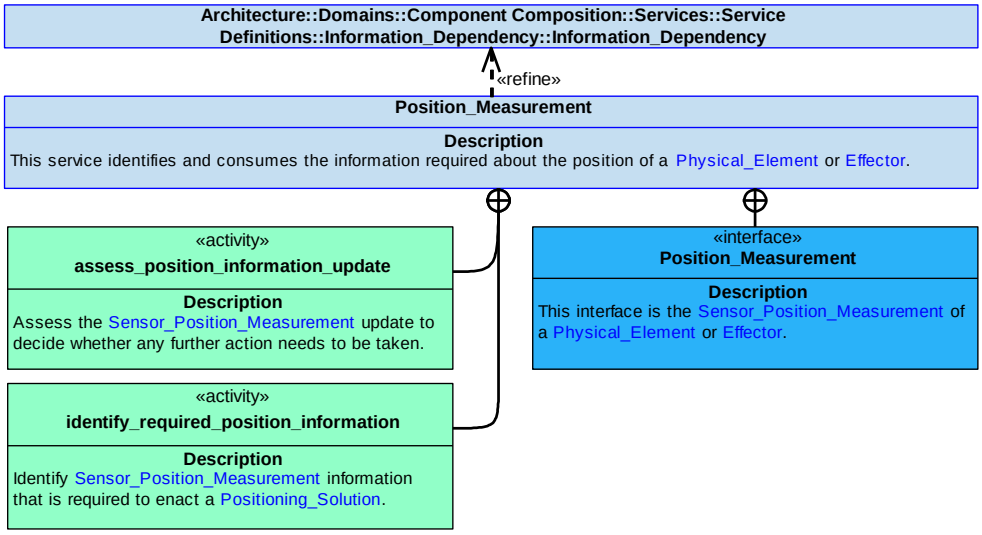

Figure 667: Position_Measurement Service Policy

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 Sensor_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 Sensor_Position_Measurement update to decide whether any further action needs to be taken.

identify_required_position_information

Identify Sensor_Position_Measurement information that is required to enact a Positioning_Solution.

B.2.34.7.1.4 Vehicle_Context

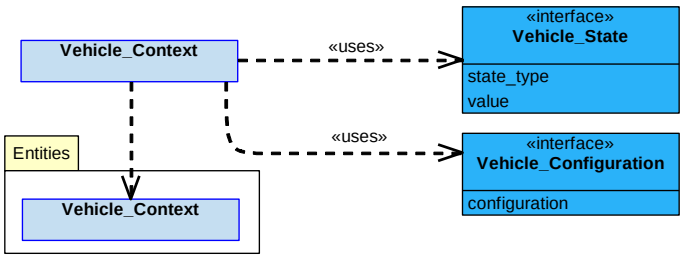

Figure 668: Vehicle_Context Service Definition

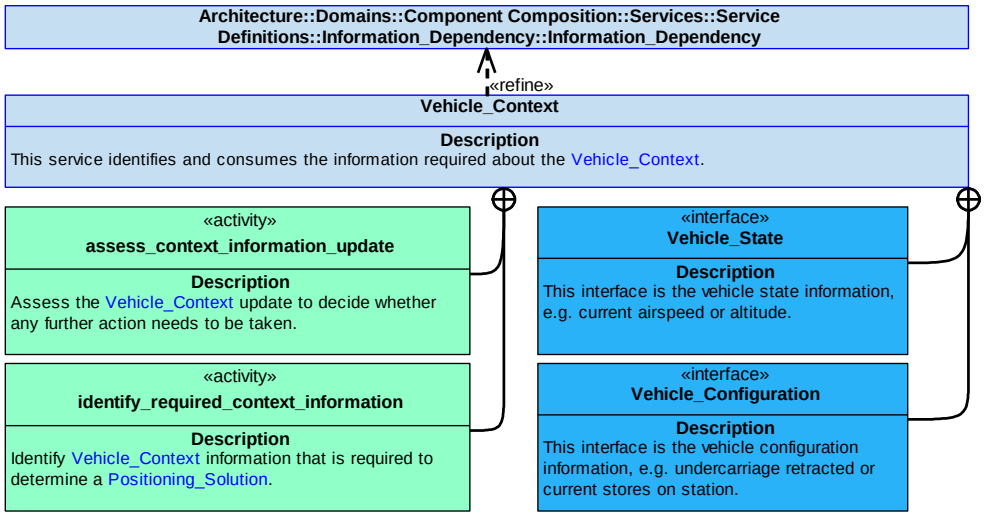

Figure 669: Vehicle_Context Service Policy

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.

B.2.34.7.1.5 Constraint

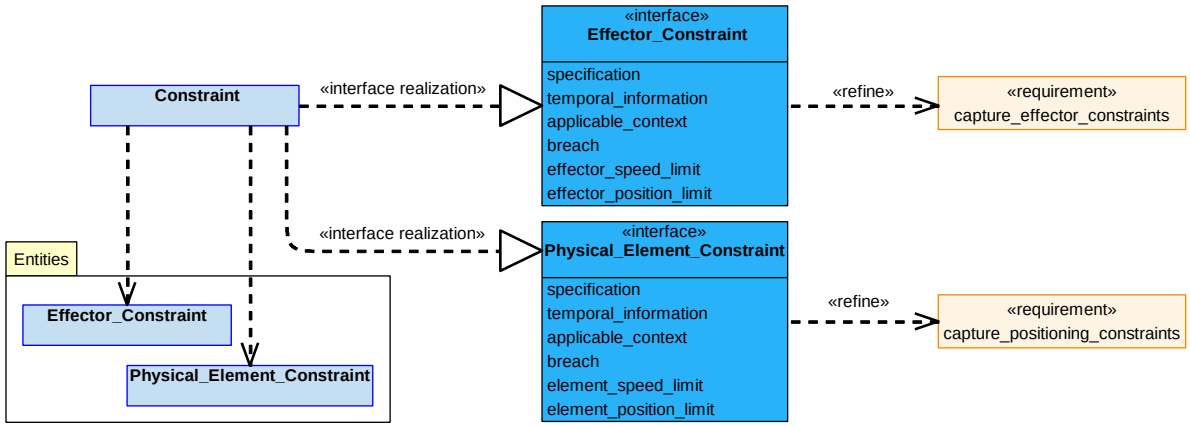

Figure 670: Constraint Service Definition

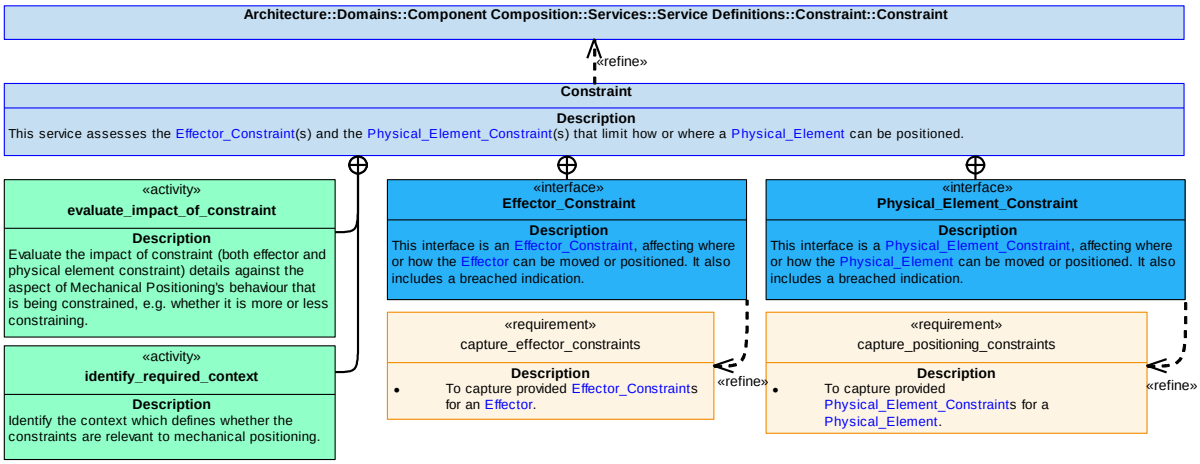

Figure 671: Constraint Service Policy

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.

B.2.34.7.1.6 Capability

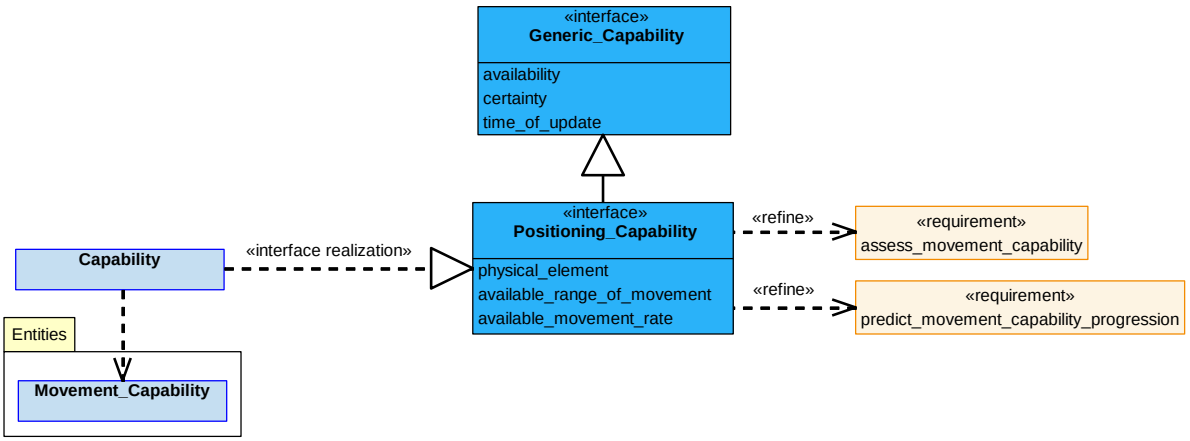

Figure 672: Capability Service Definition

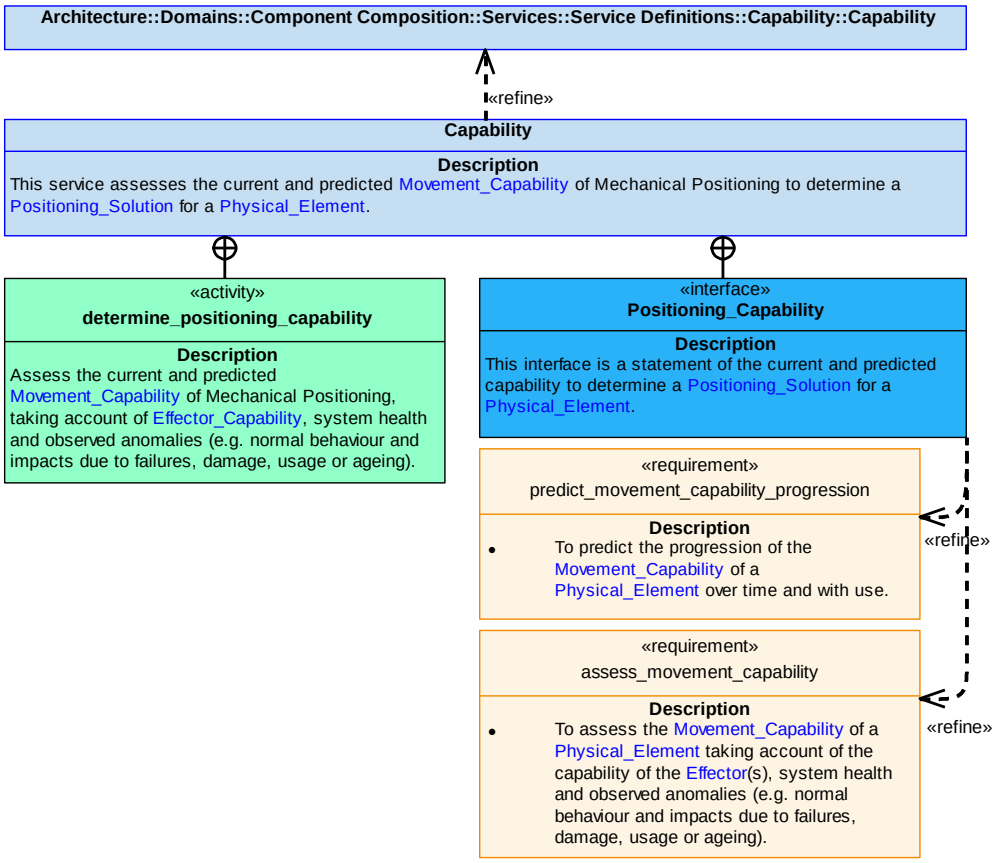

Figure 673: Capability Service Policy

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).

B.2.34.7.1.7 Capability_Evidence

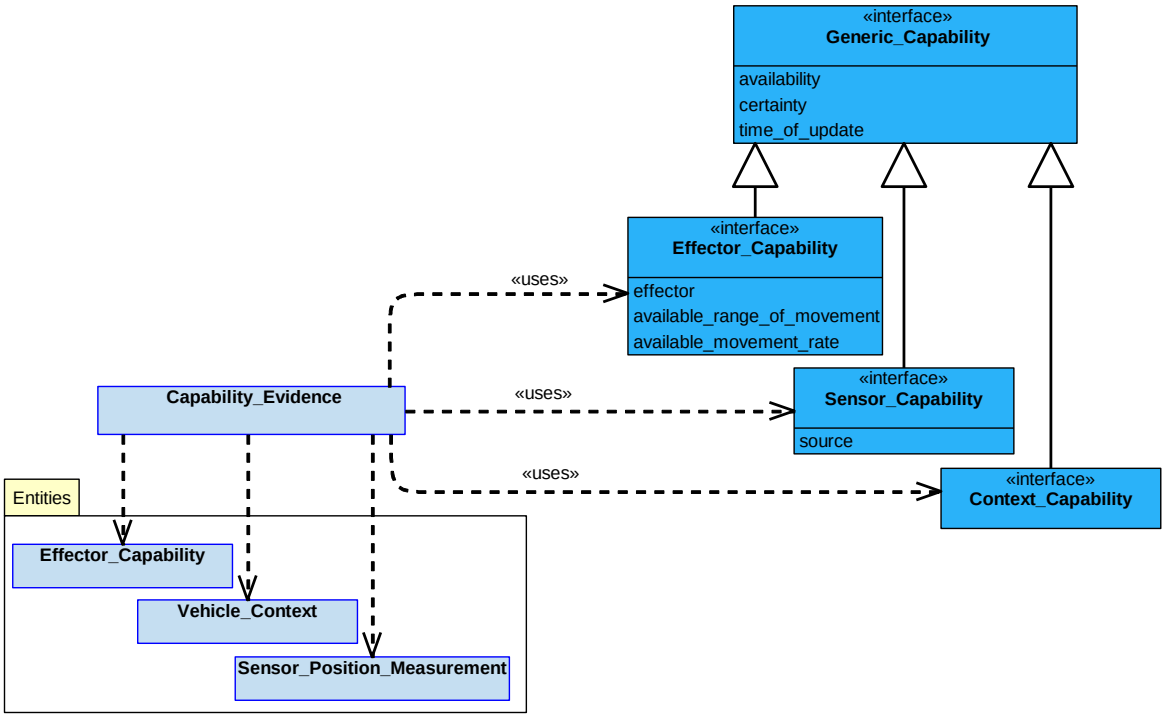

Figure 674: Capability_Evidence Service Definition

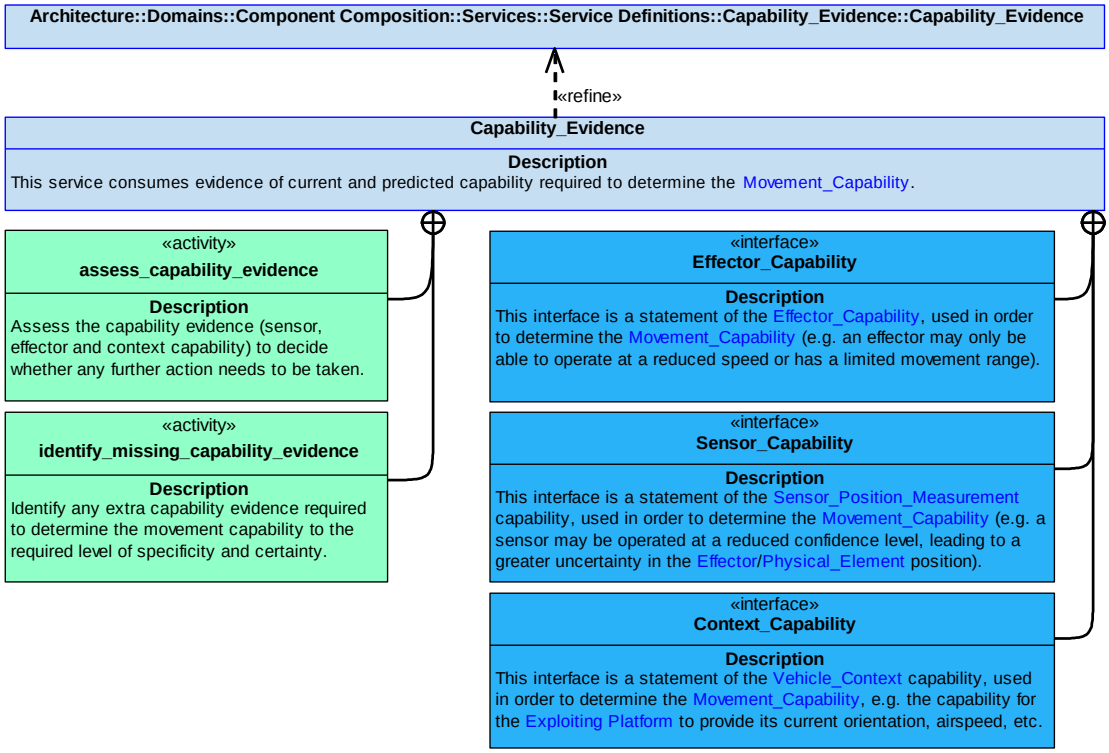

Figure 675: Capability_Evidence Service Policy

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 Sensor_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 Sensor_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.

B.2.34.7.2 Service Dependencies

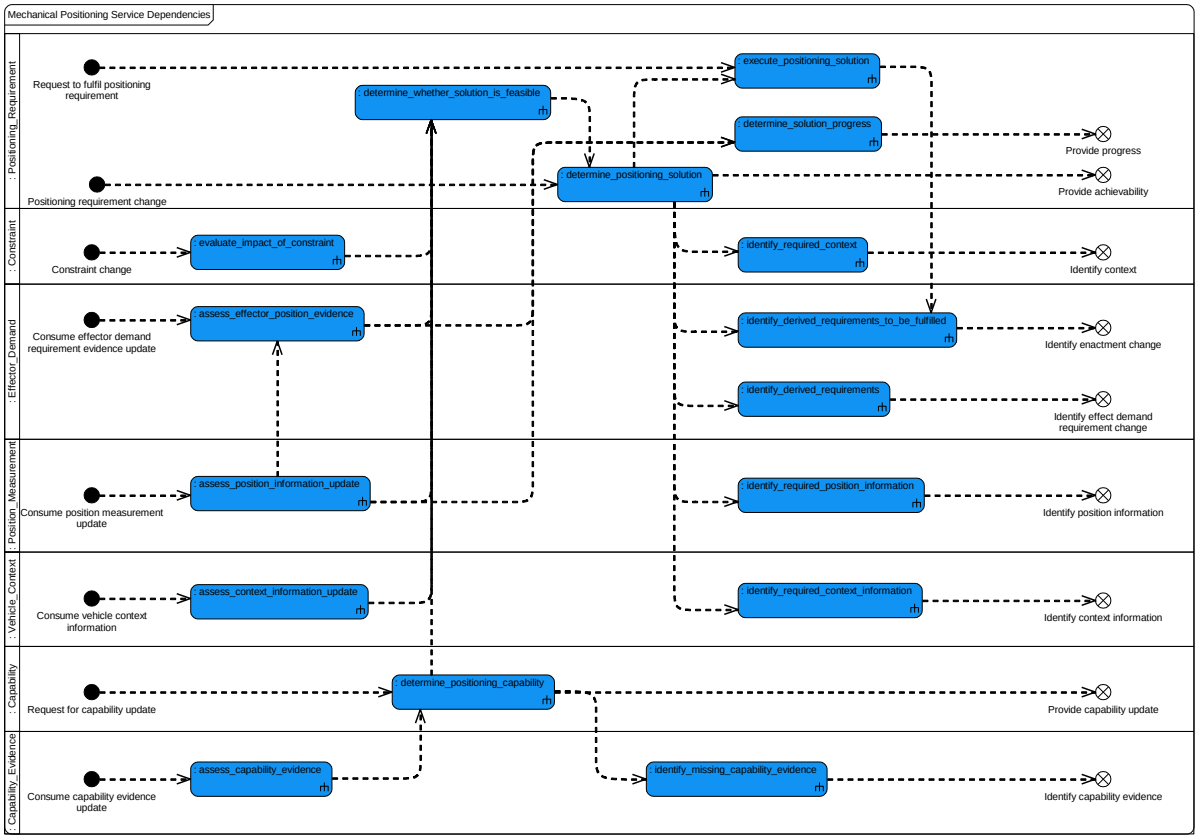

Figure 676: Mechanical Positioning Service Dependencies

B.2.35 Navigation Sensing

B.2.35.1 Role

The role of Navigation Sensing is to coordinate the activities and resources required to provide a navigation solution of the required quality.

B.2.35.2 Overview

Control Architecture

Navigation Sensing is an action component as defined in the Control Architecture policy.

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.

B.2.35.3 Service Summary

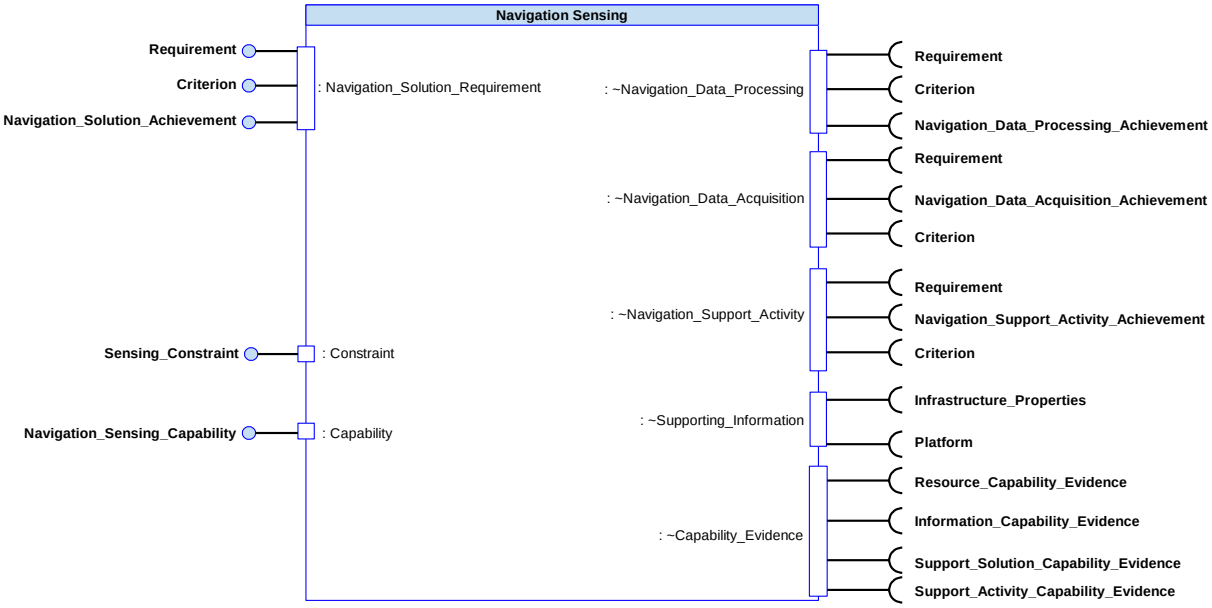

Figure 677: Navigation Sensing Service Summary

B.2.35.4 Responsibilities

capture_navigation_requirements

- To capture the Requirements for navigation, e.g. provision of the most accurate or most robust positional information.

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).

capture_solution_measurement_criteria

- To capture given Measurement_Criterion/criteria (e.g. stability or timing) against which a Navigation_Solution_Scheme will be assessed.

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_navigation_solution

- To determine a Navigation_Solution_Scheme using the available Capability, which meets the Requirements and satisfies the Constraints.

determine_solution_cost

- To determine the cost of a Navigation_Solution_Scheme against given 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).

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.

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).

predict_capability_progression

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

identify_whether_requirement_is_achievable

- To identify whether a Requirement is achievable given the current or predicted Capability and Constraints.

identify_progress_of_navigation_solution

- To identify the progress of a Navigation_Solution_Scheme and achievement against the Requirement.

determine_predicted_quality_of_navigation_solution_scheme

- To determine the predicted quality of a Navigation_Solution_Scheme against the Measurement_Criterion/criteria.

determine_solution_dependencies

- To identify dependencies to support a Navigation_Solution_Scheme or a step of the Navigation_Solution_Scheme.

identify_missing_information

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

B.2.35.5 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.

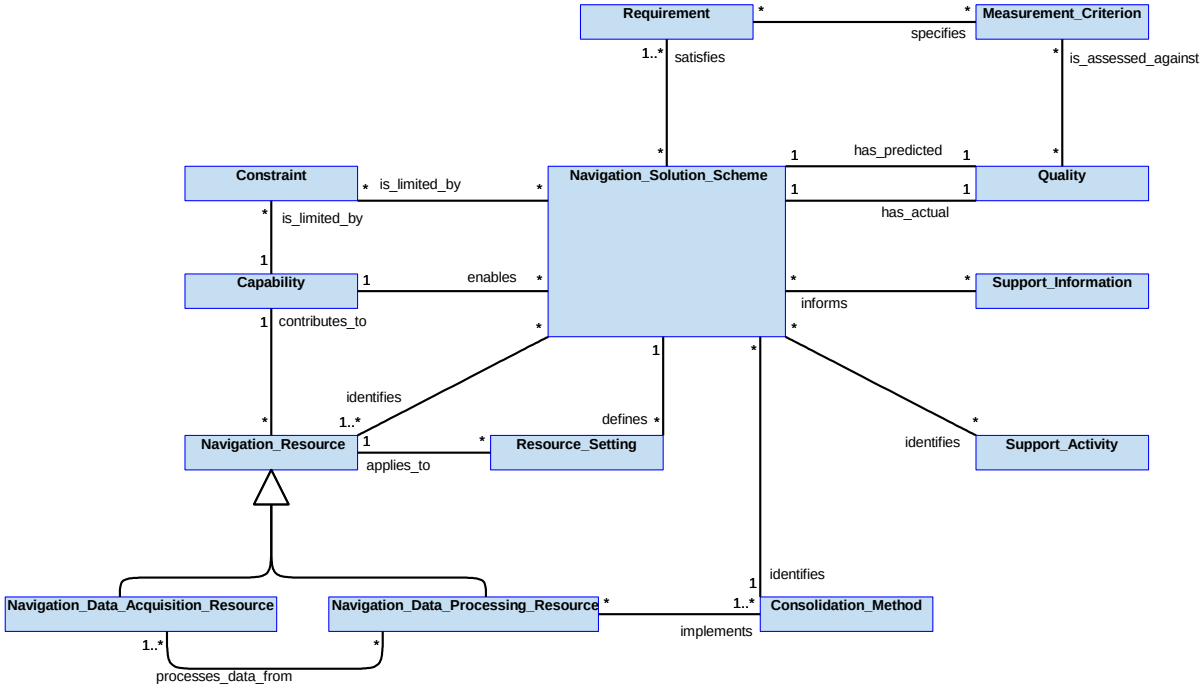

Figure 678: Navigation Sensing Semantics

B.2.35.5.1 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.

B.2.35.6 Design Rationale

B.2.35.6.1 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.

B.2.35.6.2 Design Considerations

Directly Applicable Policies

These policies 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 deployment 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 deployment 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.

B.2.35.6.3 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).

B.2.35.6.4 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.

B.2.35.7 Services

B.2.35.7.1 Service Definitions

B.2.35.7.1.1 Navigation_Solution_Requirement

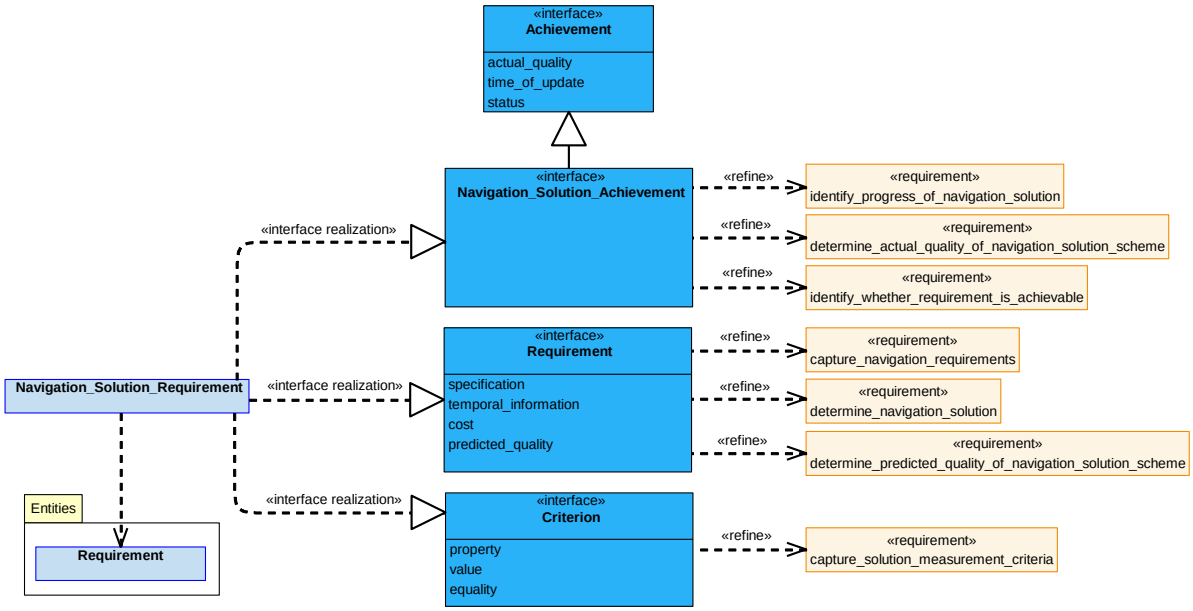

Figure 679: Navigation_Solution_Requirement Service Definition

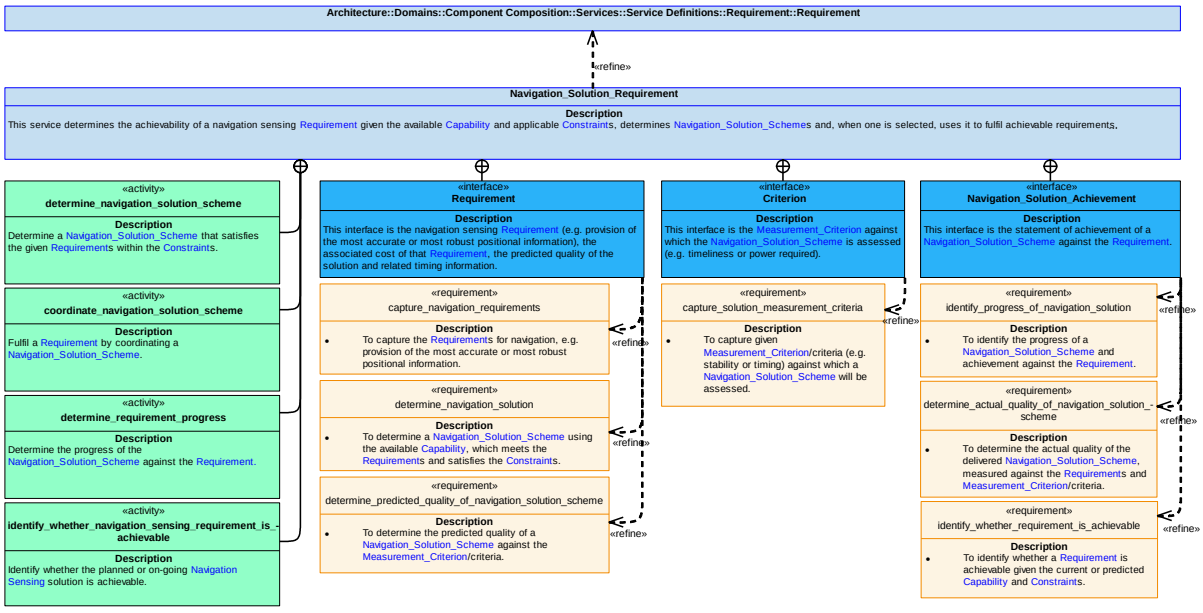

Figure 680: Navigation_Solution_Requirement Service Policy

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.

B.2.35.7.1.2 Navigation_Data_Processing

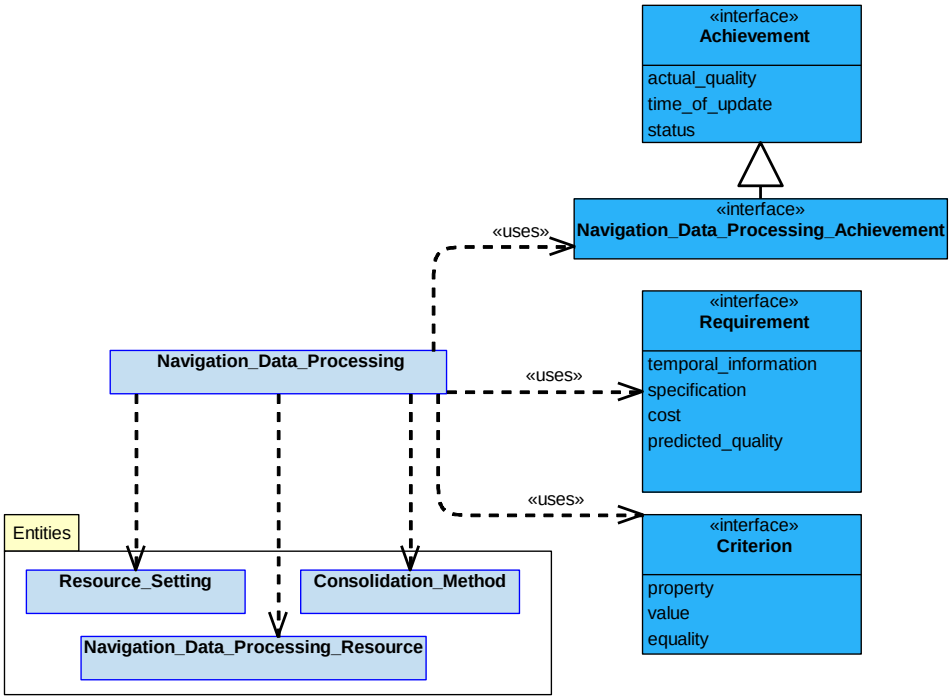

Figure 681: Navigation_Data_Processing Service Definition

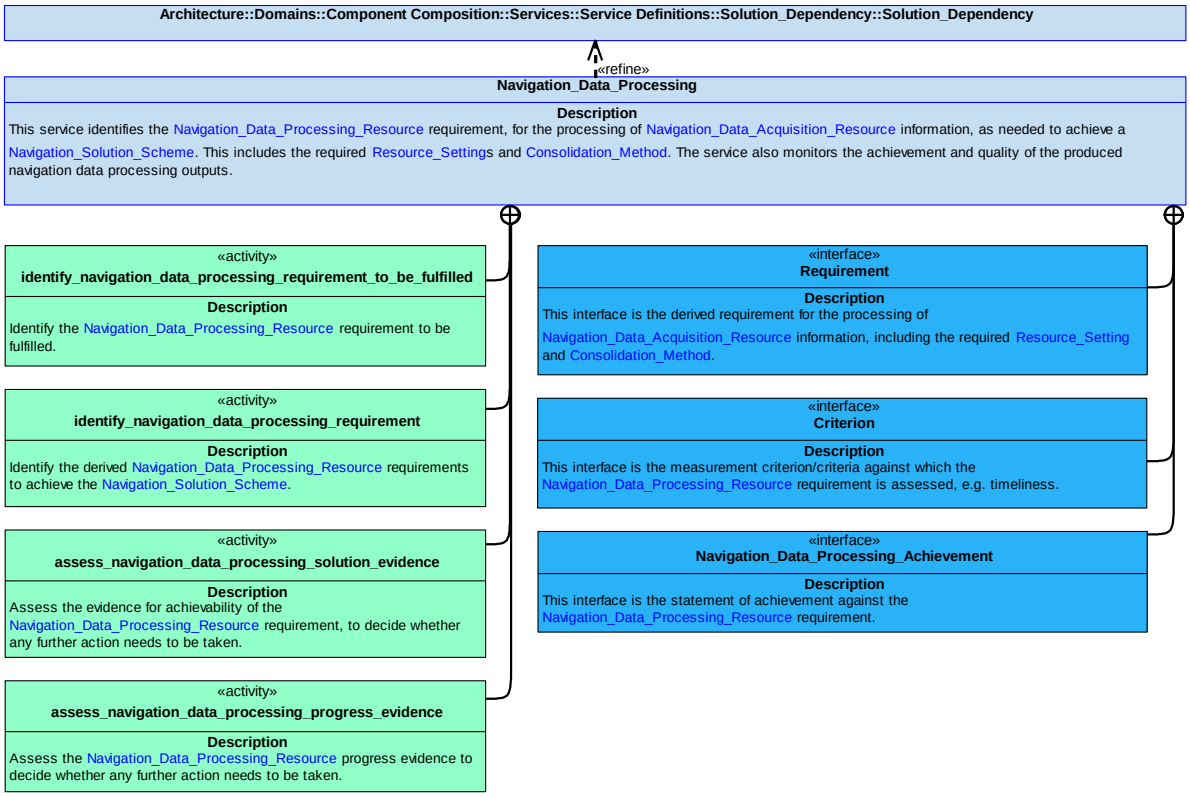

Figure 682: Navigation_Data_Processing Service Policy

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.

B.2.35.7.1.3 Navigation_Data_Acquisition

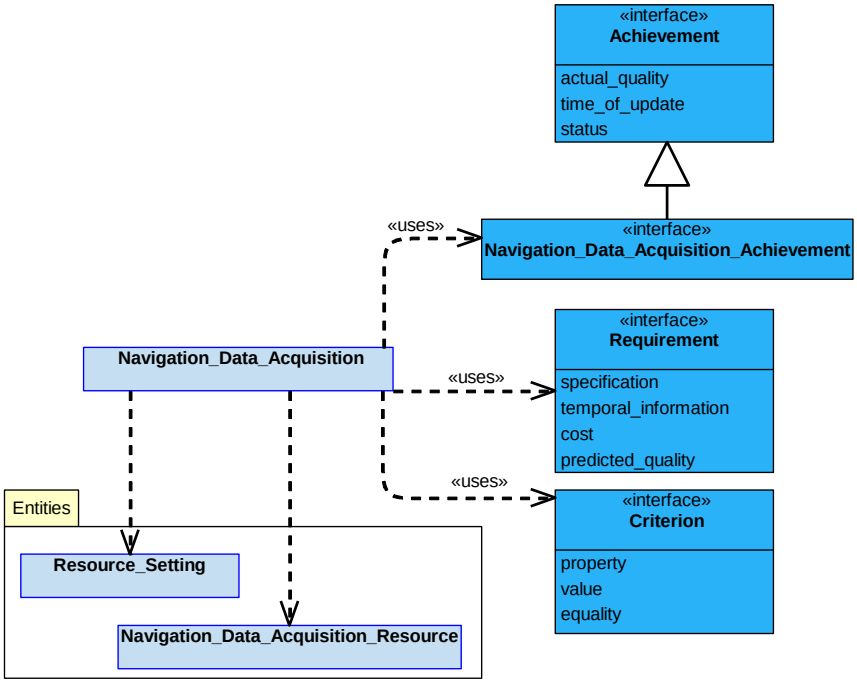

Figure 683: Navigation_Data_Acquisition Service Definition

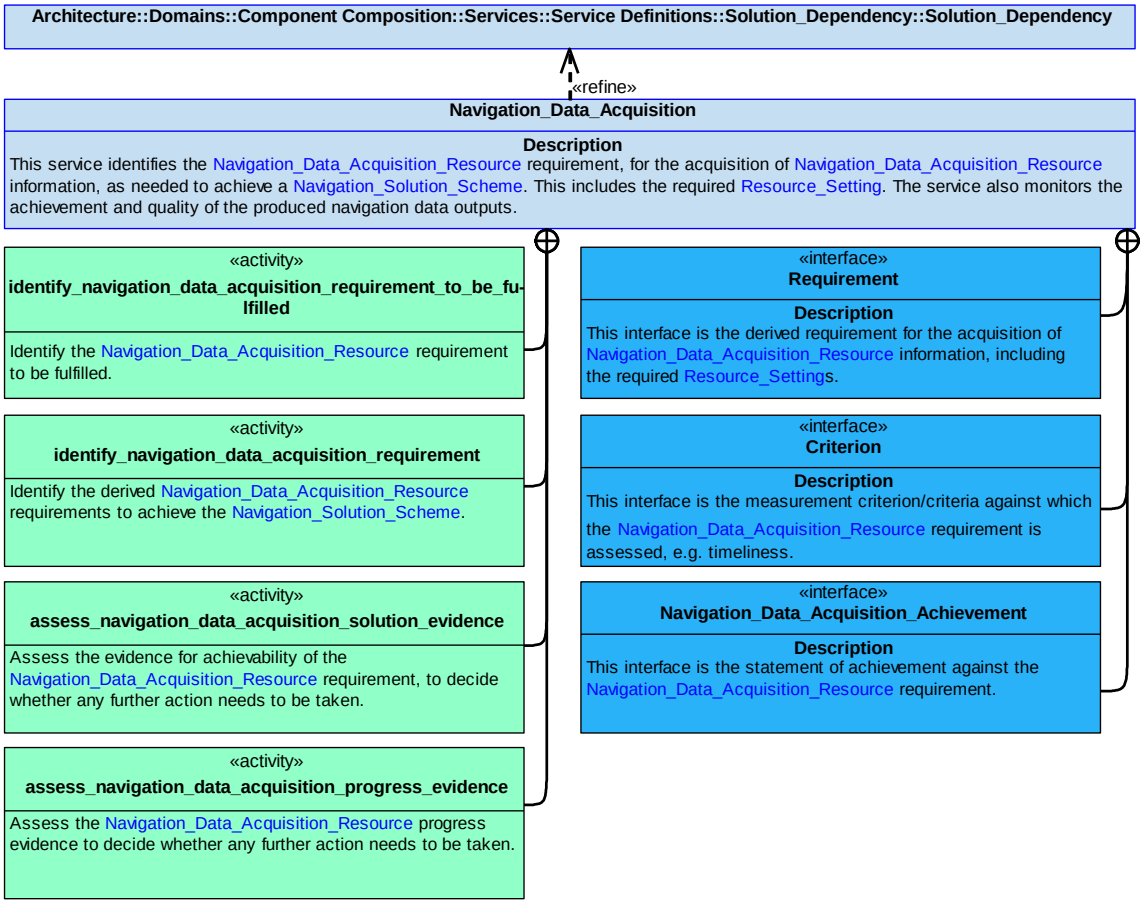

Figure 684: Navigation_Data_Acquisition Service Policy

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.

B.2.35.7.1.4 Navigation_Support_Activity

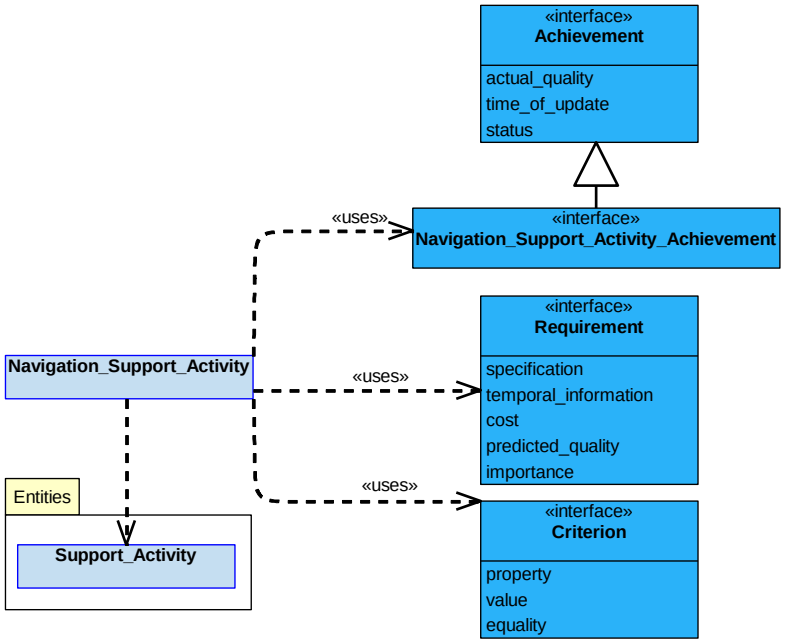

Figure 685: Navigation_Support_Activity Service Definition

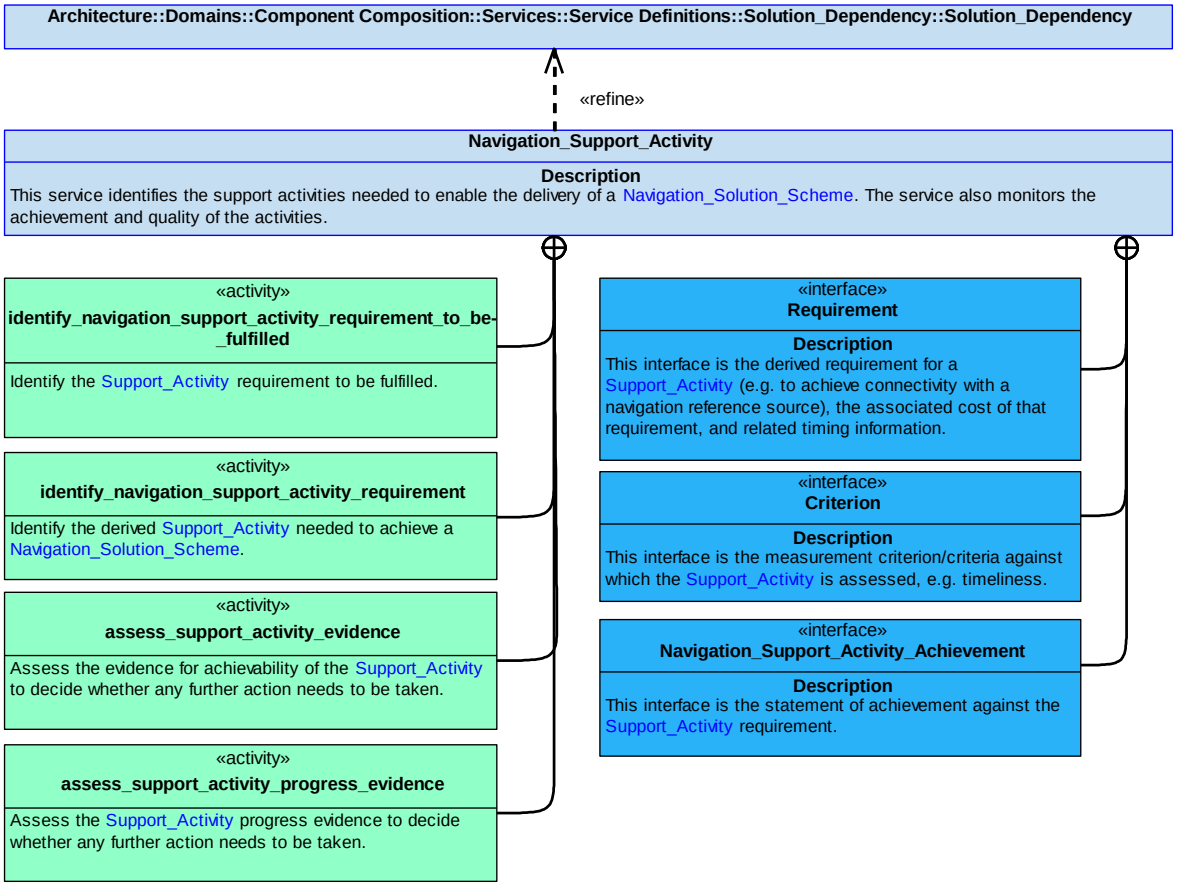

Figure 686: Navigation_Support_Activity Service Policy

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.

B.2.35.7.1.5 Supporting_Information

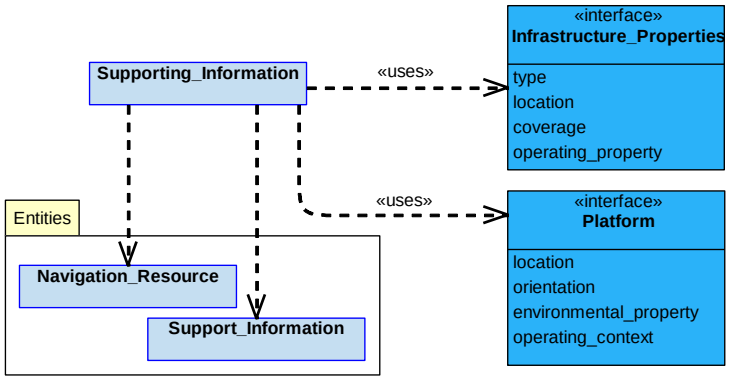

Figure 687: Supporting_Information Service Definition

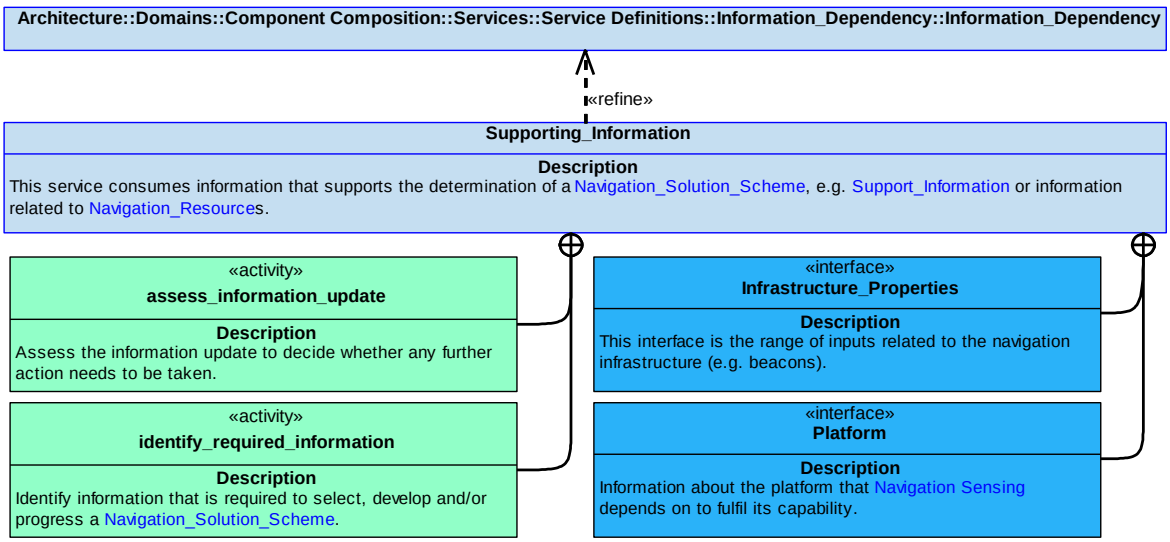

Figure 688: Supporting_Information Service Policy

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.

B.2.35.7.1.6 Constraint

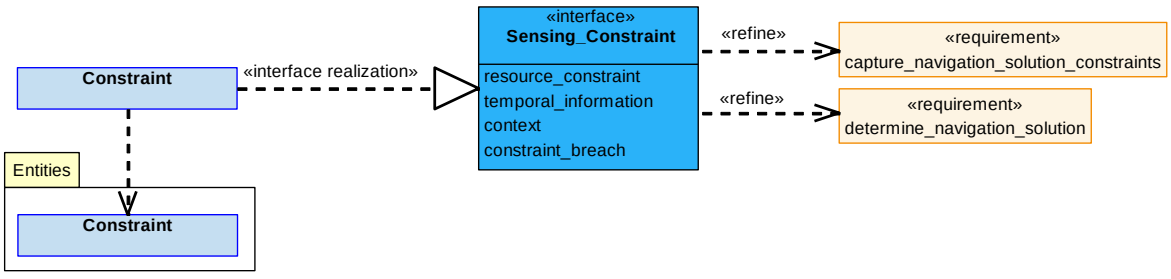

Figure 689: Constraint Service Definition

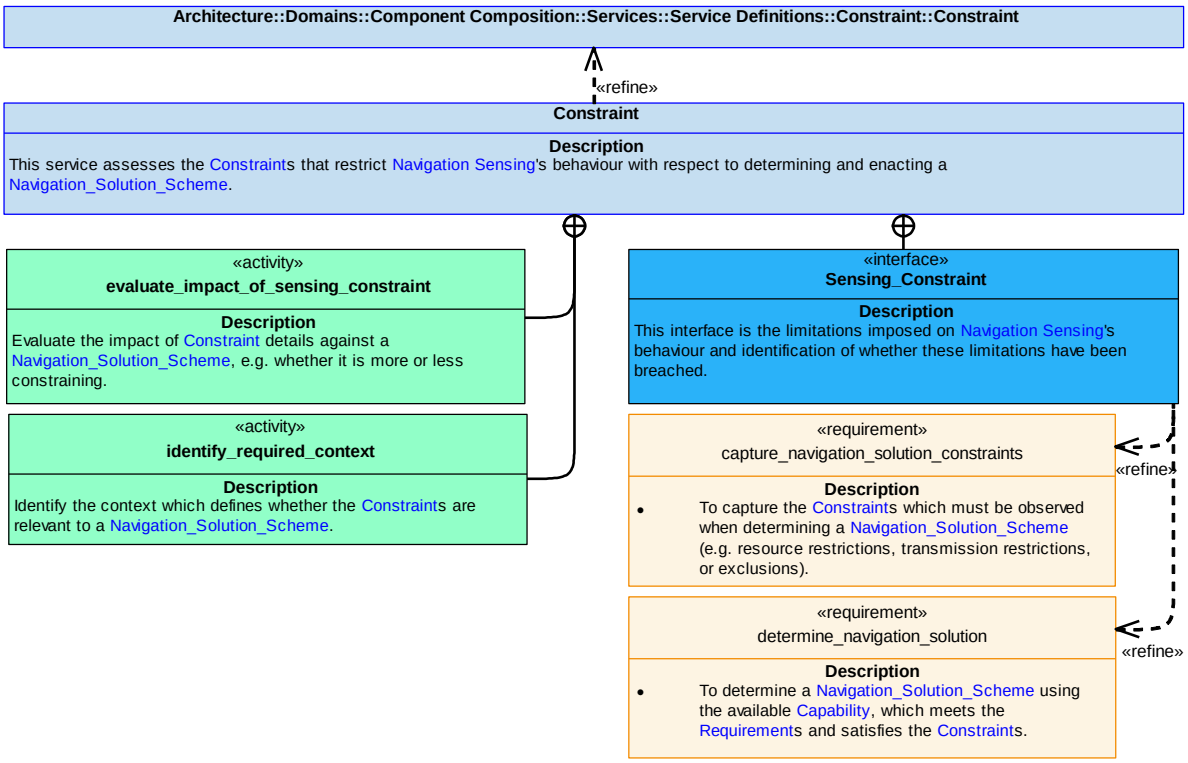

Figure 690: Constraint Service Policy

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.

B.2.35.7.1.7 Capability

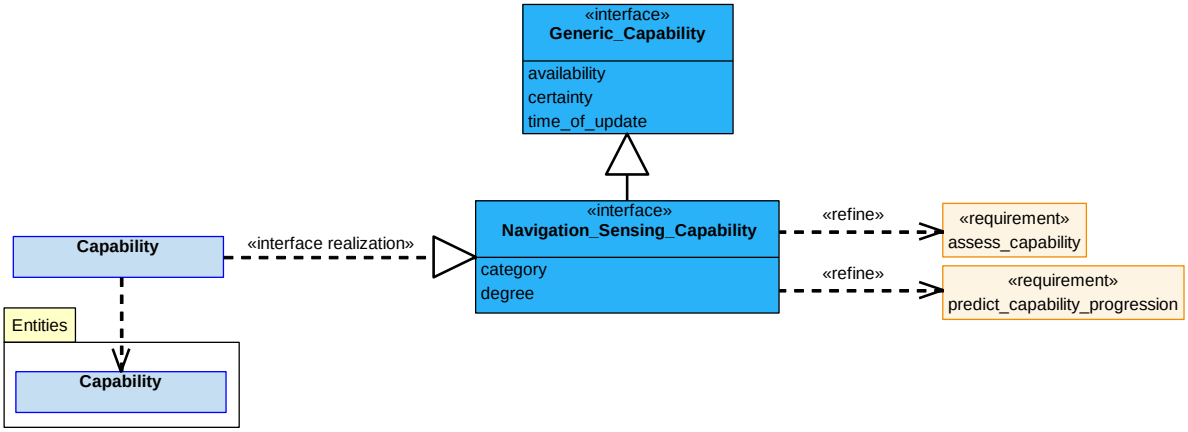

Figure 691: Capability Service Definition

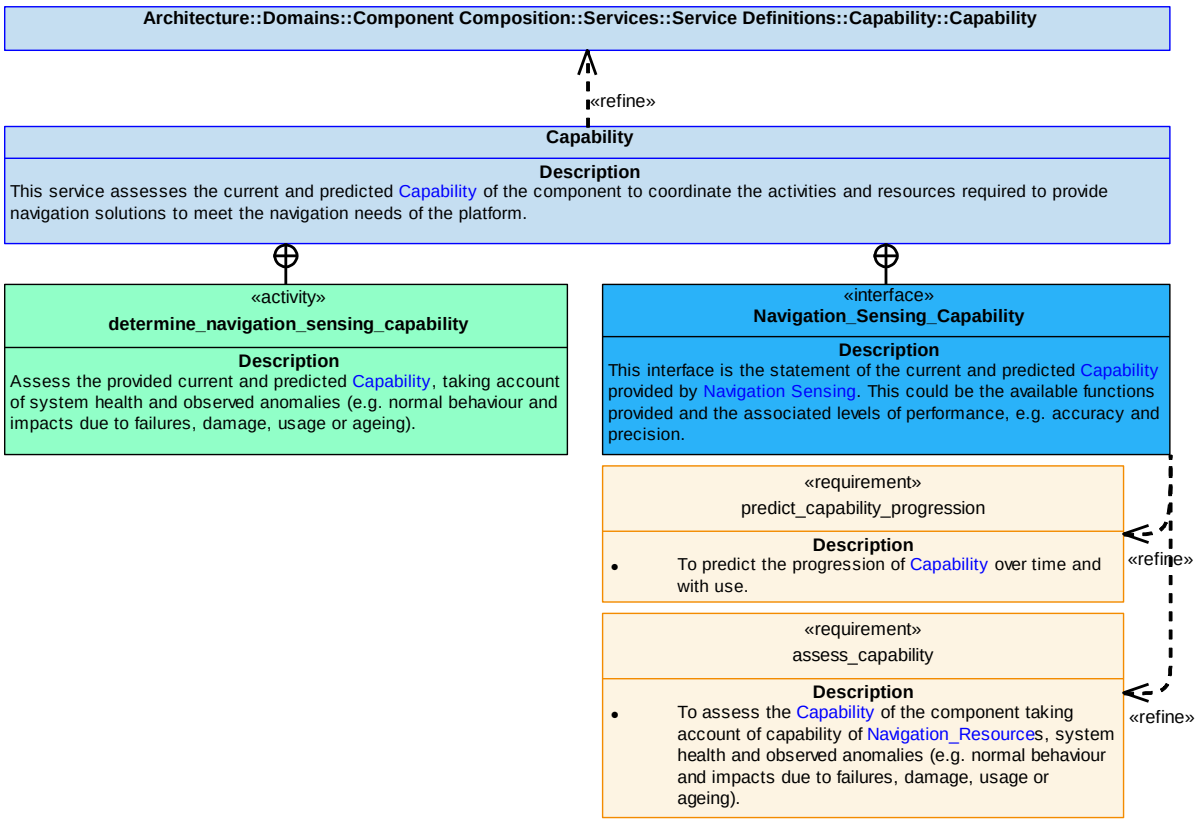

Figure 692: Capability Service Policy

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).

B.2.35.7.1.8 Capability_Evidence

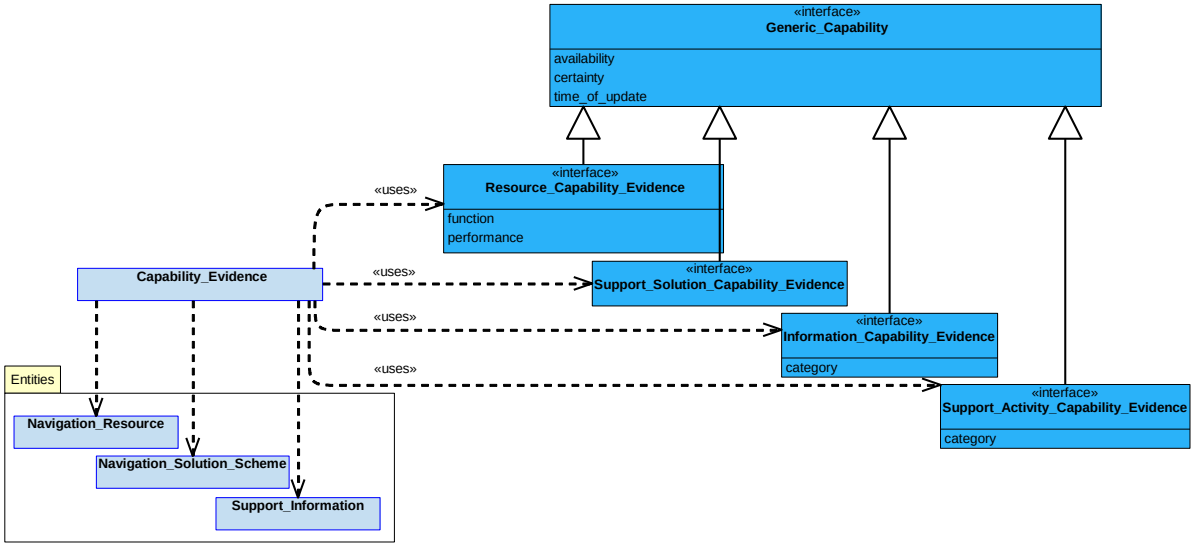

Figure 693: Capability_Evidence Service Definition

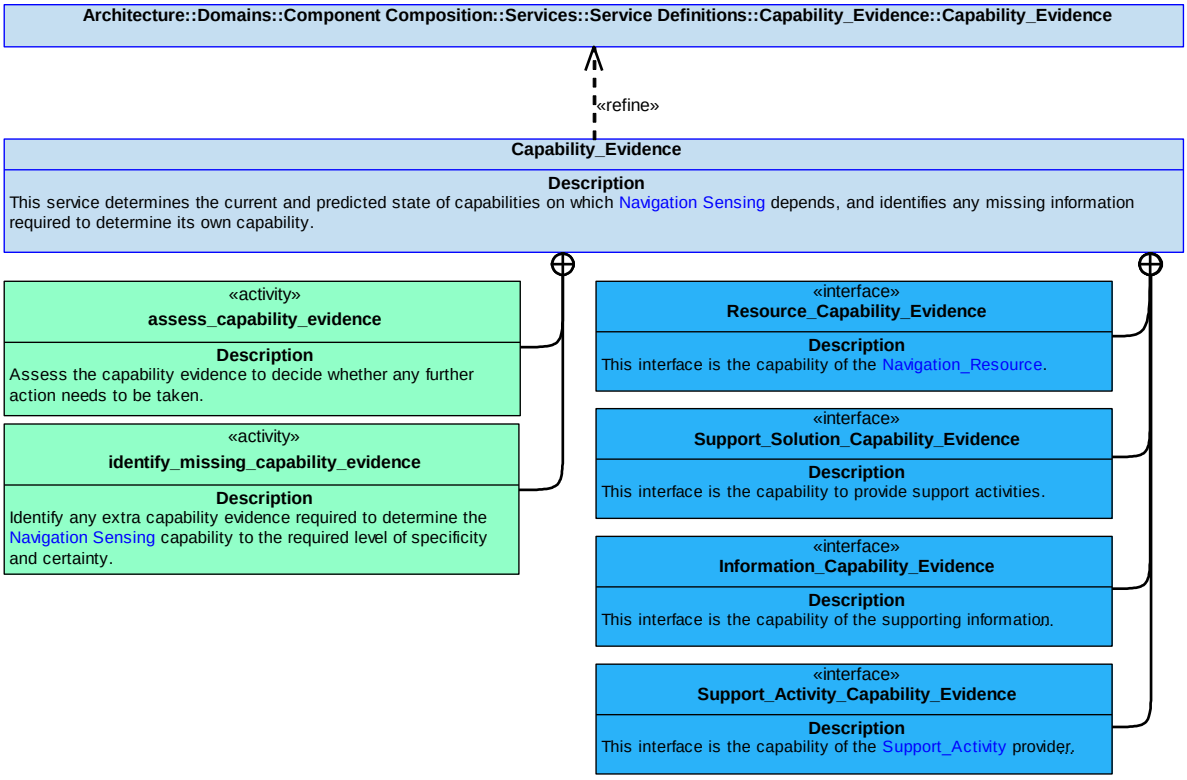

Figure 694: Capability_Evidence Service Policy

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 associated with this capability, 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.

B.2.35.7.2 Service Dependencies

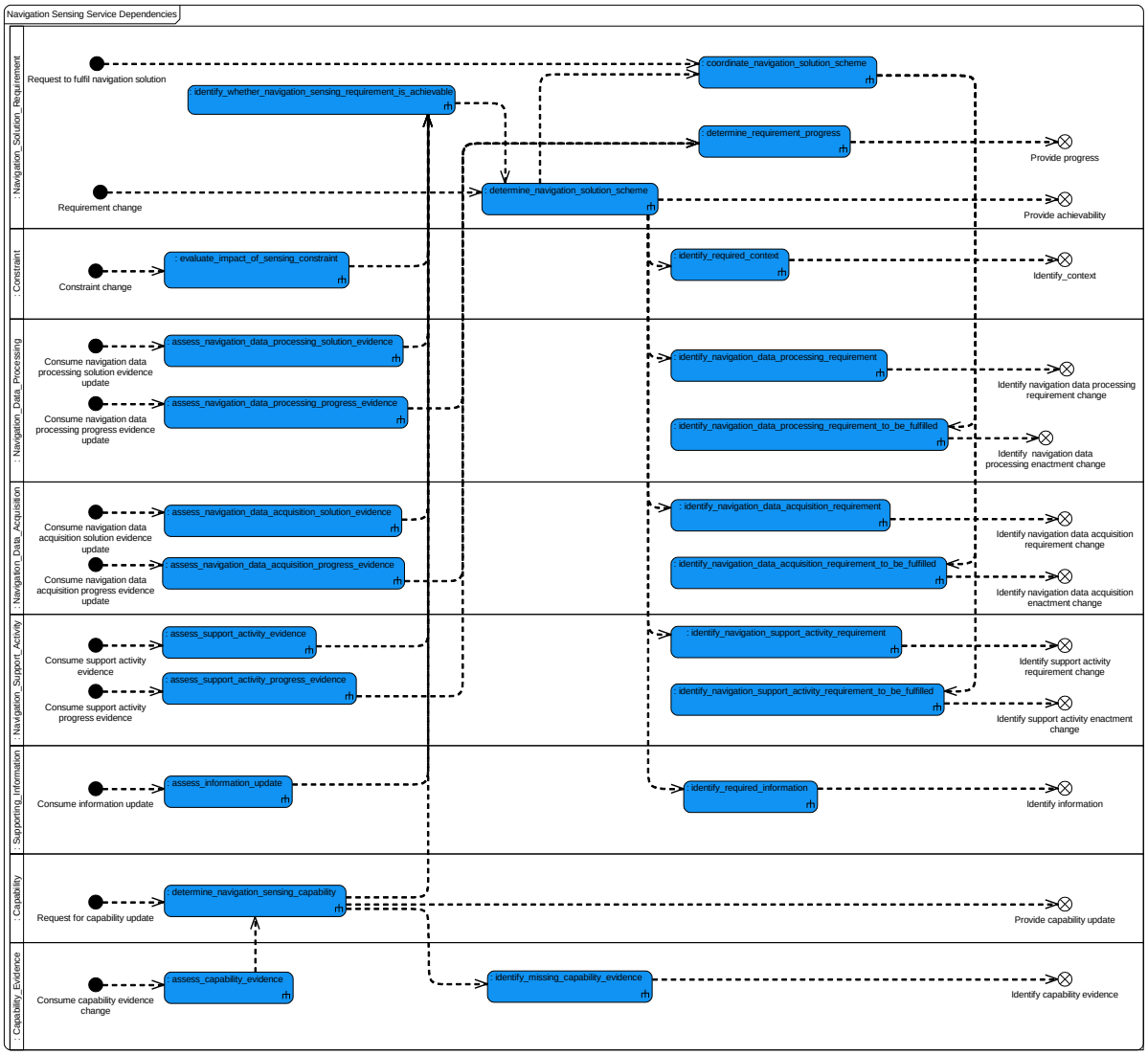

Figure 695: Navigation Sensing Service Dependencies

B.2.36 Network Routes

B.2.36.1 Role

The role of Network Routes is to deliver data over a network.

B.2.36.2 Overview

Control Architecture

Network Routes is a resource component as defined in the Control Architecture policy.

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).

B.2.36.3 Service Summary

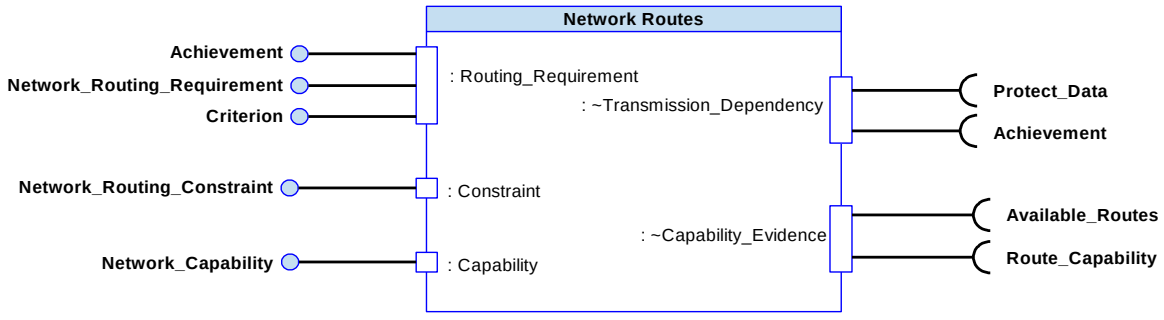

Figure 696: Network Routes Service Summary

B.2.36.4 Responsibilities

capture_requirements_for_network_routes

- To capture provided Transmission_Requirements.

capture_constraints_for_network_routes

- To capture provided Constraints for the use of network routes.

capture_measurement_criteria_for_network_routes

- To capture provided Measurement_Criterion/criteria.

manage_network_route_congestion

- To determine and manage the level of congestion within network Nodes.

determine_next_hop

- To select a Next_Hop_Solution from the Transmission_Routes.

maintain_network_configuration

- To maintain the configuration of the network.

determine_network_route_quality

- To determine the Transmission_Quality - including but not limited to bandwidth utilisation, delay over a route and packet error rate.

convey_data

- To convey data over a Next_Hop_Solution.

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).

predict_network_routes_capability_progression

- To predict the progression of the capability of network routes over time and with use.

identify_missing_information

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

determine_solution_feasibility

To determine if a planned or on-going Next_Hop_Solution remains feasible given current capability and Constraints.

B.2.36.5 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.

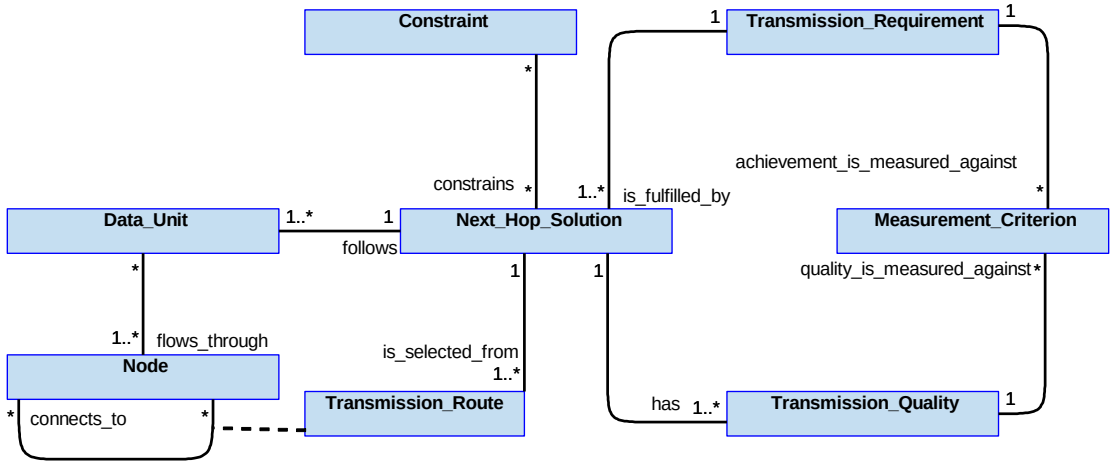

Figure 697: Network Routes Semantics

B.2.36.5.1 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

An individual element of Data to be transmitted.

Measurement_Criterion

A measurement criterion (e.g. bandwidth) against which a transmission route will be tested.

Next_Hop_Solution

The selected hop to the next node to achieve the end-to-end route across the network.

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.

B.2.36.6 Design Rationale

B.2.36.6.1 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).

B.2.36.6.2 Design Considerations

Directly Applicable Policies

These policies 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 methods.

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 forwarding (the next hop for a specific stream of traffic) information can be considered to be separate. This allows for the possibility of a Network approach where the routing information is centralised and the forwarding information is distributed.
- Network Routes can use a predetermined route, or a dynamically discovered route.

B.2.36.6.3 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 data being transferred, 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 data transfer would result in loss of "useable" data, for which DAL C is appropriate, as justified in the previous paragraph.

B.2.36.6.4 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**.

Note: The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about secure communications.

B.2.36.7 Services

B.2.36.7.1 Service Definitions

B.2.36.7.1.1 Routing_Requirement

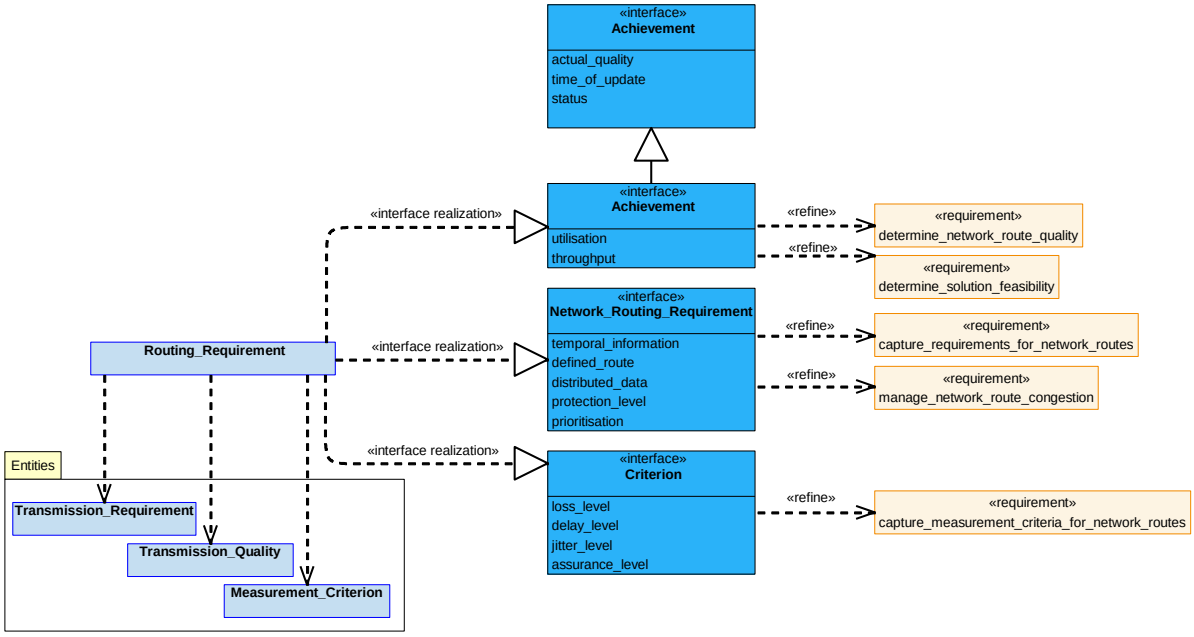

Figure 698: Routing_Requirement Service Definition

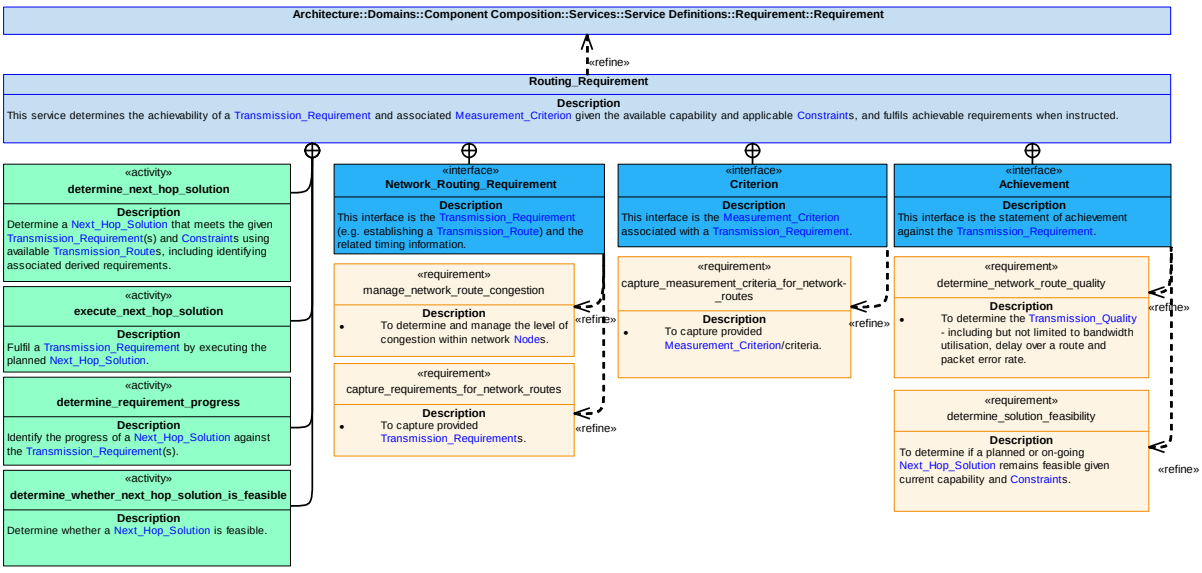

Figure 699: Routing_Requirement Service Policy

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_Solution is feasible.

determine_next_hop_solution

Determine a Next_Hop_Solution that meets the given Transmission_Requirement(s) and Constraints using available Transmission_Routes, including identifying associated derived requirements.

execute_next_hop_solution

Fulfil a Transmission_Requirement by executing the planned Next_Hop_Solution.

determine_requirement_progress

Identify the progress of a Next_Hop_Solution against the Transmission_Requirement(s).

B.2.36.7.1.2 Transmission_Dependency

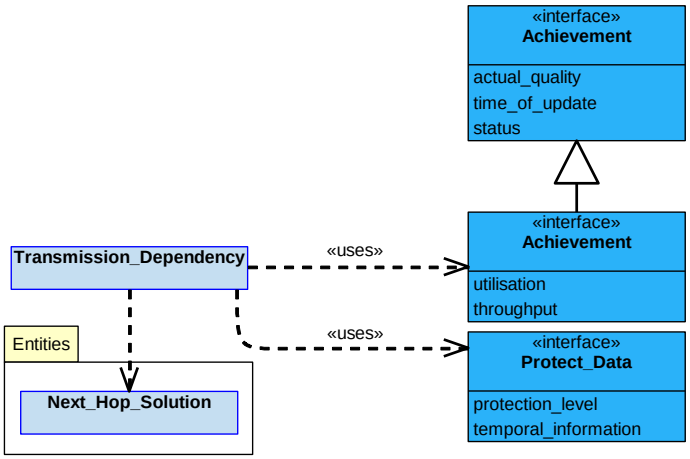

Figure 700: Transmission_Dependency Service Definition

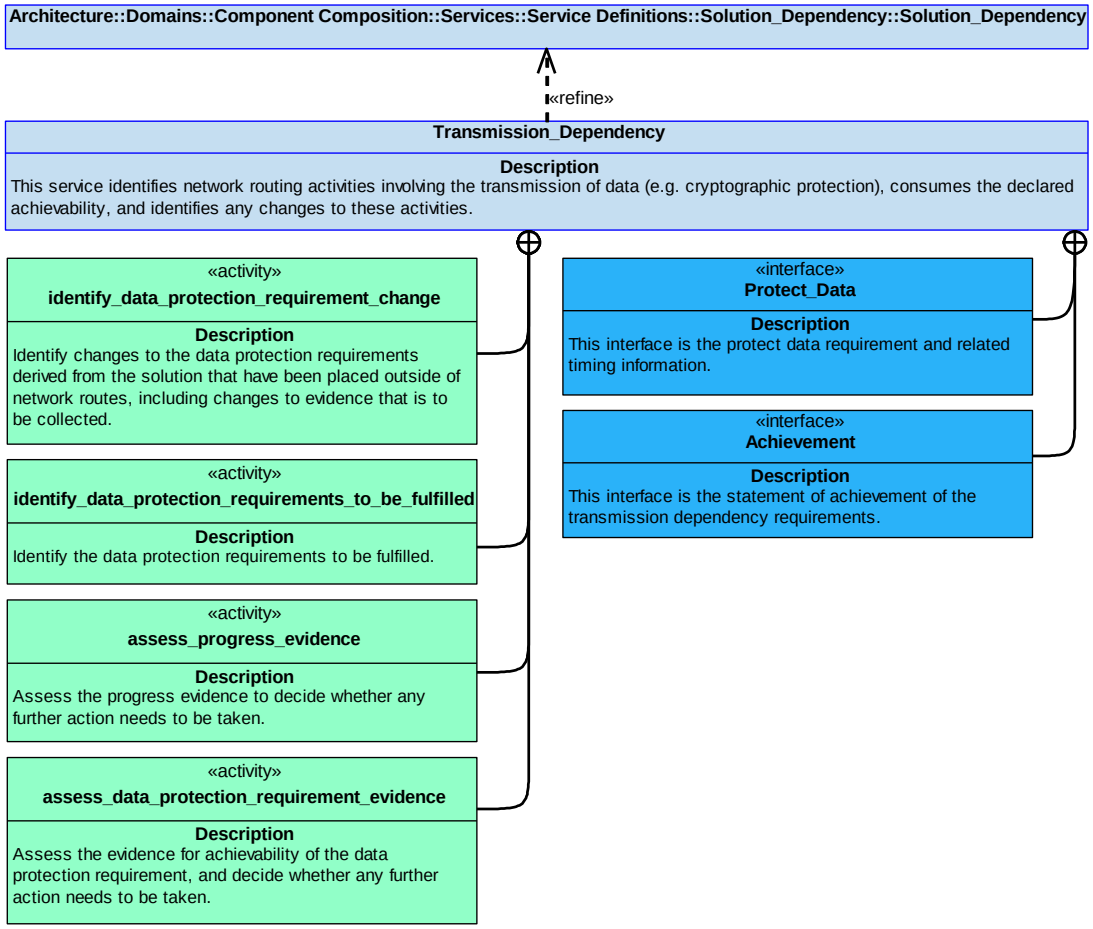

Figure 701: Transmission_Dependency Service Policy

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.

Interfaces

Protect_Data

This interface is the protect data 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.

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_data_protection_requirement_evidence

Assess the evidence for achievability of the data protection requirement, 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.

B.2.36.7.1.3 Constraint

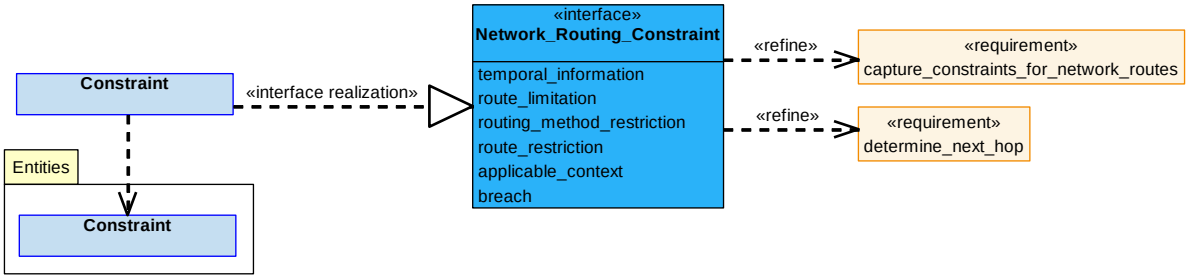

Figure 702: Constraint Service Definition

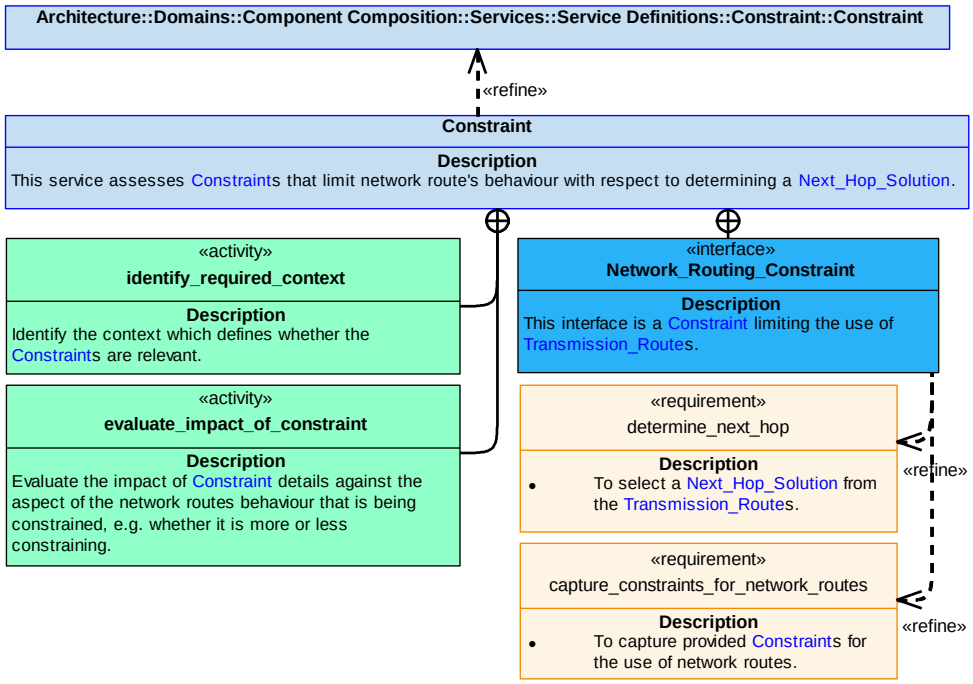

Figure 703: Constraint Service Policy

Constraint

This service assesses Constraints that limit network route's behaviour with respect to determining a Next_Hop_Solution.

Interface

Network_Routing_Constraint

This interface is a Constraint limiting the use of Transmission_Routes.

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 Transmission_Route that is not allowed to be transmitted across, this could be due to it not having the correct protection level.
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.

B.2.36.7.1.4 Capability

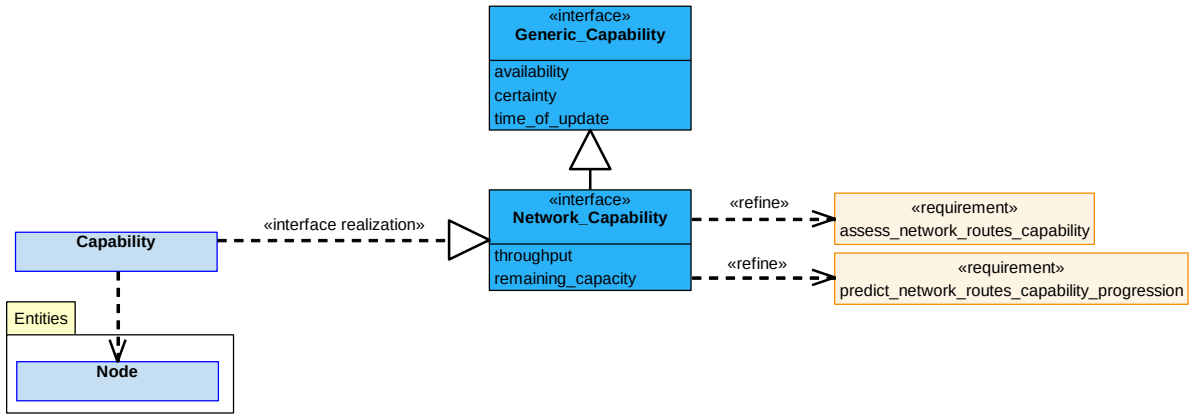

Figure 704: Capability Service Definition

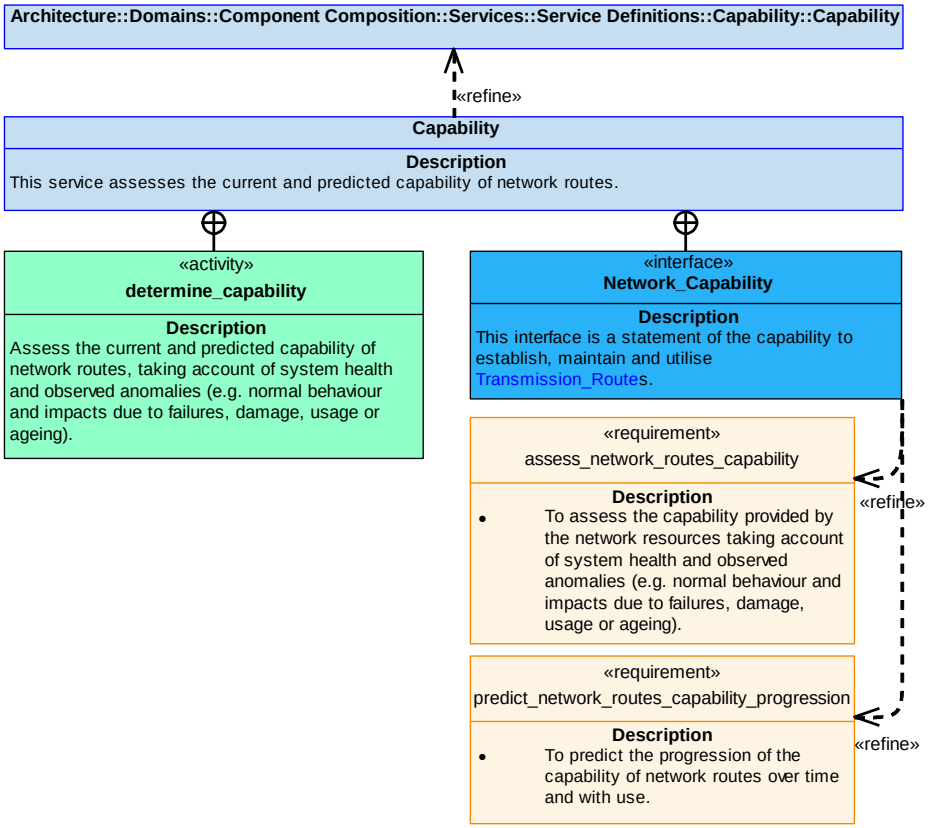

Figure 705: Capability Service Policy

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).

B.2.36.7.1.5 Capability_Evidence

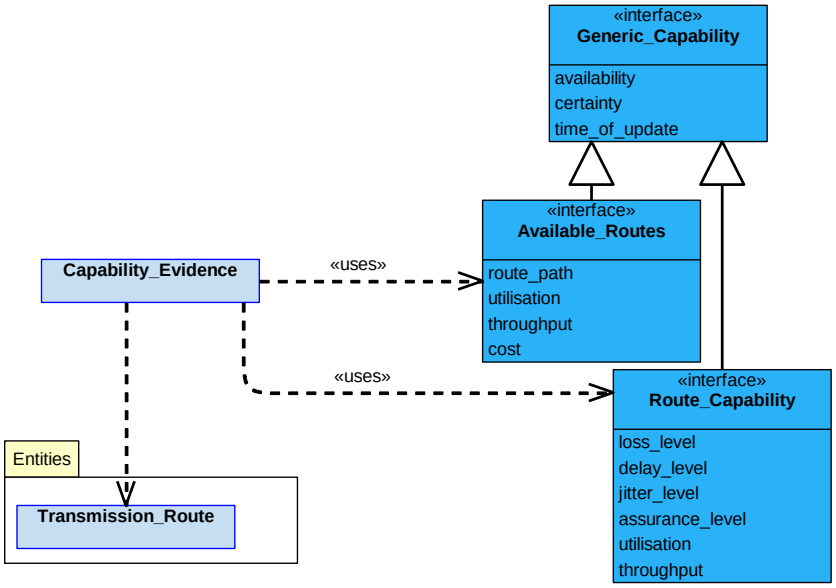

Figure 706: Capability_Evidence Service Definition

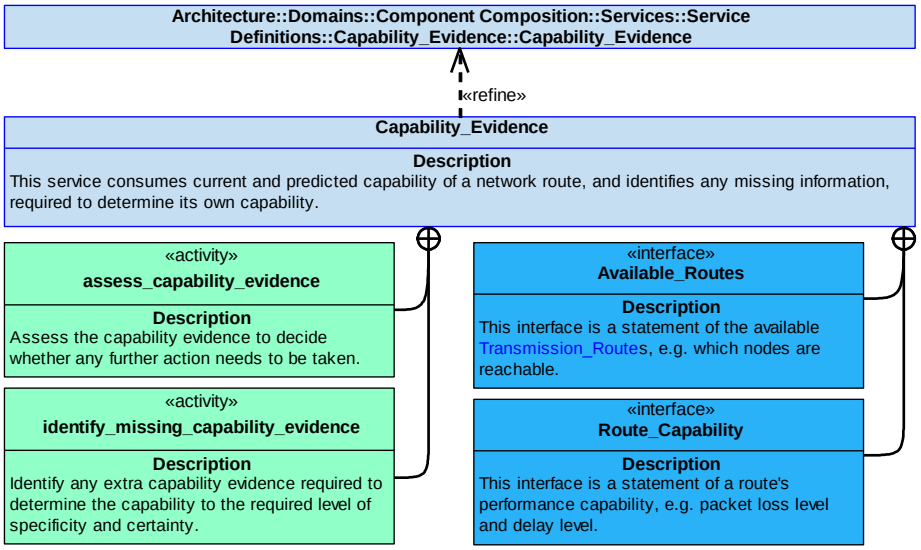

Figure 707: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability of a network route, and identifies any missing information, required to determine its own capability.

Interfaces

Available_Routes

This interface is a statement of the available Transmission_Routes, e.g. which nodes are reachable.

Attributes

route_path	A Transmission_Route that is available, such as end points that can be reached.
utilisation	The amount of traffic presented to the Transmission_Route.
throughput	The theoretical amount of traffic that can be supported on the Transmission_Route.
cost	The cost metric of a hop within a Transmission_Route.

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.

B.2.36.7.2 Service Dependencies

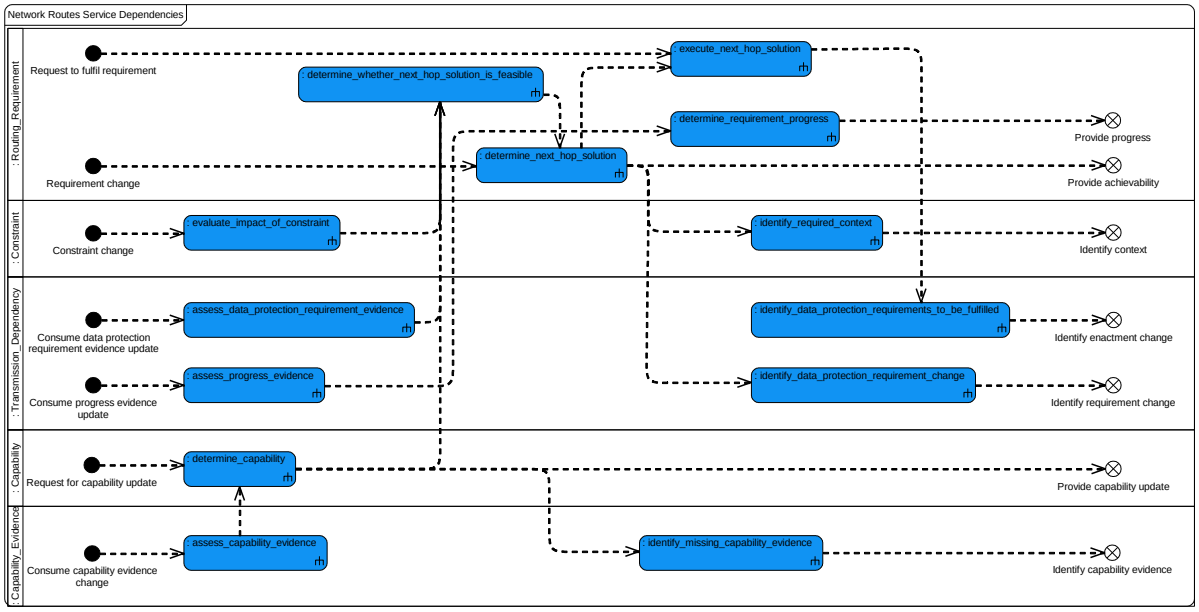

Figure 708: Network Routes Service Dependencies

B.2.37 Networks

B.2.37.1 Role

The role of Networks is to set-up, manage and optimise communications networks.

B.2.37.2 Overview

Control Architecture

Networks is an action component as defined in the Control Architecture policy.

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.

B.2.37.3 Service Summary

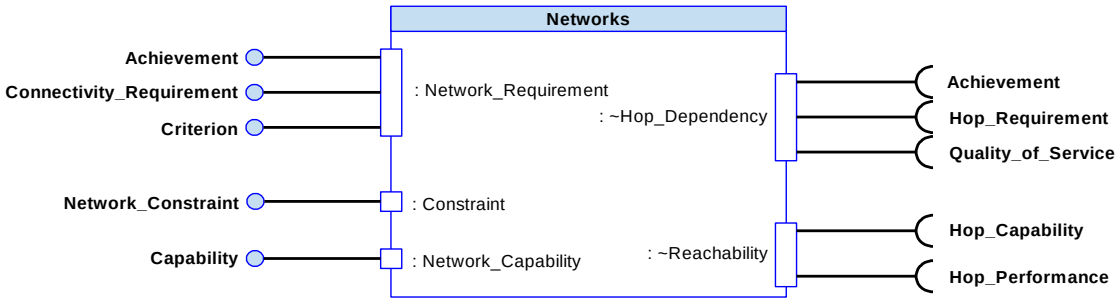

Figure 709: Networks Service Summary

B.2.37.4 Responsibilities

capture_network_requirements

- To capture given network Connectivity_Requirements (e.g. the need to establish a new Connection).

capture_network_constraints

- To capture given network Constraints (e.g. integrity, security, or safety).

capture_network_measurement_criteria

- To capture given Measurement_Criterion/criteria (e.g. bandwidth, latency, or reach) for networks.

determine_network_capability

- To determine the Capability to provide networks using available Network_Resources, taking into account observed anomalies.

predict_network_capability

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

identify_missing_capability_information

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

determine_network_solution

- Determine a network Topology and Connections that support the given network Connectivity_Requirement within Network_Resource limits and the given Constraints.

determine_network_solution_quality

- To determine the quality of a Connection.

identify_network_pre-conditions

- To identify Pre-Conditions required for a Connection.

maintain_network

- To establish and maintain a network Topology by management of Network_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_change

- To identify divergence from the expected Topology or network performance.

identify_network_solution_in_progress_remains_feasible

- To identify whether a network Connection in progress remains feasible against particular Connectivity_Requirements and Measurement_Criterion/criteria given current resources.

B.2.37.5 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).

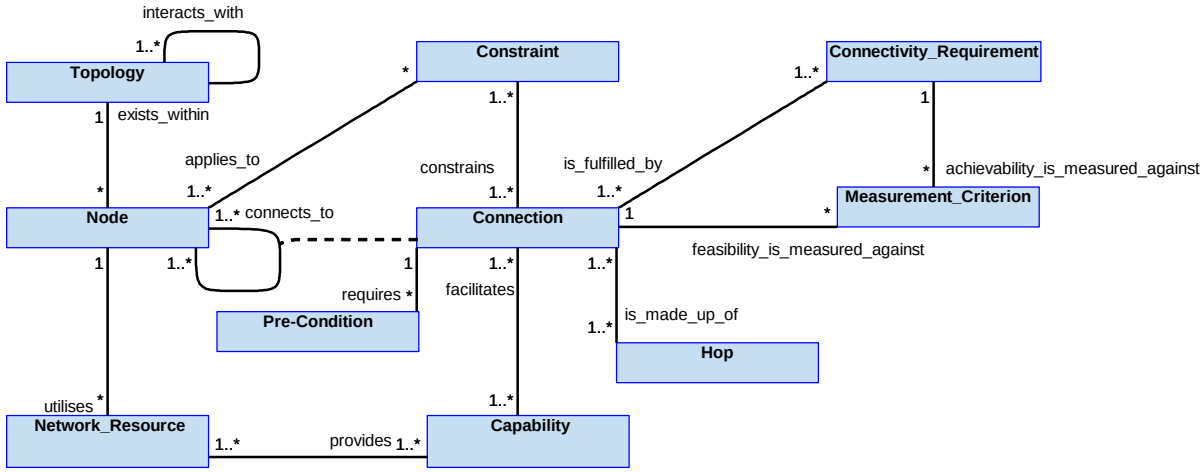

Figure 710: Networks Semantics

B.2.37.5.1 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 or indirect (e.g. as either an end-to-end connection or a series of 'hops').

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.

Measurement_Criterion

A Measurement_Criterion (e.g. bandwidth, utilisation, latency, reachability, or speed of establishment) for networks.

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 is allowed (e.g. the same deployment or the same security domain).

B.2.37.6 Design Rationale

B.2.37.6.1 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.

B.2.37.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy. 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.

B.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.
- This component does not handle the data being transferred. Therefore, this component cannot corrupt data.

B.2.37.6.4 Security Considerations

The indicative security classification is SCEO.

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.

Note: The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about secure communications.

B.2.37.7 Services

B.2.37.7.1 Service Definitions

B.2.37.7.1.1 Network_Requirement

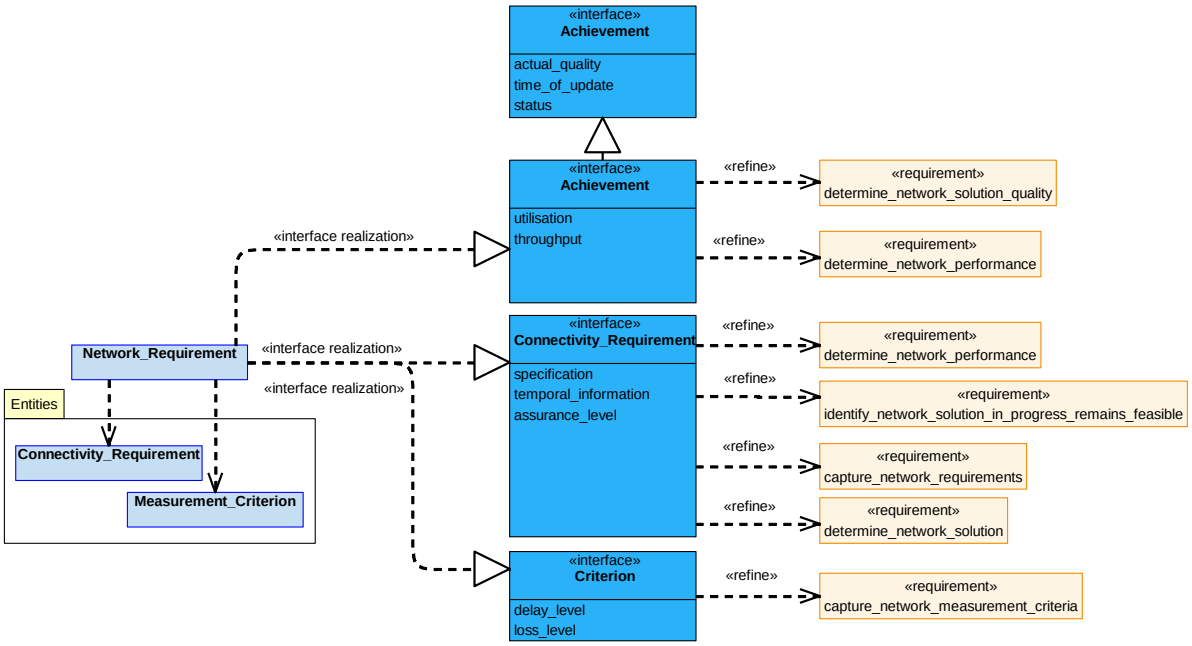

Figure 711: Network_Requirement Service Definition

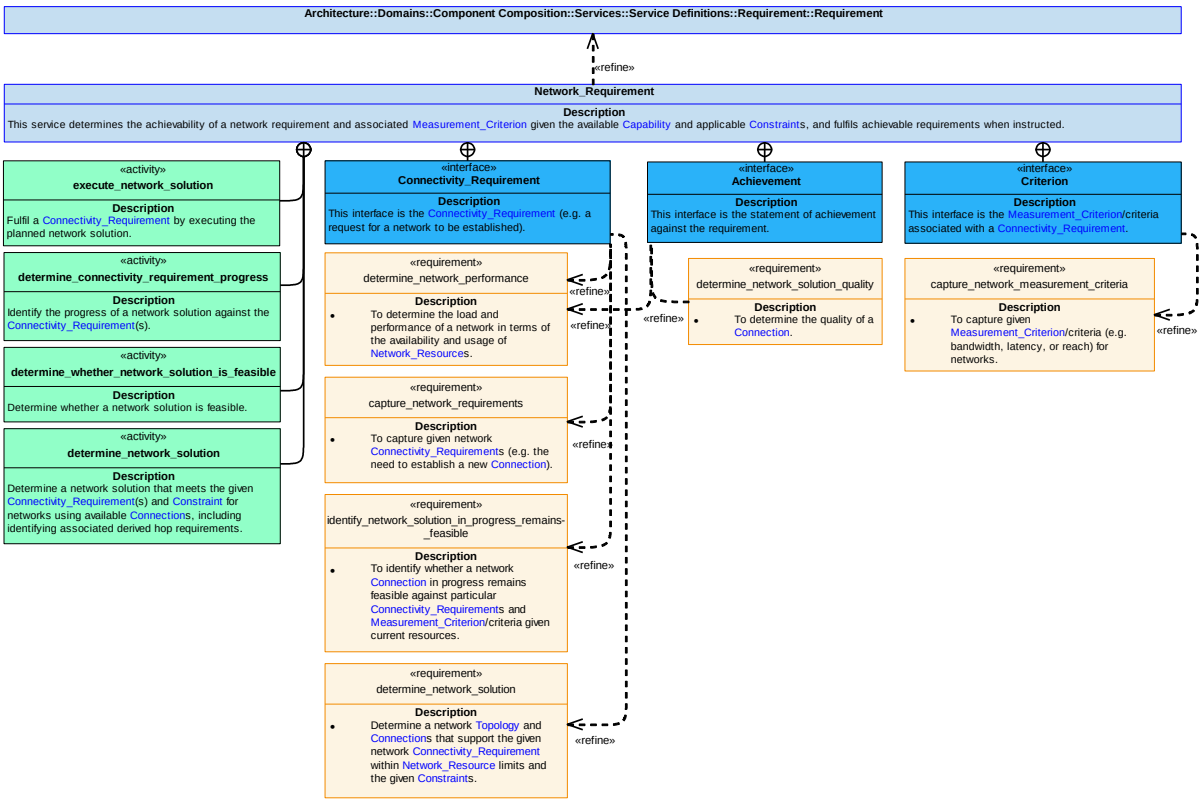

Figure 712: Network_Requirement Service Policy

Network_Requirement

This service determines the achievability of a network requirement and associated Measurement_Criterion given the available Capability and applicable Constraints, and fulfils achievable requirements when instructed.

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 of a network, 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 Measurement_Criterion/criteria 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 Constraint 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.

B.2.37.7.1.2 Hop_Dependency

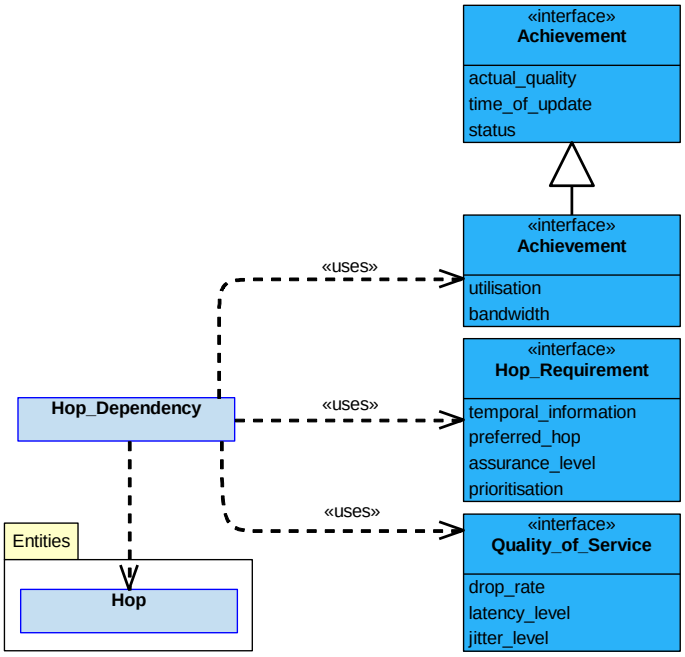

Figure 713: Hop_Dependency Service Definition

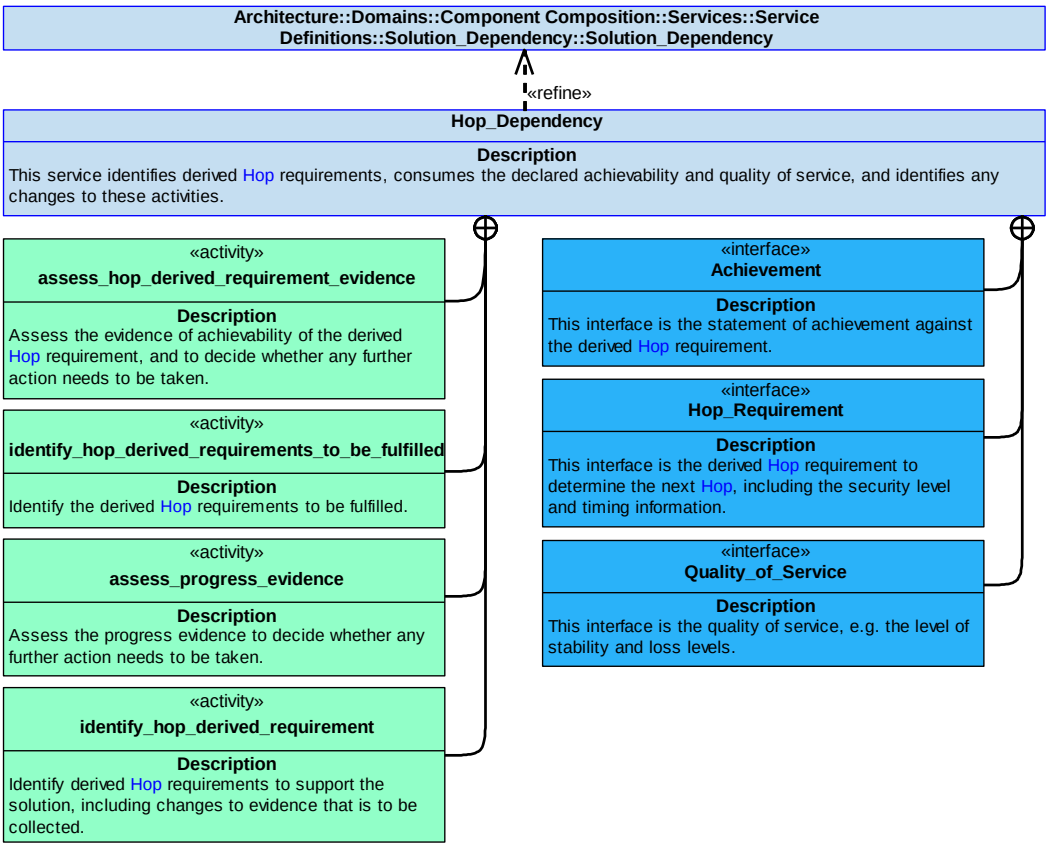

Figure 714: Hop_Dependency Service Policy

Hop_Dependency

This service identifies derived Hop requirements, consumes the declared achievability and quality of service, and identifies any changes to these activities.

Interfaces

Achievement

This interface is the statement of achievement against the derived Hop requirement.

Attributes

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.
preferred_hop	A preferred Hop to be applied.
assurance_level	The level of assurance of a Hop, e.g. whether the Hop is approved for safety critical or control traffic.
prioritisation	The priority of the 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.

B.2.37.7.1.3 Constraint

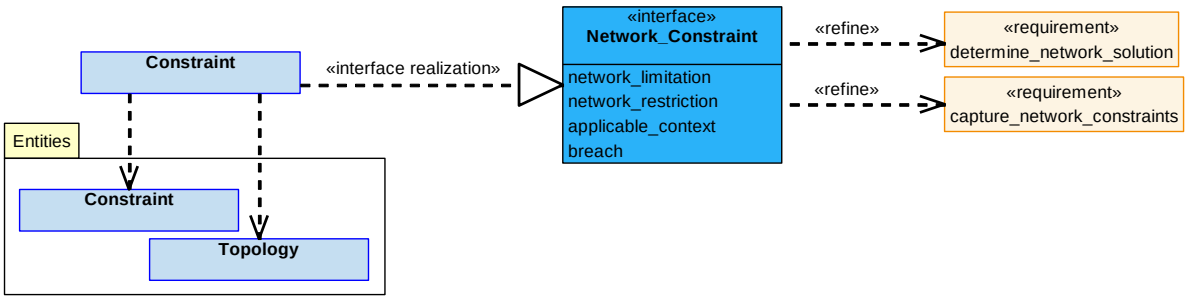

Figure 715: Constraint Service Definition

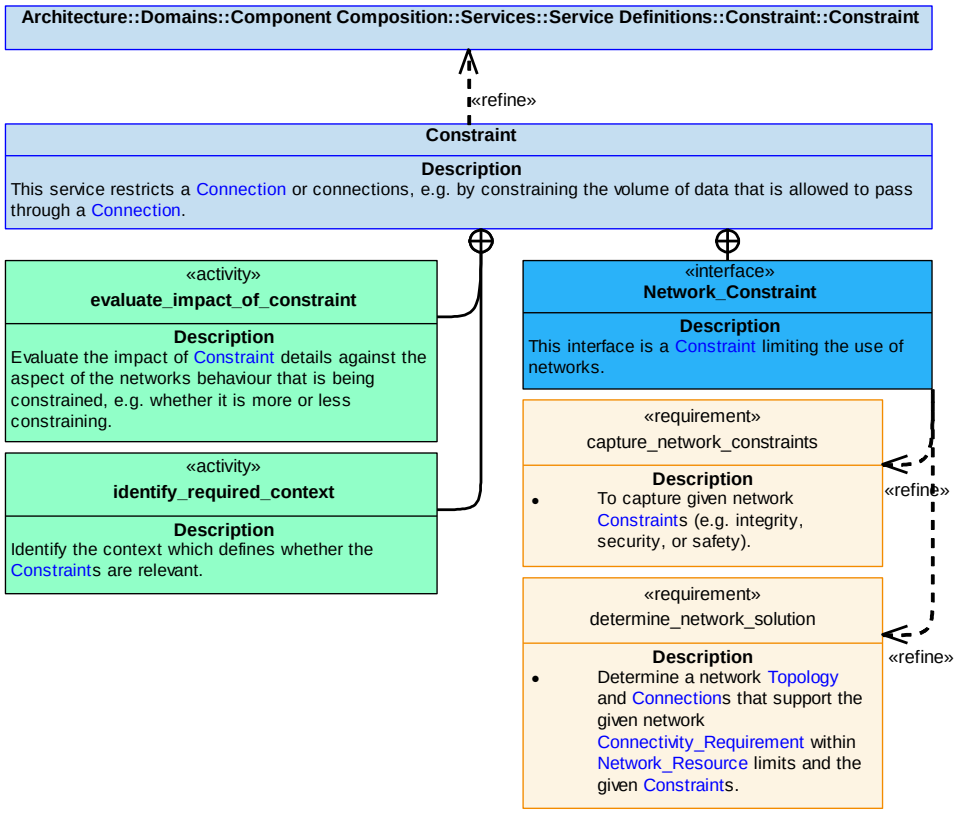

Figure 716: Constraint Service Policy

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 what network types can be used.
network_restriction	A network, or section of a network, that is not allowed to be utilised, e.g. a network 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.

B.2.37.7.1.4 Network_Capability

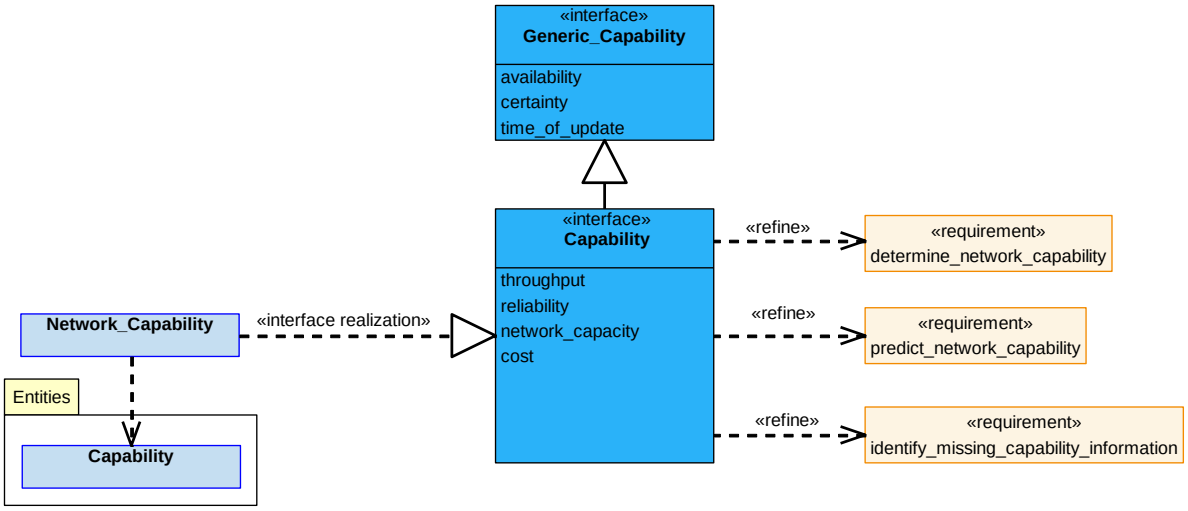

Figure 717: Network_Capability Service Definition

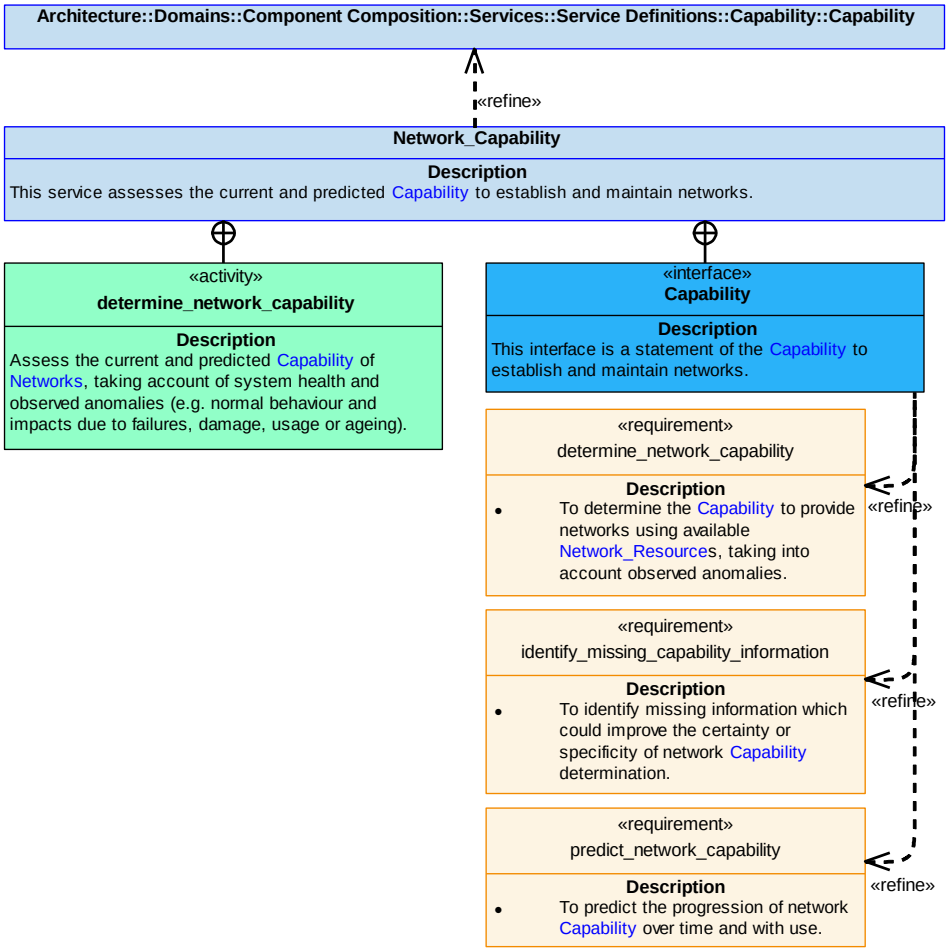

Figure 718: Network_Capability Service Policy

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 a network, e.g. whether a Connection will terminate unexpectedly.
network_capacity	The amount of the network available to be used, e.g. available bandwidth.
cost	The cost of the network Capability, inclusive of factors such as loss and delay.

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).

B.2.37.7.1.5 Reachability

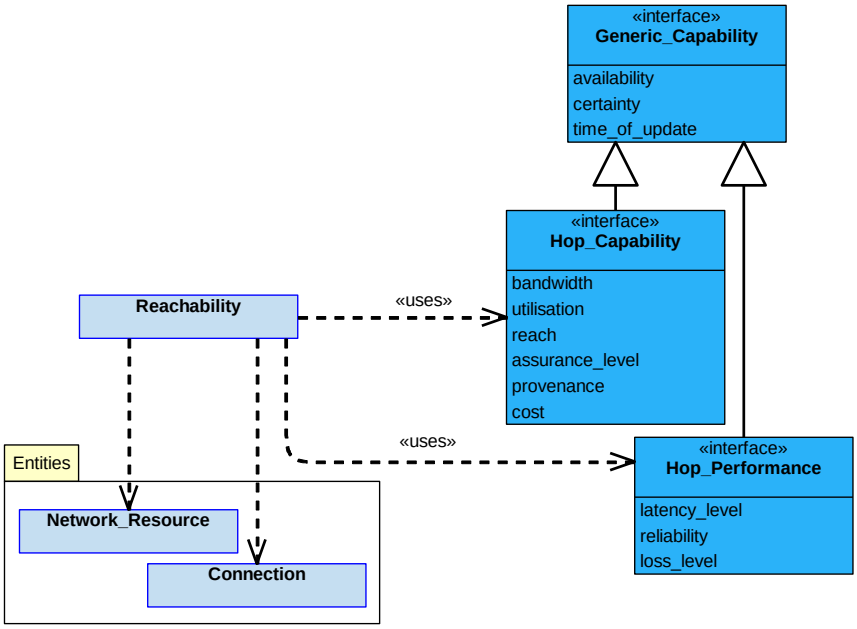

Figure 719: Reachability Service Definition

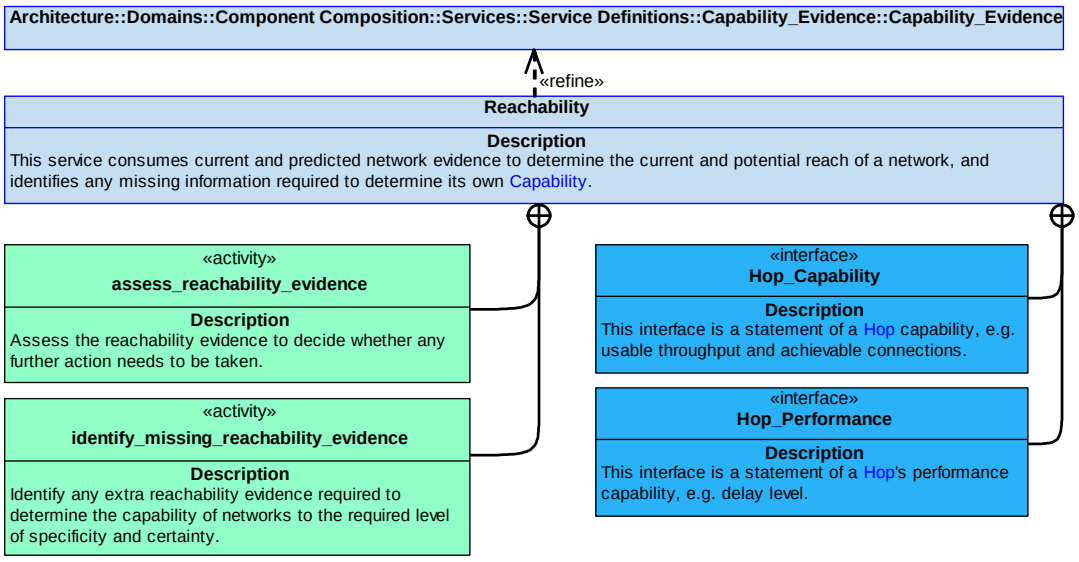

Figure 720: Reachability Service Policy

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.

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 capability, 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.

B.2.37.7.2 Service Dependencies

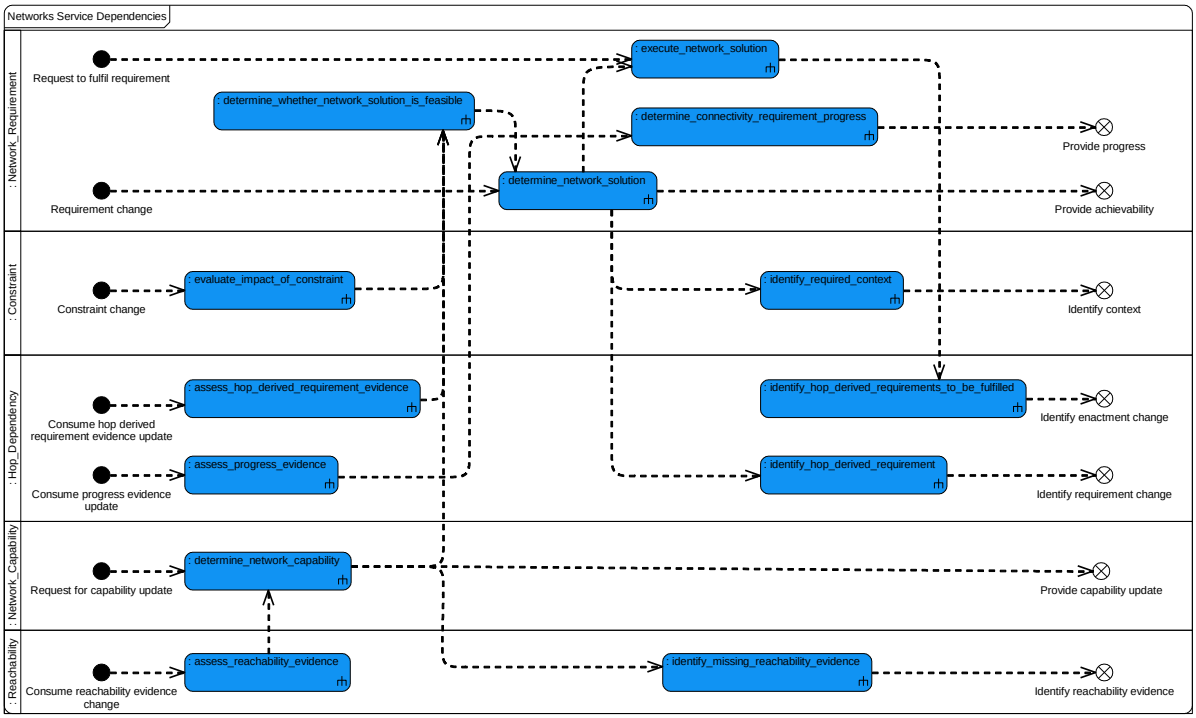

Figure 721: Networks Service Dependencies

B.2.38 Objectives

B.2.38.1 Role

The role of Objectives is to coordinate the achievement of Objectives through the execution of Tasks.

B.2.38.2 Overview

Control Architecture

Objectives is the only component in the Objective Layer, as defined in the Control Architecture policy.

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 may be 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.

B.2.38.3 Service Summary

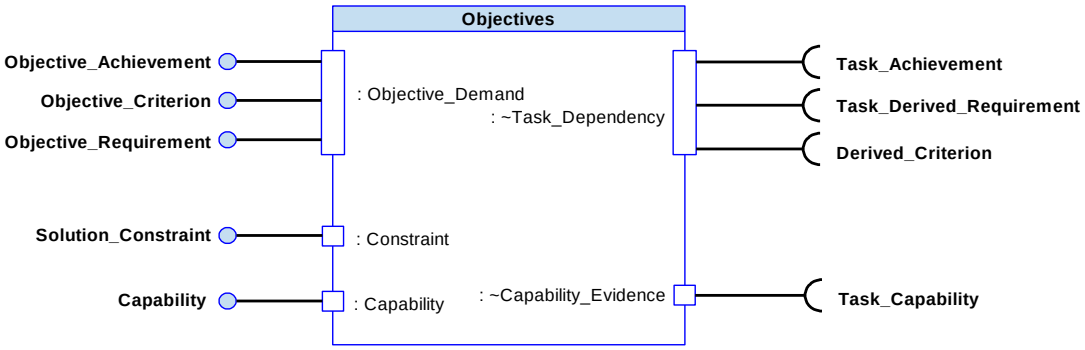

Figure 722: Objectives Service Summary

B.2.38.4 Responsibilities

capture_requirements

- To capture Objectives (e.g. including objective type, timing, imperative for success and risk profile).

capture_constraints

- To capture Constraints on how an Objective may be achieved (e.g. operator imposed restrictions and Rules of Engagement).

capture_measurement_criteria

- To capture given quality requirements for Objective_Solutions (e.g. quality, risk and robustness).

determine_capability

- To determine Flight_Capability based on Participant_Capability, taking into account observed anomalies.

determine_implementation_scheme

- To determine an Objective_Solution to achieve an Objective.

evaluate_implementation_scheme_cost

- To evaluate the costs of a planned Objective_Solution against given measurement criteria.

identify_dependencies

- To identify the solution dependencies required to support the delivery of an Objective_Solution.

satisfy_dependencies_between_tasks

- To satisfy the solution dependencies between 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.

predict_capability_progression

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

identify_whether_requirement_remains_achievable

- To identify whether an Objective is still achievable given current Flight_Capability and Constraints.

determine_predicted_quality_of_solution

- To evaluate the predicted quality of a proposed Objective_Solution against given quality requirements.

identify_missing_information

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

B.2.38.5 Subject Matter Semantics

The subject matter of Objectives is the coordination of Tasks to fulfil Objectives.

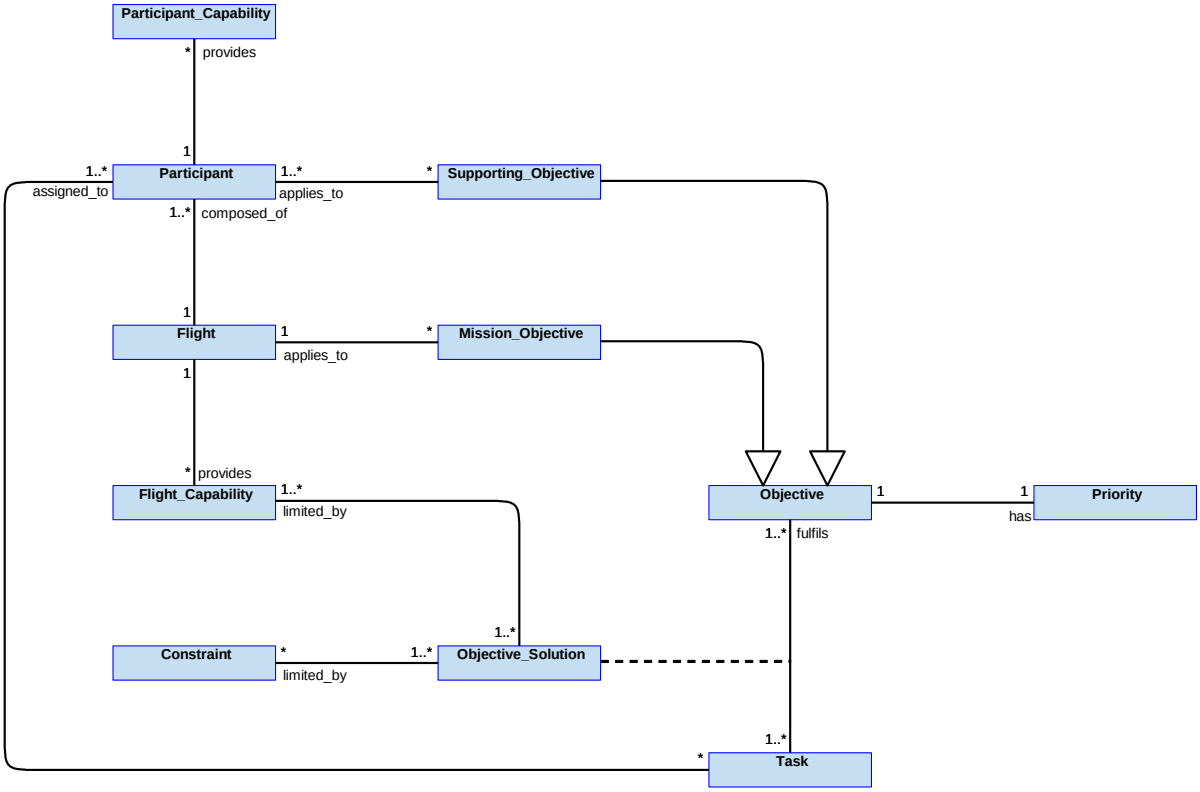

Figure 723: Objectives Semantics

B.2.38.5.1 Entities

Constraint

A restriction on the ways in which an Objective may be carried out that must not be contravened (e.g. operator imposed restrictions, Rules of Engagement).

Flight

A collection of one or more aircraft that are cooperating to perform 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 (CAS), 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.

B.2.38.6 Design Rationale

B.2.38.6.1 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.

B.2.38.6.2 Design Considerations

Directly Applicable Policies

These policies 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 deployment 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. No extensions have been defined within the PRA as these will be highly dependent on Mission Context, Operational Context and the Air Platform types and capabilities deployed.

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.
- 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.

B.2.38.6.3 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. the Contingency extension to 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.

B.2.38.6.4 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 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.

B.2.38.7 Services

B.2.38.7.1 Service Definitions

B.2.38.7.1.1 Objective_Demand

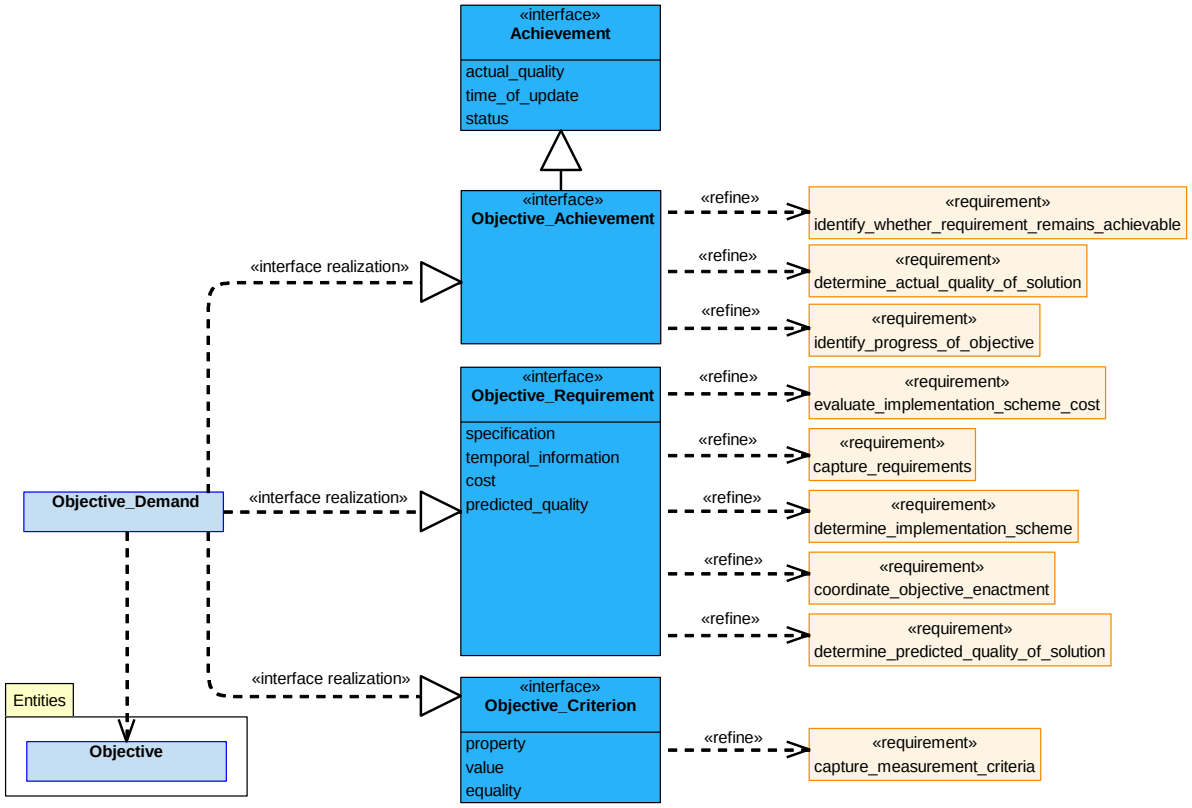

Figure 724: Objective Demand Service Definition

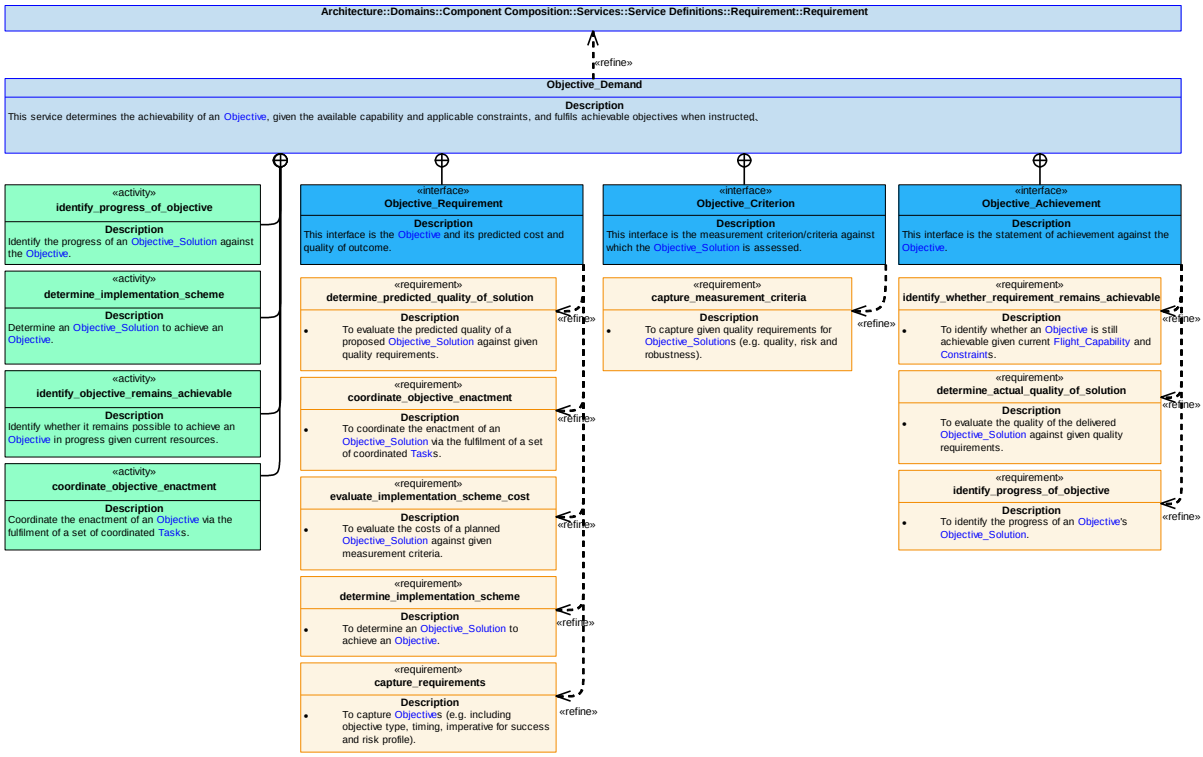

Figure 725: Objective Demand Service Policy

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_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, for example: resources used or time taken.
predicted_quality	How well the proposed Objective_Solution is predicted to satisfy the requirement.

Objective_Achievement

This interface is the statement of achievement against the Objective.

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.

B.2.38.7.1.2 Task_Dependency

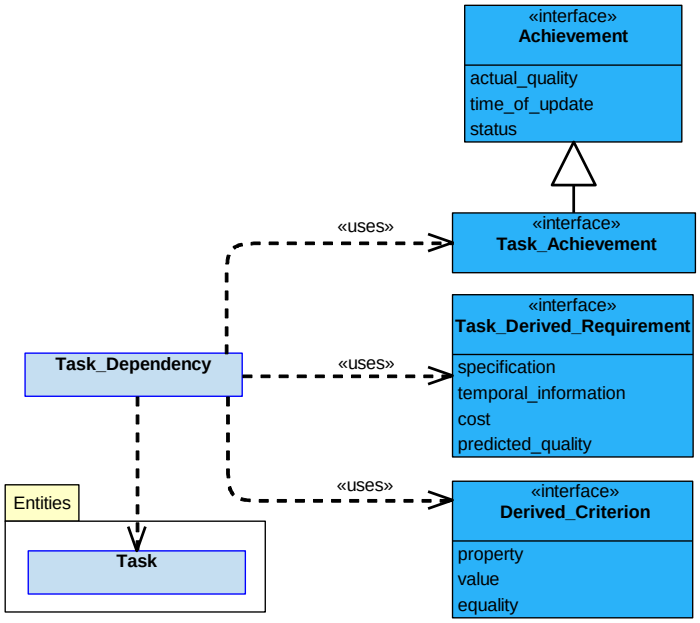

Figure 726: Task Dependency Service Definition

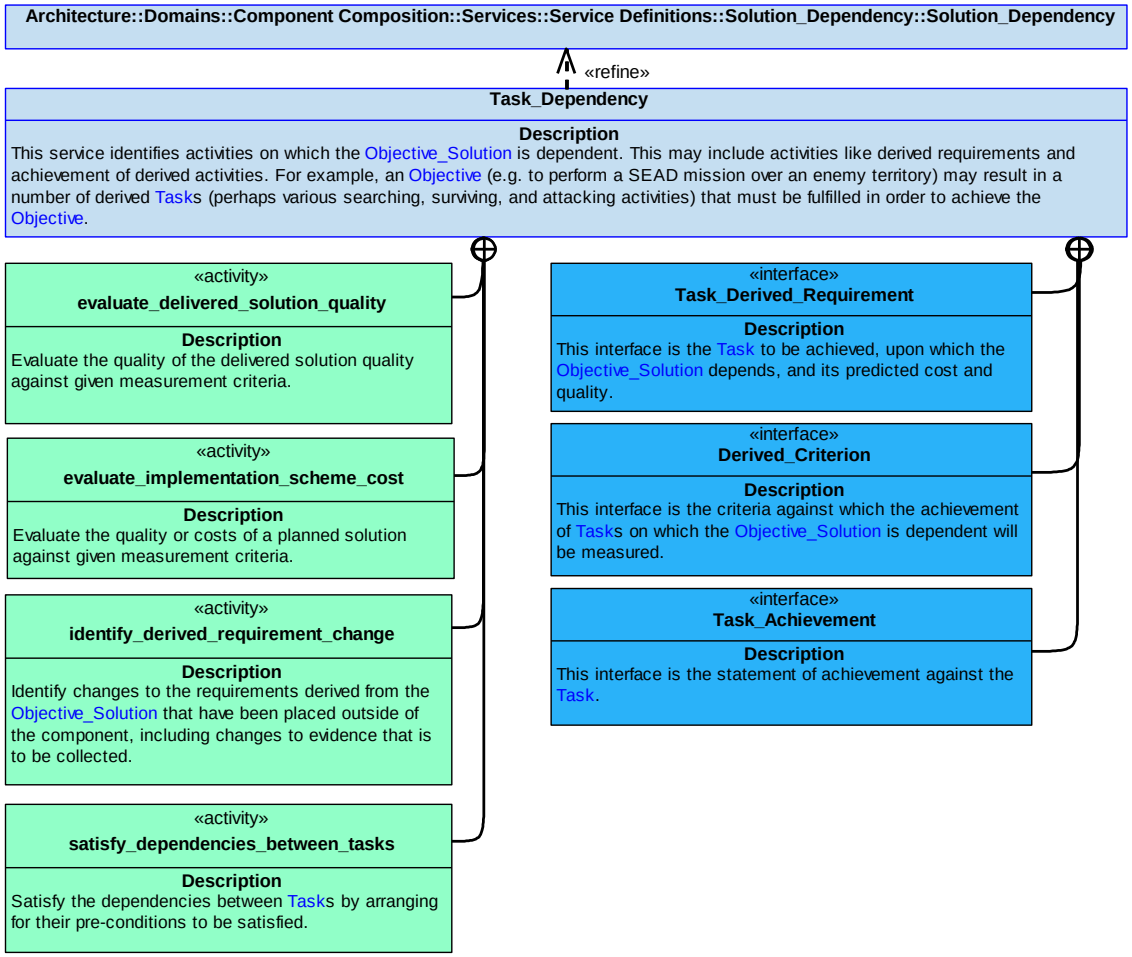

Figure 727: Task Dependency Service Policy

Task_Dependency

This service identifies activities on which the Objective_Solution is dependent. This may include activities like derived requirements and achievement of derived activities. For example, an Objective (e.g. to perform a SEAD mission over an enemy territory) may 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, for example: 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

Satisfy the dependencies between Tasks by arranging for their pre-conditions to be satisfied.

evaluate_implementation_scheme_cost

Evaluate the quality or costs of a planned solution against given measurement criteria.

B.2.38.7.1.3 Constraint

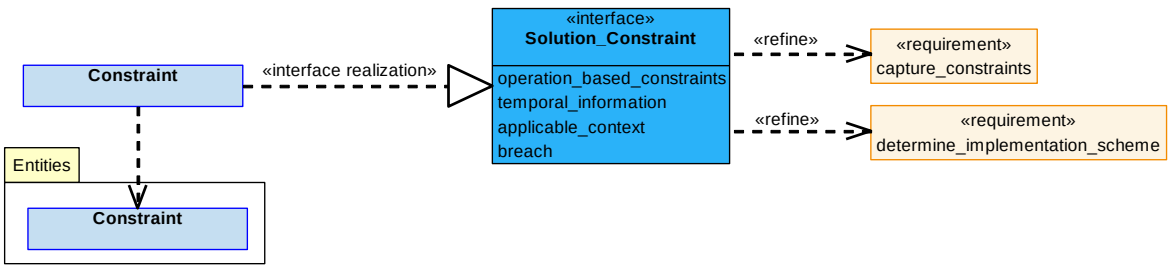

Figure 728: Constraint Service Definition

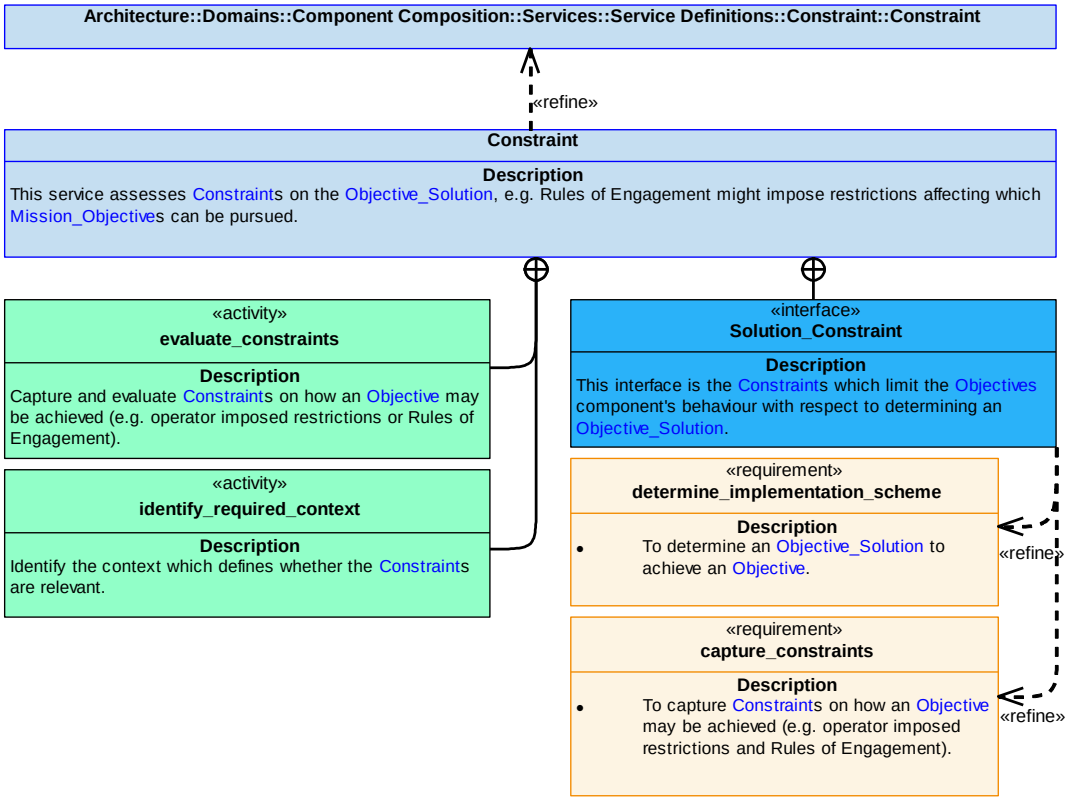

Figure 729: Constraint Service Policy

Constraint

This service assesses Constraints on the Objective_Solution, e.g. Rules of Engagement might impose restrictions affecting which 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_constraints	An attribute of the Constraint impacting the Objectives component's behaviour, e.g. RoE currently in force may dictate that certain types of Mission_Objective cannot be carried out.
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.

B.2.38.7.1.4 Capability

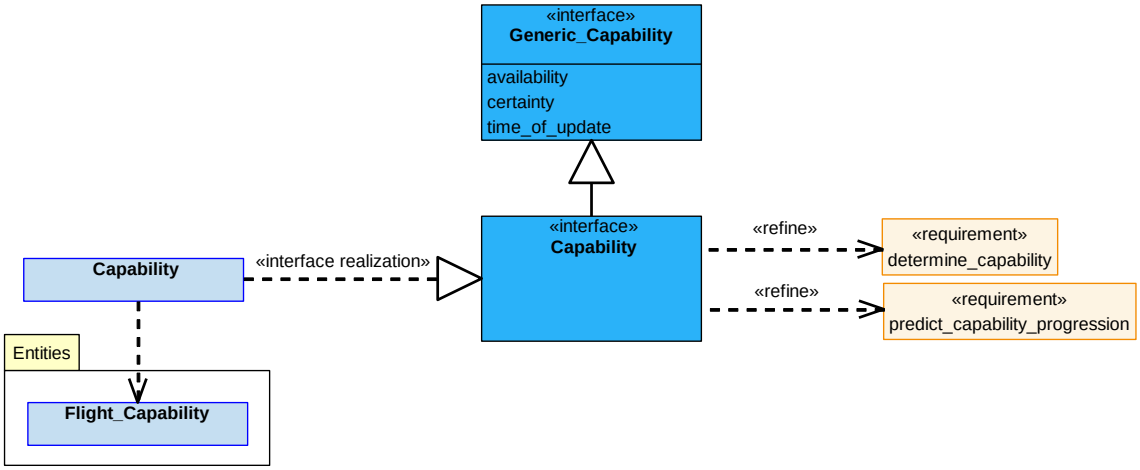

Figure 730: Capability Service Definition

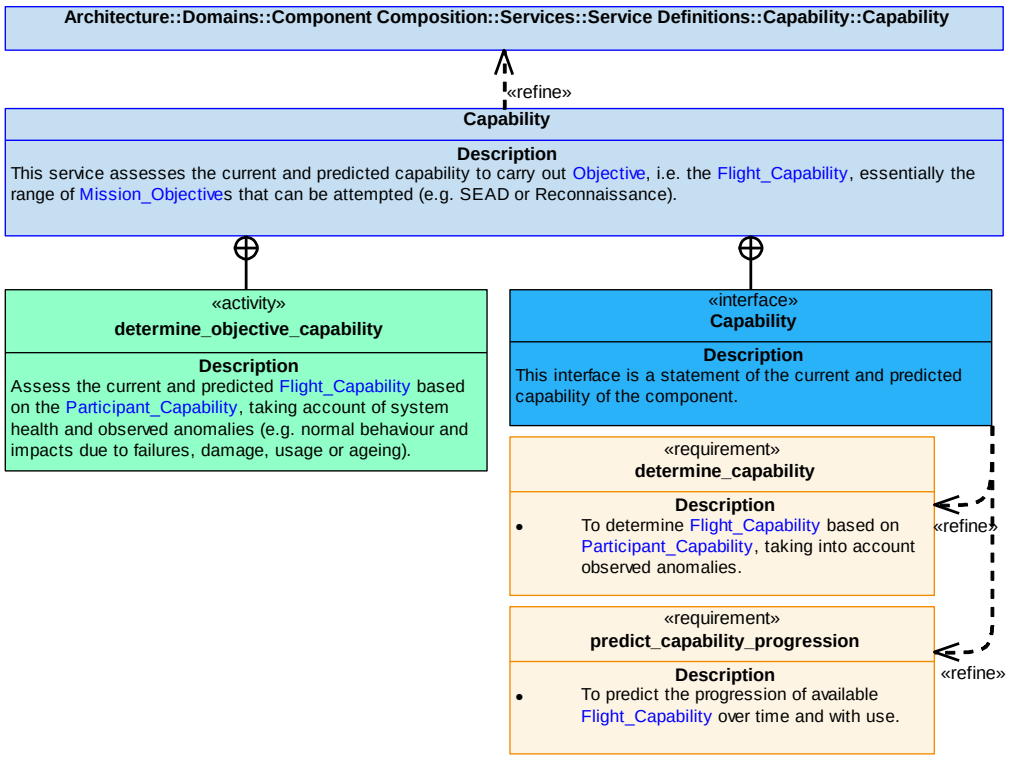

Figure 731: Capability Service Policy

Capability

This service assesses the current and predicted capability to carry out Objective, 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).

B.2.38.7.1.5 Capability_Evidence

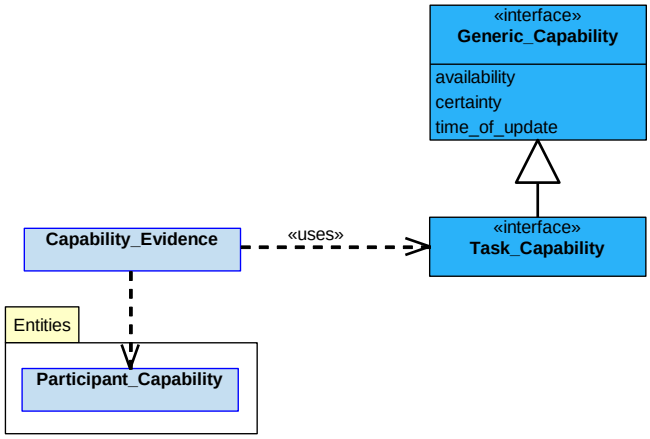

Figure 732: Capability Evidence Service Definition

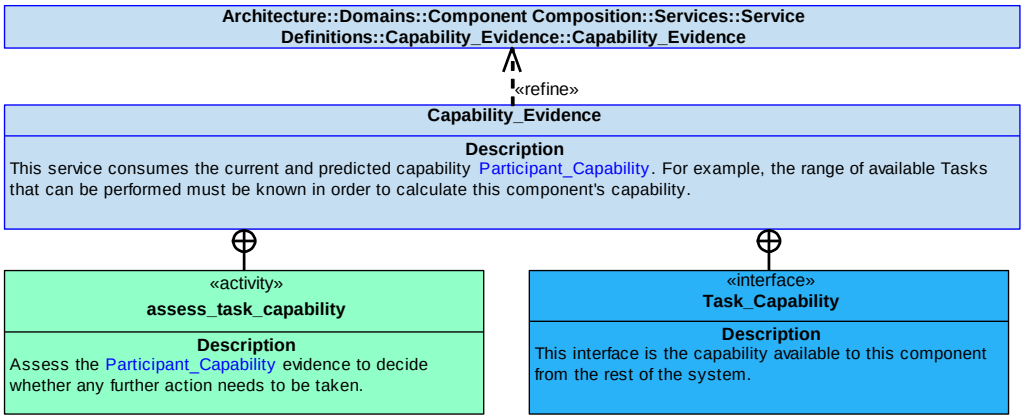

Figure 733: Capability Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capability Participant_Capability. For example, the range of available Tasks that can be performed must be known in order to calculate this component's capability.

Interface

Task_Capability

This interface is the capability available to this component from the rest of the system.

Activity

assess_task_capability

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

B.2.38.7.2 Service Dependencies

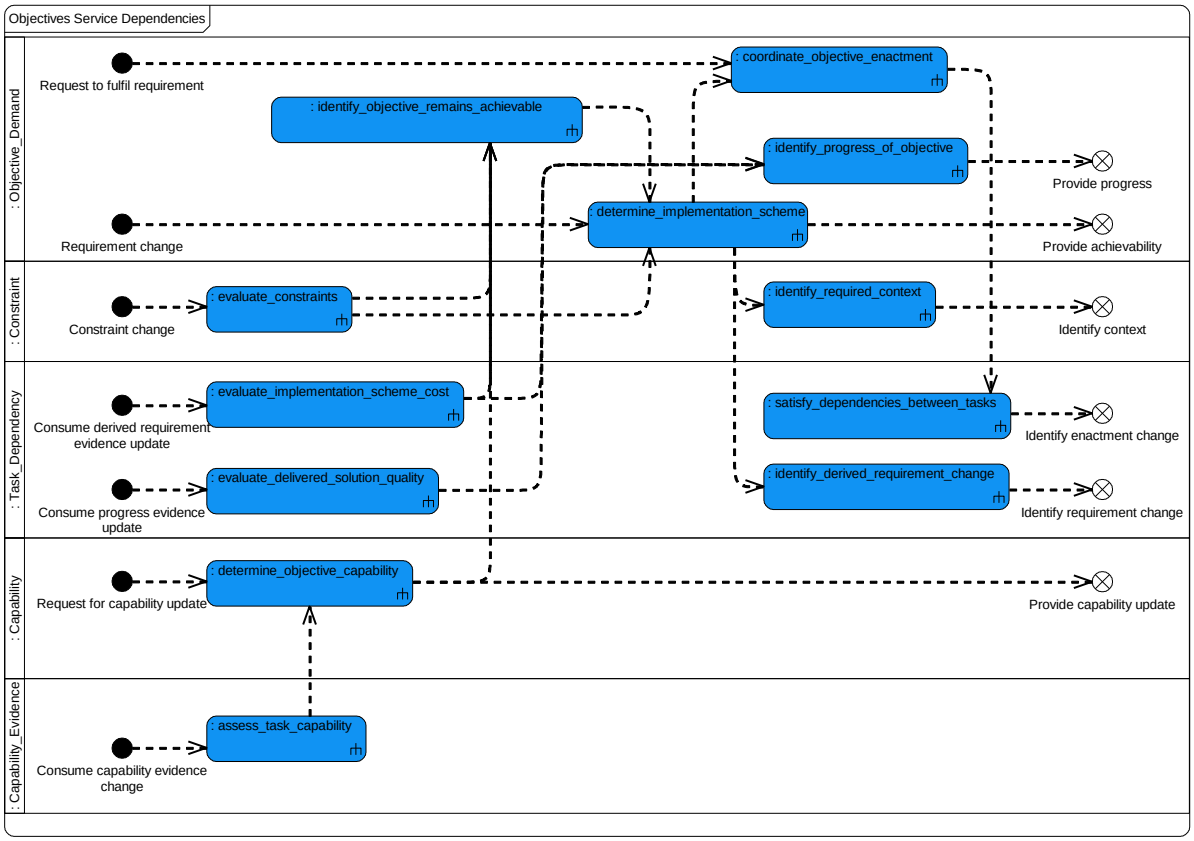

Figure 734: Objectives Service Dependencies

B.2.39 Observability

B.2.39.1 Role

The role of Observability is to evaluate a subject's observability by an observer.

B.2.39.2 Overview

Control Architecture

Observability is a service component as defined in the Control Architecture policy.

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.

B.2.39.3 Service Summary

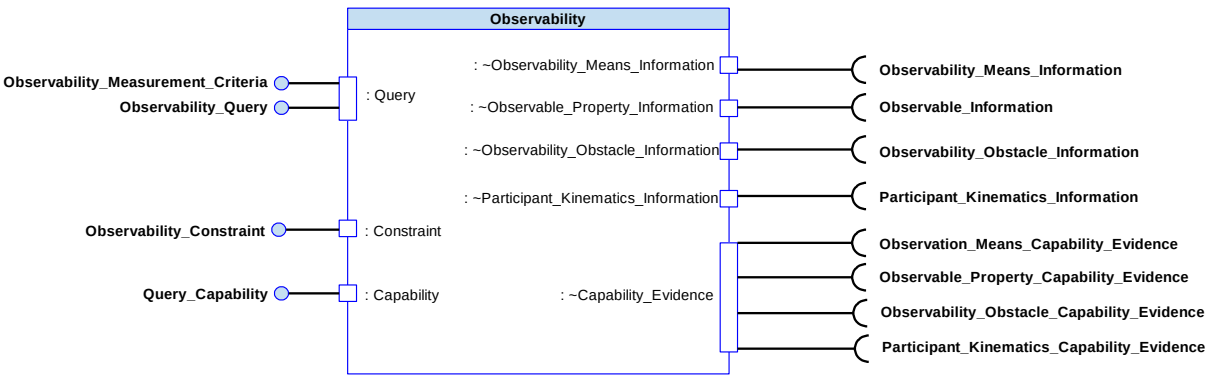

Figure 735: Observability Service Summary

B.2.39.4 Responsibilities

capture_observability_requirements

- To capture requirements for determining the Observability of a Subject.

capture_observability_constraints

- To capture Constraints on Observability (e.g. Observation_Means not to be used, effect of range and occlusion on Observation_Means).

capture_observability_measurement_criteria

- To capture Measurement_Criterion/criteria for Observability.

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_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.

determine_line_of_sight

- To determine Observation_Path between an Observer and a Subject.

predict_observability_capability_progression

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

B.2.39.5 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.

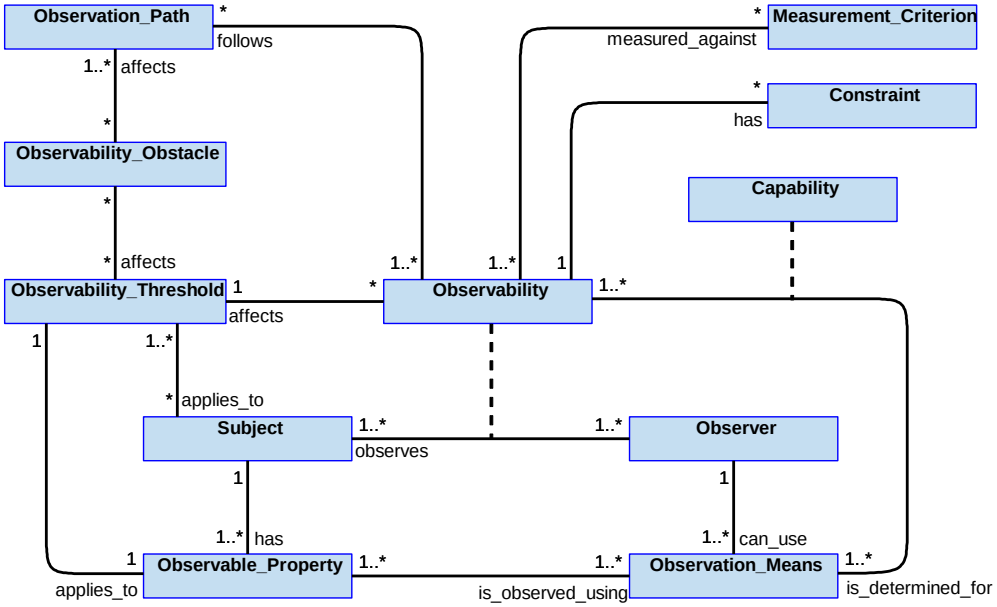

Figure 736: Observability Semantics

B.2.39.5.1 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).

B.2.39.6 Design Rationale

B.2.39.6.1 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.

B.2.39.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.39.6.3 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.

B.2.39.6.4 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.

B.2.39.7 Services

B.2.39.7.1 Service Definitions

B.2.39.7.1.1 Query

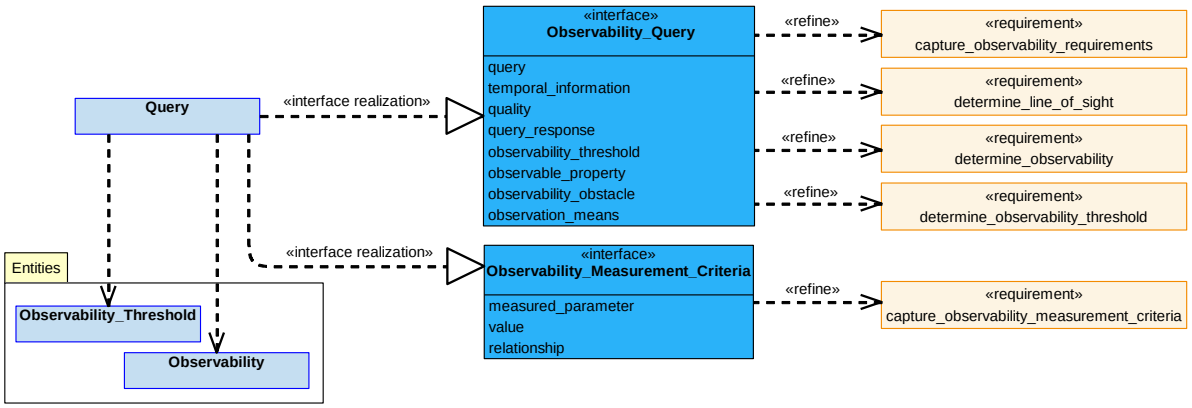

Figure 737: Query Service Definition

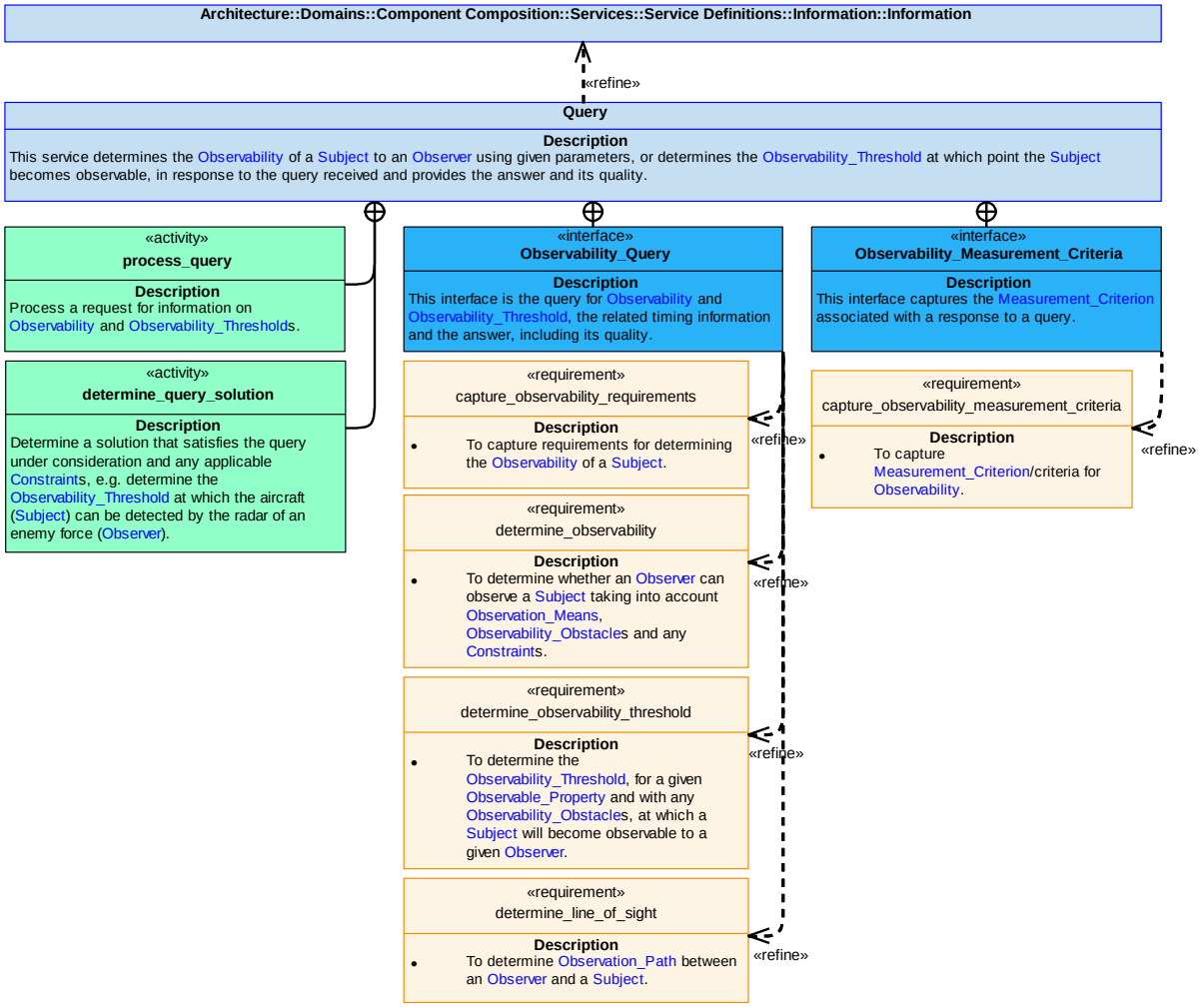

Figure 738: Query Service Policy

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).

B.2.39.7.1.2 Observability_Means_Information

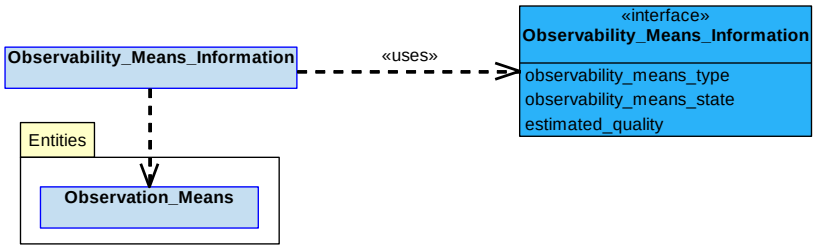

Figure 739: Observability_Means_Information Service Definition

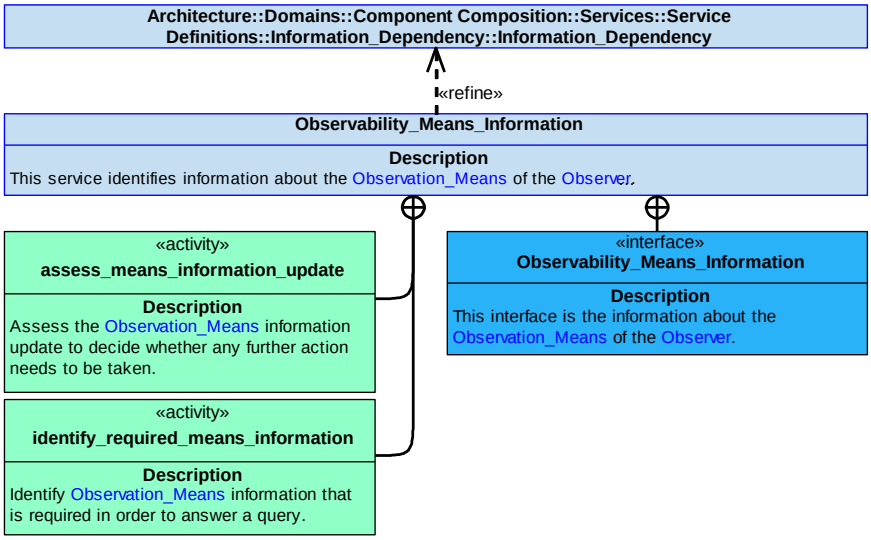

Figure 740: Observability_Means_Information Service Policy

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.

B.2.39.7.1.3 Observable_Property_Information

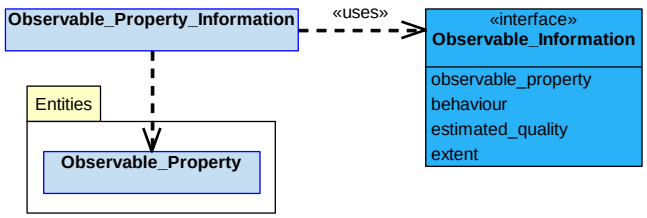

Figure 741: Observable_Property_Information Service Definition

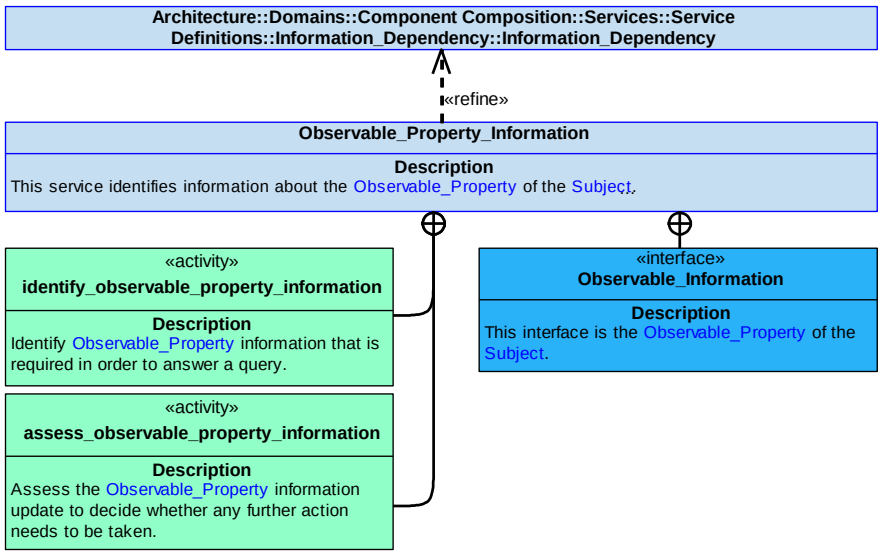

Figure 742: Observable_Property_Information Service Policy

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.

B.2.39.7.1.4 Observability_Obstacle_Information

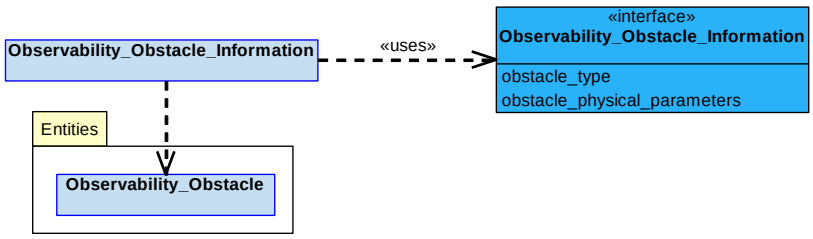

Figure 743: Observability_Obstacle_Information Service Definition

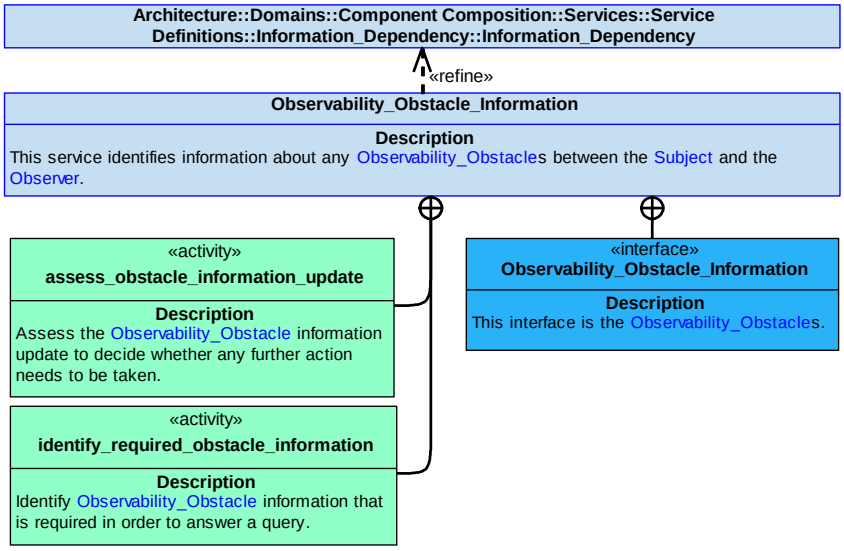

Figure 744: Observability_Obstacle_Information Service Policy

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.

B.2.39.7.1.5 Participant_Kinematics_Information

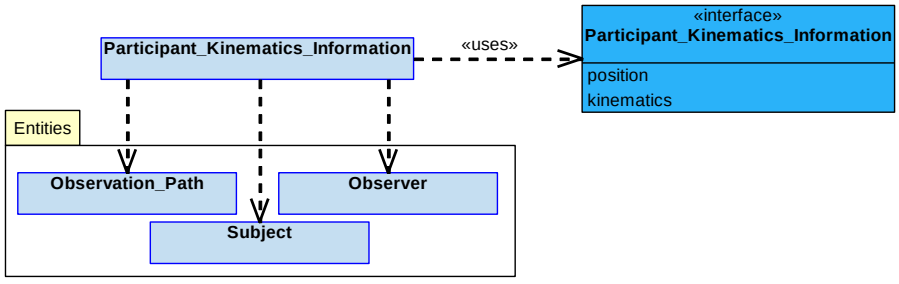

Figure 745: Participant_Kinematics_Information Service Definition

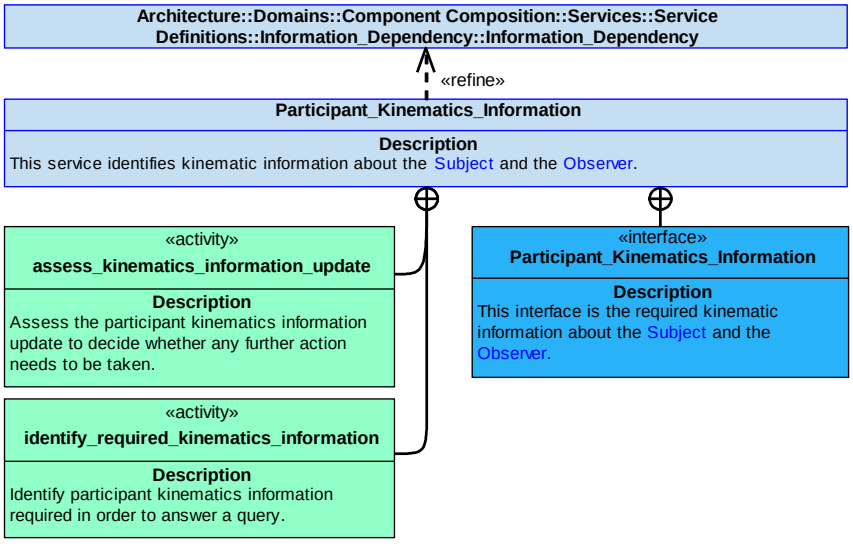

Figure 746: Participant_Kinematics_Information Service Policy

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.

B.2.39.7.1.6 Constraint

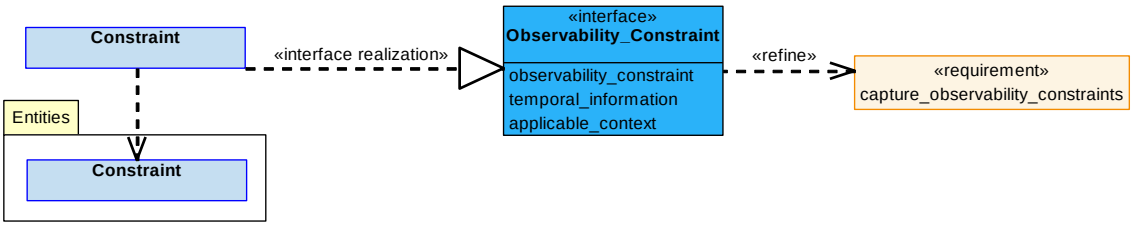

Figure 747: Constraint Service Definition

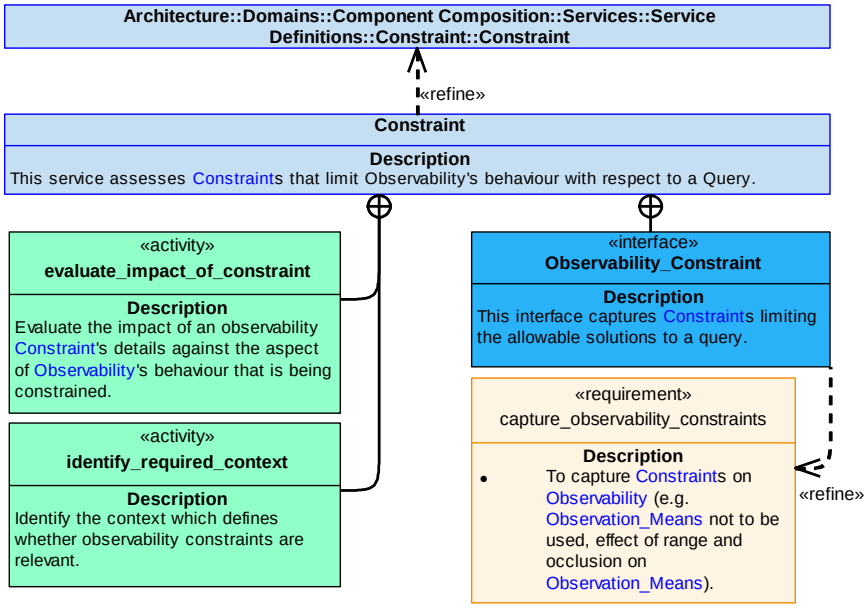

Figure 748: Constraint Service Policy

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.

B.2.39.7.1.7 Capability

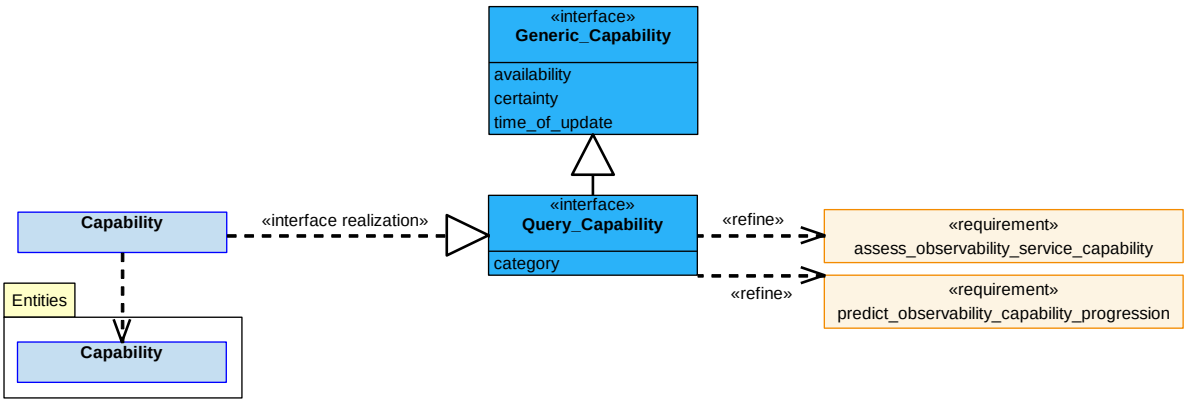

Figure 749: Capability Service Definition

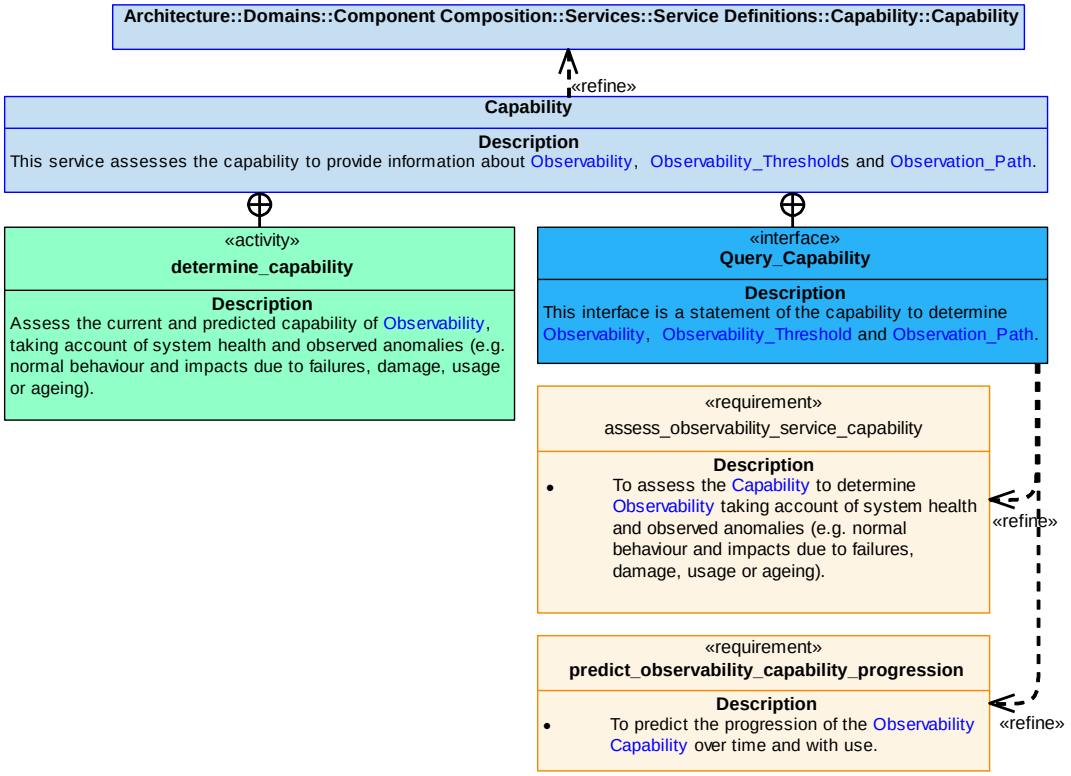

Figure 750: Capability Service Policy

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).

B.2.39.7.1.8 Capability_Evidence

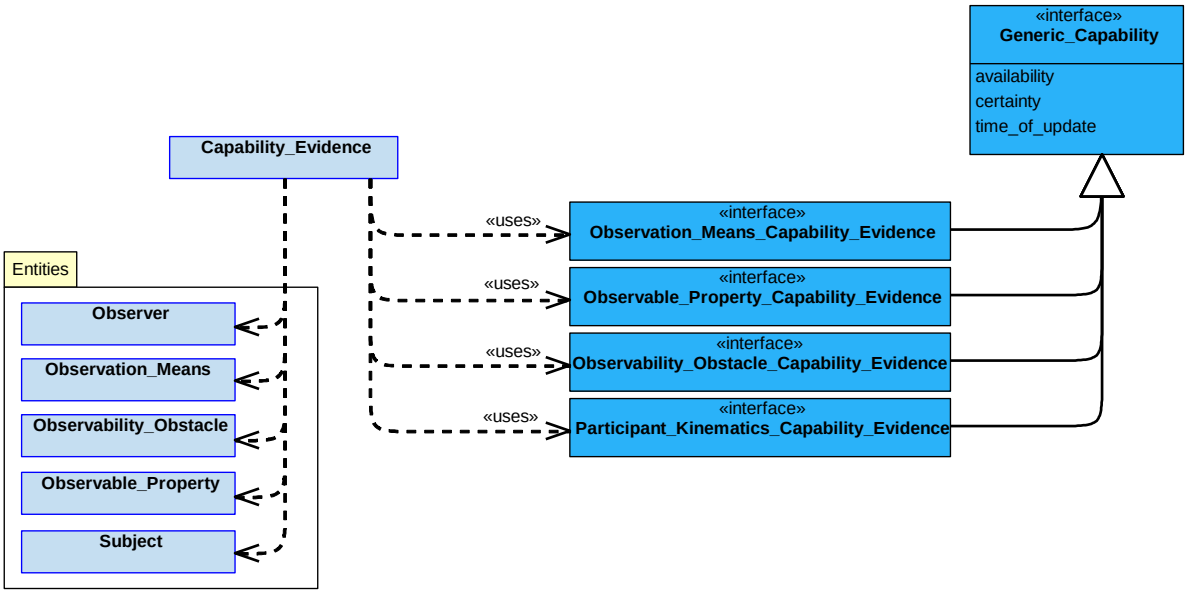

Figure 751: Capability_Evidence Service Definition

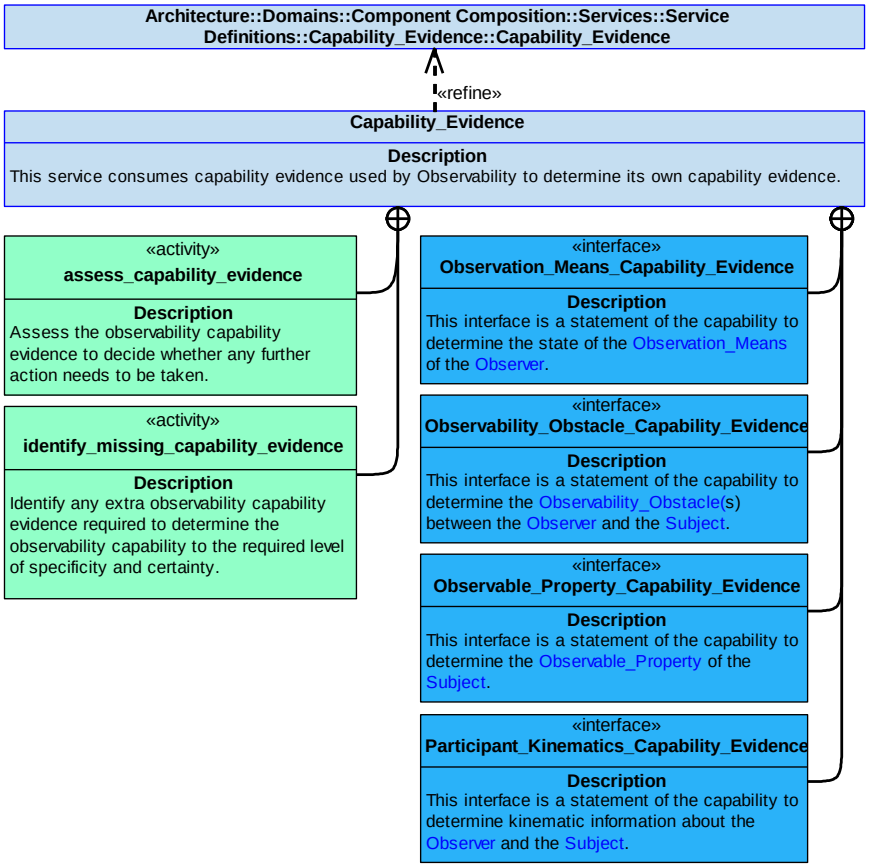

Figure 752: Capability_Evidence Service Policy

Capability_Evidence

This service consumes capability evidence used by Observability to determine its own capability evidence.

Interfaces

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.

B.2.39.7.2 Service Dependencies

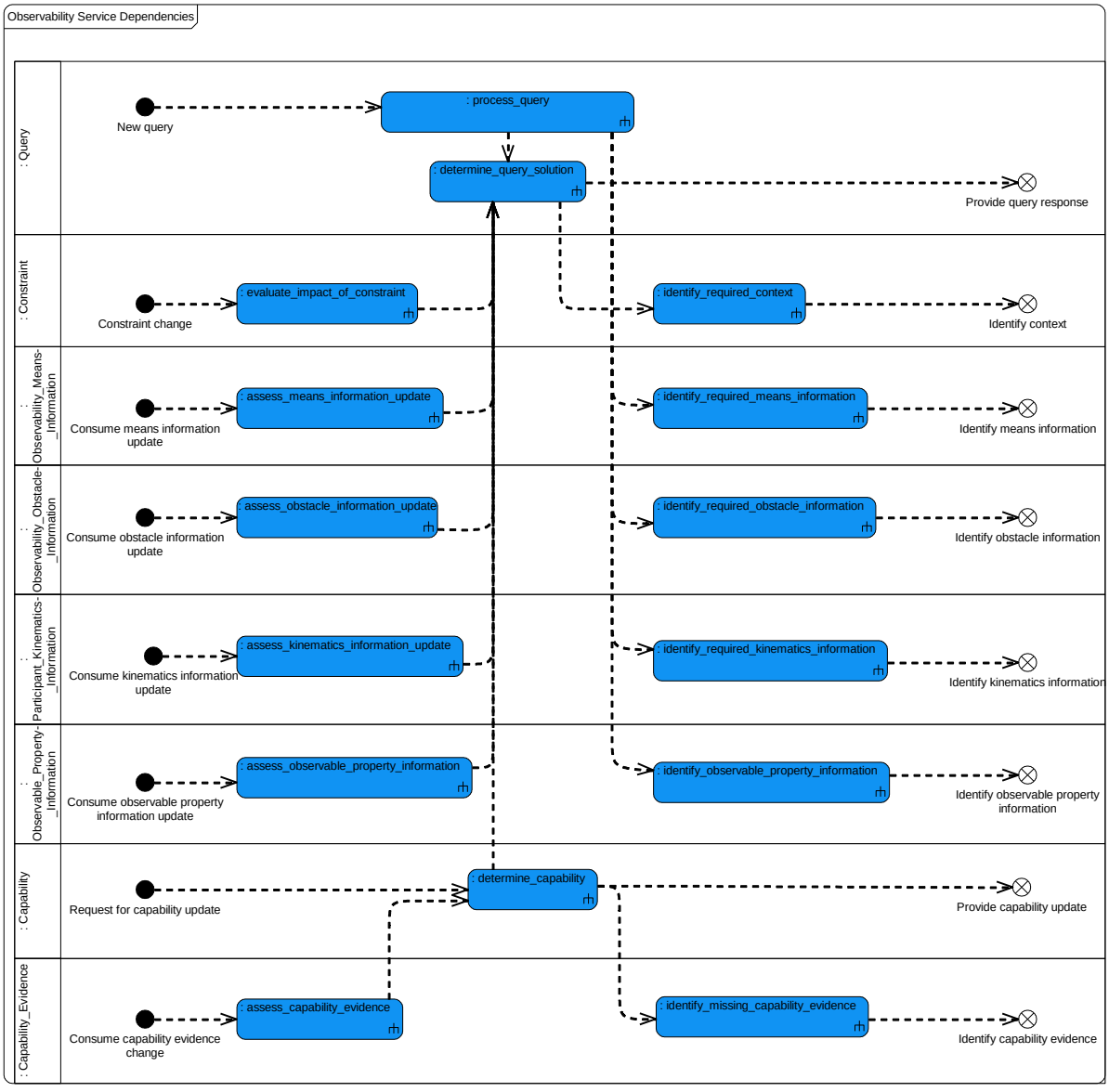

Figure 753: Observability Service Dependencies

B.2.40 Operational Rules and Limits

B.2.40.1 Role

The role of Operational Rules and Limits is to derive limits from rules.

B.2.40.2 Overview

Control Architecture

Operational Rules and Limits is a service component as defined in the Control Architecture policy.

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.

B.2.40.3 Service Summary

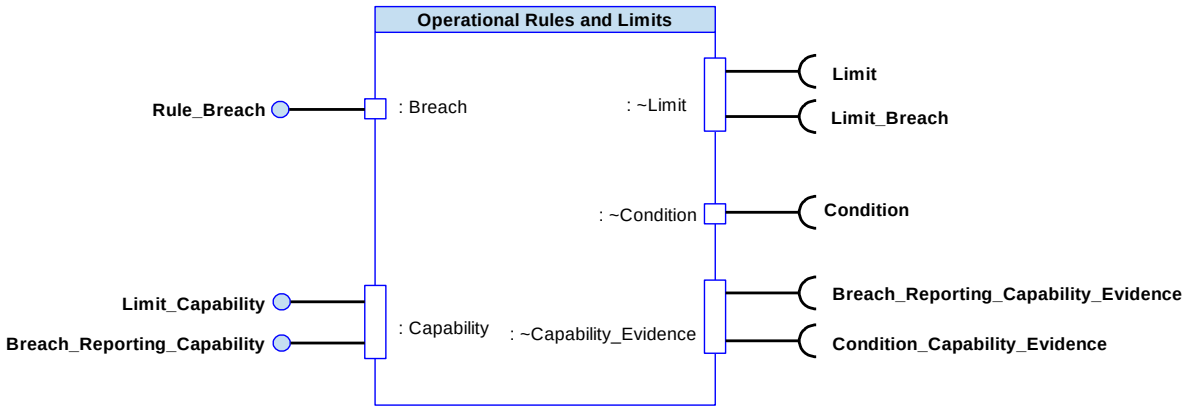

Figure 754: Operational Rules and Limits Service Summary

B.2.40.4 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.

predict_capability_progression

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

identify_missing_information

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

B.2.40.5 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.

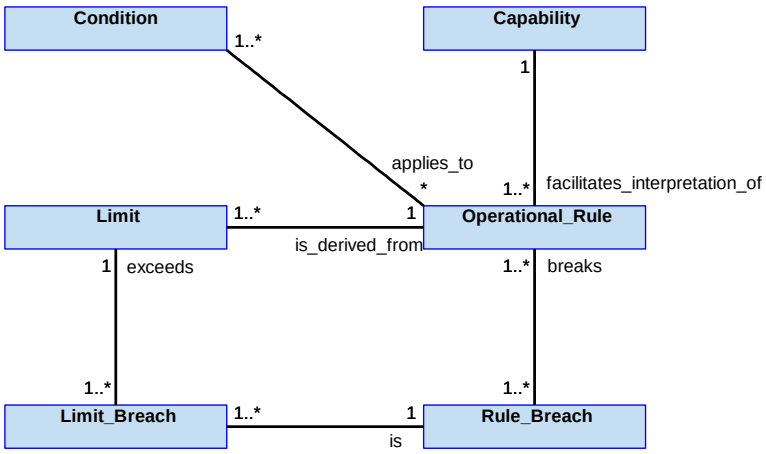

Figure 755: Operational Rules and Limits Semantics

B.2.40.5.1 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.

B.2.40.6 Design Rationale

B.2.40.6.1 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 policy, 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.

B.2.40.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Operational Rules and Limits:

- Constraint Management - Operational Rules and Limits follows the policy.
- 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 (i.e. development) 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.

B.2.40.6.3 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.

B.2.40.6.4 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).

B.2.40.7 Services

B.2.40.7.1 Service Definitions

B.2.40.7.1.1 Breach

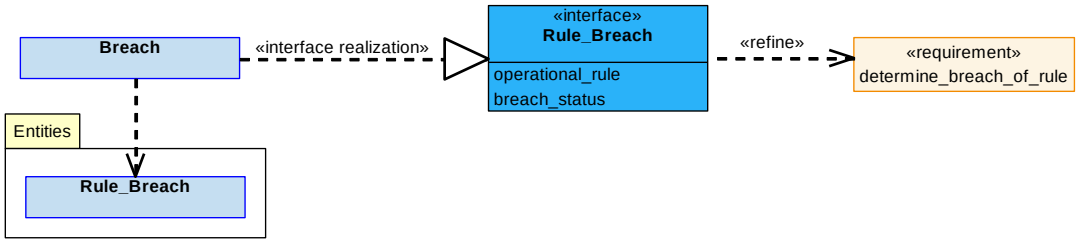

Figure 756: Breach Service Definition

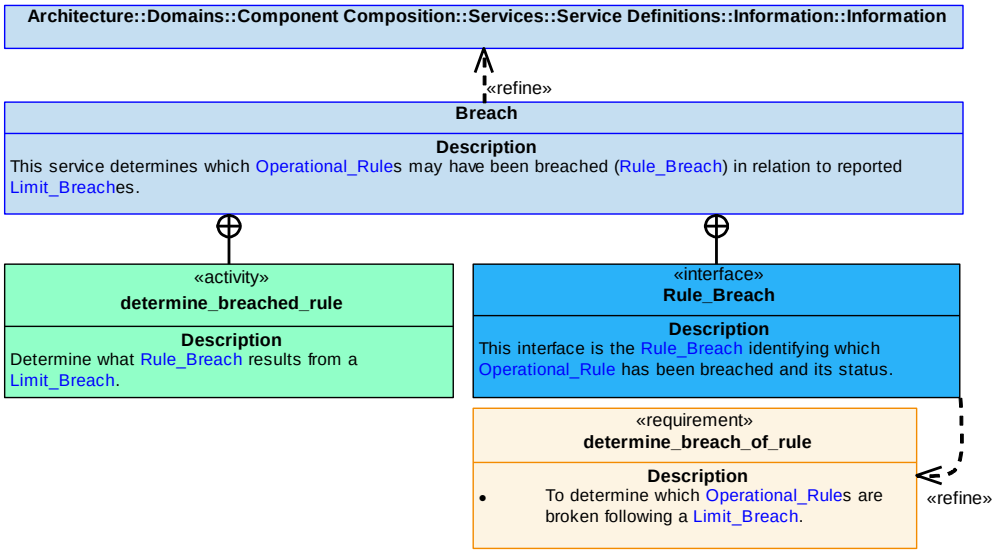

Figure 757: Breach Service Policy

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.

B.2.40.7.1.2 Limit

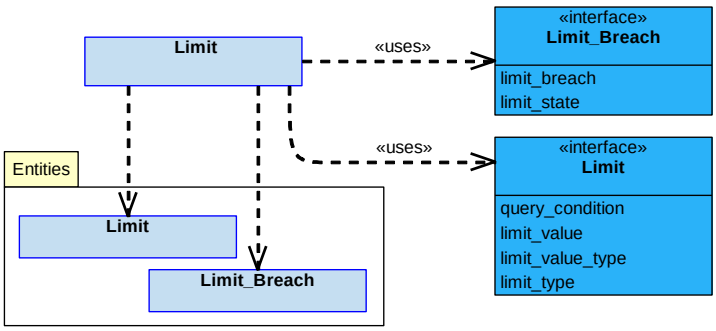

Figure 758: Limit Service Definition

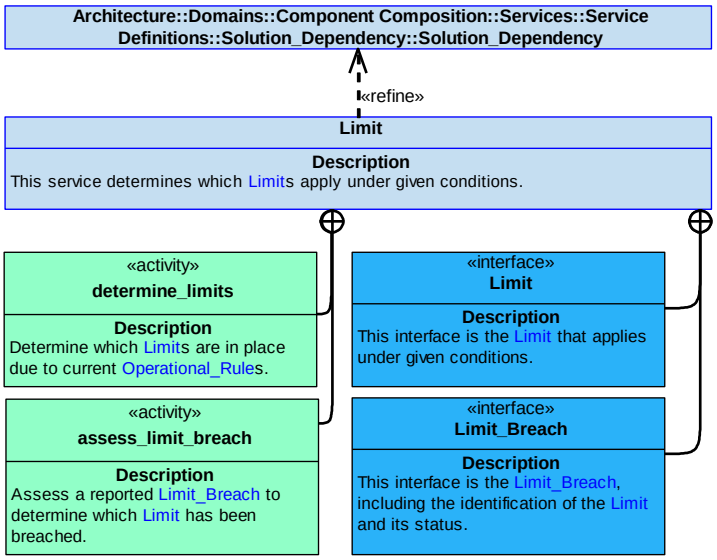

Figure 759: Limit Service Policy

Limit

This service determines which Limits apply under given conditions.

Interfaces

Limit

This interface is the Limit that applies under given conditions.

Attributes

query_condition	The conditions under which the Limit is to be determined, e.g. location, time, or altitude.
limit_value	The value of the Limit, e.g. 50 m/s, where the limit is a maximum speed of 50 m/s.
limit_value_type	The type of the Limit value, e.g. speed, where the limit is a maximum speed of 50 m/s.
limit_type	The way the value applies to the value type, e.g. maximum, where the limit is a maximum speed of 50 m/s.

Limit_Breach

This interface is the Limit_Breach, including the identification of the Limit and its status.

Attributes

limit_breach	The Limit which has been breached (Limit_Breach), e.g. max speed, min altitude, use of a prohibited frequency band.
limit_state	The reported state of the actual Limit, e.g. breached (outside limit boundaries).

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.

B.2.40.7.1.3 Condition

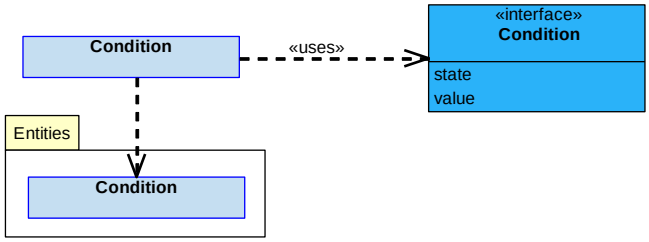

Figure 760: Condition Service Definition

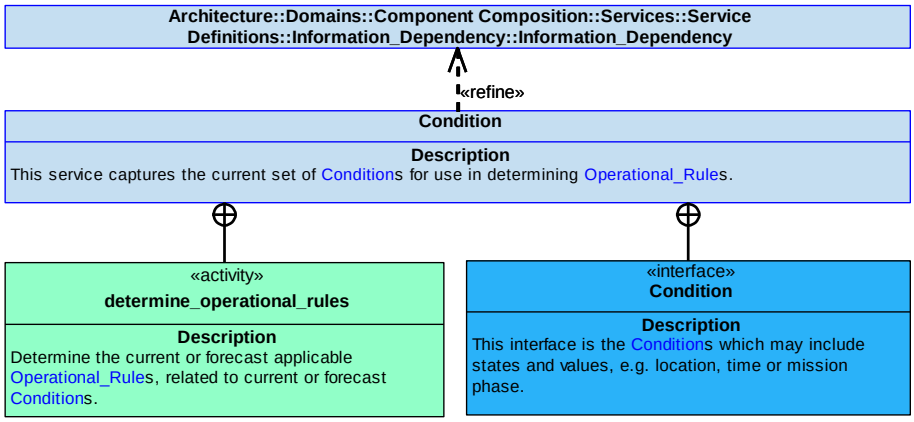

Figure 761: Condition Service Policy

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.

B.2.40.7.1.4 Capability

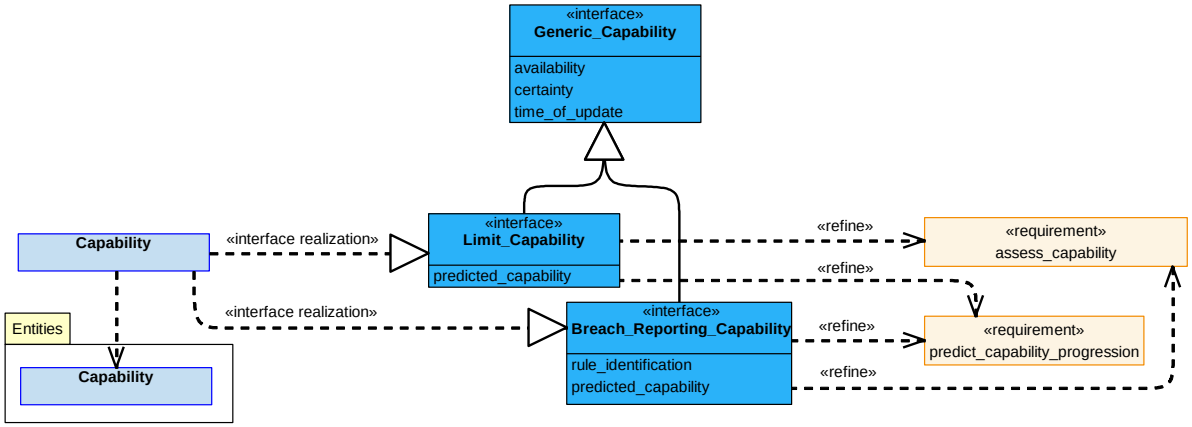

Figure 762: Capability Service Definition

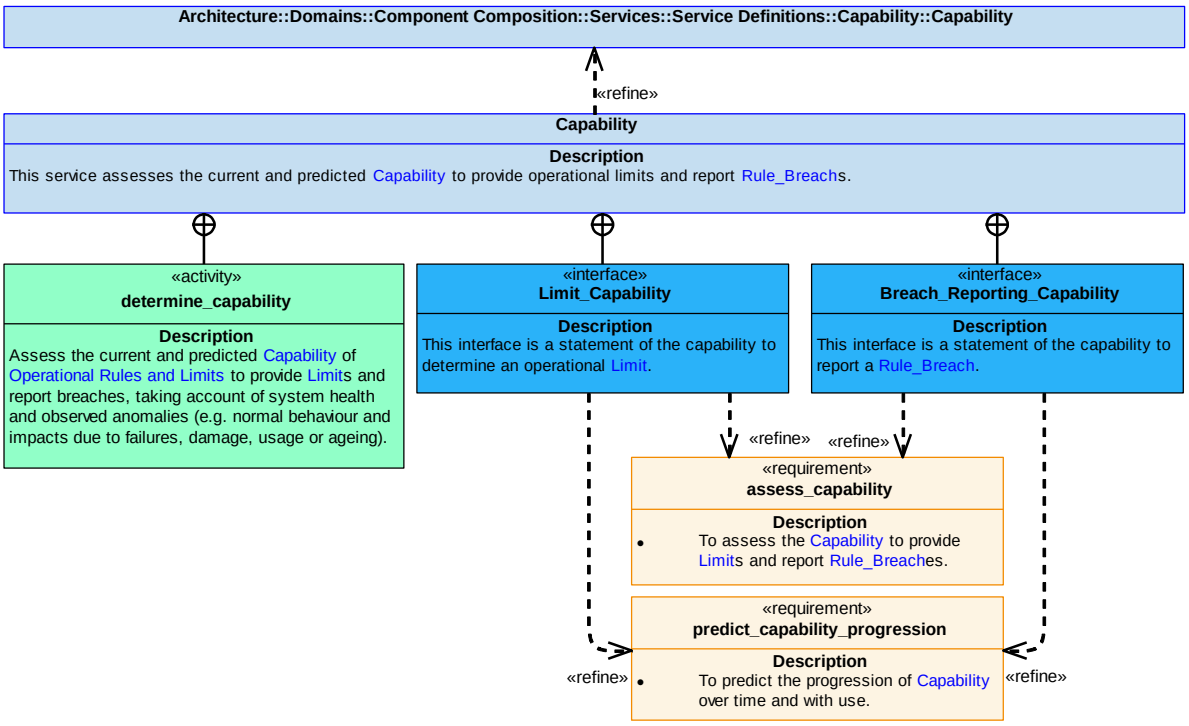

Figure 763: Capability Service Policy

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).

B.2.40.7.1.5 Capability_Evidence

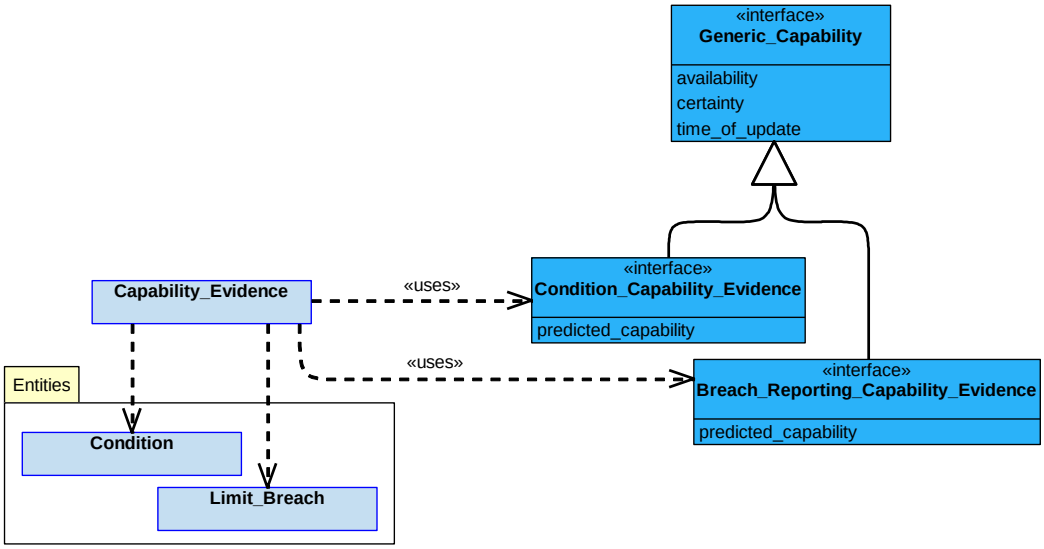

Figure 764: Capability_Evidence Service Definition

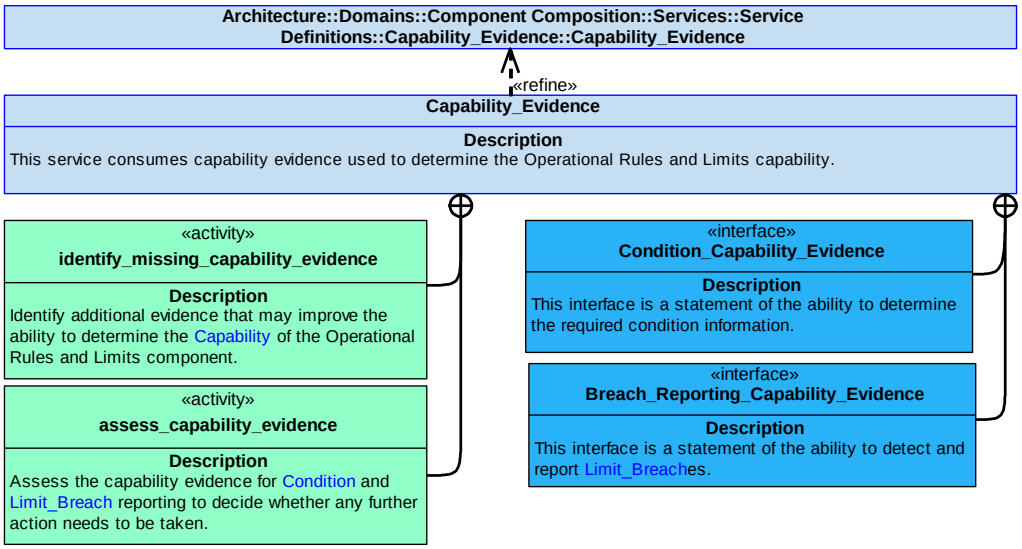

Figure 765: Capability_Evidence Service Policy

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.

B.2.40.7.2 Service Dependencies

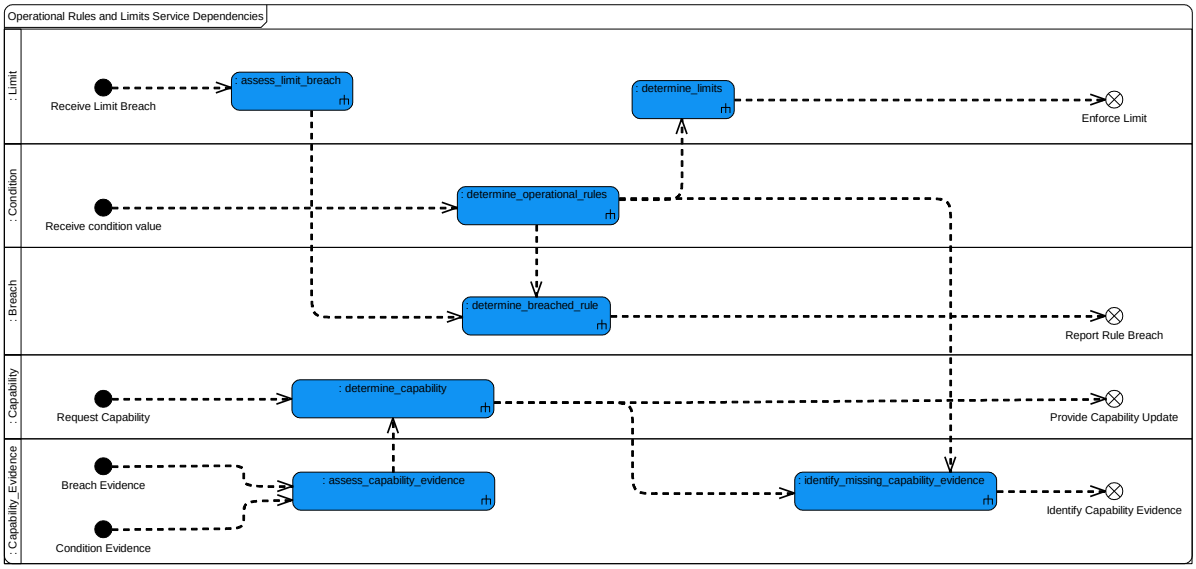

Figure 766: Operational Rules and Limits Service Dependencies

B.2.41 Path Demands

B.2.41.1 Role

The role of Path Demands is to coordinate and arbitrate requests to control the position and motion of the vehicle.

B.2.41.2 Overview

Control Architecture

Path Demands is an action component as defined in the Control Architecture policy.

Standard Pattern of Use

When a Path_Demand is received from a recognised Path_Demand source, Path Demands checks the Path_Demand against applicable Path_Constraints and all other concurrent Path_Demands and issues the identity of a prioritised Path_Demand for execution by the associated Vehicle. The prioritisation can be performed on separate elements of a Path_Demand (e.g. in lateral, vertical and speed channels).

Planned sequences of Path_Demands are also checked to determine whether they are achievable (as a continuous sequence) by a Vehicle**.**

Examples of Use

Path Demands will be used in any system that has:

- Multiple sources of path control for its Vehicles
- Automatic checking that demands for path control fall within system defined constraints (e.g. for flight safety reasons).

B.2.41.3 Service Summary

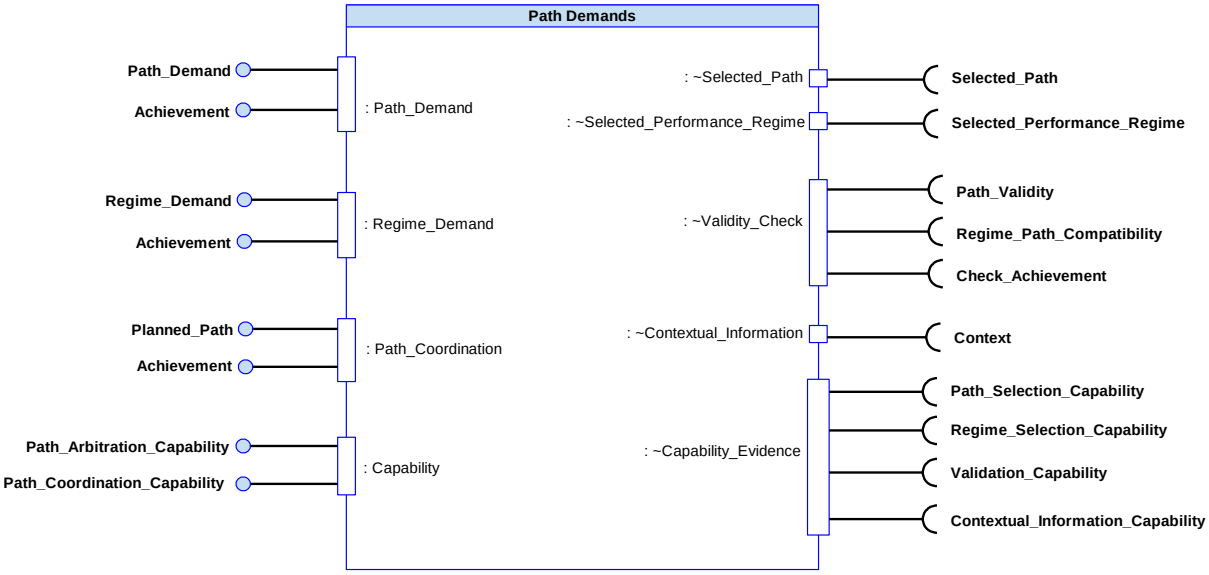

Figure 767: Path Demands Service Summary

B.2.41.4 Responsibilities

capture_path_demands

- To capture Vehicle Path_Demands.

ensure_path_demands_within_constraints

- To ensure Vehicle Path_Demands chosen for implementation are within Path_Constraints.

arbitrate_path_demands

- To arbitrate conflicting Vehicle Path_Demands in the path channels (e.g. lateral, vertical and speed).

coordinate_vehicle_path

- To coordinate the Planned_Path of the Vehicle (e.g. from taxi, take-off, departure, transit, route, approach, landing and taxi).

ensure_path_demand_continuity

- To ensure that a complete set of Vehicle Path_Demands exists to guide the Vehicle through a Planned_Path.

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.

capture_performance_regime_demands

- To capture Vehicle Regime_Demand.

arbitrate_performance_regime_demands

- To arbitrate conflicting Regime_Demand.

predict_capability_progression

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

B.2.41.5 Subject Matter Semantics

The subject matter of Path Demands is demands to control the path of a Vehicle, and their priorities.

Exclusions

The subject matter of Path Demands does not include:

- The generation of Path_Demands.
- Execution of the checks that Path_Demands are within constraints.
- The execution of a chosen path.
- Reporting progress against the detailed position and motion requirement associated with a path.

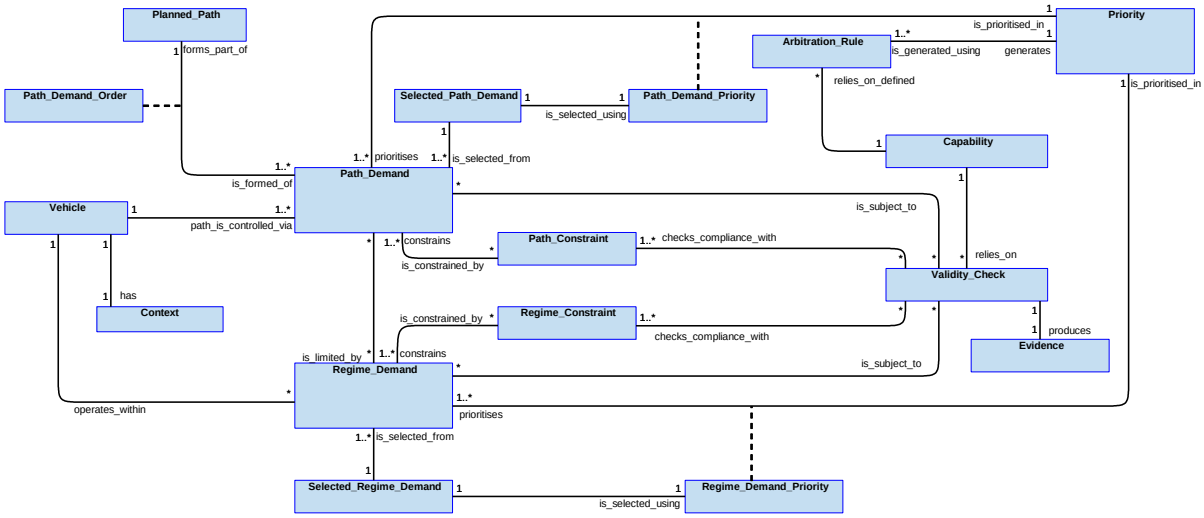

Figure 768: Path Demands Semantics

B.2.41.5.1 Entities

Capability

The ability to coordinate the Planned_Path of the Vehicle, including ensuring the validity of Path_Demands and Regime_Demands, and to arbitrate conflicting demands.

Evidence

Evidence that a constraint is satisfied.

Arbitration_Rule

A method or procedure used (in isolation or in conjunction with others) to determine the priority ordering of a set of demands.

Path_Constraint

A limitation on whether a Path_Demand is executed, e.g. a performance limitation (such as a speed limit), a spatial limitation (such as a volume of space that cannot be entered during a certain time) or a constraint on the validity of transitions between different types of Path_Demands.

Path_Demand

A request to position or move a Vehicle.

Path_Demand_Order

The order of a Path_Demand within a sequence of Path_Demands.

Path_Demand_Priority

The priority of a Path_Demand within a list of demands.

Planned_Path

A planned sequence of Path_Demands.

Priority

The priorities of current demands, used to decide which Path_Demand should be executed and/or which Regime_Demand should be applied.

Validity_Check

A check to determine if a constraint is satisfied.

Vehicle

A moveable object whose path can be controlled.

Regime_Demand

A request to change the particular mode of operation that defines a set of minimum, maximum or optimum performance characteristics, e.g. for take-off, optimum cruise range or during stores release.

Selected_Regime_Demand

The Regime_Demand selected for activation, based on the Regime_Demand_Priority.

Regime_Demand_Priority

The priority of a Regime_Demand within a list of demands.

Selected_Path_Demand

The Path_Demand selected for execution, based on the Path_Demand_Priority.

Regime_Constraint

A limitation on whether a Regime_Demand is activated, e.g. an optimum cruise performance regime cannot be activated when the undercarriage is lowered.

Context

Information relating to the vehicle configuration, state, activity or environment.

B.2.41.6 Design Rationale

B.2.41.6.1 Assumptions

- There can be multiple potential sources of Path_Demands for a Vehicle.
- The number of potential sources of Path_Demands will be dependent on Vehicle type.

B.2.41.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Path Demands:

- Data Driving - The types of Arbitration_Rules, Path_Constraints and Regime_Constraints used by this component could be data-driven.
- Recording and Logging - Logging will be performed for audit and investigation purposed.

Extensions

- It is unlikely that extensions will be appropriate as the basic methods of checking and prioritising demands are not likely to change.

Exploitation Considerations

- This component will be used within an exploitation to ensure vehicle path demands are safe; it is envisaged that this component would be data-driven with the relationships between types of vehicle path demands and the safety checks that need to be performed on them. This makes the component more reusable as it does not have embedded knowledge of the vehicle path demands being managed.
- This component is solely responsible for managing vehicle path demands: this limits what it needs to reason about to the possible types of vehicle path demand, their relative priority under different circumstances, the requirements for them to be considered valid and safe and the set of vehicle path demands which need to be present. By divorcing this component from detailed knowledge of the vehicle path demands and how they are made up it is more reusable. It can be configured to manage any vehicle path demand types for which it is provided with the necessary data.
- This component only requires sufficient path demand information to make an informed arbitration decision between conflicting path demands. It is not envisioned that this component will be provided with detailed demand (position and motion) information for the purpose of path arbitration and execution.
- In some cases components may demand a particular performance regime that is not explicitly associated with a path demand. It is expected that this component would arbitrate whether these requests are consistent with the current demand before passing them to the Vehicle Performance component.
- This component may need to consider any performance regimes currently active, before switching to new path demands to avoid unsafe behaviour. For example, a requested collision avoidance manoeuvre may not be acted on immediately if a stores release performance regime is already active in order to prevent inappropriate manoeuvres during release.
- Information that may be used to prioritise between path demands includes the vehicle operational phase (e.g. take-off or attack), the current performance regime, the mission objective, the type of region that the vehicle is operating in, the configuration of the vehicle (e.g. whether or not the landing gear is lowered), etc.
- Path Demands must always be able to select a path for the vehicle. It is up to the Exploiting Platform to define the rules to ensure that this can happen, for example, by defining what should happen when no behaviourally safe path demand exists.

B.2.41.6.3 Safety Considerations

The indicative IDAL is DAL A**.**

The rationale behind this is:

- Failure of this component would cause uncontrolled flight of the associated 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 fully controlled - i.e. not implement changes to 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.

B.2.41.6.4 Security Considerations

The indicative security classification is O.

This component understands the Exploiting Platform's position and which path that it is following/planned to follow. It is assumed these will be derived using relative coordinates to give an indicative security classification of O. Should absolute coordinates be used that give access to present position etc. a higher classification will apply. Paths will be arbitrated with consideration of own vehicle performance data and any applicable rules, etc. from which mission objectives may be deduced which may similarly raise the classification.

The integrity of this component is paramount for the platform and it is a probable target for cyber attack; tamper with it and you can override safety arbitration and control the vehicle path. Additionally, protections would be required to ensure confidentiality of the path and the availability of functions when required.

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

- **Maintaining Audit Records** of path arbitration decisions for accountability purposes**.**
- **Supporting Safe Operation** of safety critical functions (see Safety Considerations) and will need to be protected to assure continued airworthiness.
- **System Status and Monitoring**, an unexplainable deviation from the chosen path may indicate the component has been compromised by a cyber attack.
- Generation of **Warnings and Notifications** that may provide awareness of unexpected path activity and therefore possible cyber attack.
- Where used in an unmanned platform, this component will be fundamental in **Supporting Secure Remote Operation** as the point that will interface with autonomous or remote piloting commands.

This component is considered unlikely to directly implement any security enforcing functions, although it is dependent on the integrity of the demanded paths.

B.2.41.7 Services

B.2.41.7.1 Service Definitions

B.2.41.7.1.1 Path_Demand

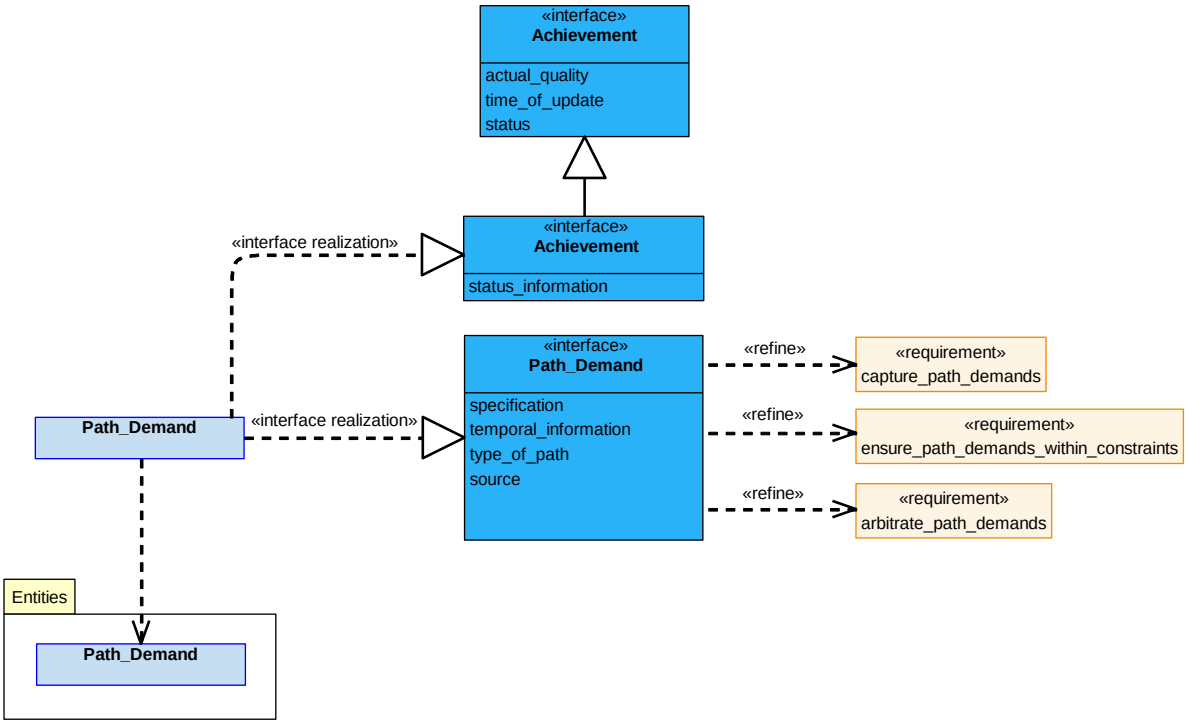

Figure 769: Path_Demand Service Definition

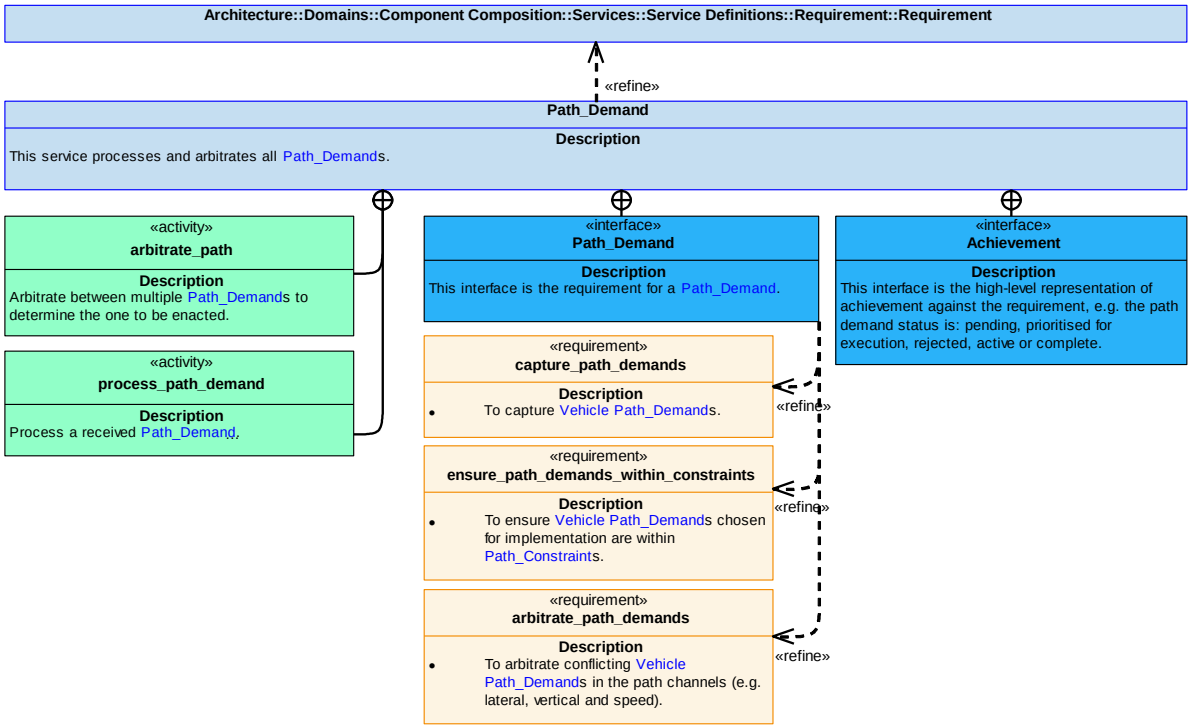

Figure 770: Path_Demand Service Policy

Path_Demand

This service processes and arbitrates all Path_Demands.

Interfaces

Path_Demand

This interface is the requirement for a Path_Demand.

Attributes

specification	The definition of a Path_Demand.
temporal_information	Information covering timing, such as start time and duration, or end time. For example, the time when the path can commence, or must be completed by.
type_of_path	Type of Path_Demand (e.g. collision avoidance or planned route).
source	The source of the Path_Demand.

Achievement

This interface is the high-level representation of achievement against the requirement, e.g. the path demand status is: pending, prioritised for execution, rejected, active or complete.

Attribute

status_information	Supporting information regarding a Path_Demand, e.g. why a request was rejected.

Activities

process_path_demand

Process a received Path_Demand.

arbitrate_path

Arbitrate between multiple Path_Demands to determine the one to be enacted.

B.2.41.7.1.2 Regime_Demand

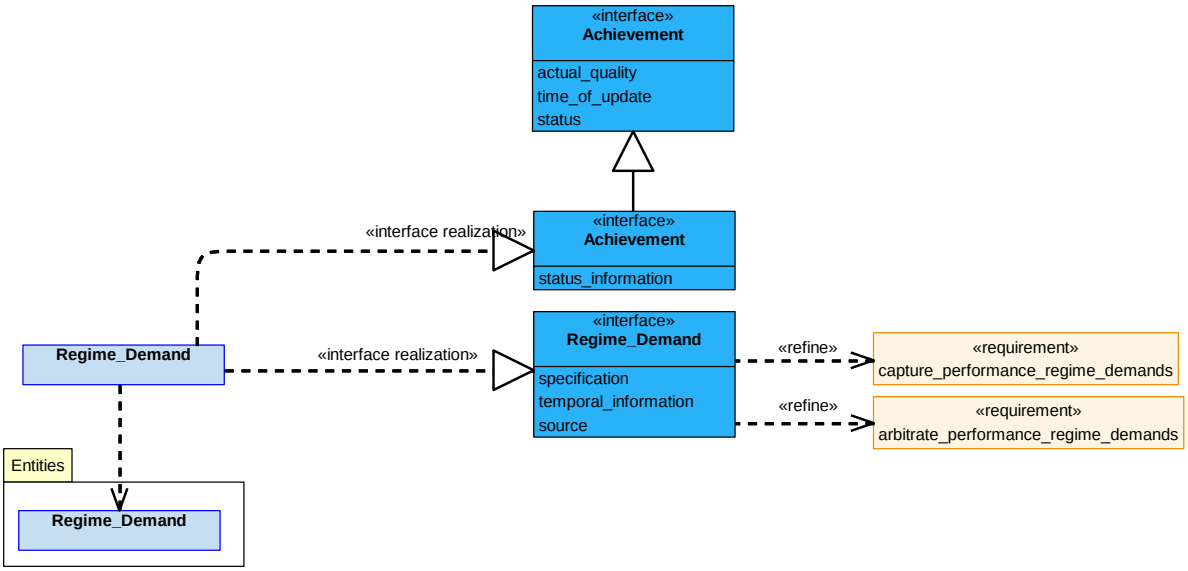

Figure 771: Regime_Demand Service Definition

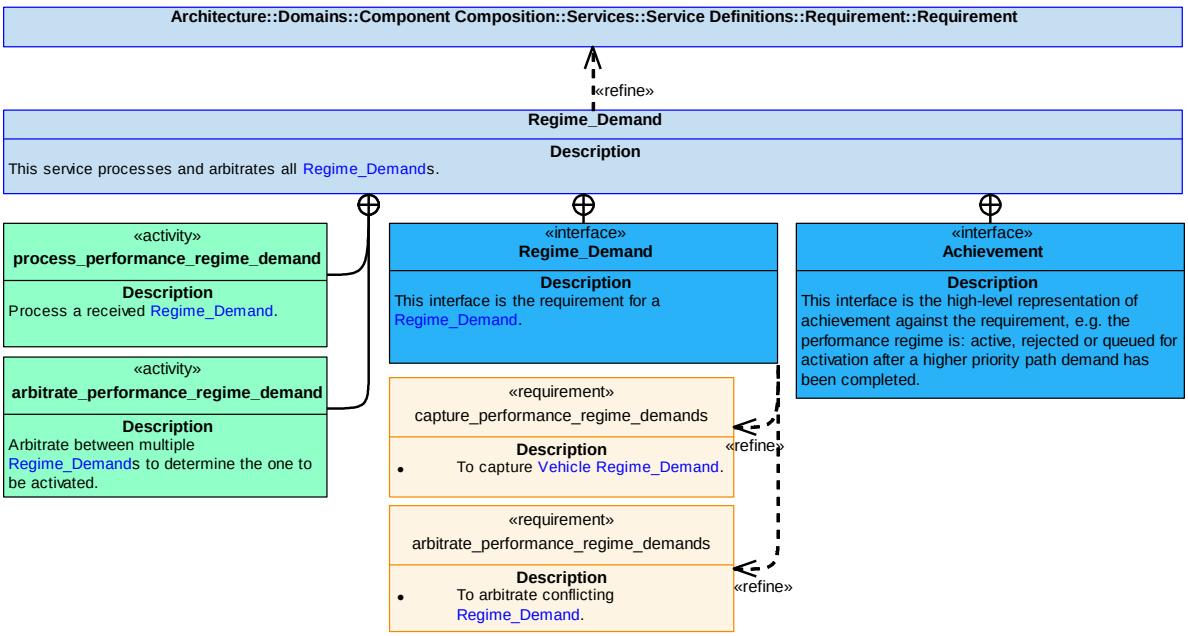

Figure 772: Regime_Demand Service Policy

Regime_Demand

This service processes and arbitrates all Regime_Demands.

Interfaces

Regime_Demand

This interface is the requirement for a Regime_Demand.

Attributes

specification	The definition of a Regime_Demand.
temporal_information	Information covering timing, such as start time and duration, or end time.
source	The source of the Regime_Demand.

Achievement

This interface is the high-level representation of achievement against the requirement, e.g. the performance regime is: active, rejected or queued for activation after a higher priority path demand has been completed.

Attribute

status_information	Supporting information regarding a Regime_Demand, e.g. why a request was rejected.

Activities

process_performance_regime_demand

Process a received Regime_Demand.

arbitrate_performance_regime_demand

Arbitrate between multiple Regime_Demands to determine the one to be activated.

B.2.41.7.1.3 Path_Coordination

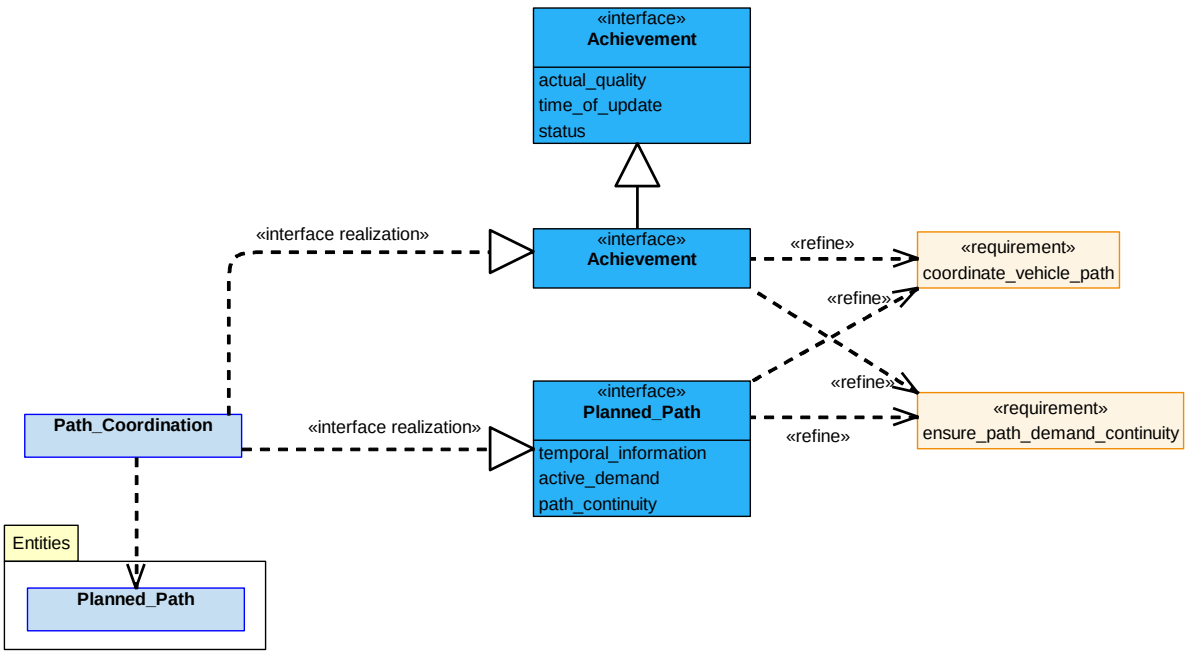

Figure 773: Path_Coordination Service Definition

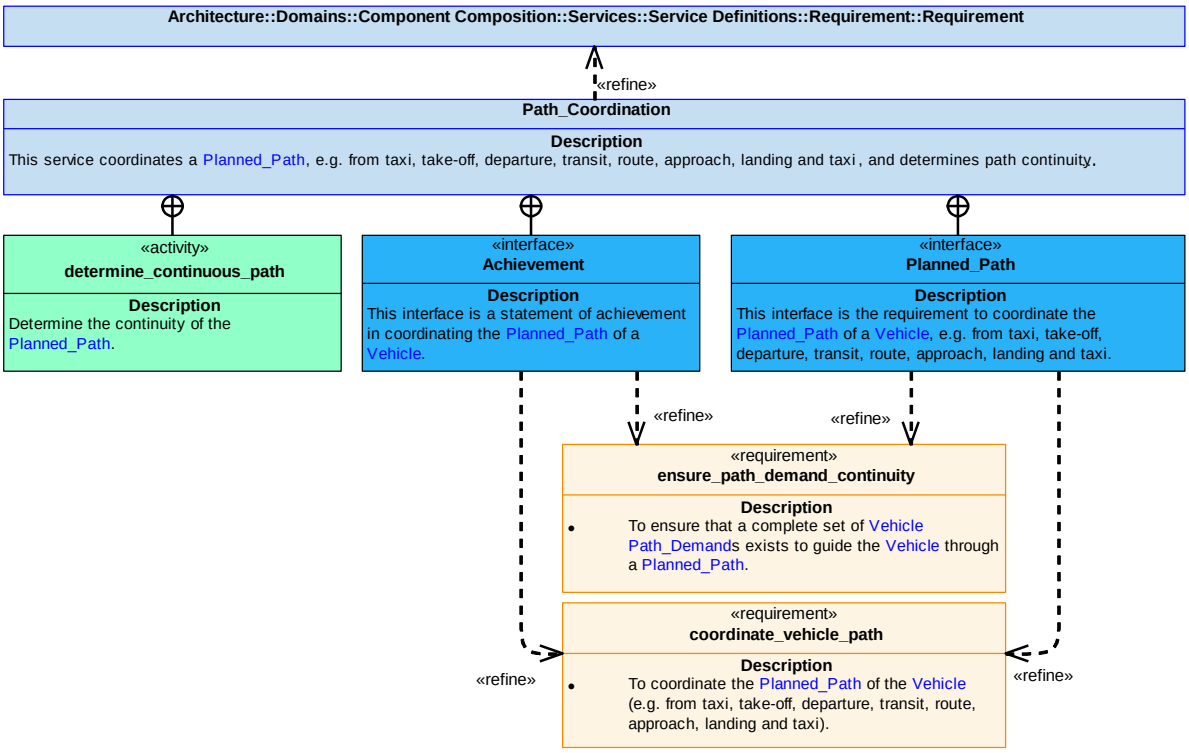

Figure 774: Path_Coordination Service Policy

Path_Coordination

This service coordinates a Planned_Path, e.g. from taxi, take-off, departure, transit, route, approach, landing and taxi, and determines path continuity.

Interfaces

Planned_Path

This interface is the requirement to coordinate the Planned_Path of a Vehicle, e.g. from taxi, take-off, departure, transit, route, approach, landing and taxi.

Attributes

temporal_information	Information covering timing, including start time, duration, or end time, e.g. the time when the path can commence, or must be completed by.
active_demand	The currently active demand within a Planned_Path.
path_continuity	A statement on the continuity/non-continuity of the Planned_Path.

Achievement

This interface is a statement of achievement in coordinating the Planned_Path of a Vehicle.

Activity

determine_continuous_path

Determine the continuity of the Planned_Path.

B.2.41.7.1.4 Selected_Path

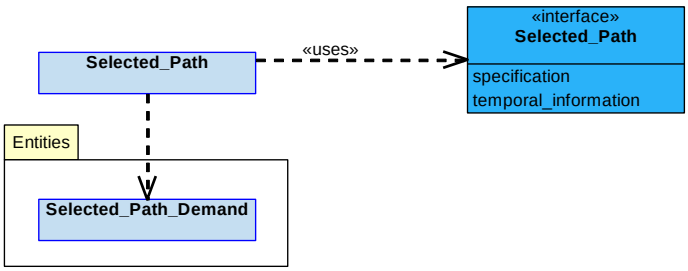

Figure 775: Selected_Path Service Definition

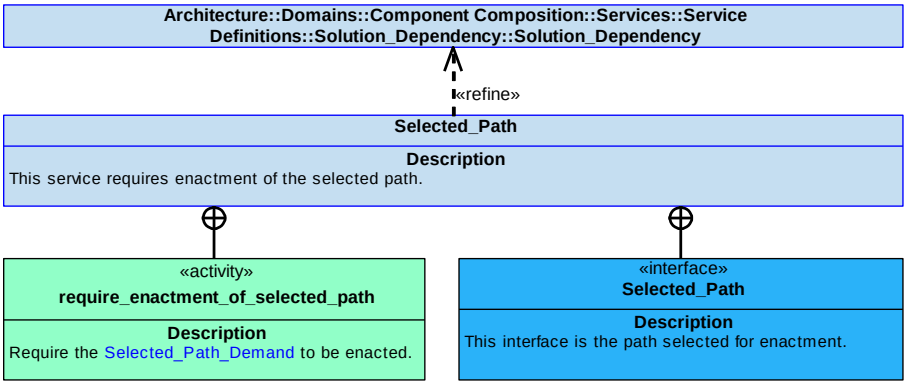

Figure 776: Selected_Path Service Policy

Selected_Path

This service requires enactment of the selected path.

Interface

Selected_Path

This interface is the path selected for enactment.

Attributes

specification	Details of the selected path.
temporal_information	Information covering timing, such as start time and duration, or end time. For example, the time when the path can commence, or must be completed by.

Activity

require_enactment_of_selected_path

Require the Selected_Path_Demand to be enacted.

B.2.41.7.1.5 Selected_Performance_Regime

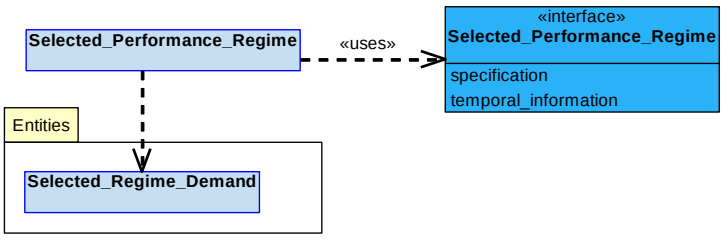

Figure 777: Selected_Performance_Regime Service Definition

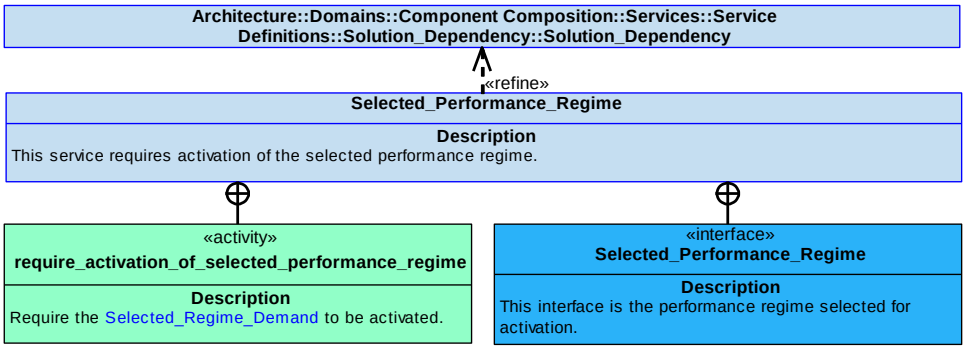

Figure 778: Selected_Performance_Regime Service Policy

Selected_Performance_Regime

This service requires activation of the selected performance regime.

Interface

Selected_Performance_Regime

This interface is the performance regime selected for activation.

Attributes

specification	Details of the selected performance regime.
temporal_information	Information covering timing, such as start time and duration, or end time.

Activity

require_activation_of_selected_performance_regime

Require the Selected_Regime_Demand to be activated.

B.2.41.7.1.6 Validity_Check

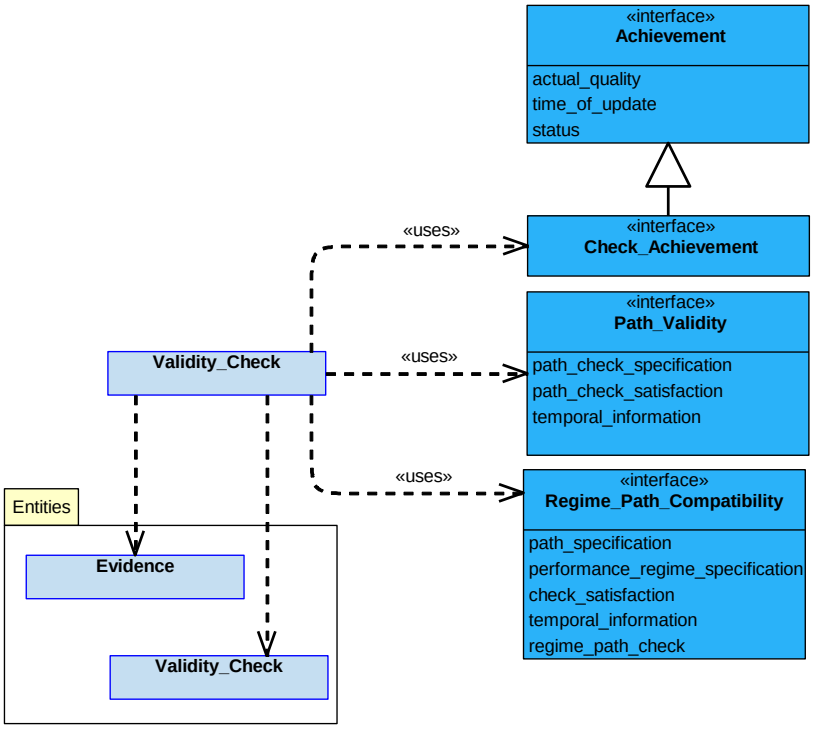

Figure 779: Validity_Check Service Definition

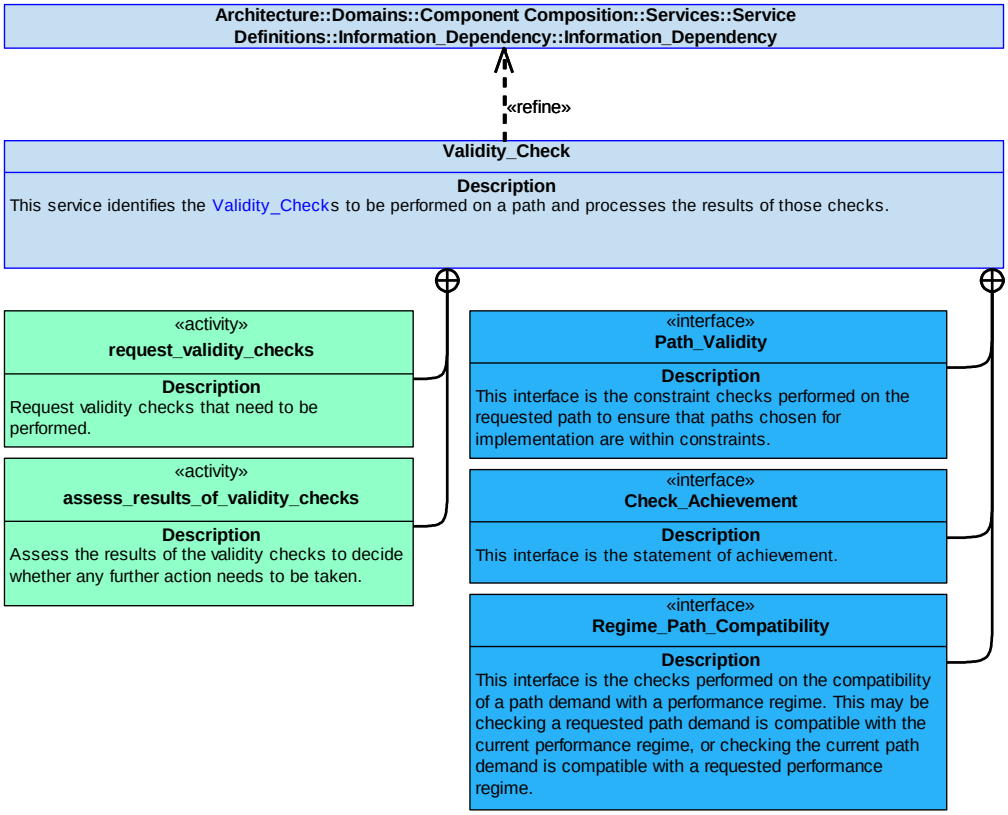

Figure 780: Validity_Check Service Policy

Validity_Check

This service identifies the Validity_Checks to be performed on a path and processes the results of those checks.

Interfaces

Path_Validity

This interface is the constraint checks performed on the requested path to ensure that paths chosen for implementation are within constraints.

Attributes

path_check_specification	Details of the check required to be carried out on the path, e.g. a check that the path does not infringe a no-fly zone.
path_check_satisfaction	A measure of whether or not the path passed all the necessary path checks (such as safety).
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.

Regime_Path_Compatibility

This interface is the checks performed on the compatibility of a path demand with a performance regime. This may be checking a requested path demand is compatible with the current performance regime, or checking the current path demand is compatible with a requested performance regime.

Attributes

path_specification	The definition of a path.
performance_regime_specification	The definition of a performance regime.
check_satisfaction	A measure of whether or not necessary checks have been passed.
temporal_information	Information covering timing, such as when the check was carried out, and for how long it remains valid.
regime_path_check	Details of the check required to be carried out, e.g. a check that the active path demand is compatible with a requested performance regime.

Activities

request_validity_checks

Request 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.

B.2.41.7.1.7 Contextual_Information

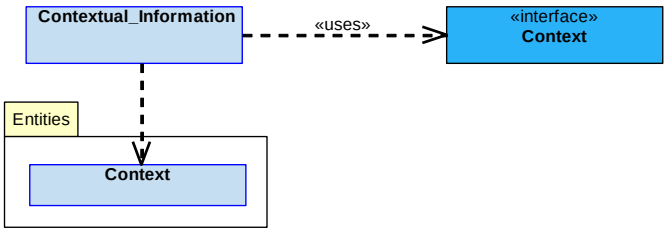

Figure 781: Contextual_Information Service Definition

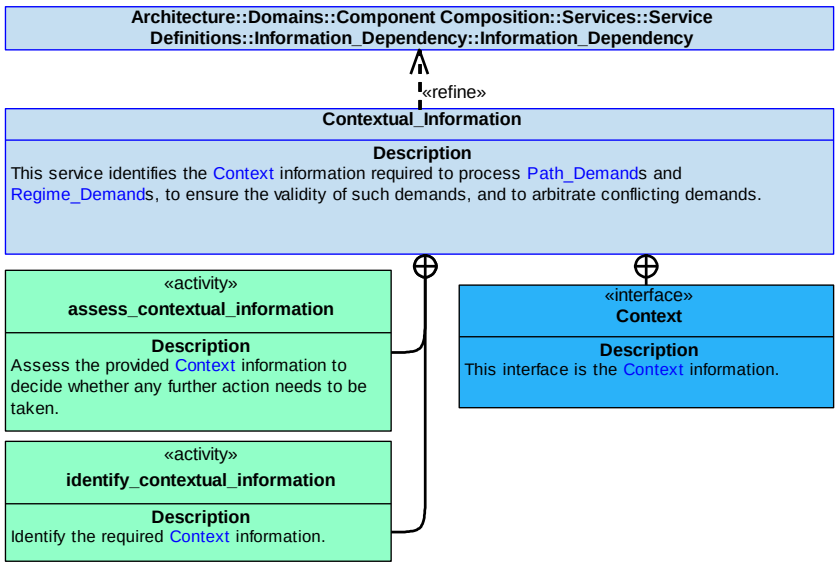

Figure 782: Contextual_Information Service Policy

Contextual_Information

This service identifies the Context information required to process Path_Demands and Regime_Demands, to ensure the validity of such demands, and to arbitrate conflicting demands.

Interface

Context

This interface is the Context information.

Activities

identify_contextual_information

Identify the required Context information.

assess_contextual_information

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

B.2.41.7.1.8 Capability

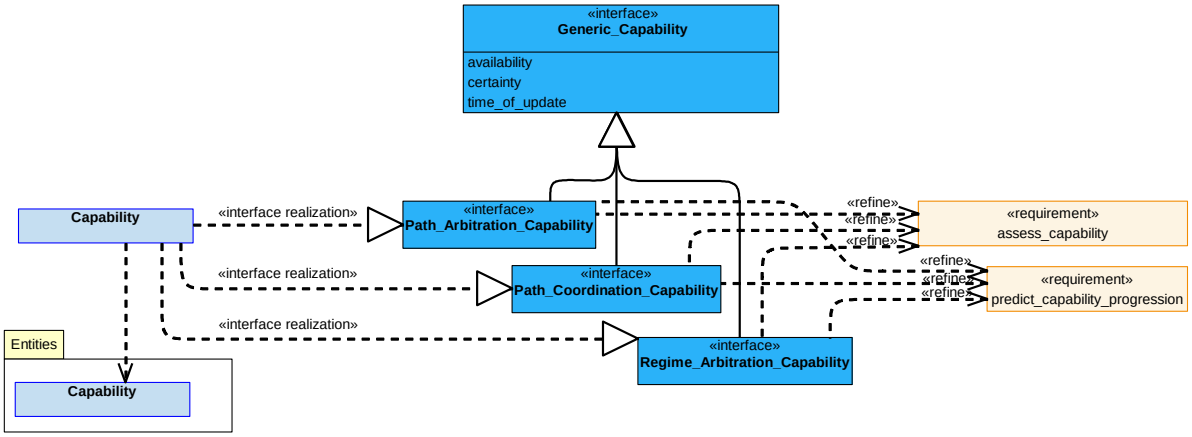

Figure 783: Capability Service Definition

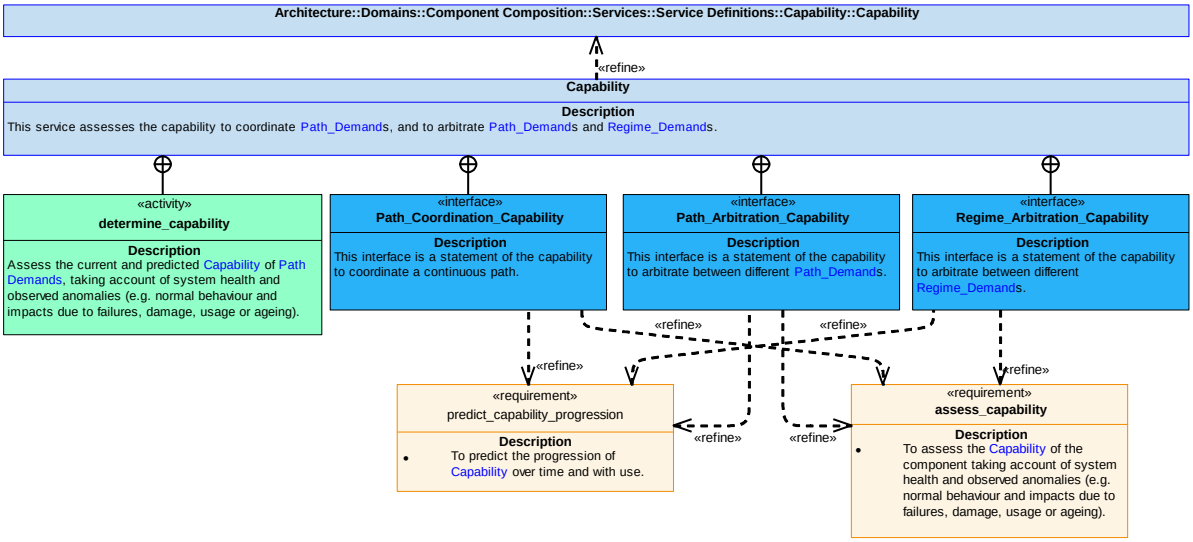

Figure 784: Capability Service Policy

Capability

This service assesses the capability to coordinate Path_Demands, and to arbitrate Path_Demands and Regime_Demands.

Interfaces

Path_Arbitration_Capability

This interface is a statement of the capability to arbitrate between different Path_Demands.

Path_Coordination_Capability

This interface is a statement of the capability to coordinate a continuous path.

Regime_Arbitration_Capability

This interface is a statement of the capability to arbitrate between different Regime_Demands.

Activity

determine_capability

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

B.2.41.7.1.9 Capability_Evidence

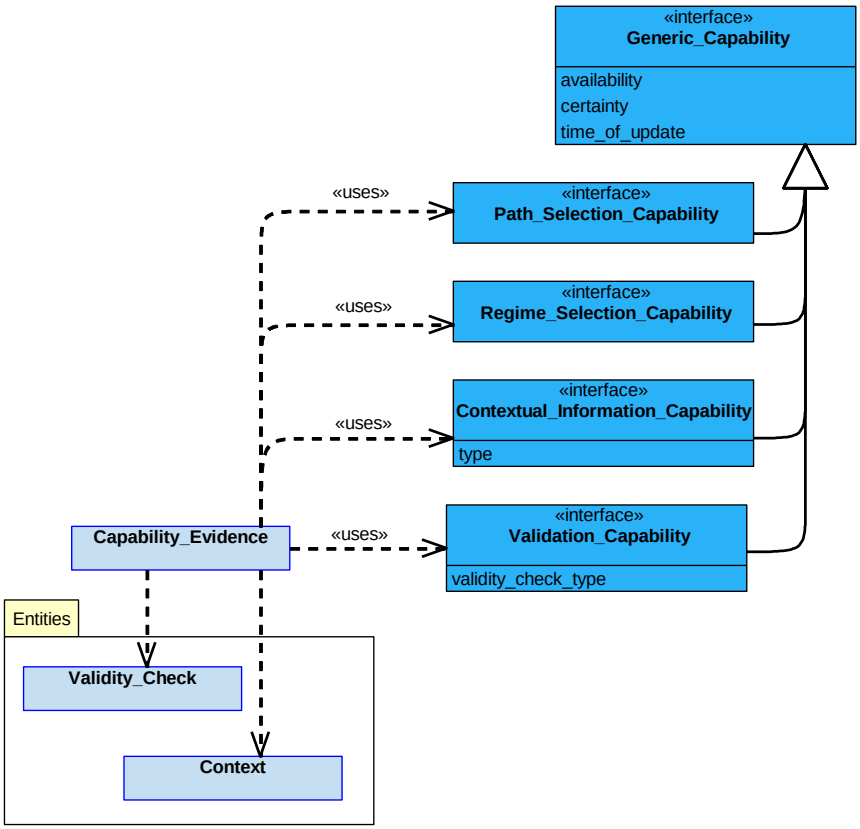

Figure 785: Capability_Evidence Service Definition

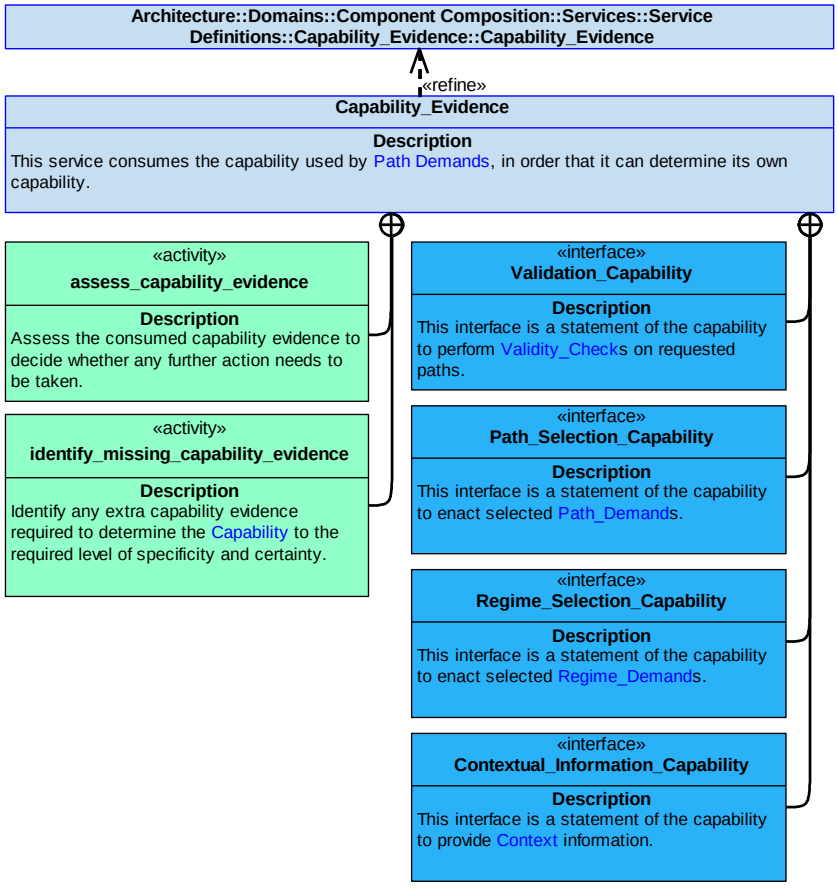

Figure 786: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the capability used by Path Demands, in order that it can determine its own capability.

Interfaces

Validation_Capability

This interface is a statement of the capability to perform Validity_Checks on requested paths.

Attribute

validity_check_type The type of Validity_Check.

Path_Selection_Capability

This interface is a statement of the capability to enact selected Path_Demands.

Regime_Selection_Capability

This interface is a statement of the capability to enact selected Regime_Demands.

Contextual_Information_Capability

This interface is a statement of the capability to provide Context information.

Attribute

type The type of Context information.

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.

B.2.41.7.2 Service Dependencies

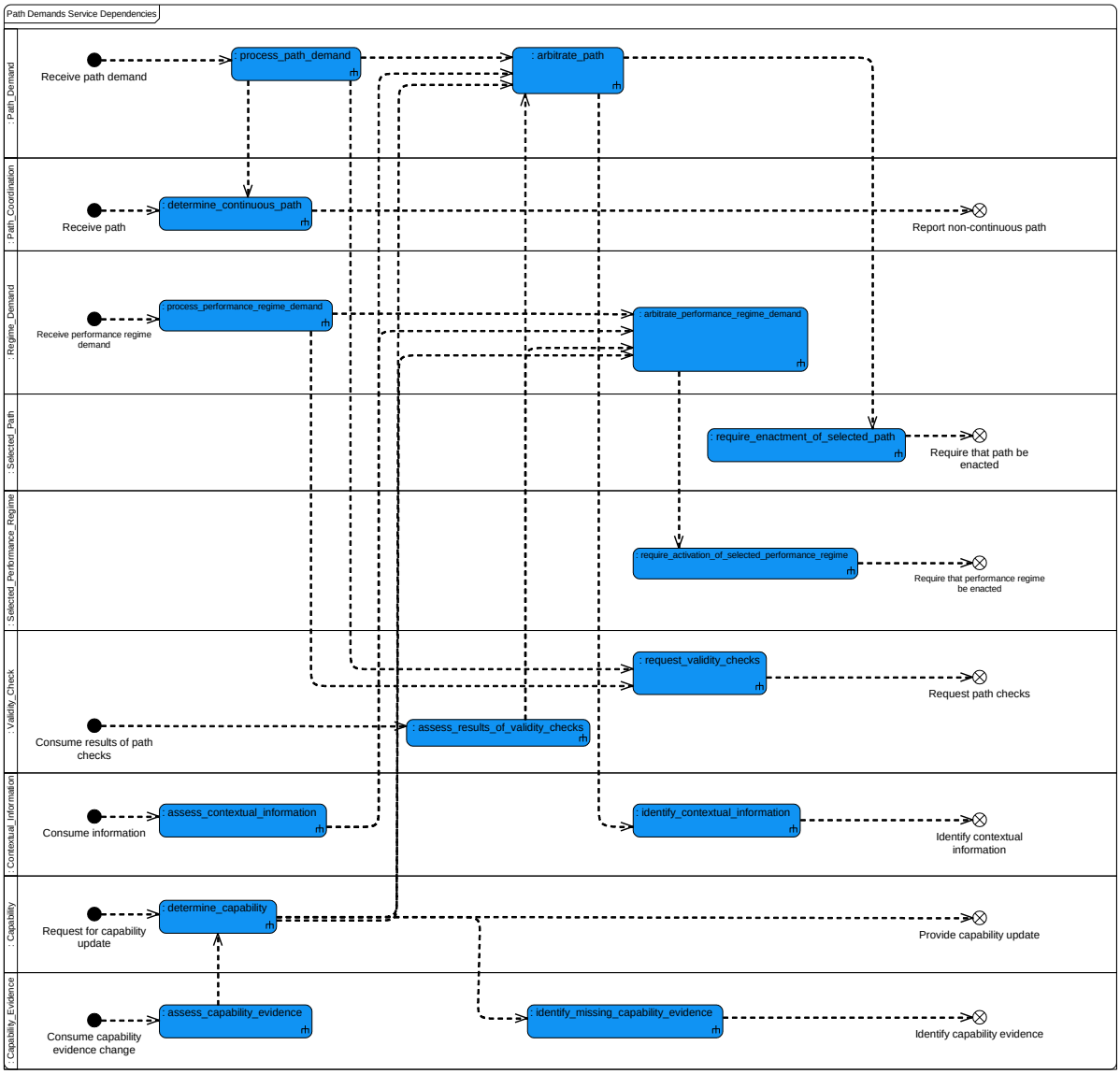

Figure 787: Path Demands Service Dependencies

B.2.42 Pointing

B.2.42.1 Role

The role of Pointing is to determine and control the orientation of equipment that can be directed in its orientation.

B.2.42.2 Overview

Control Architecture

Pointing is an action component as defined in the Control Architecture policy.

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.

B.2.42.3 Service Summary

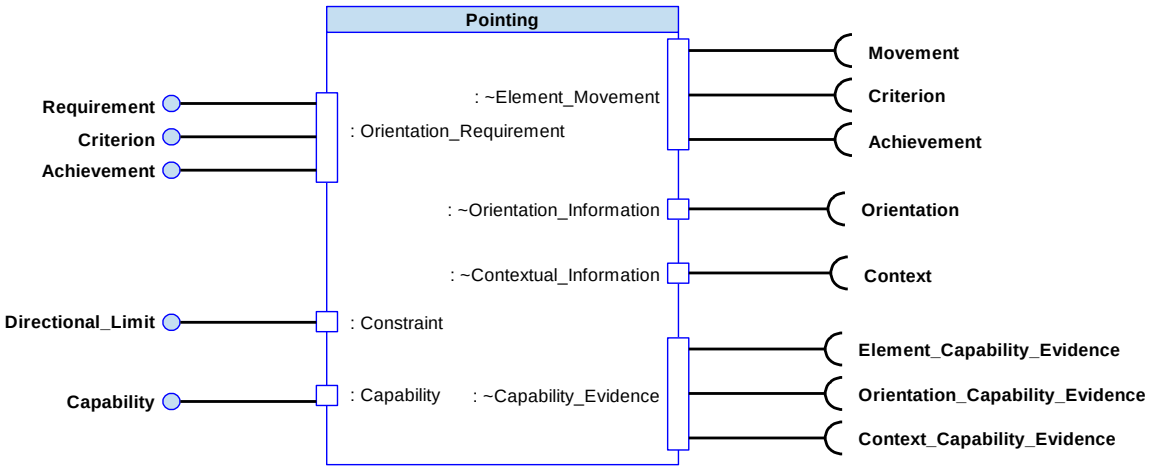

Figure 788: Pointing Service Summary

B.2.42.4 Responsibilities

capture_orientation_requirements

- To capture the orientation Requirements (e.g. required position or turn rate).

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.

coordinate_orientation_solution

- To coordinate the Orientation_Solution to ensure the individual Controllable_Element is oriented correctly (either mechanically or electrically).

determine_current_orientation

- To determine the current Orientation of a Controllable_Element.

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).

predict_capability_progression

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

identify_missing_information

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

identify_orientation_solution_in_progress_remains_feasible

- To identify whether an Orientation_Solution in progress remains feasible.

capture_orientation_measurement_criteria

- To capture provided criteria that an Orientation_Solution will be measured against.

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.

B.2.42.5 Subject Matter Semantics

The subject matter of Pointing is the orientation of Controllable_Elements.

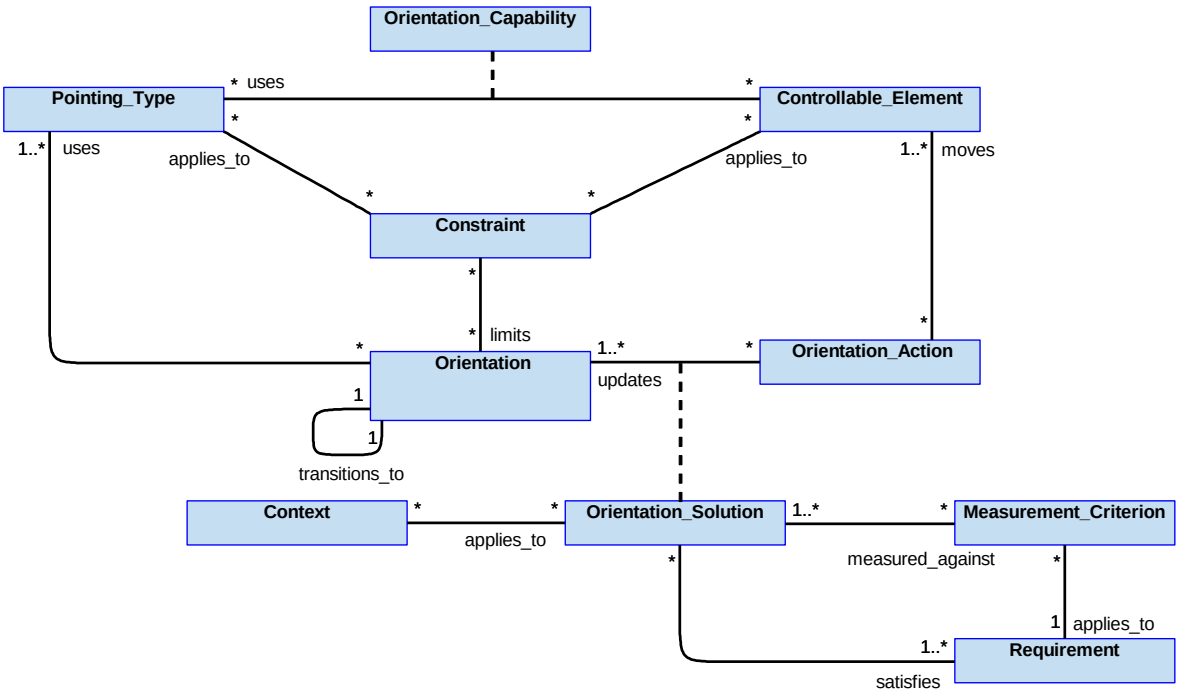

Figure 789: Pointing Semantics

B.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.

B.2.42.6 Design Rationale

B.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.

B.2.42.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Pointing:

- Data Driving - This policy will allow the component to be configured to orient different types of Controllable_Element.
- Interaction with Equipment - This policy 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 of pointing for different types of pointing (e.g. two or three axis rotation pointing).

B.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.

B.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.

B.2.42.7 Services

B.2.42.7.1 Service Definitions

B.2.42.7.1.1 Orientation_Requirement

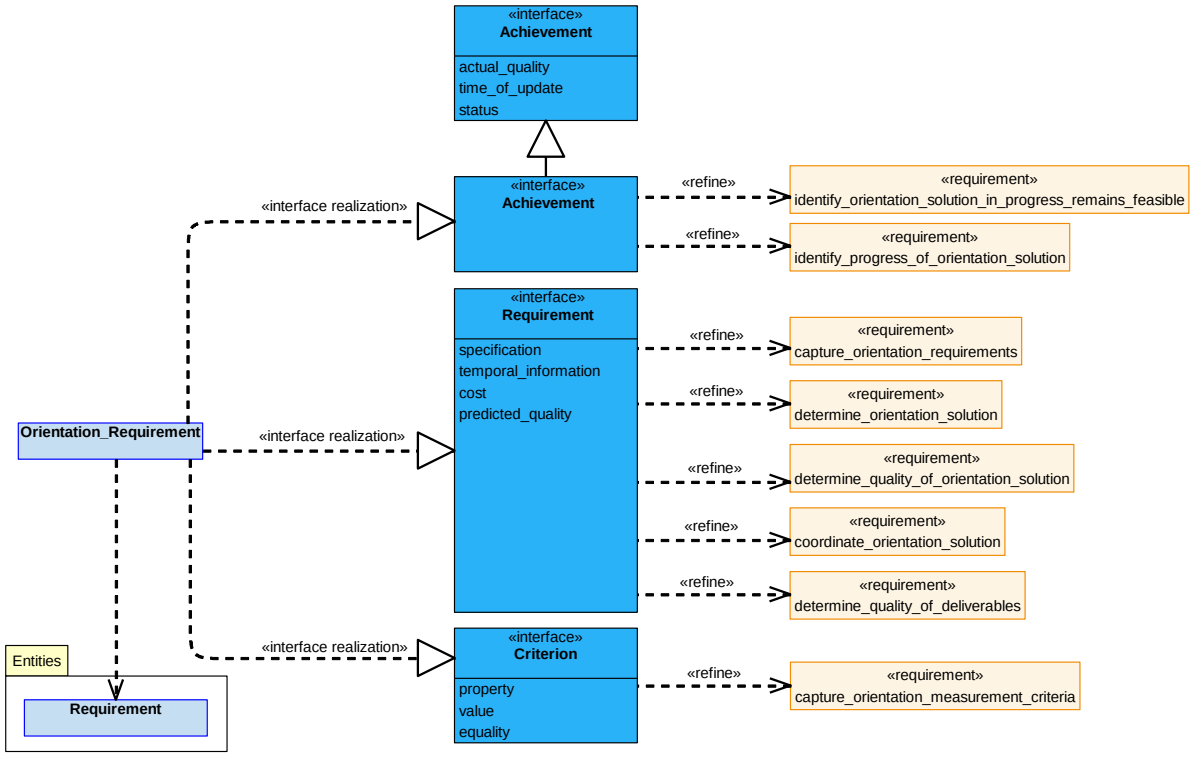

Figure 790: Orientation_Requirement Service Definition

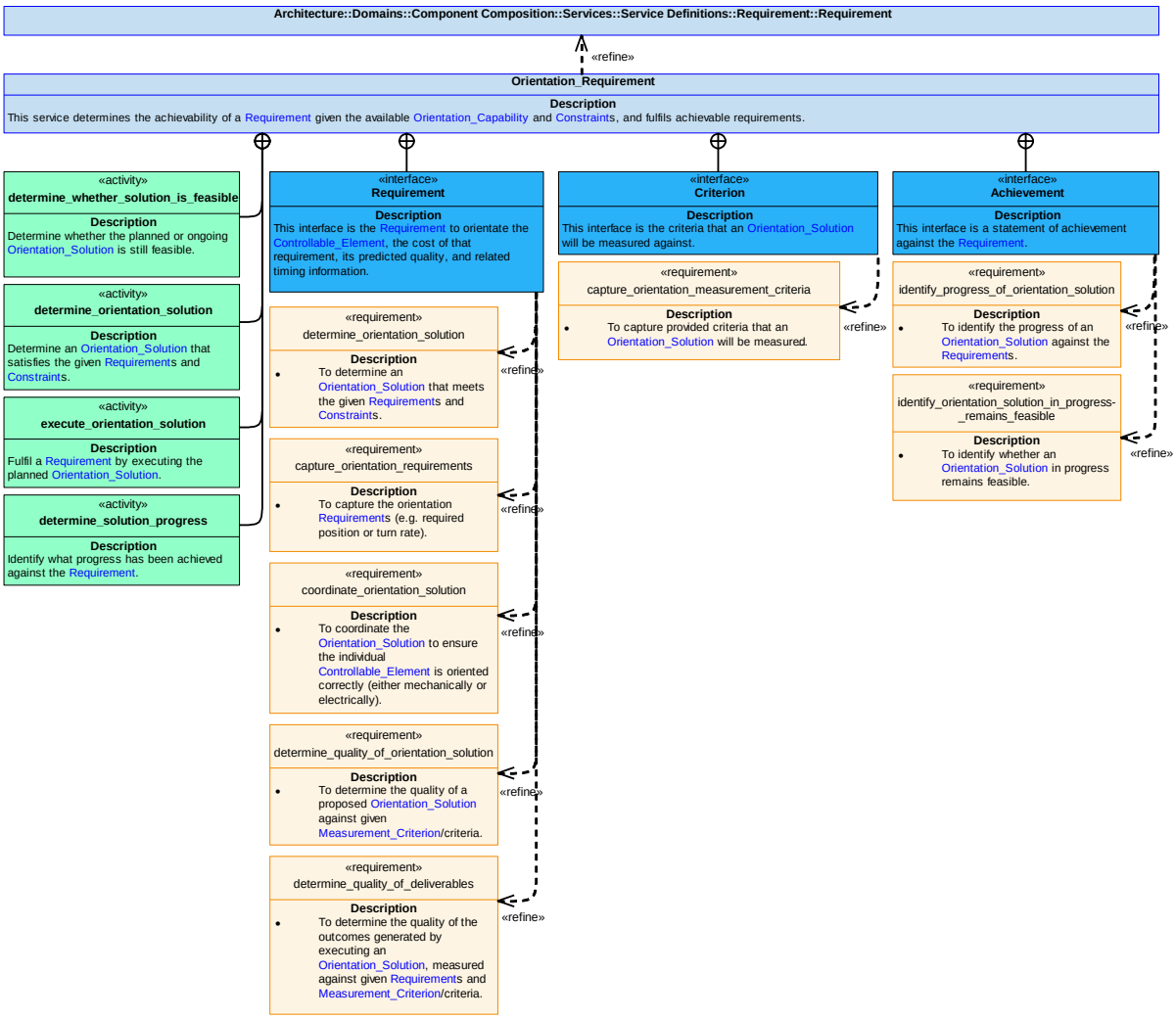

Figure 791: Orientation_Requirement Service Policy

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.

B.2.42.7.1.2 Element_Movement

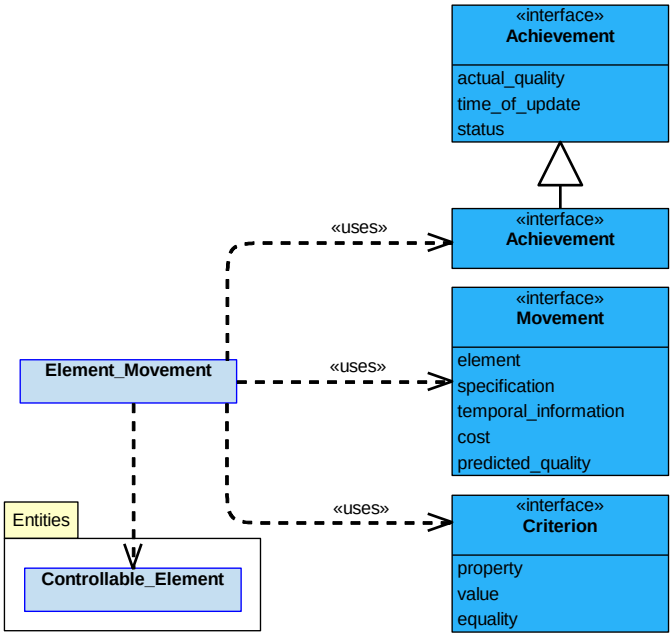

Figure 792: Element_Movement Service Definition

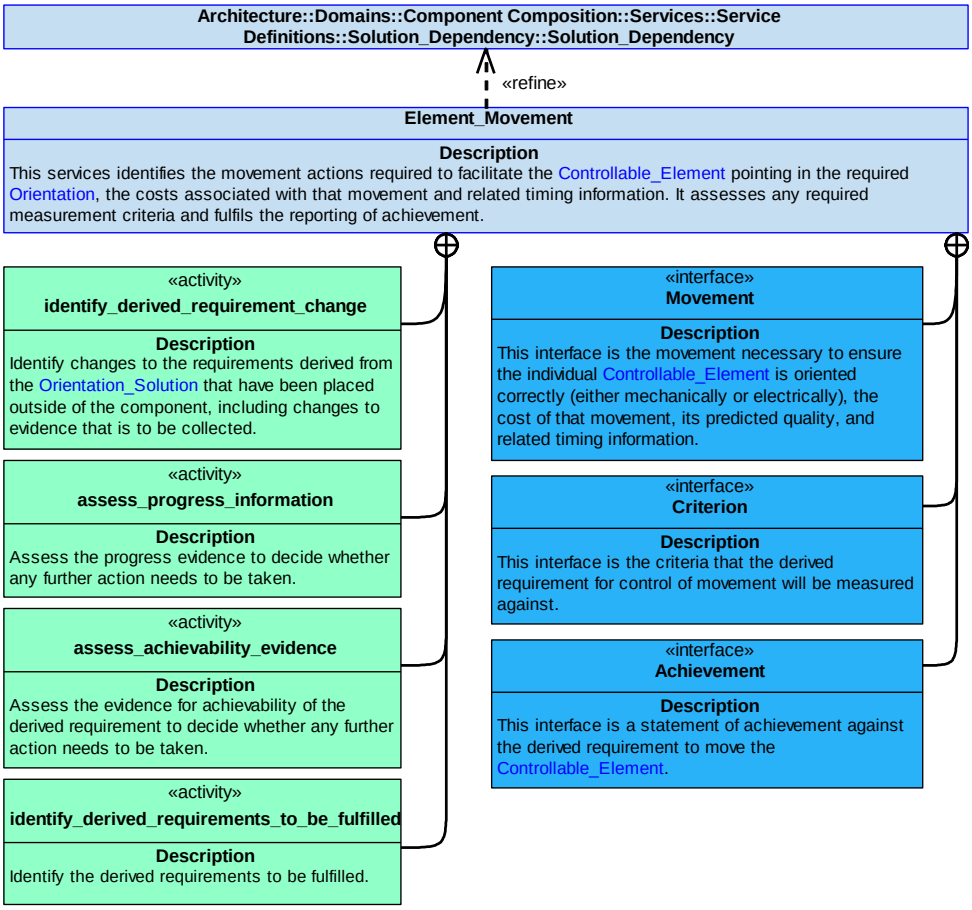

Figure 793: Element_Movement Service Policy

Element_Movement

This services 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 electrically), 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.

B.2.42.7.1.3 Contextual_Information

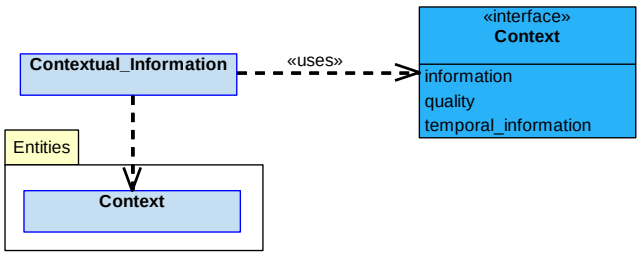

Figure 794: Contextual_Information Service Definition

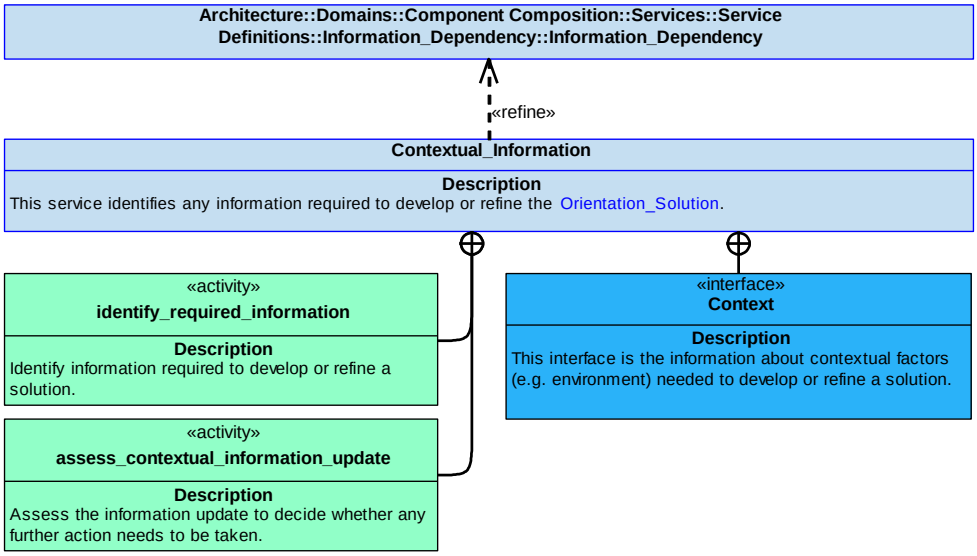

Figure 795: Contextual_Information Service Policy

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.

B.2.42.7.1.4 Orientation_Information

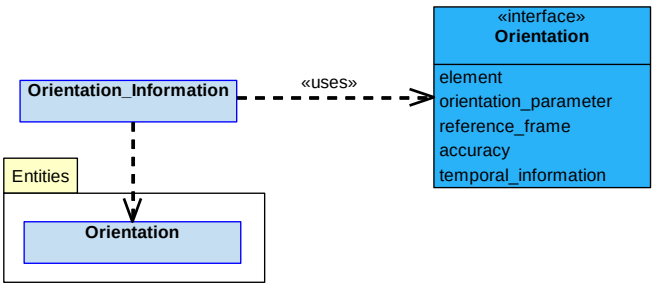

Figure 796: Orientation_Information Service Definition

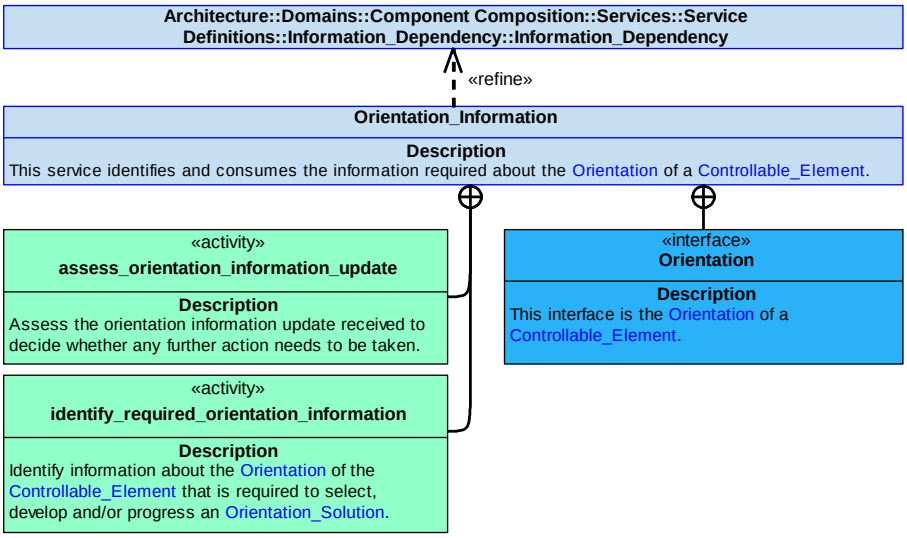

Figure 797: Orientation_Information Service Policy

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.

B.2.42.7.1.5 Constraint

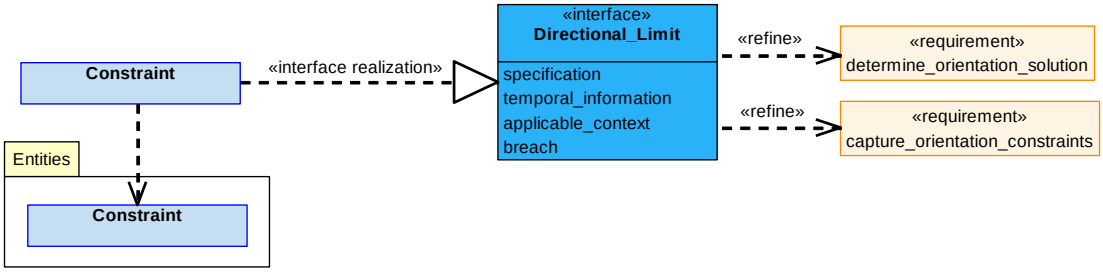

Figure 798: Constraint Service Definition

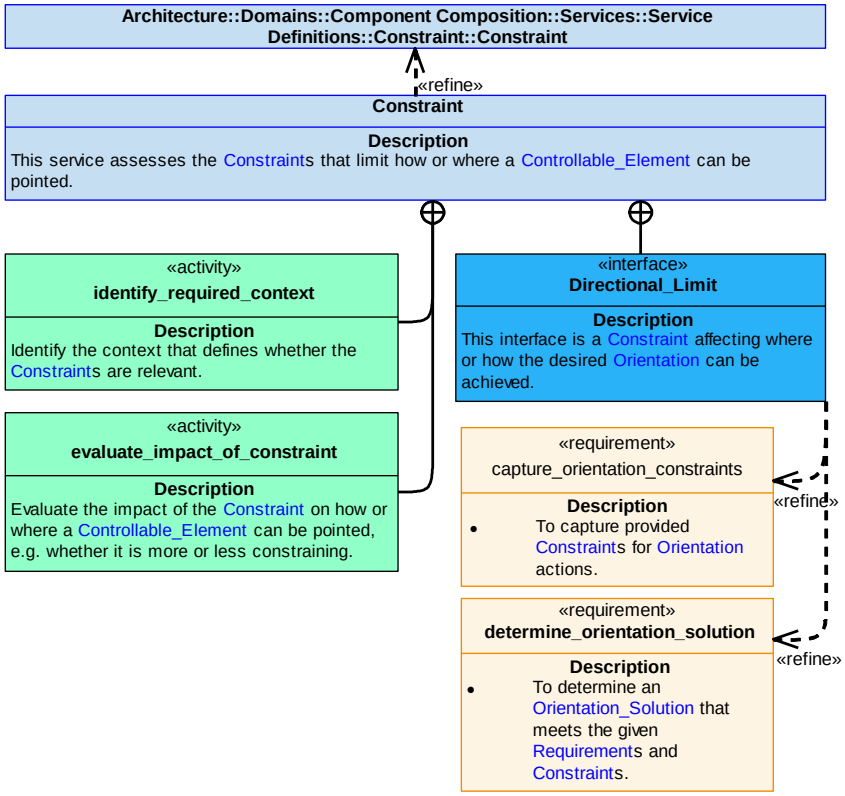

Figure 799: Constraint Service Policy

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.

B.2.42.7.1.6 Capability

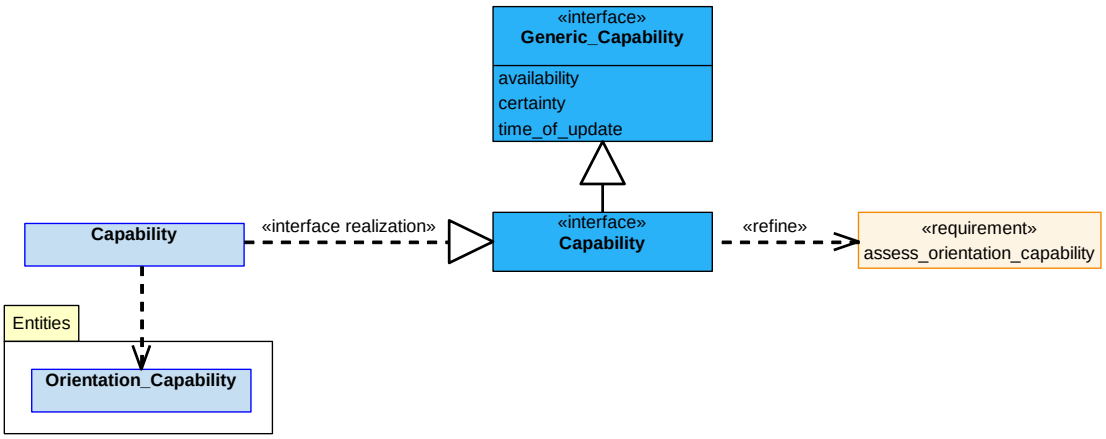

Figure 800: Capability Service Definition

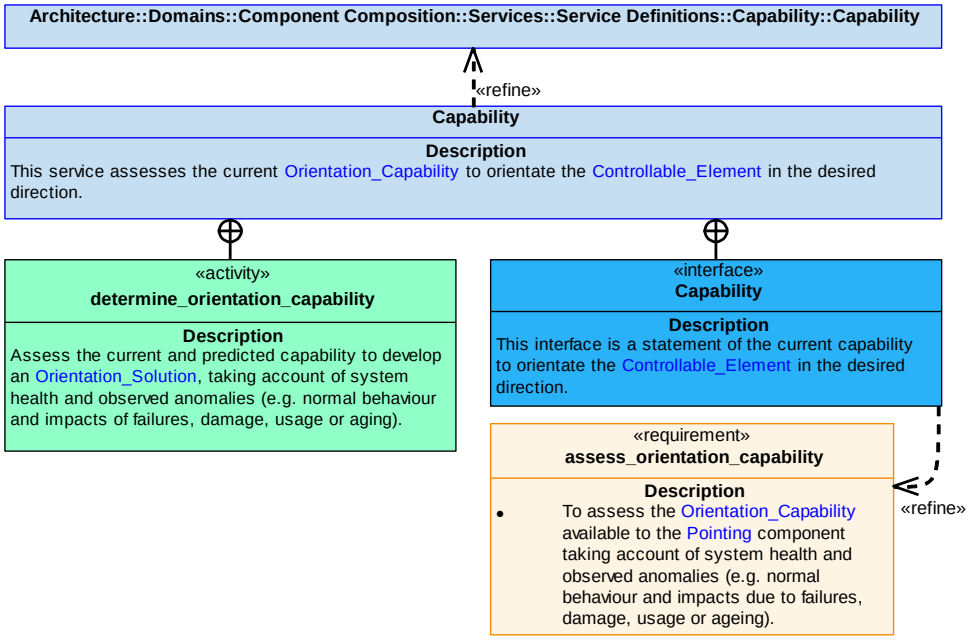

Figure 801: Capability Service Policy

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).

B.2.42.7.1.7 Capability_Evidence

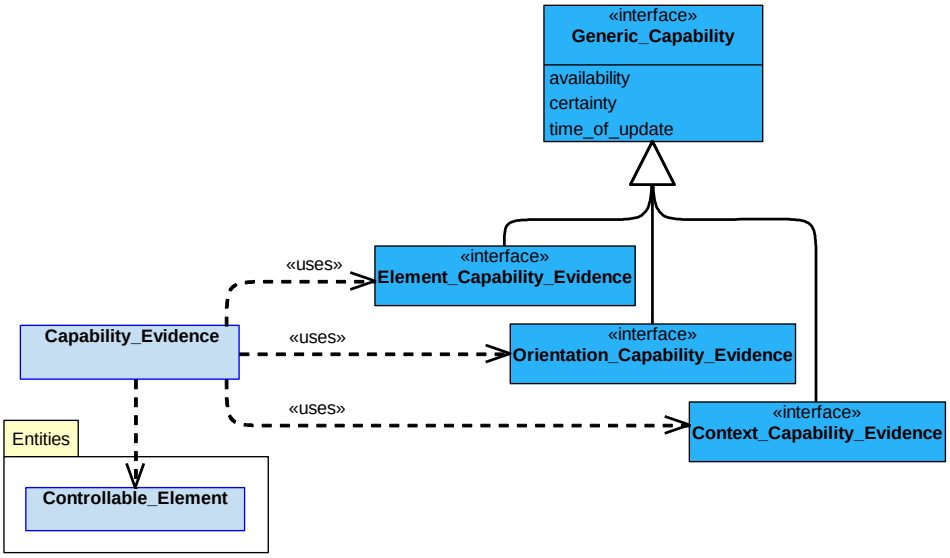

Figure 802: Capability_Evidence Service Definition

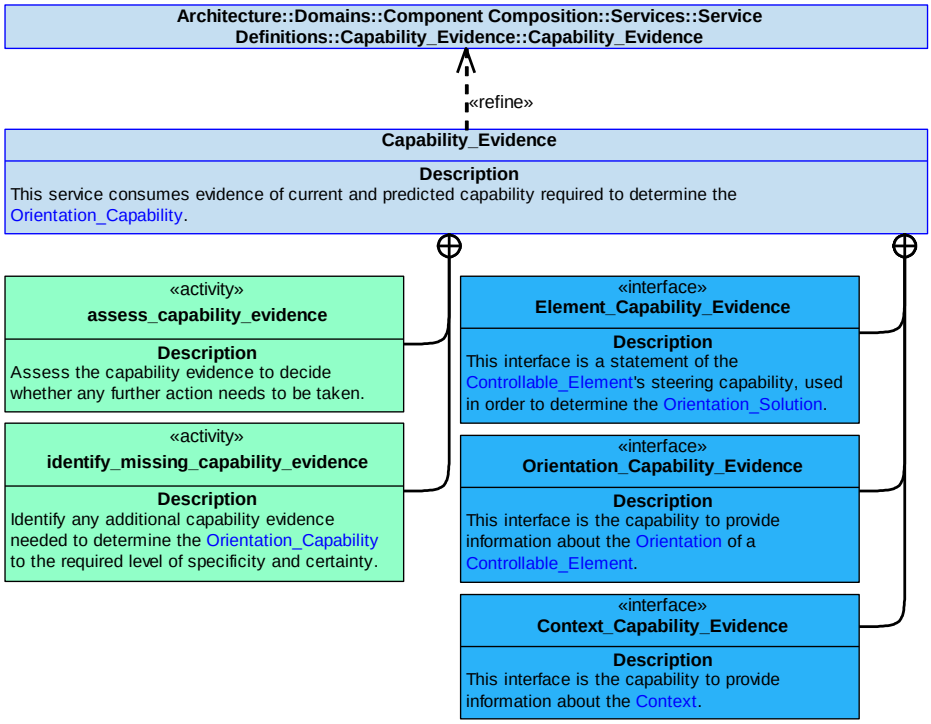

Figure 803: Capability_Evidence Service Policy

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.

B.2.42.7.2 Service Dependencies

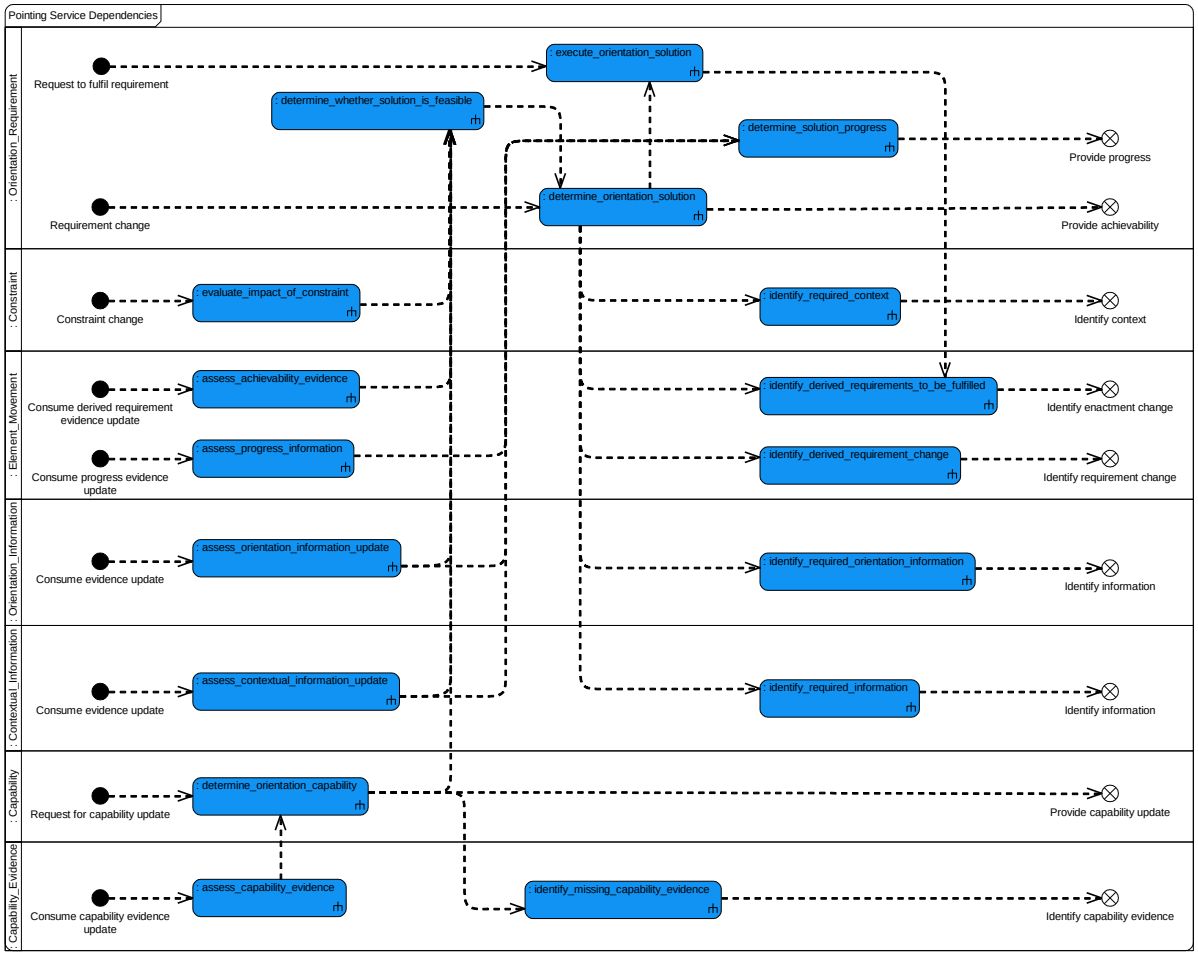

Figure 804: Pointing Service Dependencies

B.2.43 Power

B.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.

B.2.43.2 Overview

Control Architecture

Power is a resource component as defined in the Control Architecture policy.

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.

B.2.43.3 Service Summary

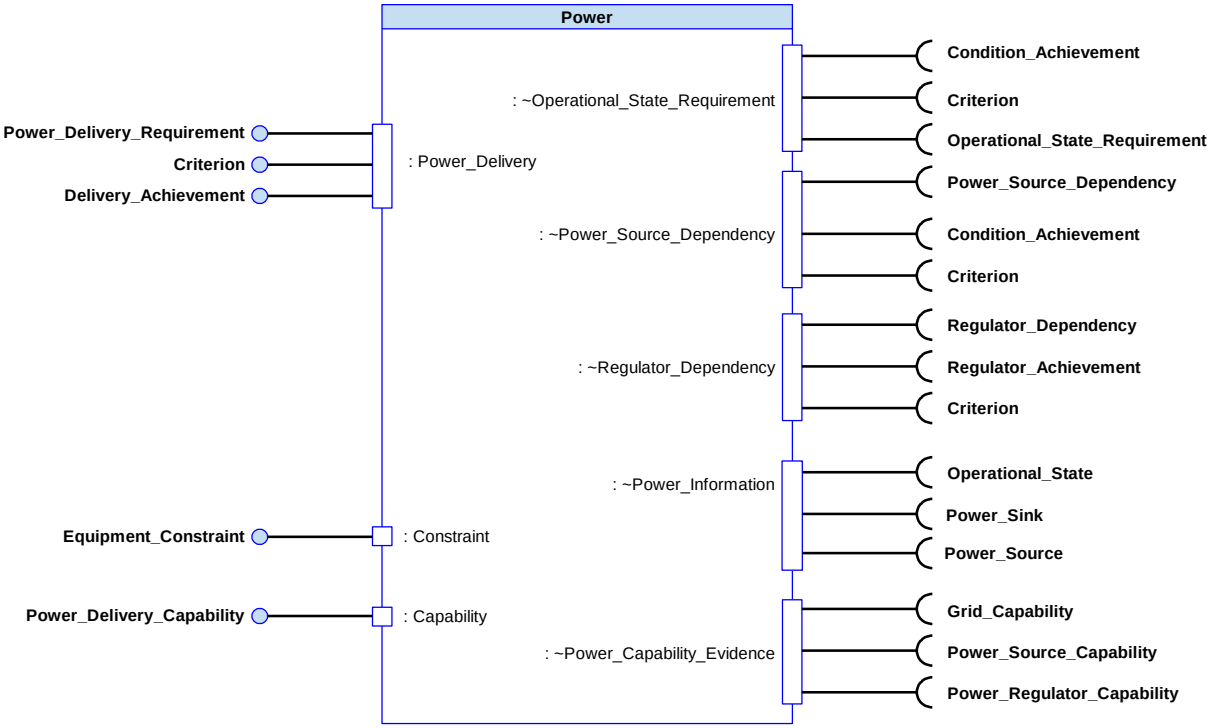

Figure 805: Power Service Summary

B.2.43.4 Responsibilities

assess_power_capability

- To assess the current capability to manage Power_Sources, Power_Sinks and Power_Regulators.

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.

capture_power_solution_requirements

- To capture Power_Requirement for power Power_Delivery_Solutions.

coordinate_power_resources

- To coordinate the use of Power_Sources, Power_Sinks and Power_Regulators to meet the requirements of a Power_Delivery_Solution.

determine_power_solution

- To determine the appropriate Power_Delivery_Solution to meet captured Power_Requirements and Equipment_Constraints.

determine_quality_of_delivered_solution

- To determine the Delivered_Quality of the delivered Power_Delivery_Solution measured against Power_Requirement in terms of given Measurement_Criterion.

identify_missing_information

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

identify_whether_power_requirement_remains_achievable

- To identify if a Power_Requirement is still achievable given current resources.

identify_progress_of_power_solution

- To identify the progress of a Power_Delivery_Solution against the Power_Requirements.

predict_power_capability_progression

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

determine_quality_of_designed_solution

- To determine the Designed_Quality of the Power_Delivery_Solution against one or more given Measurement_Criterion.

B.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.

Exclusions

The subject matter of Power does not include:

- The allocation of power, only its distribution.

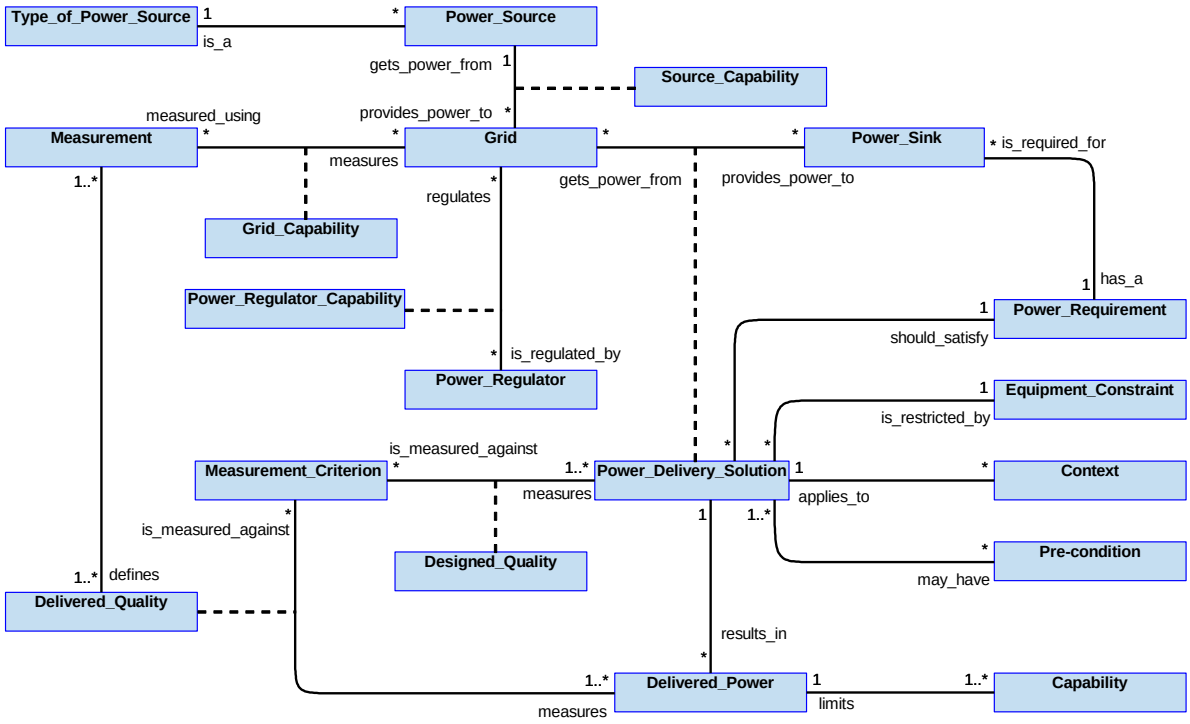

Figure 806: Power Semantics

B.2.43.5.1 Entities

Capability

The capability to provide the required power to a Power_Sink.

Context

The situation within which a Power_Delivery_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_Delivery_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_Delivery_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. minimum vehicle speed to enable a wind turbine to rotate.

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).

B.2.43.6 Design Rationale

B.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.

B.2.43.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.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.

B.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).

B.2.43.7 Services

B.2.43.7.1 Service Definitions

B.2.43.7.1.1 Power_Delivery

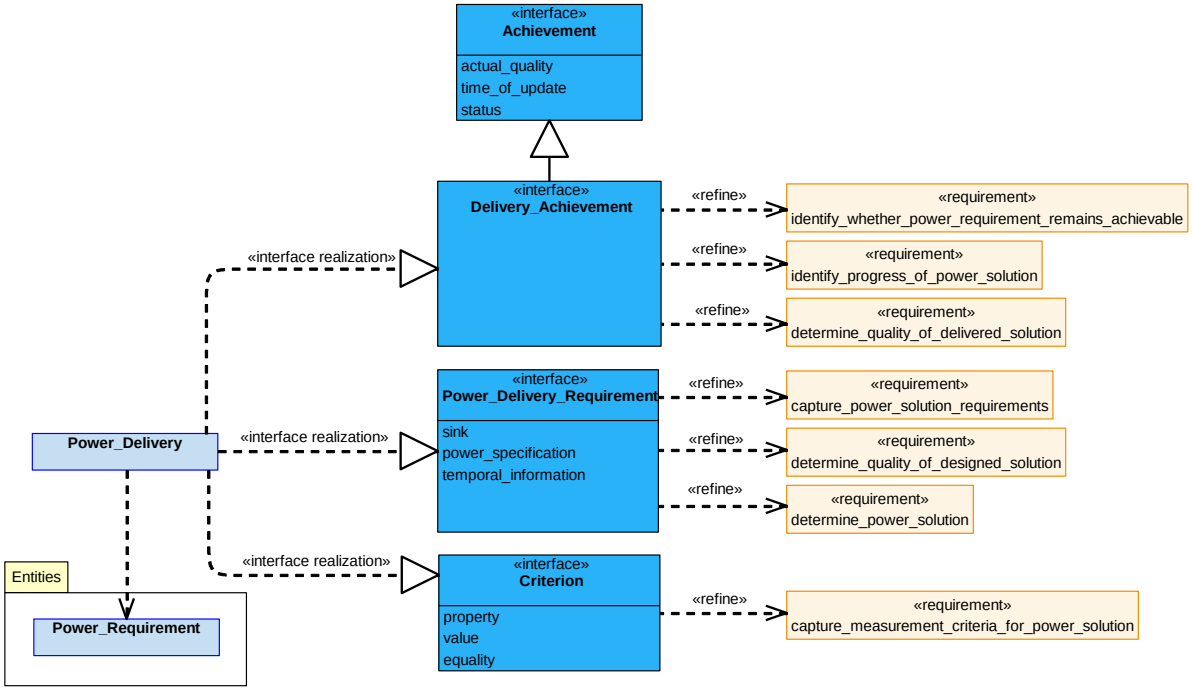

Figure 807: Power_Delivery Service Definition

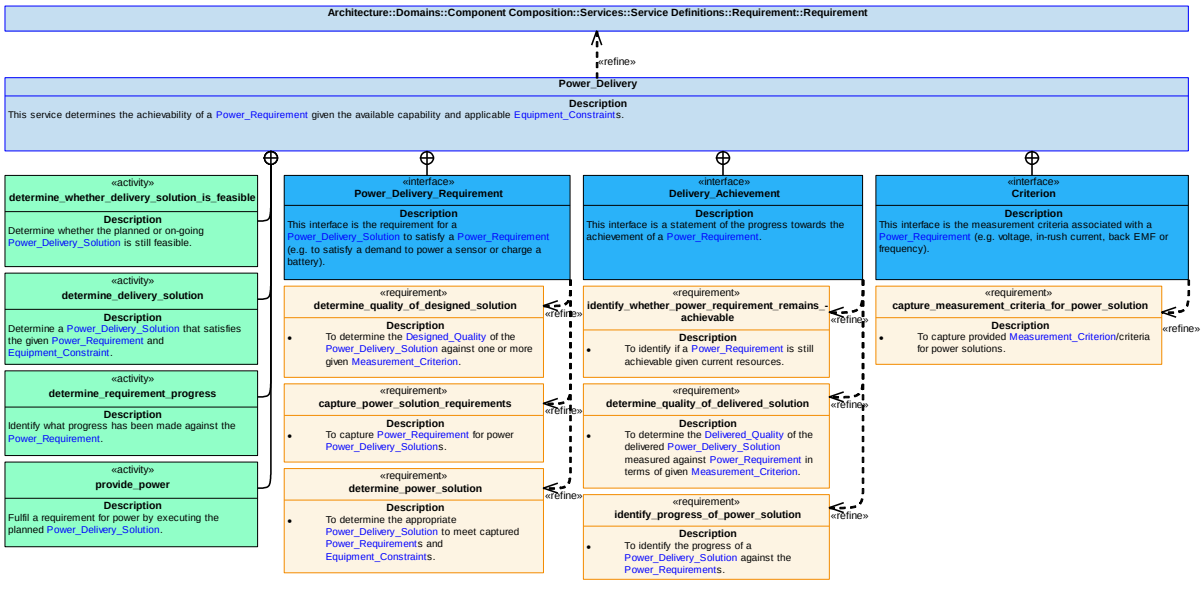

Figure 808: Power_Delivery Service Policy

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_Delivery_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_Delivery_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_Delivery_Solution.

determine_delivery_solution

Determine a Power_Delivery_Solution that satisfies the given Power_Requirement and Equipment_Constraint.

determine_whether_delivery_solution_is_feasible

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

determine_requirement_progress

Identify what progress has been made against the Power_Requirement.

B.2.43.7.1.2 Operational_State_Requirement

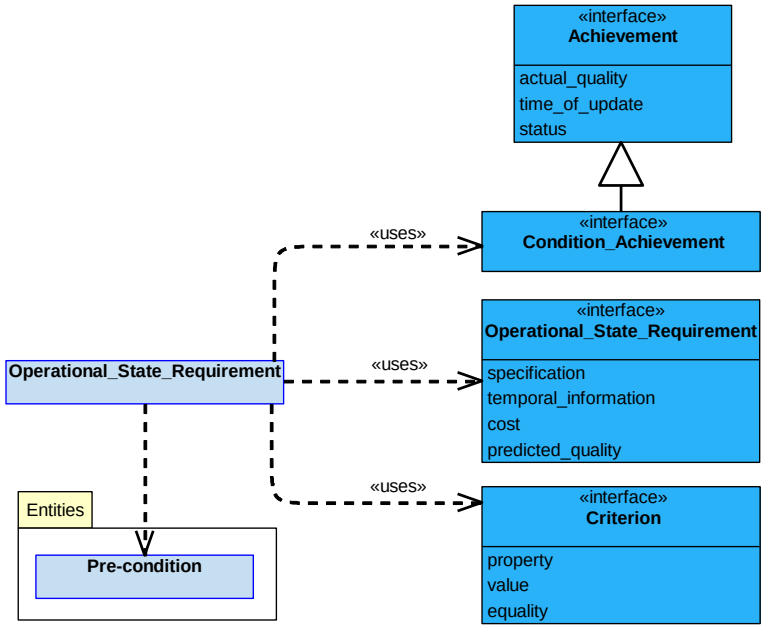

Figure 809: Operational_State_Requirement Service Definition

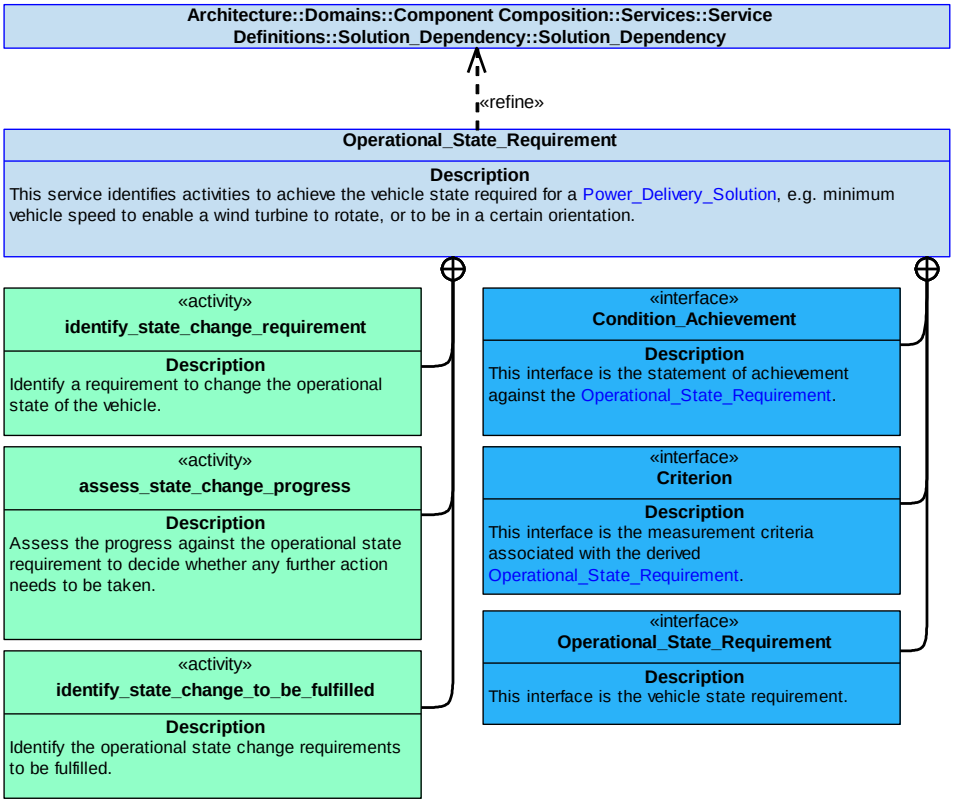

Figure 810: Operational_State_Requirement Service Policy

Operational_State_Requirement

This service identifies activities to achieve the vehicle state required for a Power_Delivery_Solution, e.g. minimum vehicle speed to enable a wind turbine to rotate, or to be in a certain orientation.

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.

B.2.43.7.1.3 Power_Source_Dependency

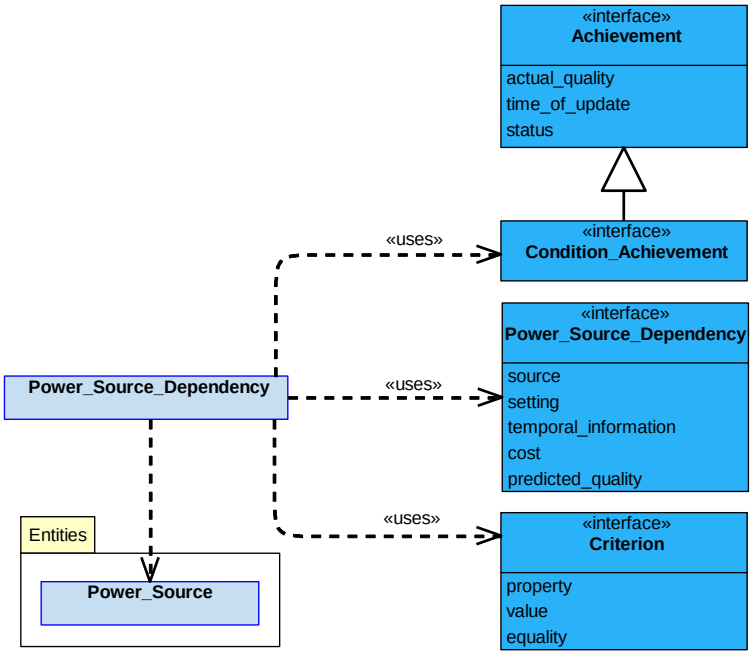

Figure 811: Power_Source_Dependency Service Definition

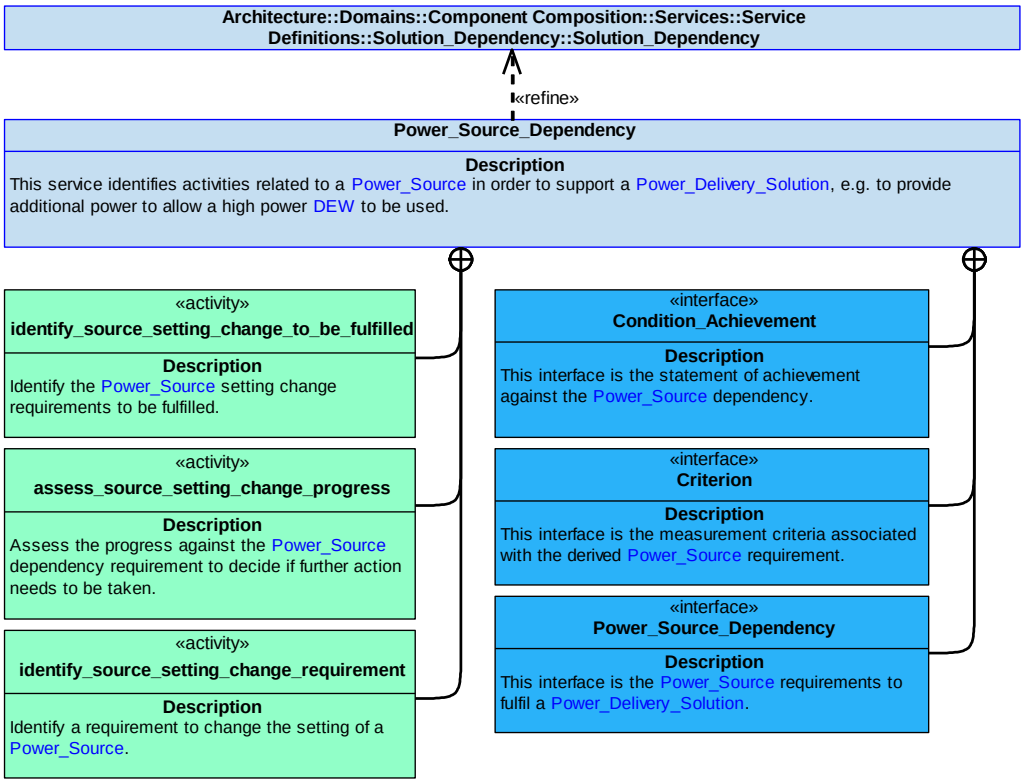

Figure 812: Power_Source_Dependency Service Policy

Power_Source_Dependency

This service identifies activities related to a Power_Source in order to support a Power_Delivery_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_Delivery_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.

B.2.43.7.1.4 Regulator_Dependency

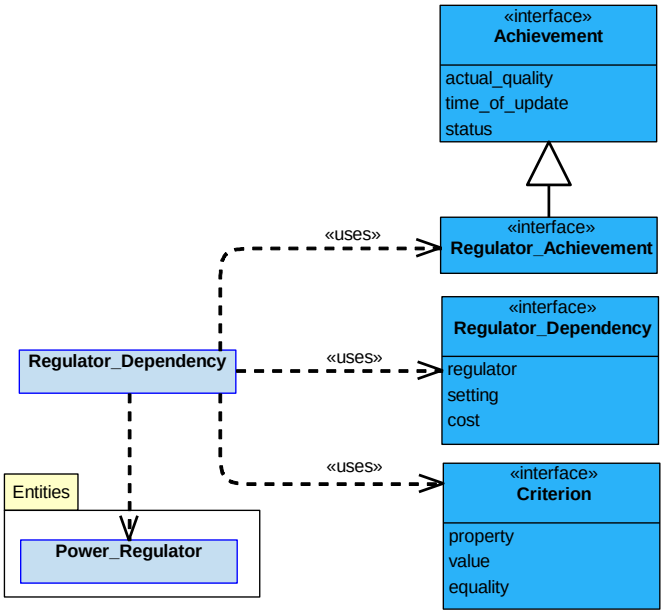

Figure 813: Regulator_Dependency Service Definition

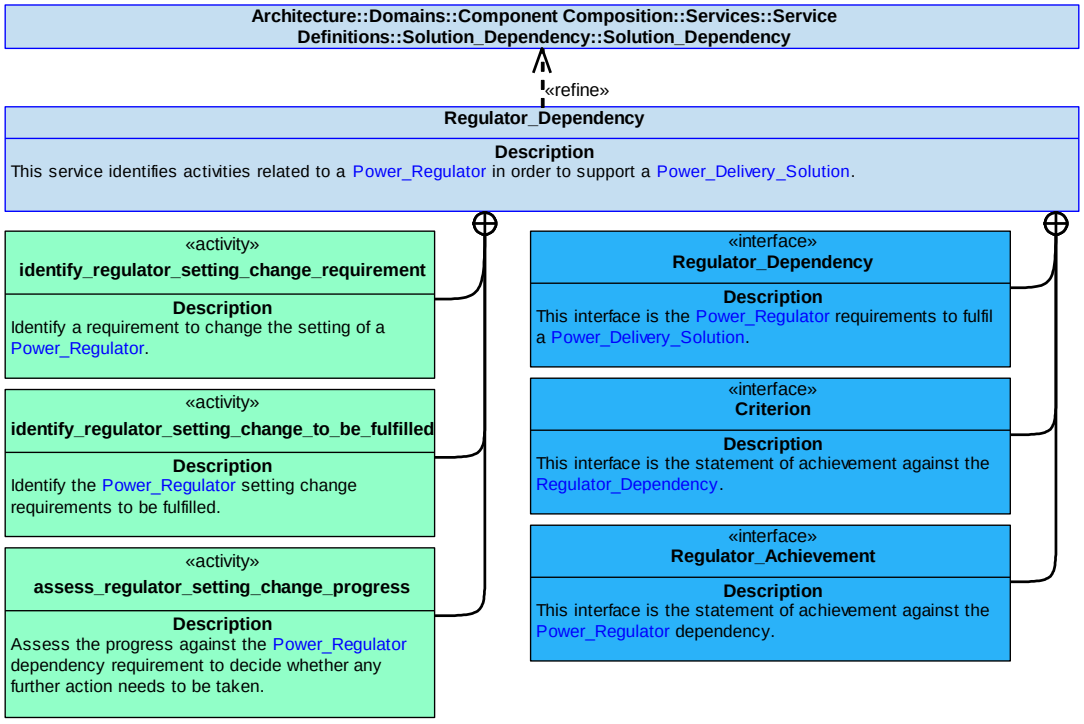

Figure 814: Regulator_Dependency Service Policy

Regulator_Dependency

This service identifies activities related to a Power_Regulator in order to support a Power_Delivery_Solution.

Interfaces

Regulator_Dependency

This interface is the Power_Regulator requirements to fulfil a Power_Delivery_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.

B.2.43.7.1.5 Power_Information

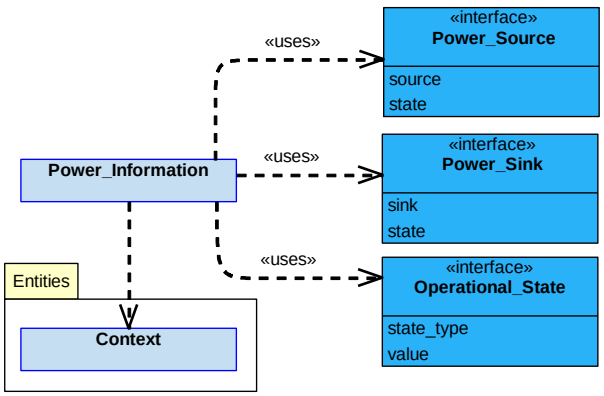

Figure 815: Power_Information Service Definition

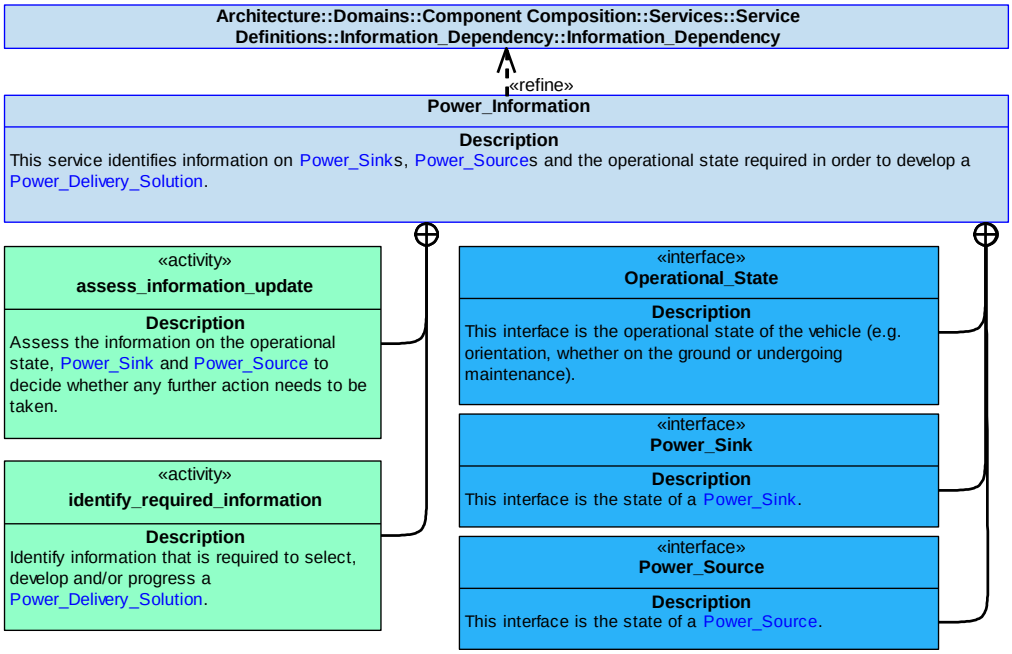

Figure 816: Power_Information Service Policy

Power_Information

This service identifies information on Power_Sinks, Power_Sources and the operational state required in order to develop a Power_Delivery_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.

Activities

assess_information_update

Assess the information on the operational state, Power_Sink and Power_Source 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 Power_Delivery_Solution.

B.2.43.7.1.6 Constraint

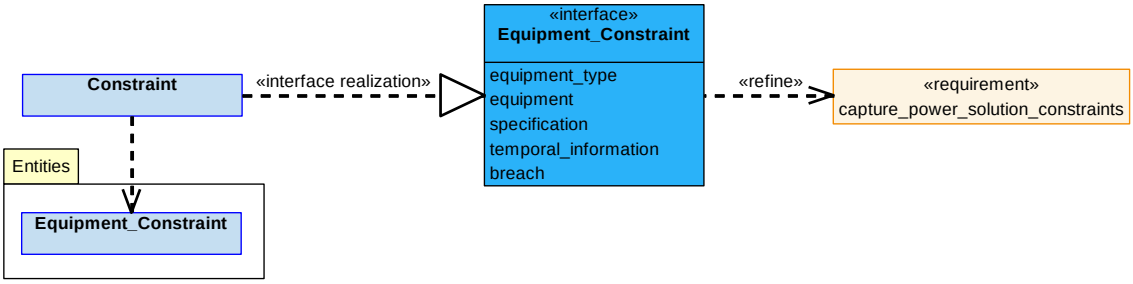

Figure 817: Constraint Service Definition

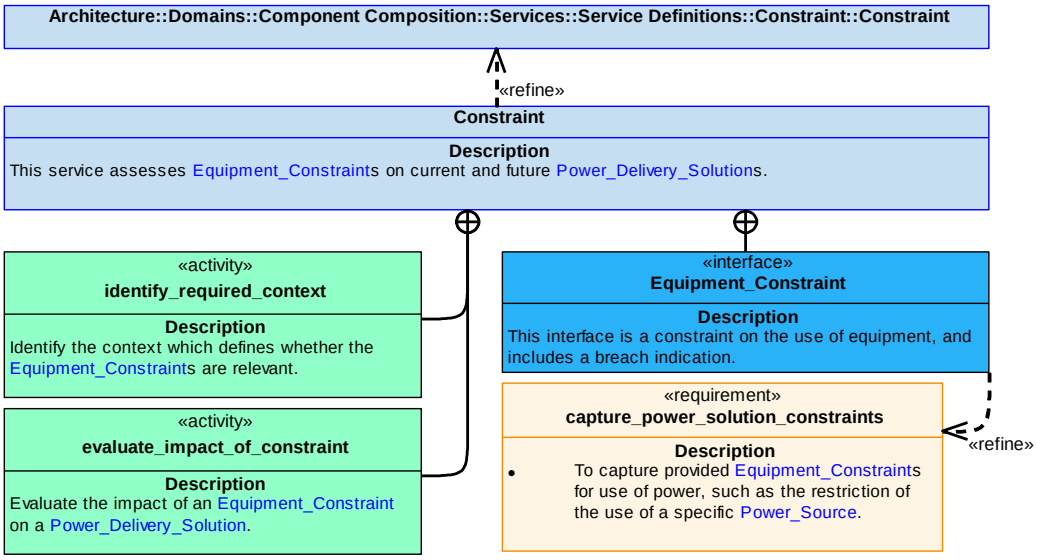

Figure 818: Constraint Service Policy

Constraint

This service assesses Equipment_Constraints on current and future Power_Delivery_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_Delivery_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_Delivery_Solution.

identify_required_context

Identify the context which defines whether the Equipment_Constraints are relevant.

B.2.43.7.1.7 Capability

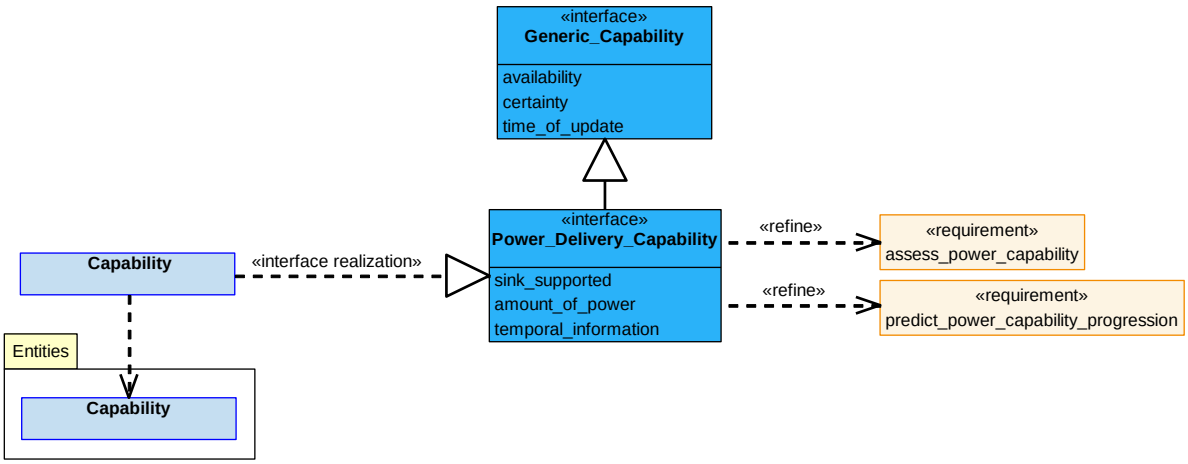

Figure 819: Capability Service Definition

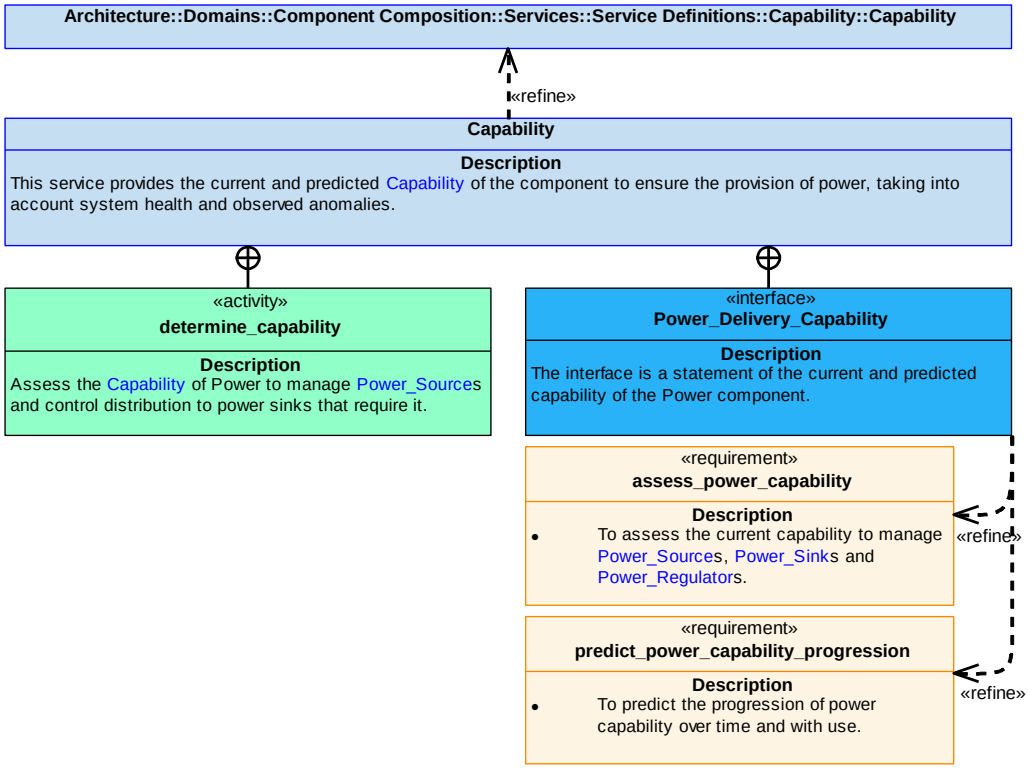

Figure 820: Capability Service Policy

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.

B.2.43.7.1.8 Power_Capability_Evidence

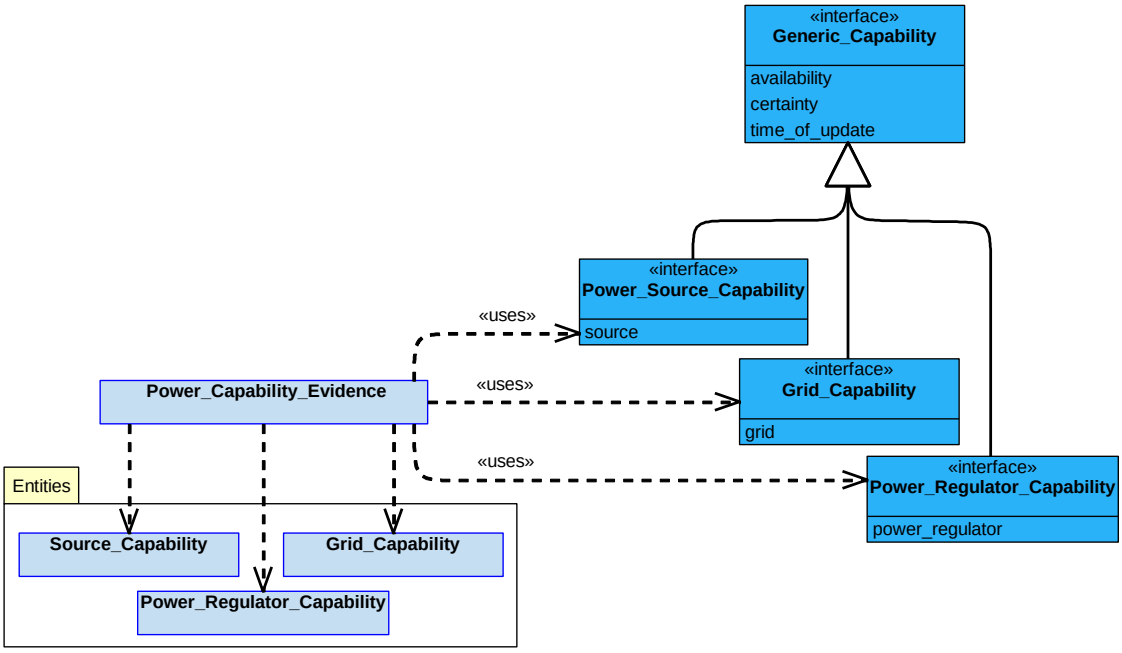

Figure 821: Power_Capability_Evidence Service Definition

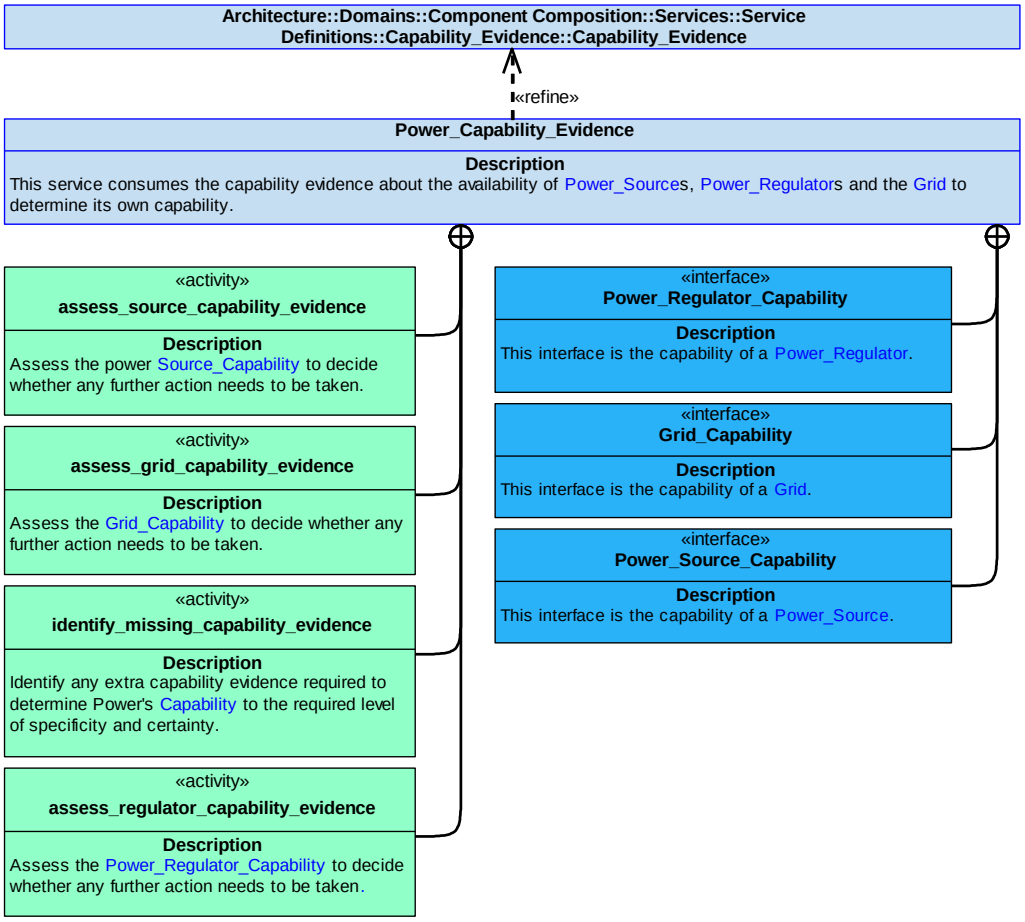

Figure 822: Power_Capability_Evidence Service Policy

Power_Capability_Evidence

This service consumes the capability evidence about the availability of Power_Sources, Power_Regulators and the Grid 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.

Activities

assess_source_capability_evidence

Assess the power Source_Capability to decide whether any further action needs to be taken.

assess_grid_capability_evidence

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

assess_regulator_capability_evidence

Assess the Power_Regulator_Capability 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.

B.2.43.7.2 Service Dependencies

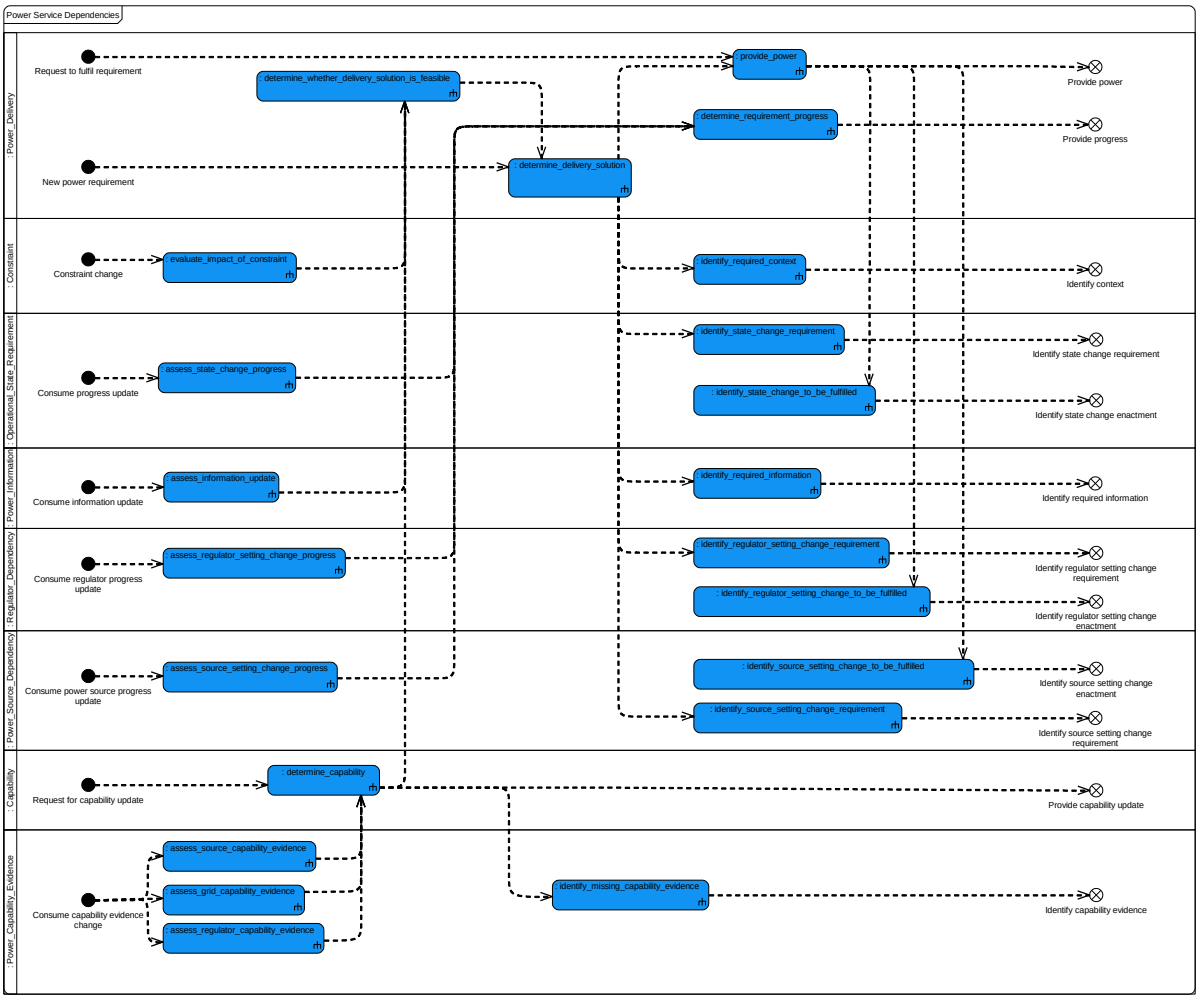

Figure 823: Power Service Dependencies

B.2.44 Propulsion

B.2.44.1 Role

The role of Propulsion is to control the propulsion of an air vehicle and the state of the propulsion units.

B.2.44.2 Overview

Control Architecture

Propulsion is a resource component as defined in the Control Architecture policy.

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.

B.2.44.3 Service Summary

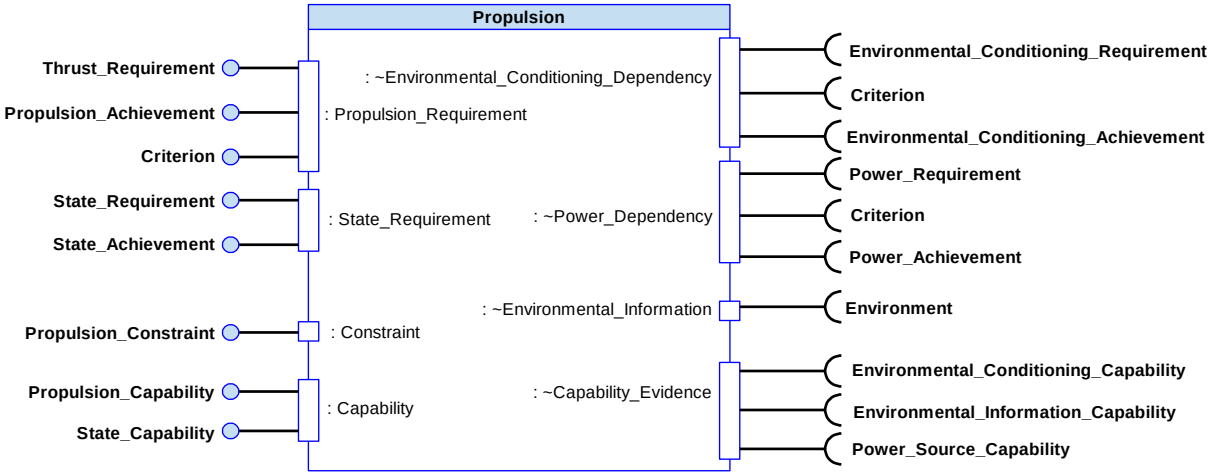

Figure 824: Propulsion Service Summary

B.2.44.4 Responsibilities

capture_propulsion_requirements

- To capture Thrust_Requirements.

capture_state_requirements

- To capture State_Requirements.

capture_propulsion_constraints

- To capture provided Constraints for use of propulsion resources.

control_propulsion

- To control the Propulsion_Unit_Setting(s) for the output required.

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).

capture_measurement_criteria_for_propulsion_solution

- To capture provided Measurement_Criterion for propulsion solutions.

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.

identify_progress_of_delivered_thrust

- To identify the progress of Thrust against the Thrust_Requirements.

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.

B.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.

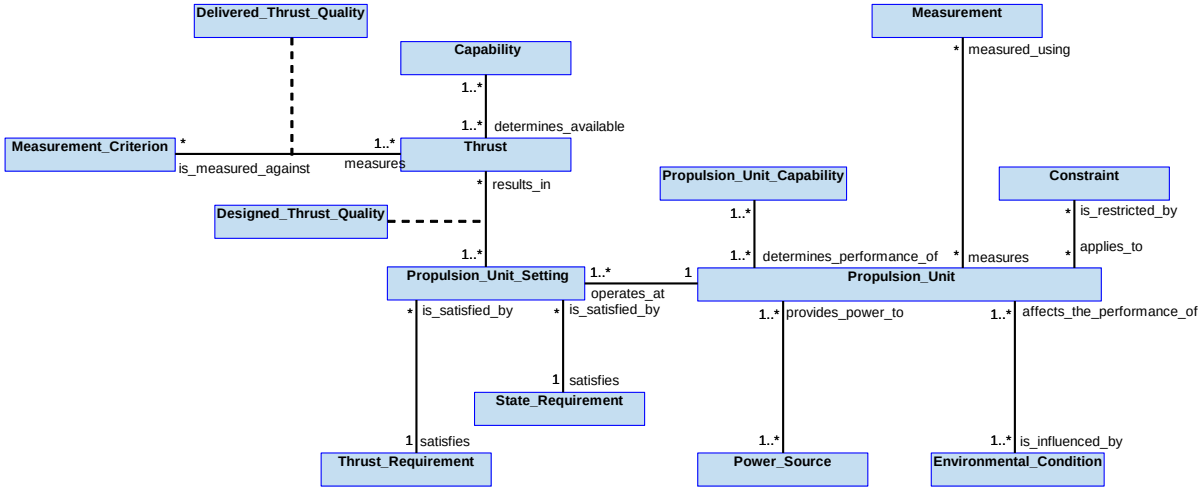

Figure 825: Propulsion Semantics

B.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 representation of a measurement device, including its capabilities.

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).

B.2.44.6 Design Rationale

B.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.

B.2.44.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.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.

B.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.

B.2.44.7 Services

B.2.44.7.1 Service Definitions

B.2.44.7.1.1 Propulsion_Requirement

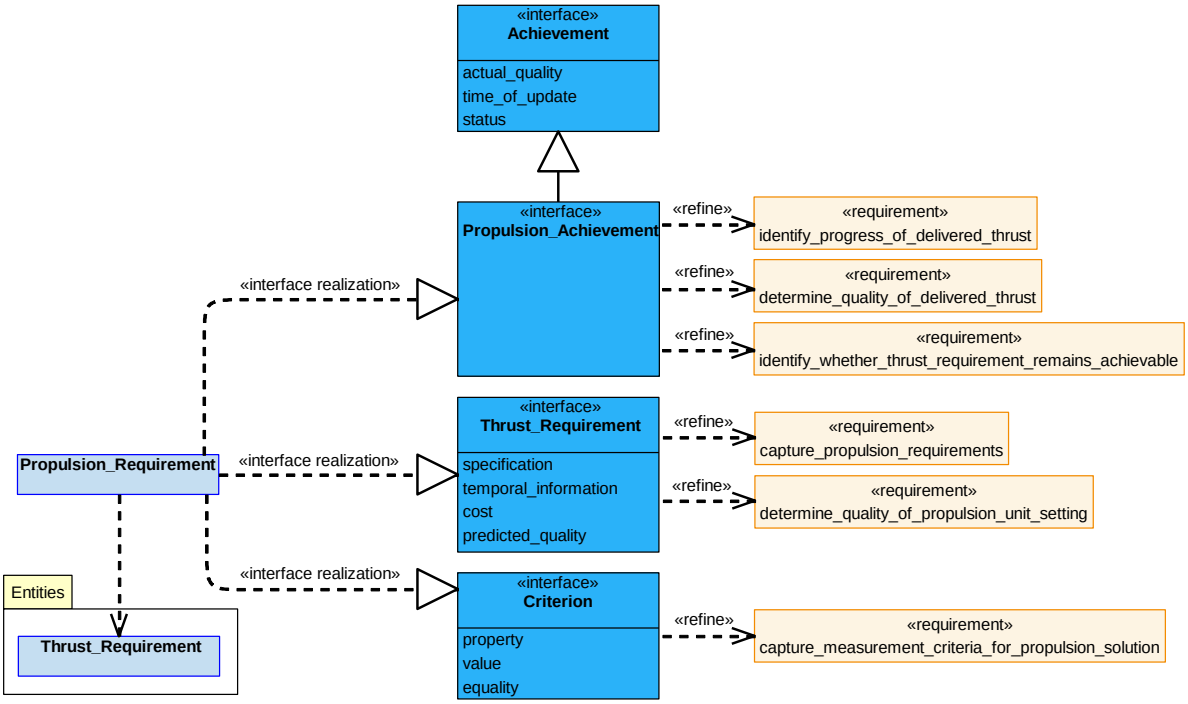

Figure 826: Propulsion_Requirement Service Definition

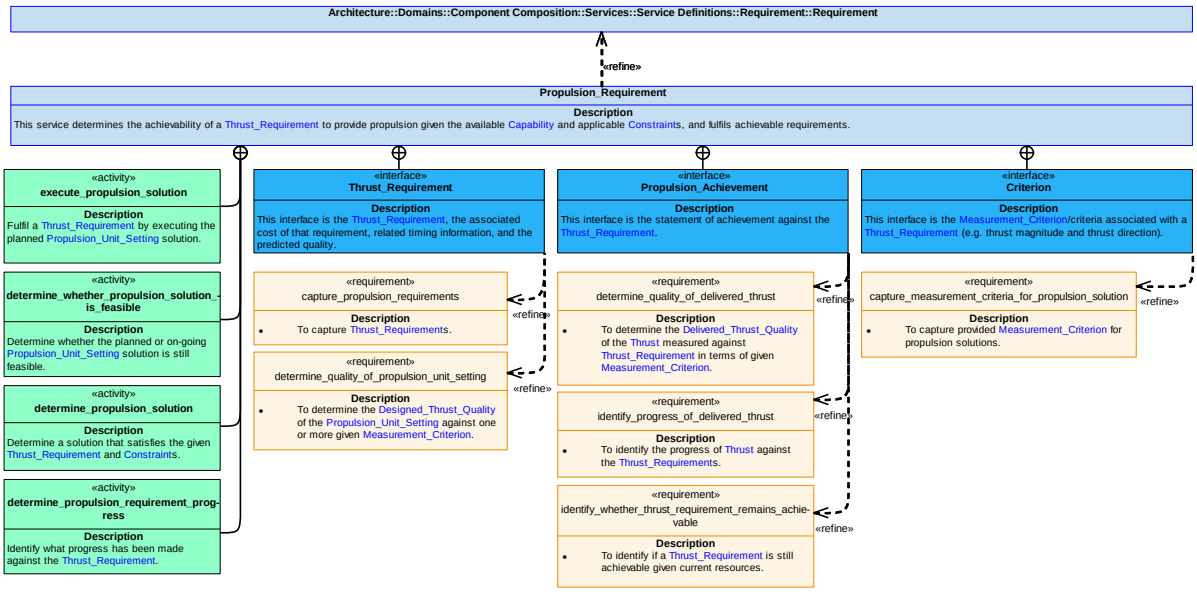

Figure 827: Propulsion_Requirement Service Policy

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_requirement_is_achievable

Determine whether the planned or on-going Thrust_Requirement is still achievable.

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.

B.2.44.7.1.2 State_Requirement

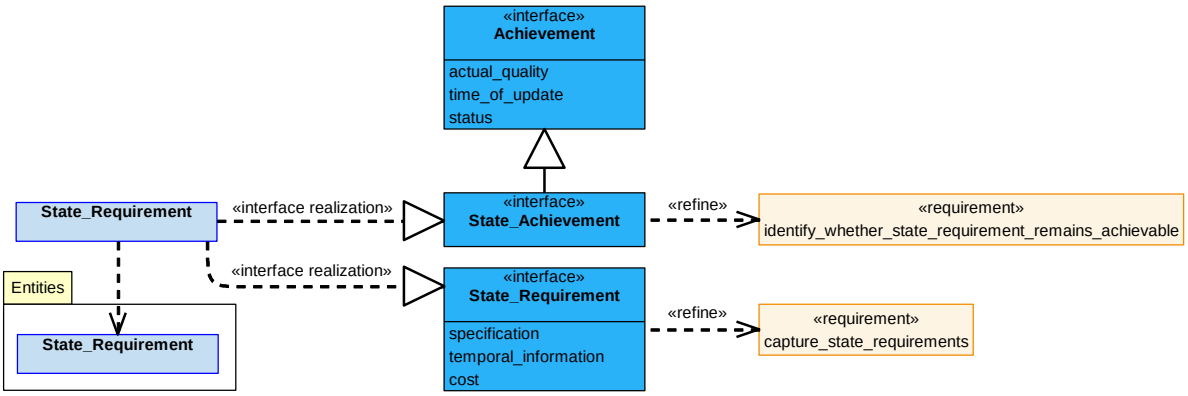

Figure 828: State_Requirement Service Definition

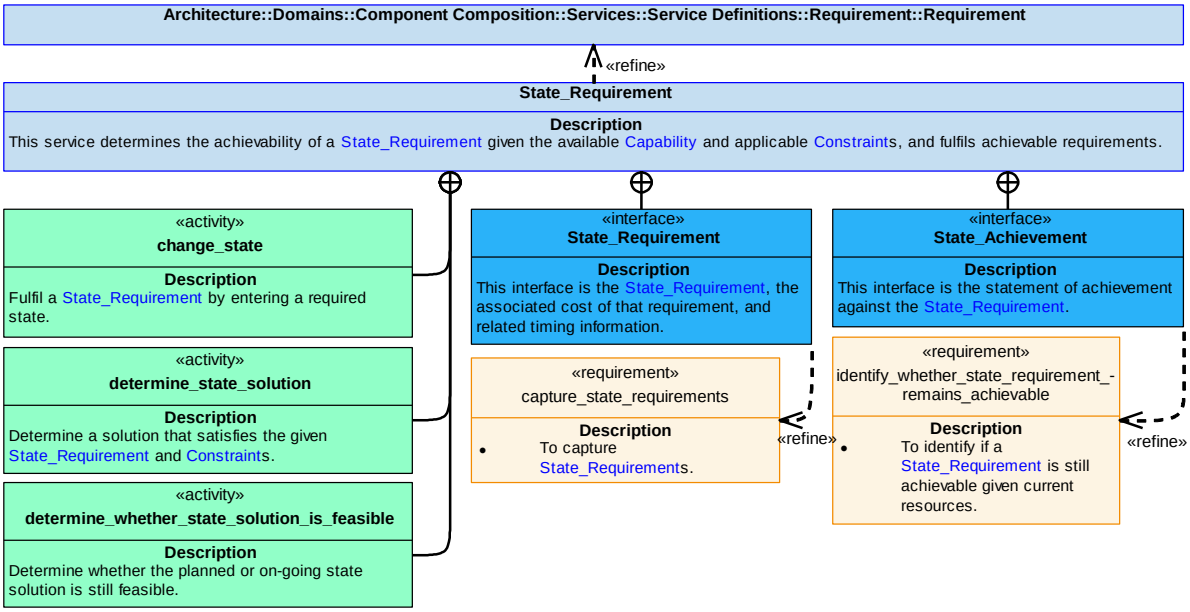

Figure 829: State_Requirement Service Policy

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.

B.2.44.7.1.3 Environmental_Conditioning_Dependency

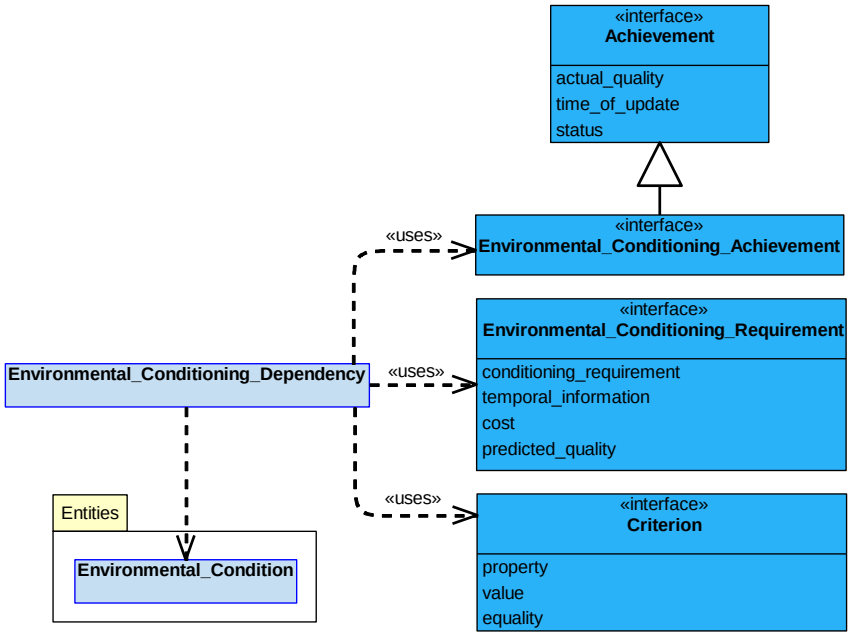

Figure 830: Environmental_Conditioning_Dependency Service Definition

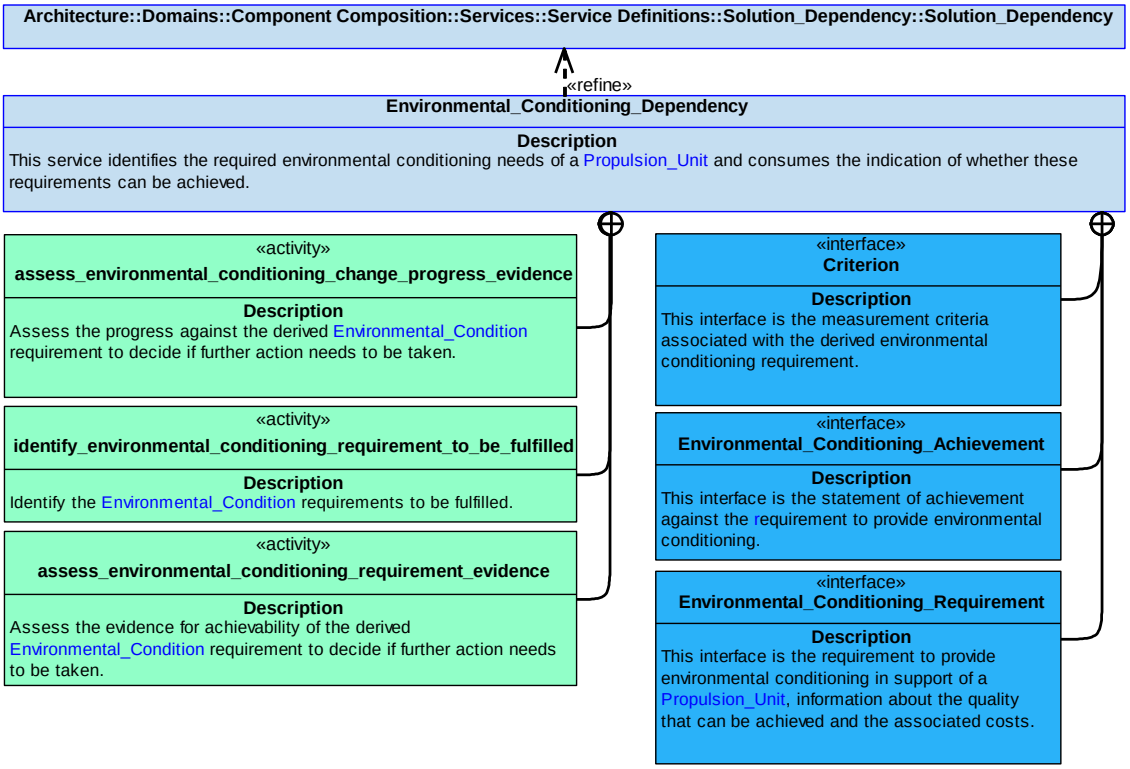

Figure 831: Environmental_Conditioning_Dependency Service Policy

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.

B.2.44.7.1.4 Power_Dependency

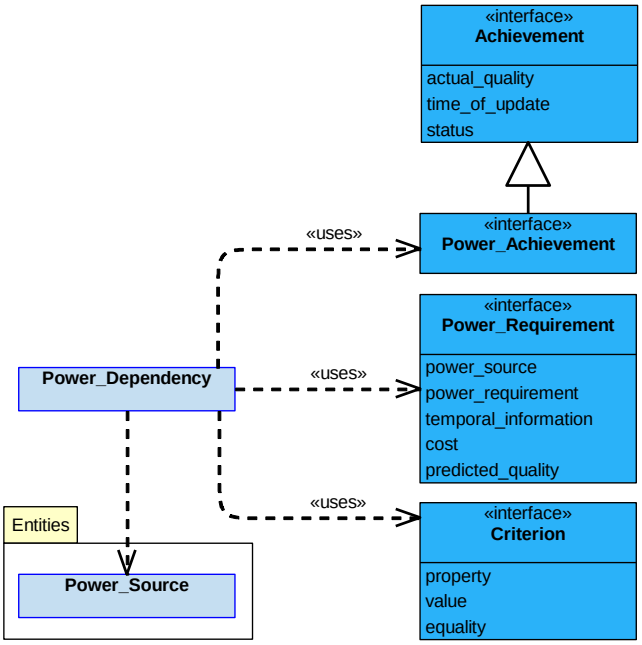

Figure 832: Power_Dependency Service Definition

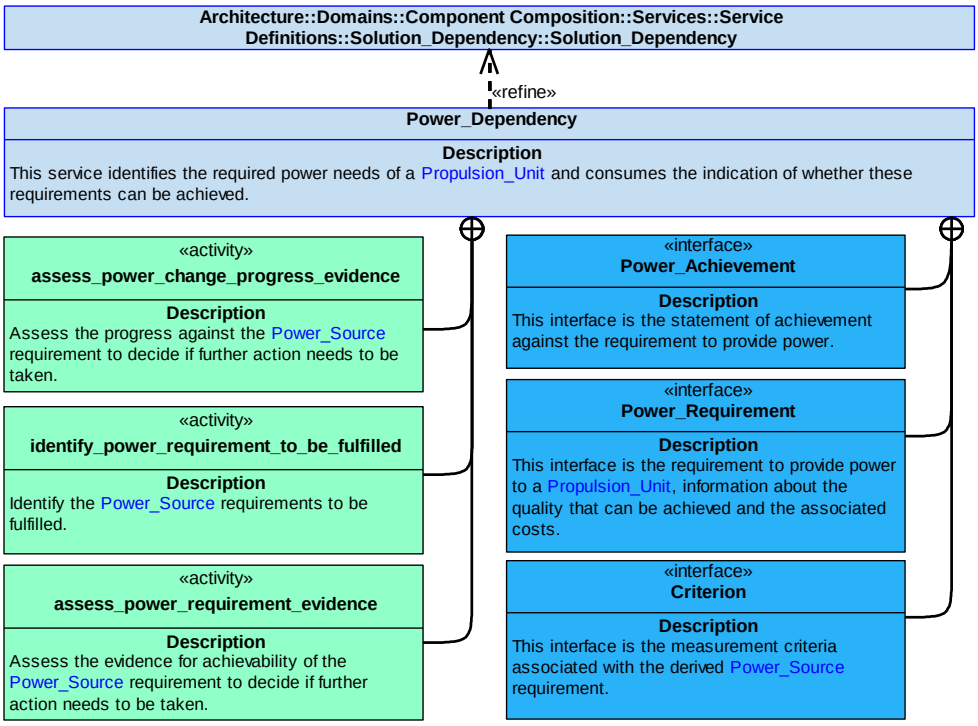

Figure 833: Power_Dependency Service Policy

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.

B.2.44.7.1.5 Environmental_Information

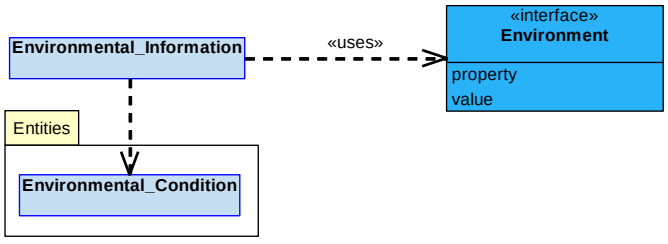

Figure 834: Environmental_Information Service Definition

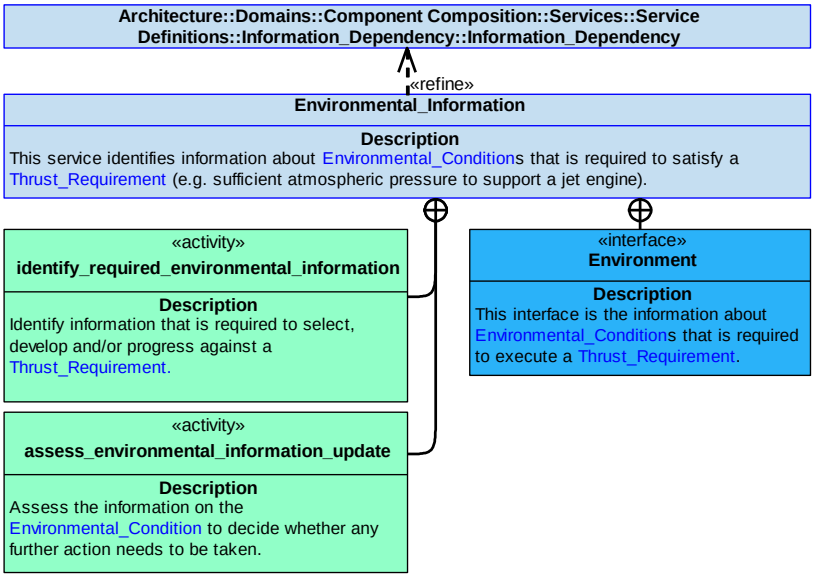

Figure 835: Environmental_Information Service Policy

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.

B.2.44.7.1.6 Constraint

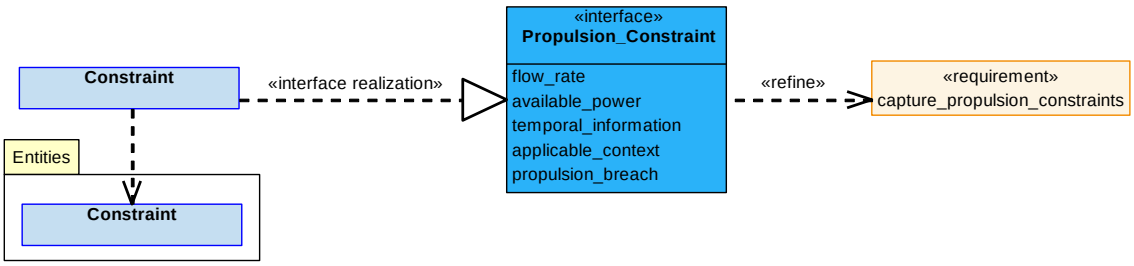

Figure 836: Constraint Service Definition

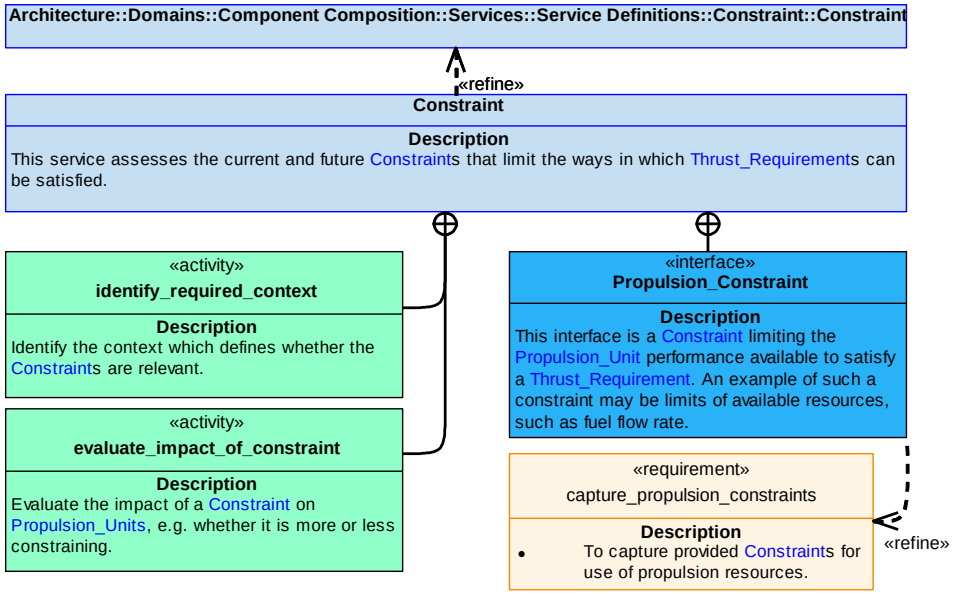

Figure 837: Constraint Service Policy

Constraint

This service assesses the current and future Constraints that limit the ways in which Thrust_Requirements can be satisfied.

Interface

Propulsion_Constraint

This interface is a Constraint limiting the Propulsion_Unit performance available to satisfy a Thrust_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.
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.

B.2.44.7.1.7 Capability

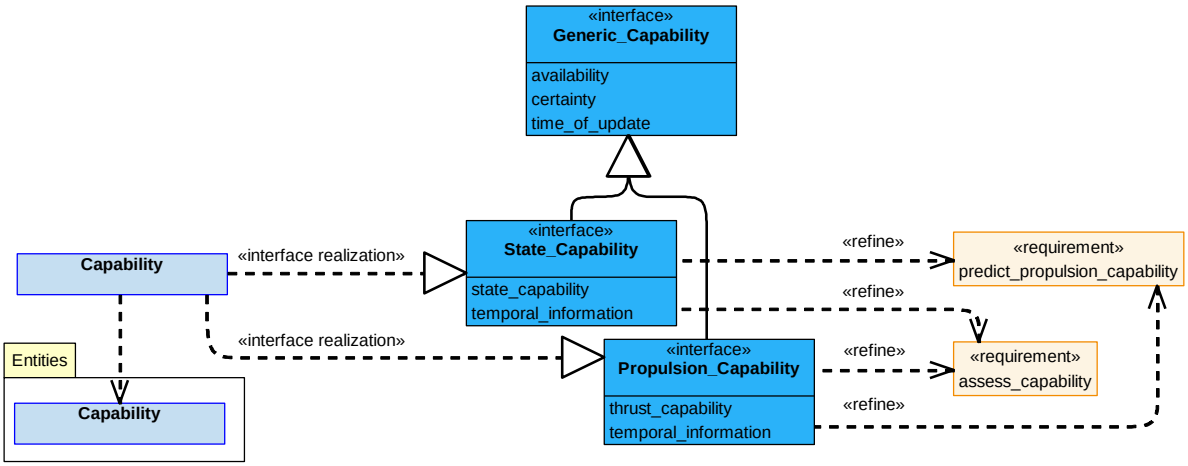

Figure 838: Capability Service Definition

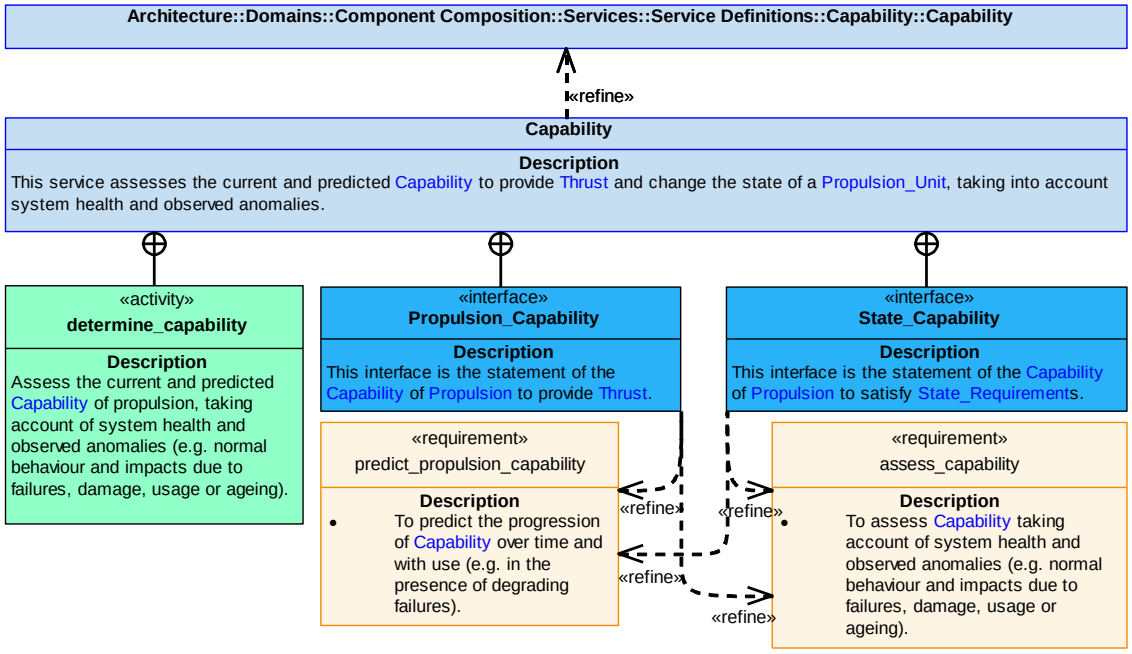

Figure 839: Capability Service Policy

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).

B.2.44.7.1.8 Capability_Evidence

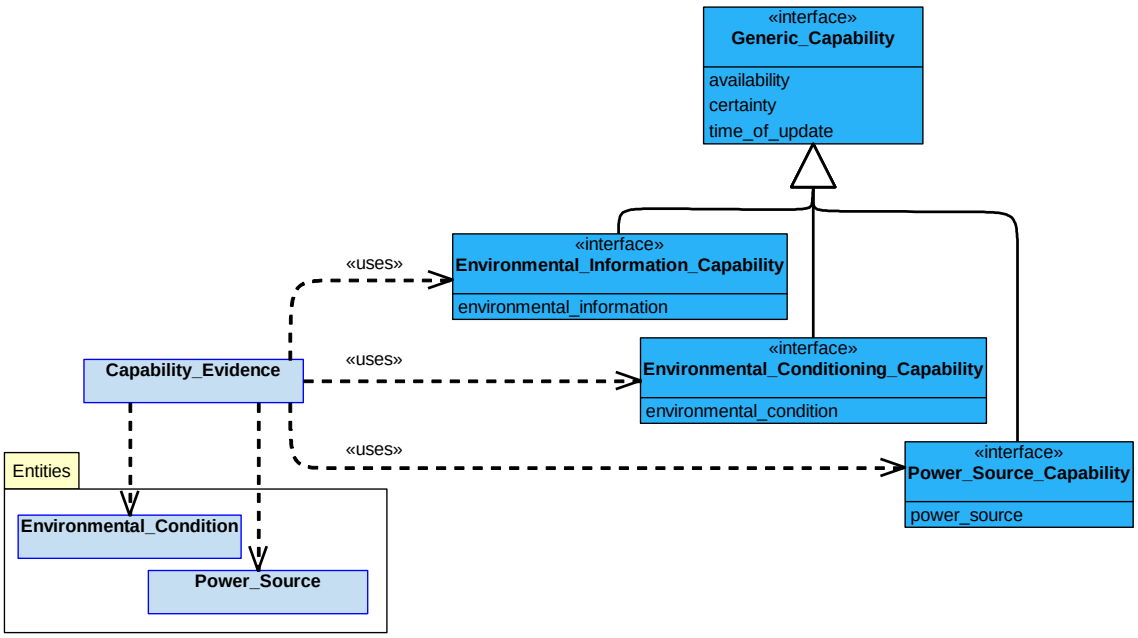

Figure 840: Capability_Evidence Service Definition

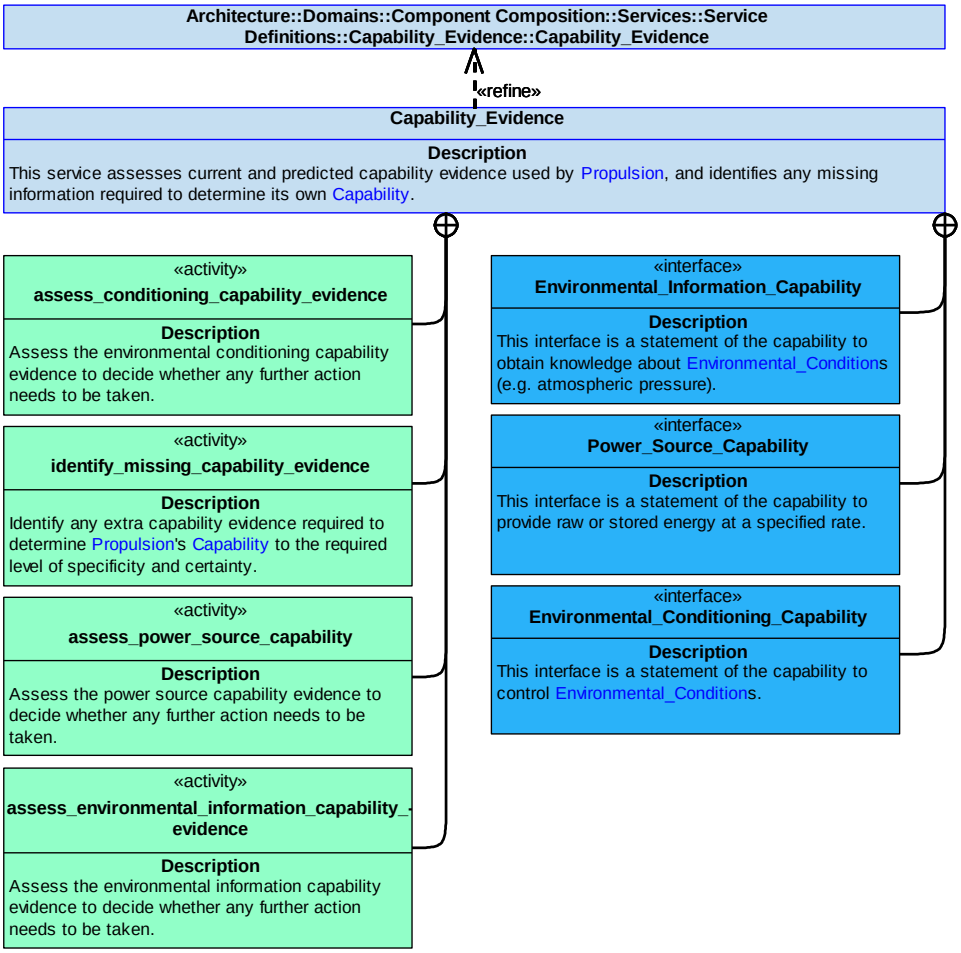

Figure 841: Capability_Evidence Service Policy

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.

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_conditioning_capability_evidence

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

assess_environmental_information_capability_evidence

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

assess_power_source_capability

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

B.2.44.7.2 Service Dependencies

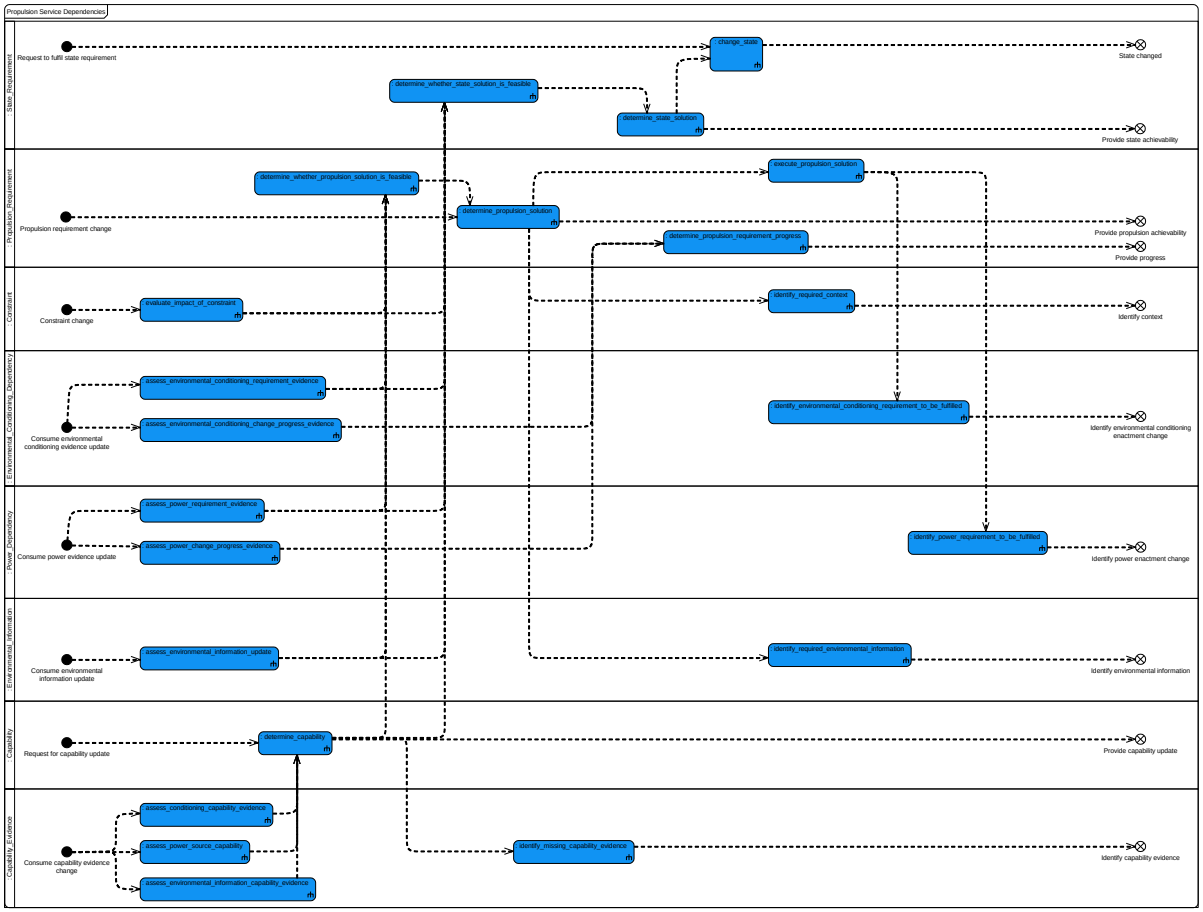

Figure 842: Propulsion Service Dependencies

B.2.45 Reference Times

B.2.45.1 Role

The role of Reference Times is to represent the references used to express time and their relationships.

B.2.45.2 Overview

Control Architecture

Reference Times is a service component as defined in the Control Architecture policy.

Standard Pattern of Use

This component assesses Reference_Times and provides the Difference between them. This includes factoring the Reference_Times and considering their Confidence level, to determine a specific Reference_Time (e.g. current or system time) as a Difference from a given Reference_Time.

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.

B.2.45.3 Service Summary

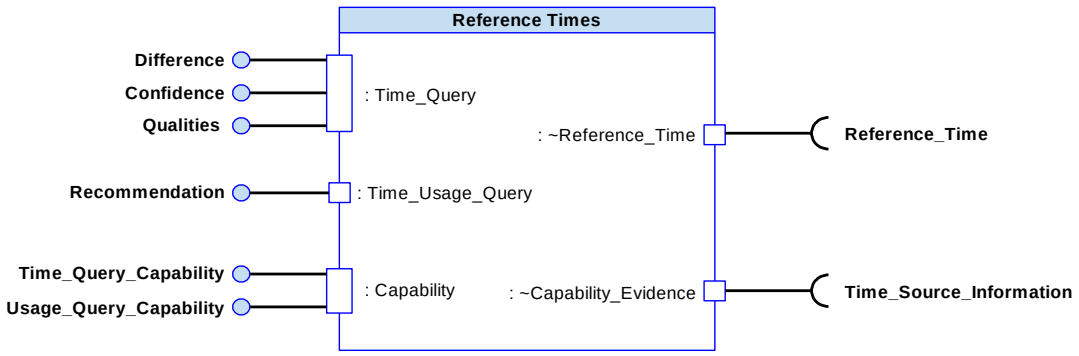

Figure 843: Reference Times Service Summary

B.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.

predict_capability_progression

- To predict the progression of Reference Times Capability over time.

assess_capability

- To assess the capability to represent Reference_Times, their Quality, and their Use.

determine_confidence_in_a_time

- To determine the reported Confidence in a Reference_Time, relative to one or more of the declared Accuracy, declared Precision, or declared Confidence.

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.

identify_missing_information

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

B.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.

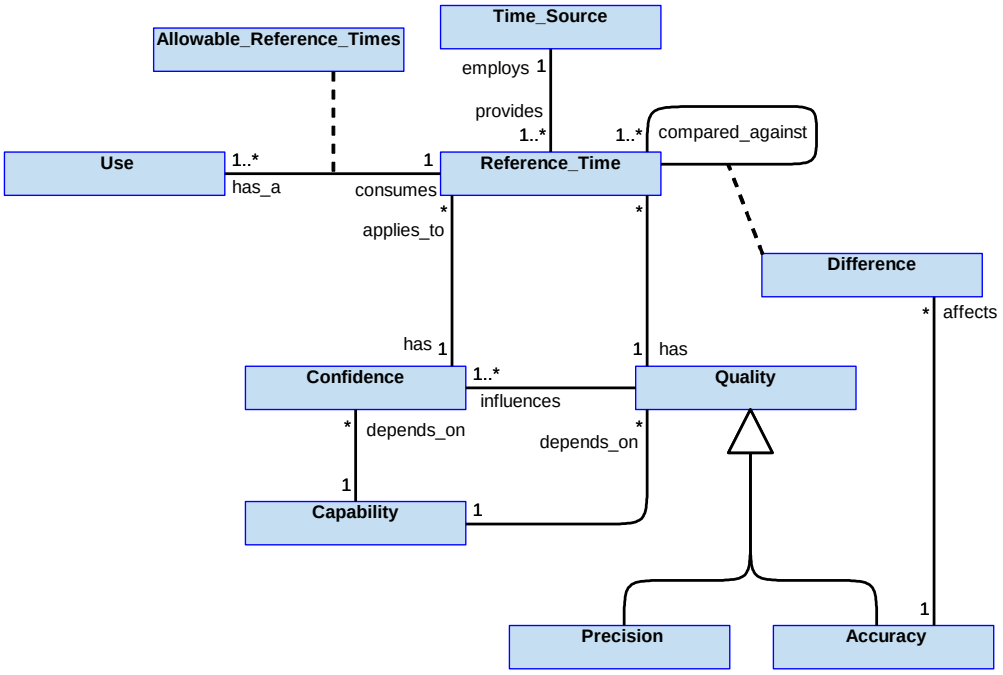

Figure 844: Reference Times Semantics

B.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

The allowable Reference_Times for a specific Use.

Capability

The capability to represent Reference_Times, their qualities, and what they're used for.

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).

B.2.45.6 Design Rationale

B.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).
- This component reasons about the Differences between Reference_Times.
- This component reasons about the Use of Reference_Times.
- This component will determine the Differences between the clocks of two time references, but it does not provide any clocks.
- The Difference between Reference_Times are defined as an offset from a datum Reference_Time.

B.2.45.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Reference Times:

- Data Driving - For 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).

B.2.45.6.3 Safety Considerations

The indicative IDAL is DAL A.

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).

B.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.

B.2.45.7 Services

B.2.45.7.1 Service Definitions

B.2.45.7.1.1 Time_Query

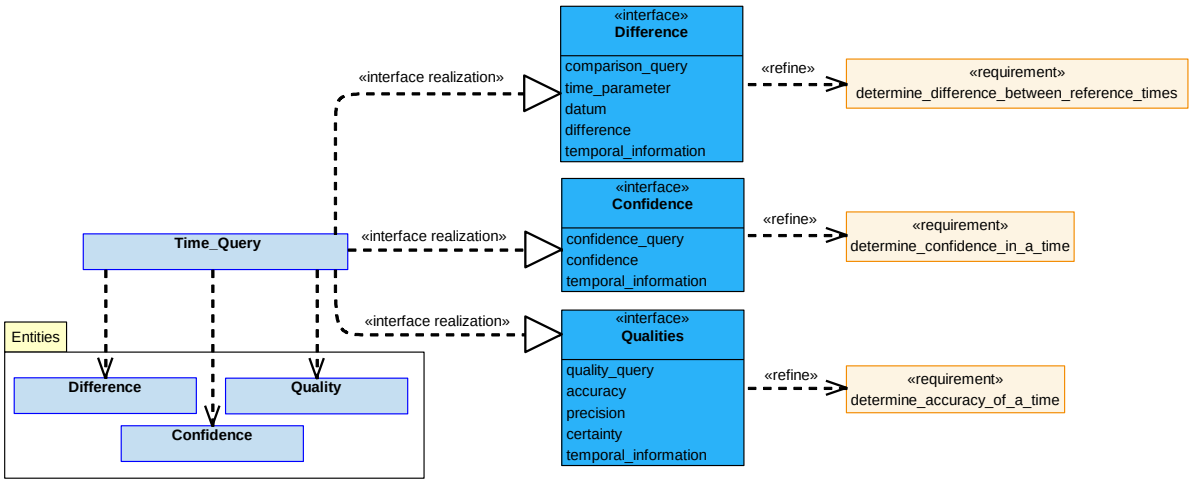

Figure 845: Time_Query Service Definition

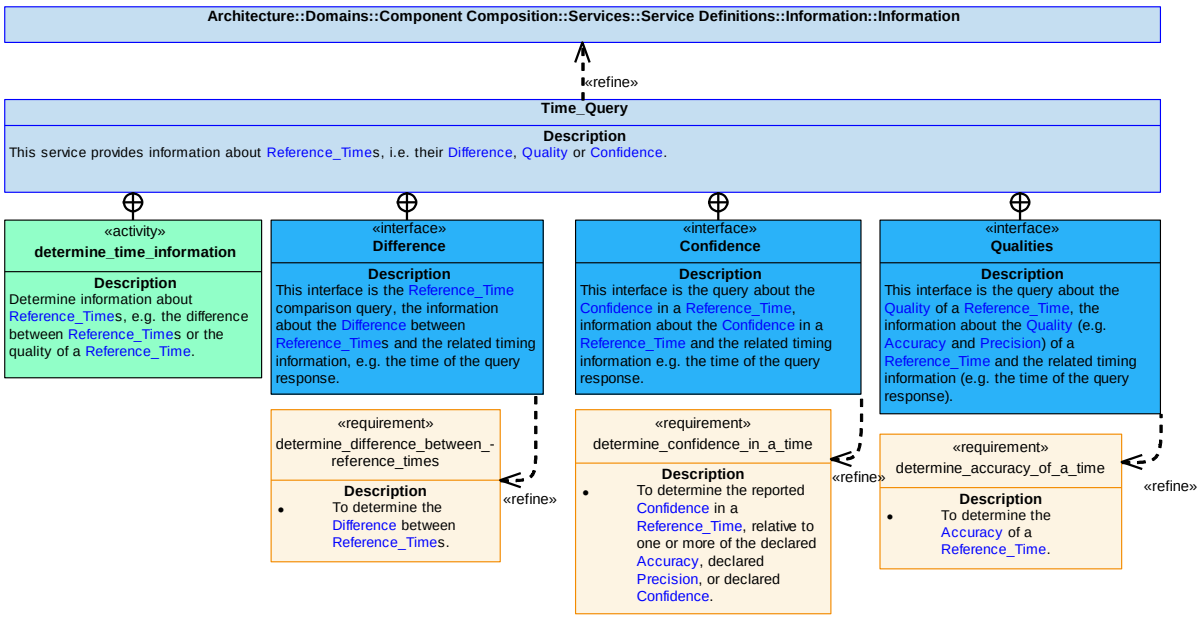

Figure 846: Time_Query Service Policy

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.

B.2.45.7.1.2 Time_Usage_Query

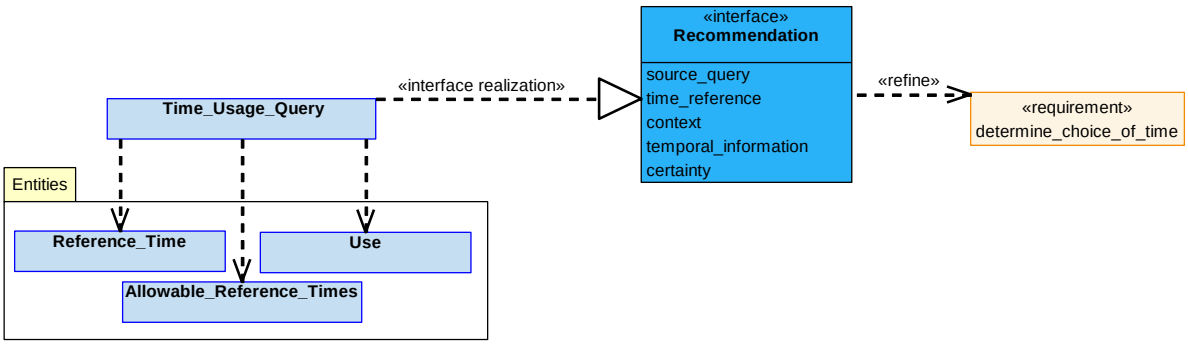

Figure 847: Time_Usage_Query Service Definition

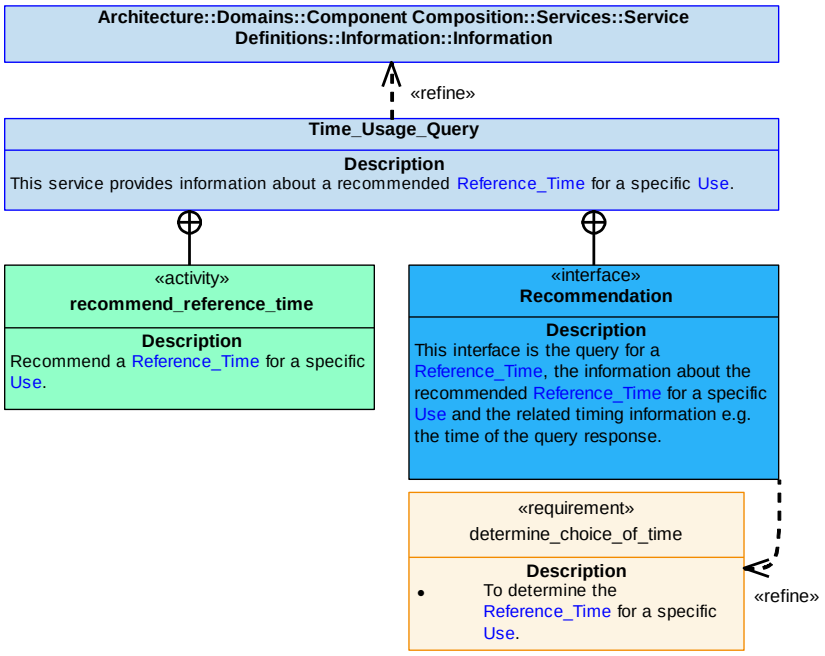

Figure 848: Time_Usage_Query Service Policy

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 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.

B.2.45.7.1.3 Reference_Time

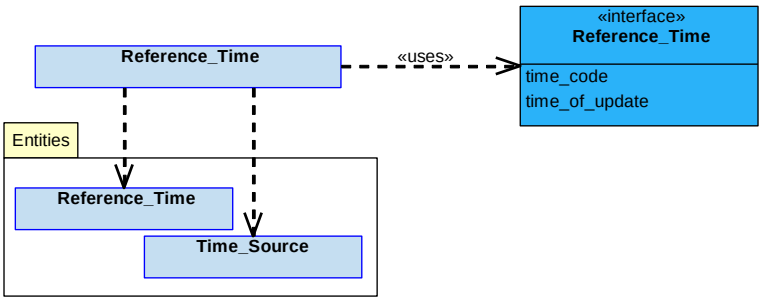

Figure 849: Reference_Time Service Definition

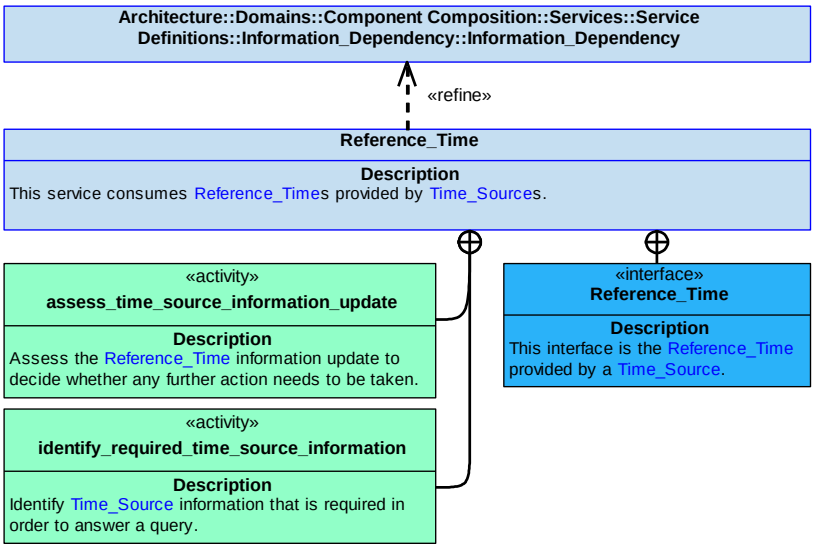

Figure 850: Reference_Time Service Policy

Reference_Time

This service consumes Reference_Times provided by Time_Sources.

Interface

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.

Activities

assess_time_source_information_update

Assess the Reference_Time 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.

B.2.45.7.1.4 Capability

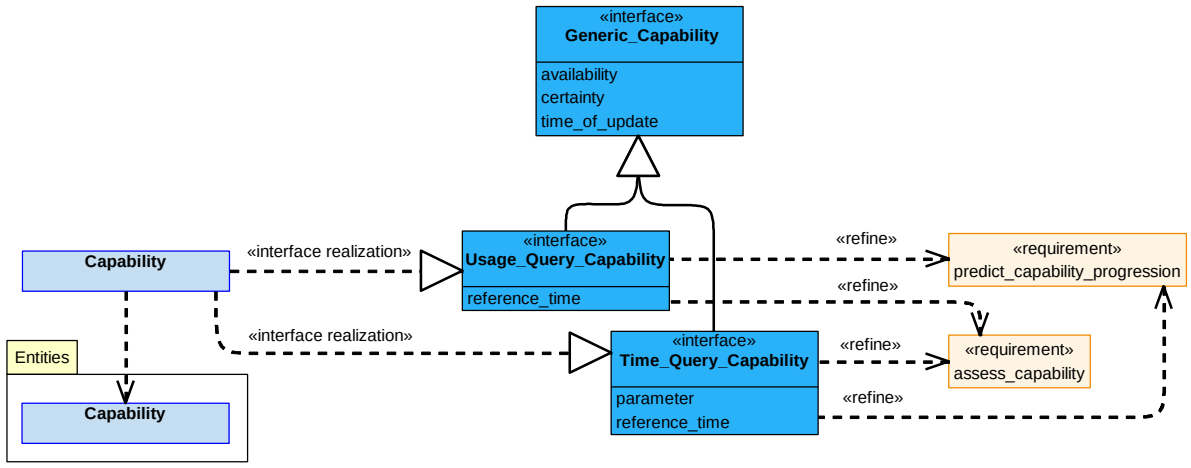

Figure 851: Capability Service Definition

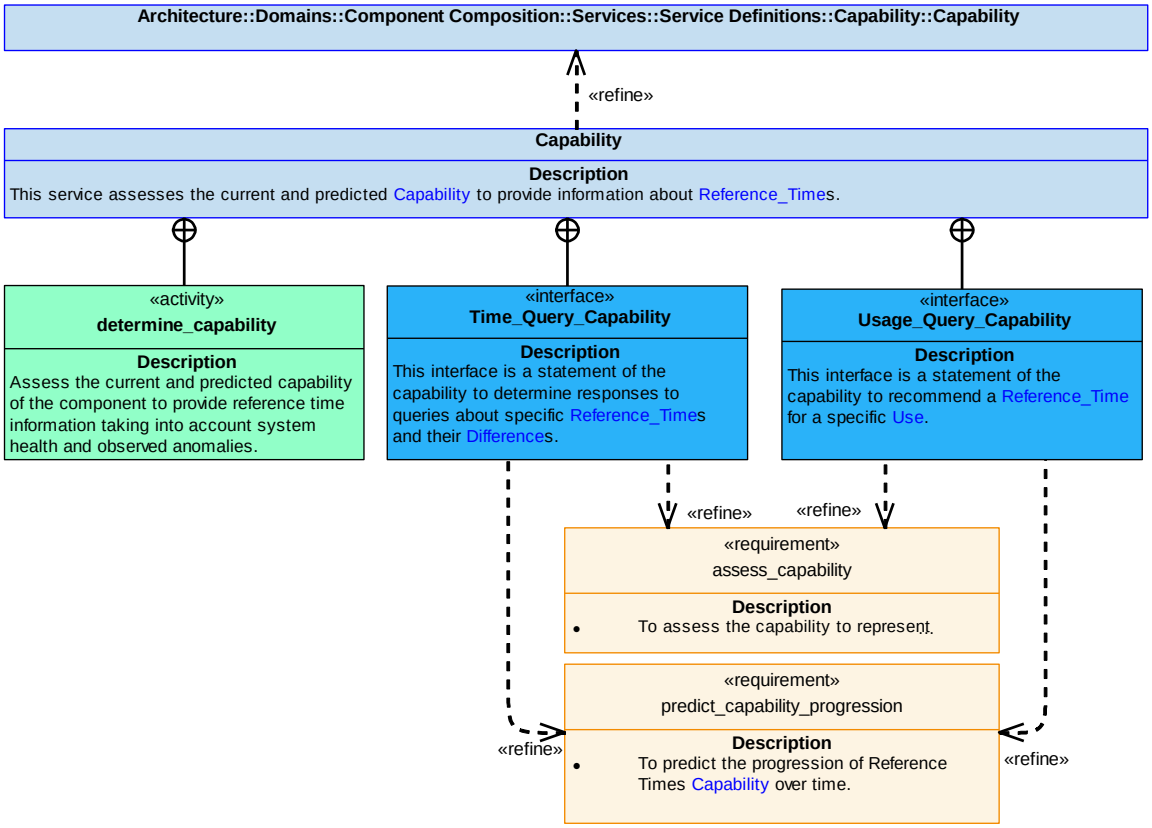

Figure 852: Capability Service Policy

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.

Attribute

reference_time	The Reference_Times the component is able to recommend, e.g. GPS 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.

B.2.45.7.1.5 Capability_Evidence

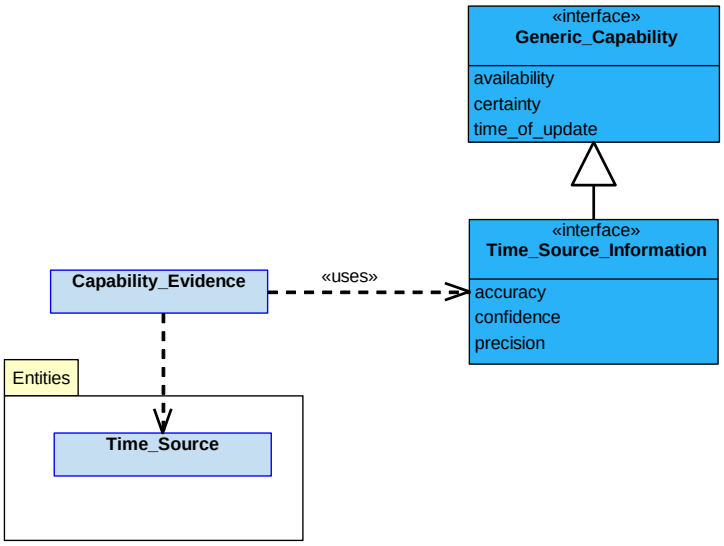

Figure 853: Capability_Evidence Service Definition

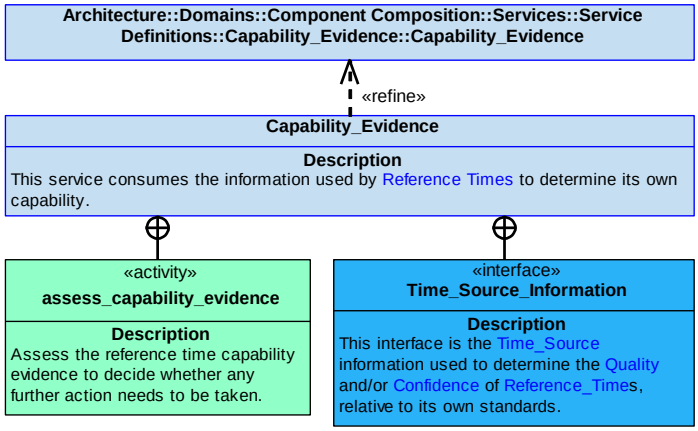

Figure 854: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the information used by Reference Times to determine its own capability.

Interface

Time_Source_Information

This interface is the Time_Source information used to determine the Quality and/or Confidence of Reference_Times, relative to its own standards.

Attributes

accuracy	The degree to which a time measurement aligns to the passing of time.
confidence	The degree of certainty in the Time_Source information.
precision	The granularity of the time code.

Activity

assess_capability_evidence

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

B.2.45.7.2 Service Dependencies

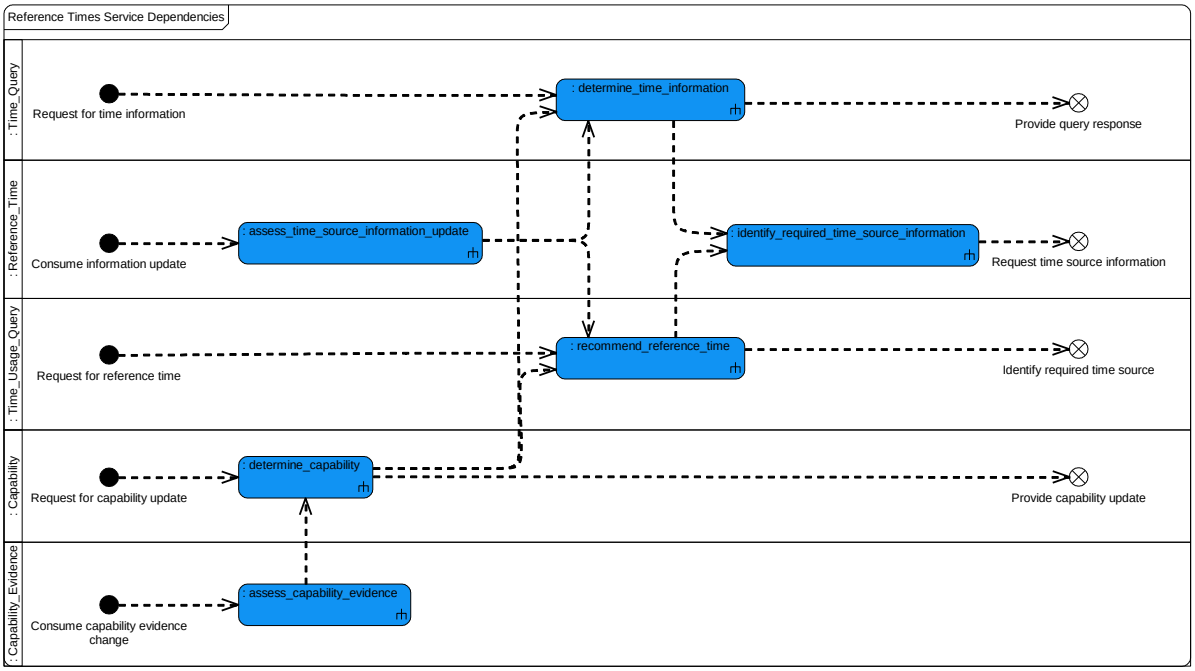

Figure 855: Reference Times Service Dependencies

B.2.46 Release Aiming

B.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.

B.2.46.2 Overview

Control Architecture

Release Aiming is a service component as defined in the Control Architecture policy.

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:

- Aiming of Stores is required.

B.2.46.3 Service Summary

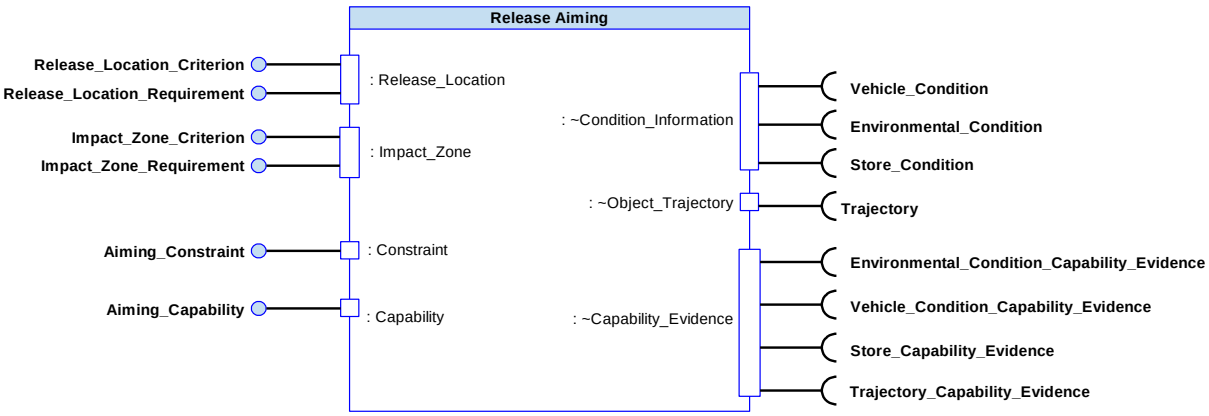

Figure 856: Release Aiming Service Summary

B.2.46.4 Responsibilities

determine_aiming_solutions

- To determine Aiming_Solutions.

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).

predict_aiming_capability_progression

- To predict the progression of Release Aiming's aiming Capability over time and with use.

identify_aiming_solution_in_progress_remains_feasible

- To identify whether an Aiming_Solution in progress remains feasible given current resources.

capture_release_constraints

- To capture the externally imposed Constraints that limit where or how the store can be released.

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.

determine_predicted_quality_of_aiming_solution

- To determine the predicted quality of the Aiming_Solution against provided Measurement_Criterion/criteria.

identify_missing_information

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

B.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.

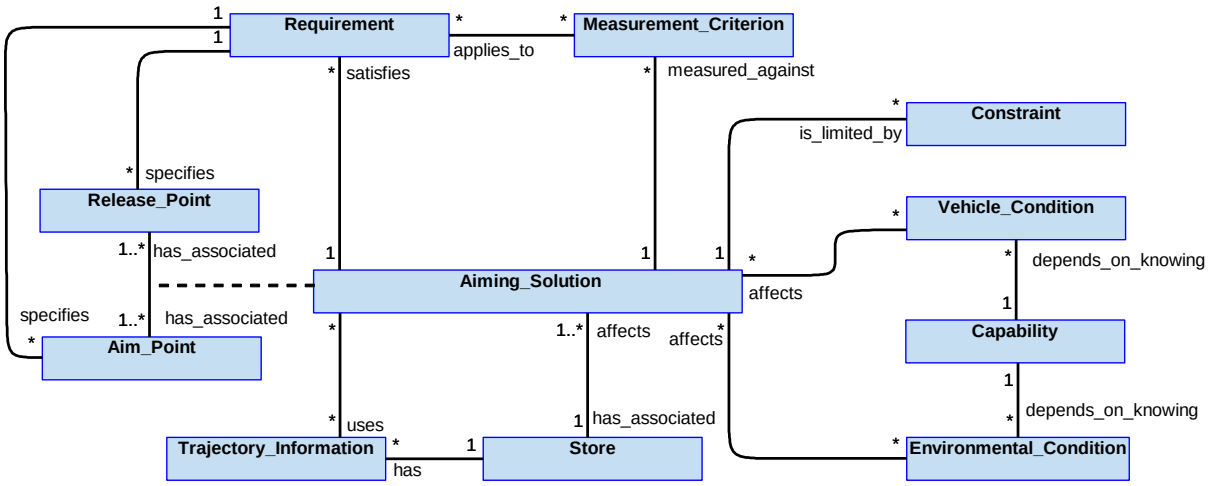

Figure 857: Release Aiming Semantics

B.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).

B.2.46.6 Design Rationale

B.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.

B.2.46.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Release Aiming:

- Data Driving - Release Aiming follows this policy 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 policy.

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 policy).

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.

B.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. In accordance with UK MoD direction (see the Safety Analysis policy) this drives a DAL B indicative IDAL.

B.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.

B.2.46.7 Services

B.2.46.7.1 Service Definitions

B.2.46.7.1.1 Release_Location

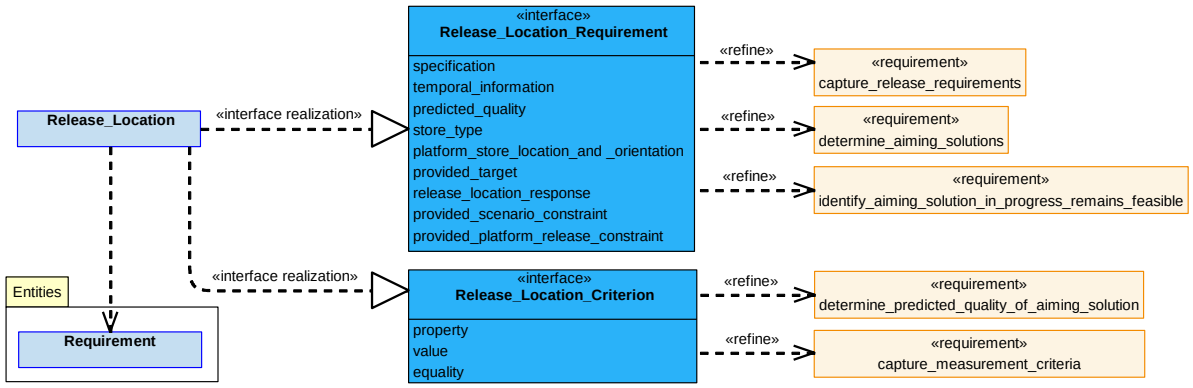

Figure 858: Release_Location Service Definition

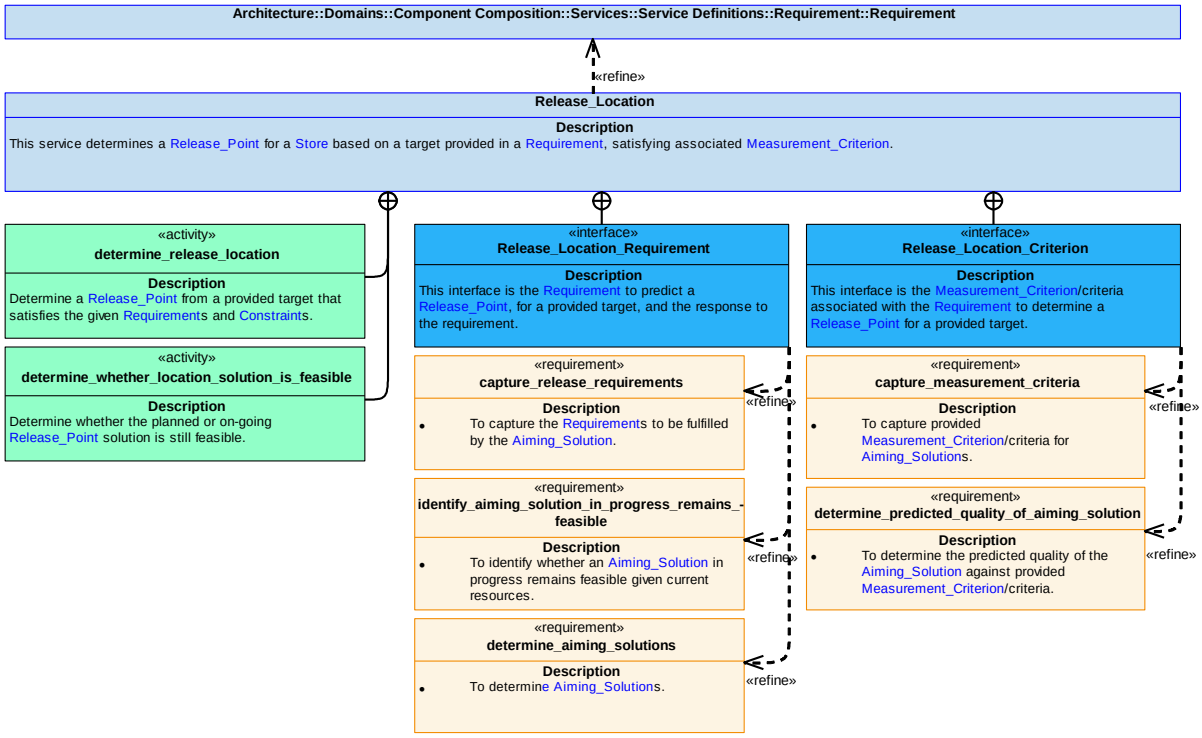

Figure 859: Release_Location Service Policy

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.

B.2.46.7.1.2 Impact_Zone

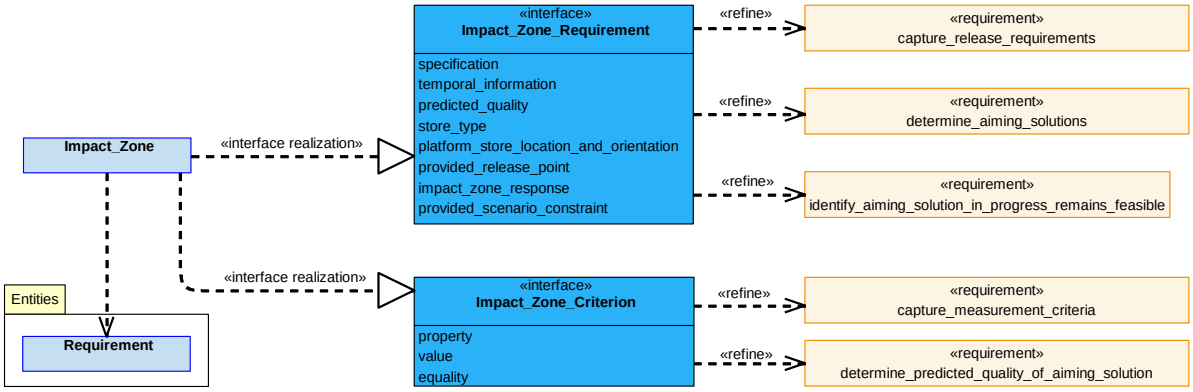

Figure 860: Impact_Zone Service Definition

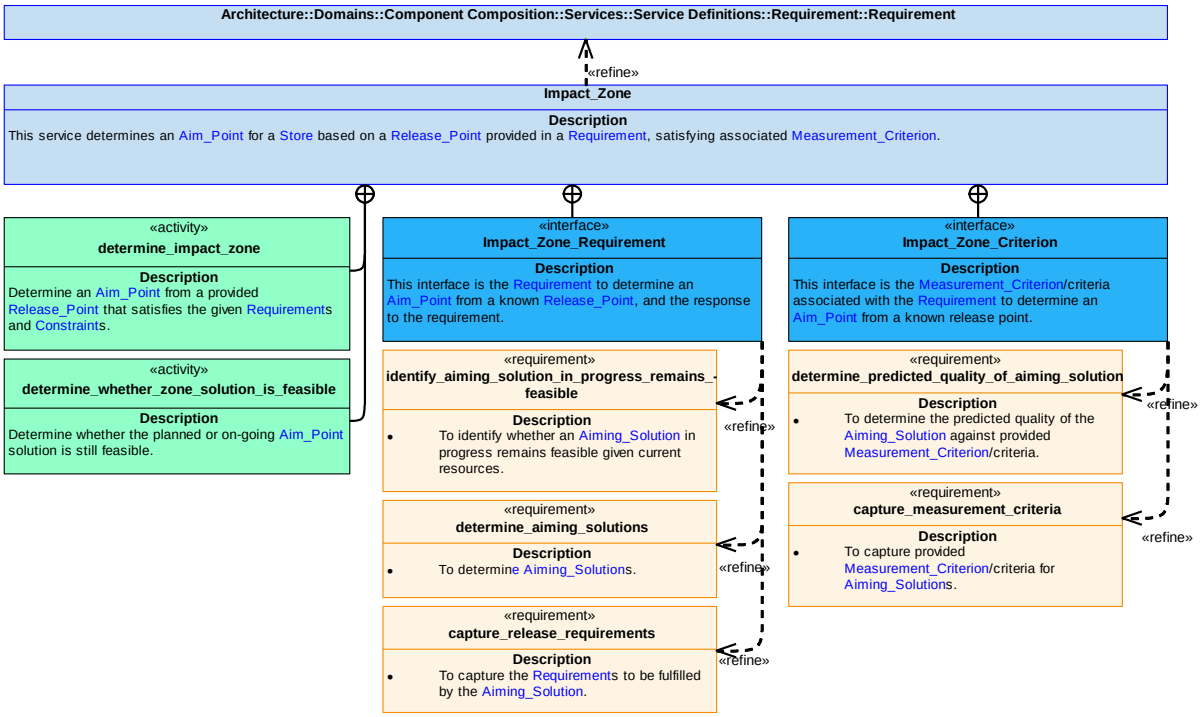

Figure 861: Impact_Zone Service Policy

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.

B.2.46.7.1.3 Condition_Information

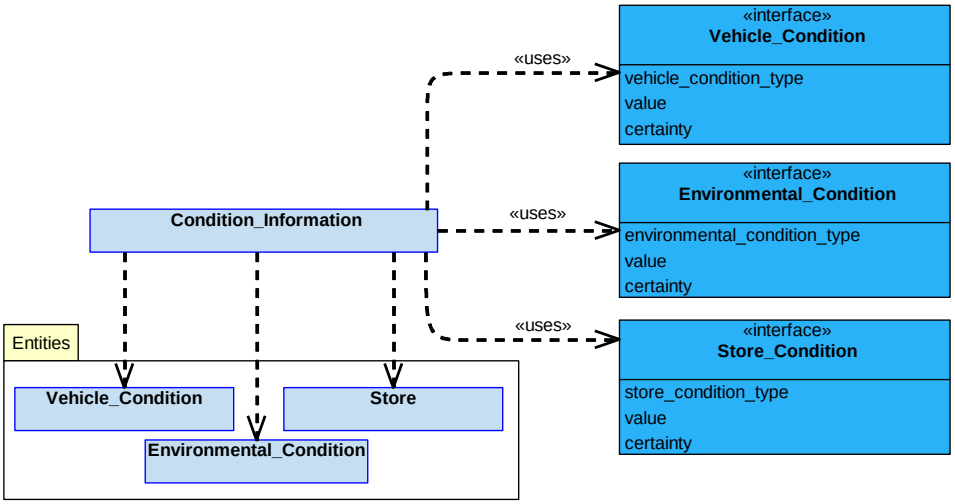

Figure 862: Condition_Information Service Definition

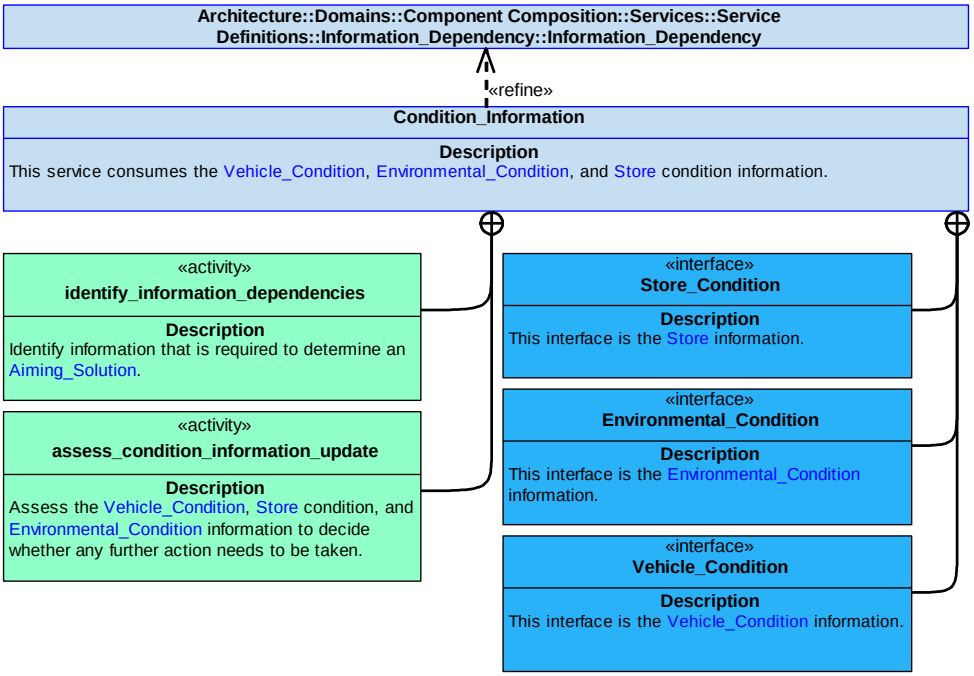

Figure 863: Condition_Information Service Policy

Condition_Information

This service consumes the Vehicle_Condition, Environmental_Condition, and Store condition information.

Interfaces

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.

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.

Store_Condition

This interface is the Store information.

Attributes

store_condition_type	The type of information about the 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

assess_condition_information_update

Assess the Vehicle_Condition, Store condition, and Environmental_Condition information to decide whether any further action needs to be taken.

identify_information_dependencies

Identify information that is required to determine an Aiming_Solution.

B.2.46.7.1.4 Object_Trajectory

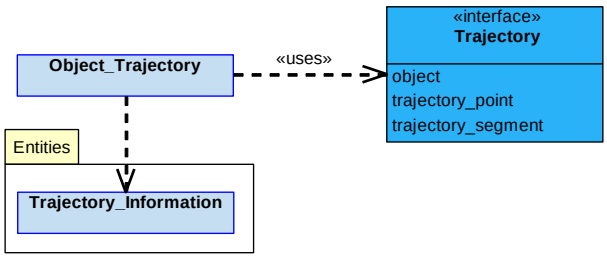

Figure 864: Object_Trajectory Service Definition

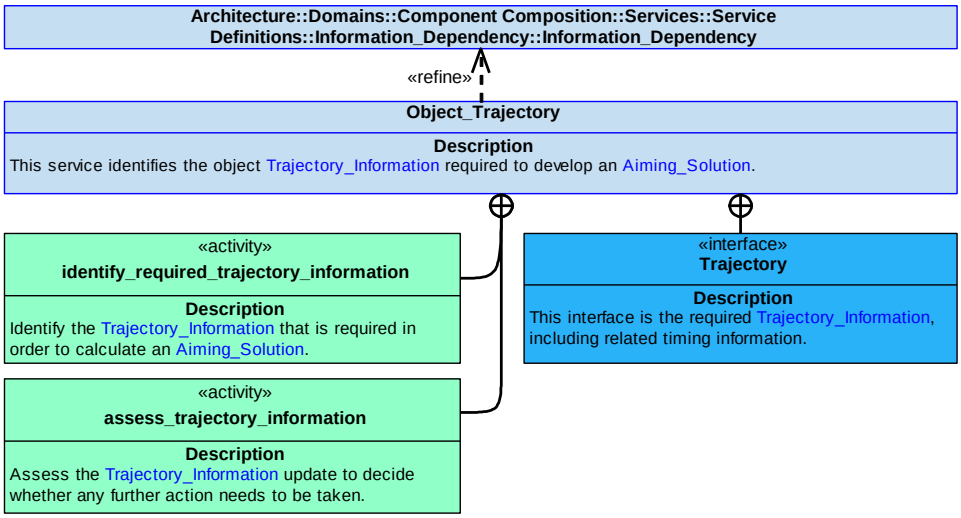

Figure 865: Object_Trajectory Service Policy

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.

B.2.46.7.1.5 Constraint

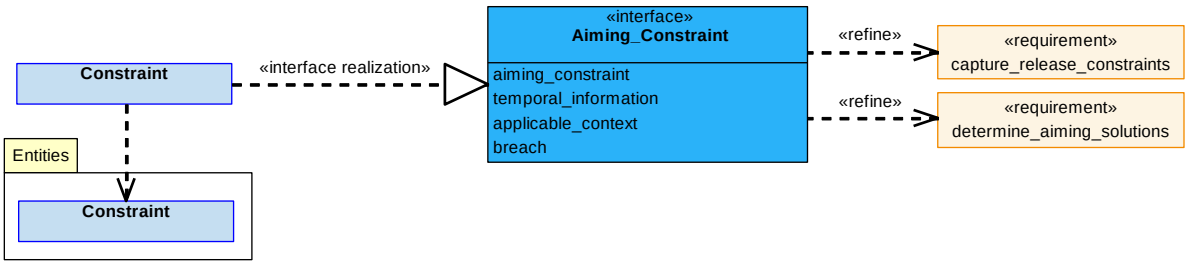

Figure 866: Constraint Service Definition

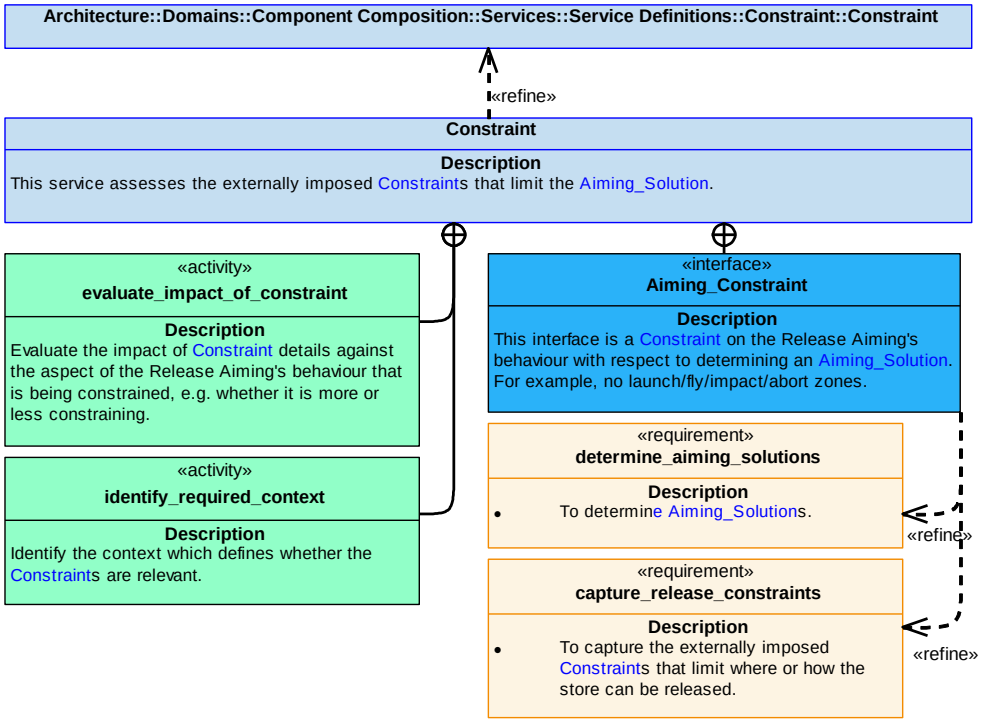

Figure 867: Constraint Service Policy

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.

B.2.46.7.1.6 Capability

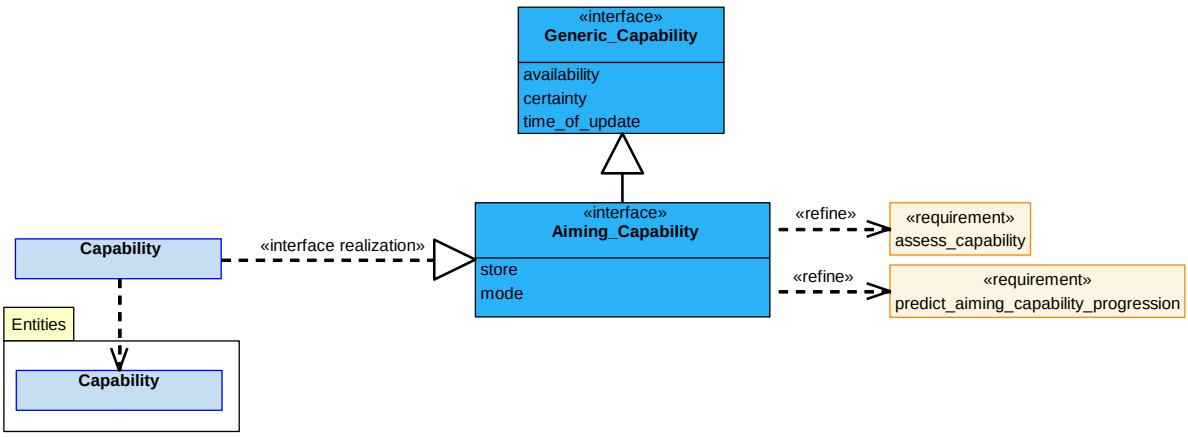

Figure 868: Capability Service Definition

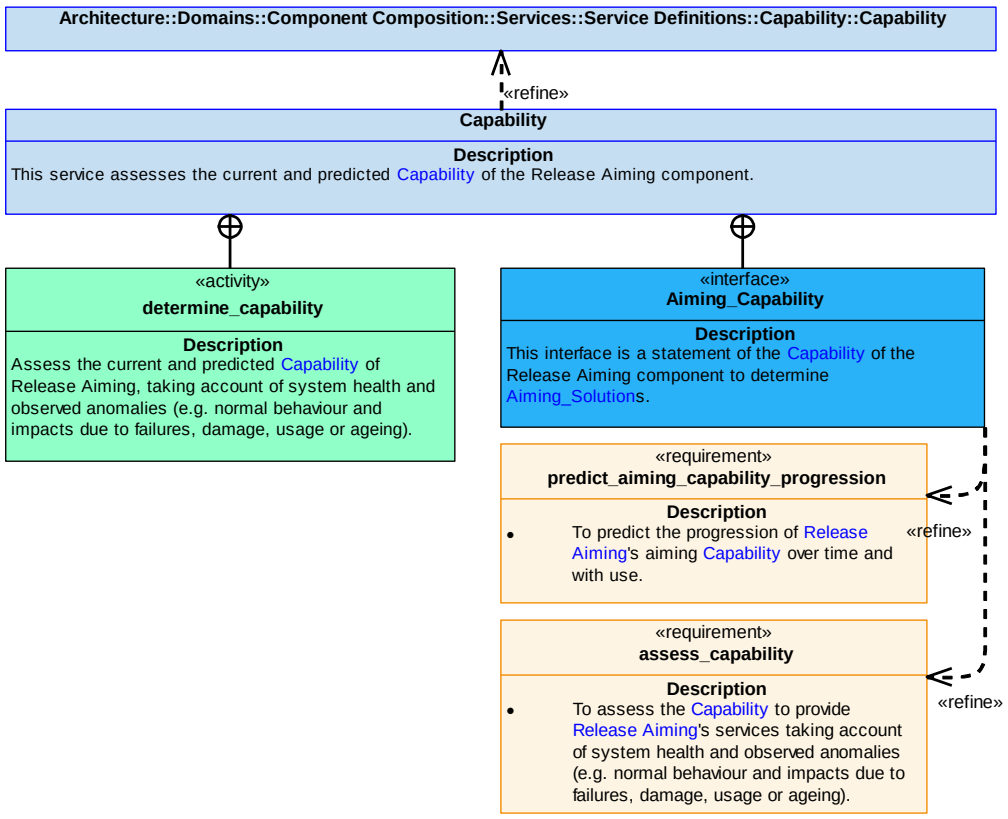

Figure 869: Capability Service Policy

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).

B.2.46.7.1.7 Capability_Evidence

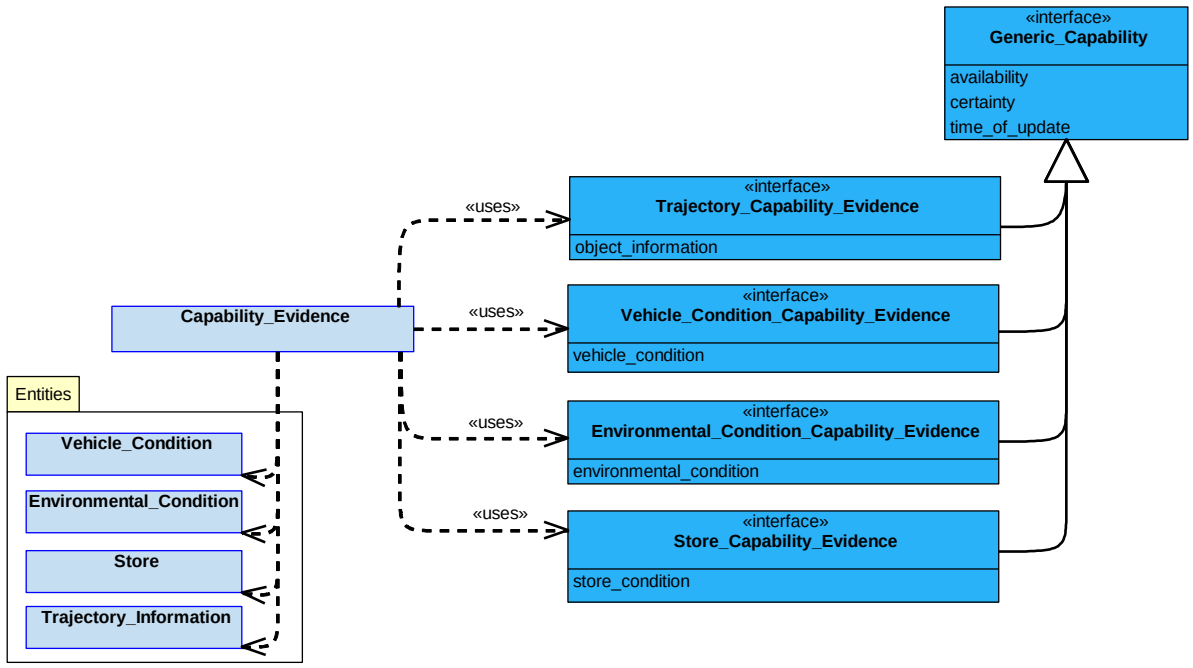

Figure 870: Capability_Evidence Service Definition

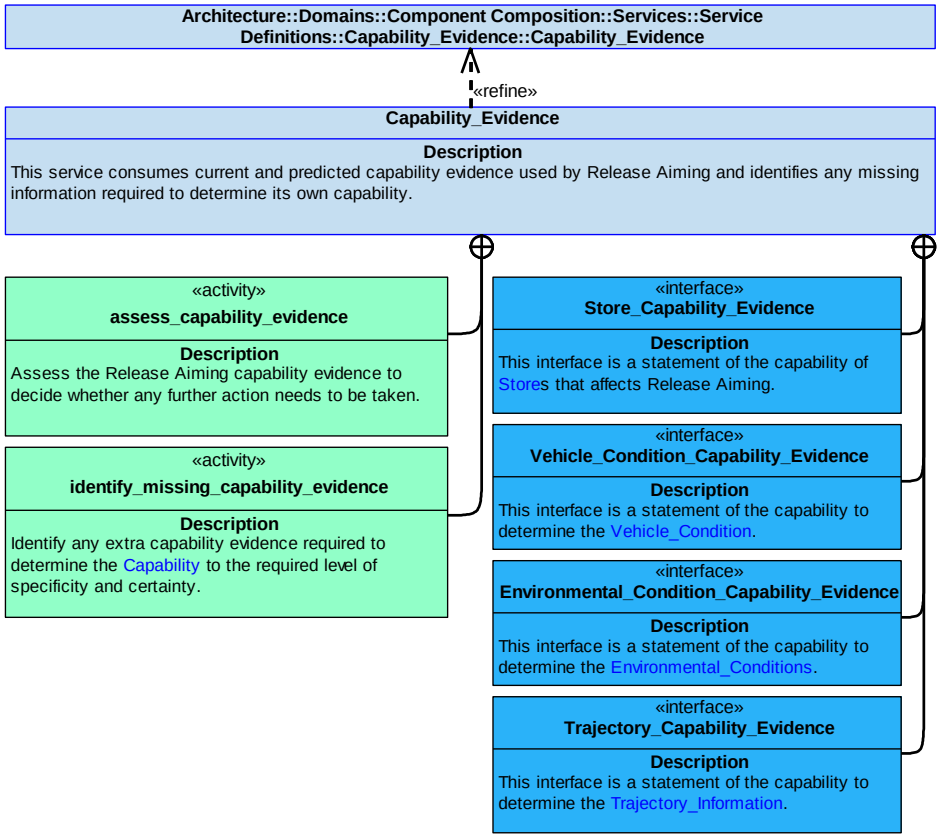

Figure 871: Capability_Evidence Service Policy

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.

B.2.46.7.2 Service Dependencies

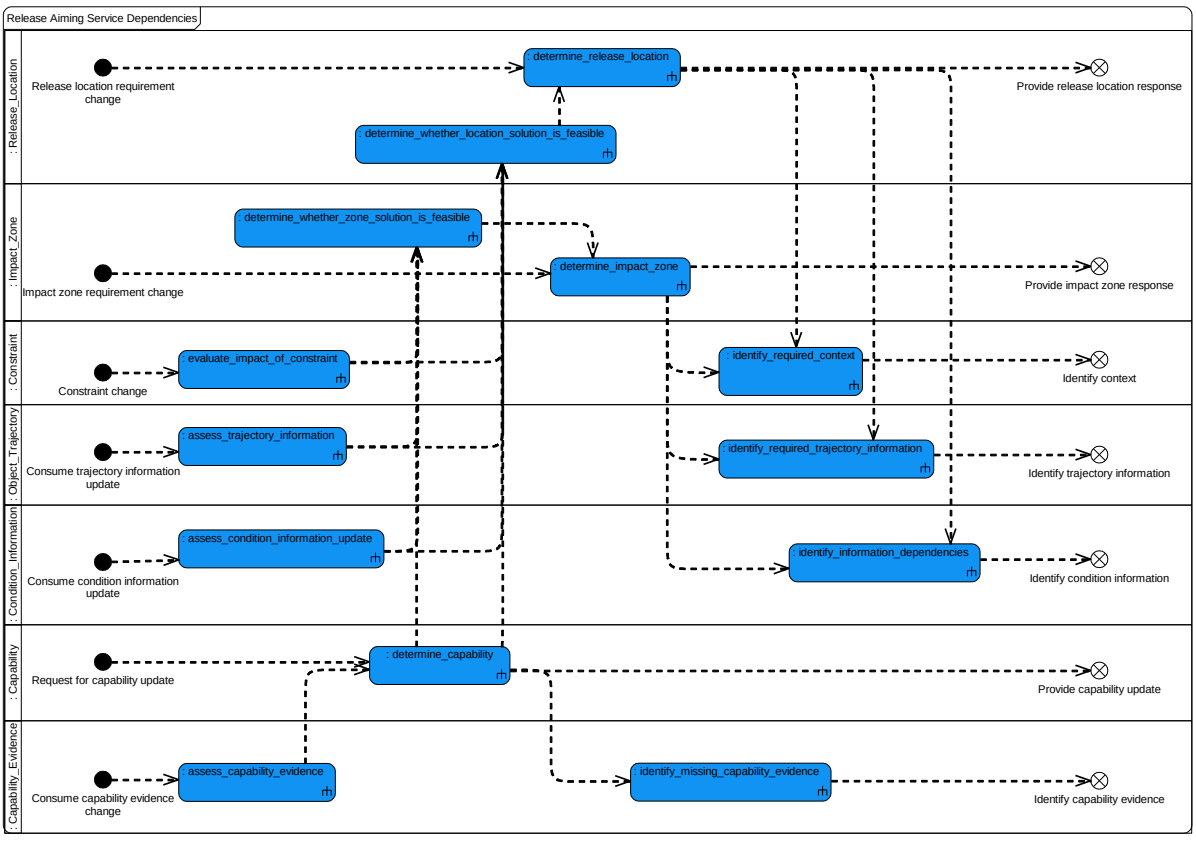

Figure 872: Release Aiming Service Dependencies

B.2.47 Release Effecting

B.2.47.1 Role

The role of Release Effecting is to effect the release of a Store.

B.2.47.2 Overview

Control Architecture

Release Effecting is a resource component as defined in the Control Architecture policy.

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.

B.2.47.3 Service Summary

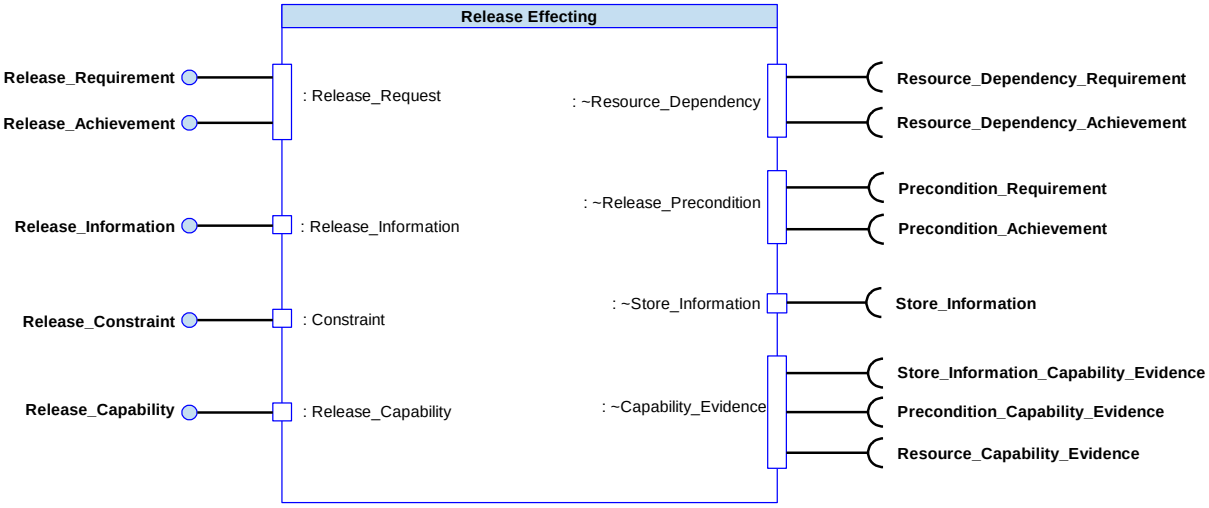

Figure 873: Release Effecting Service Summary

B.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).

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.

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).

predict_capability_progression

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

identify_missing_information

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

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_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).

identify_whether_release_requirement_is_achievable

- To identify whether a Store release Requirement is achievable given current Release_Effecting_Capability.

identify_pre-conditions

- To identify Pre-conditions required to support Release_Effecting_Steps that fulfil the Release_Timeline.

capture_release_effecting_constraints

- To capture provided Constraints, e.g. Stores are not to be released for training activities.

determine_release_element_states

- To determine the current state of release information on Stores and Release_Effecting_Resources.

B.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 coordination between multiple Store stations and multiple releases.
- The preparation of a Store for release (e.g. setting fusing options or priming with targeting data), only the Release_Timeline.

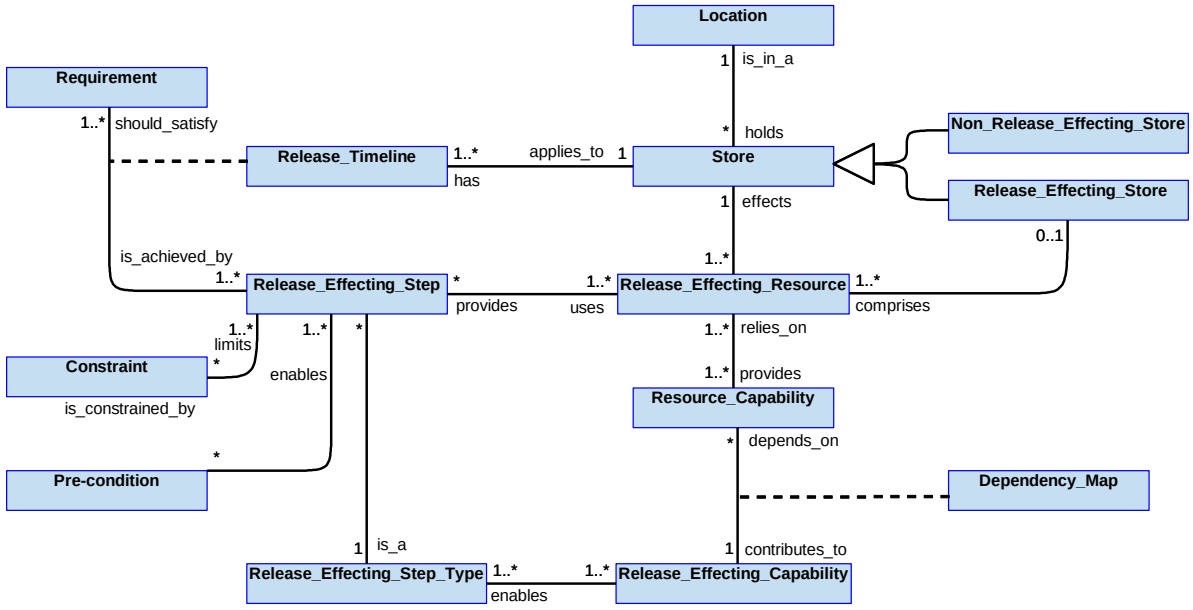

Figure 874: Release Effecting Semantics

B.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).

B.2.47.6 Design Rationale

B.2.47.6.1 Assumptions

None.

B.2.47.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy 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).

B.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.

B.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.

B.2.47.7 Services

B.2.47.7.1 Service Definitions

B.2.47.7.1.1 Release_Request

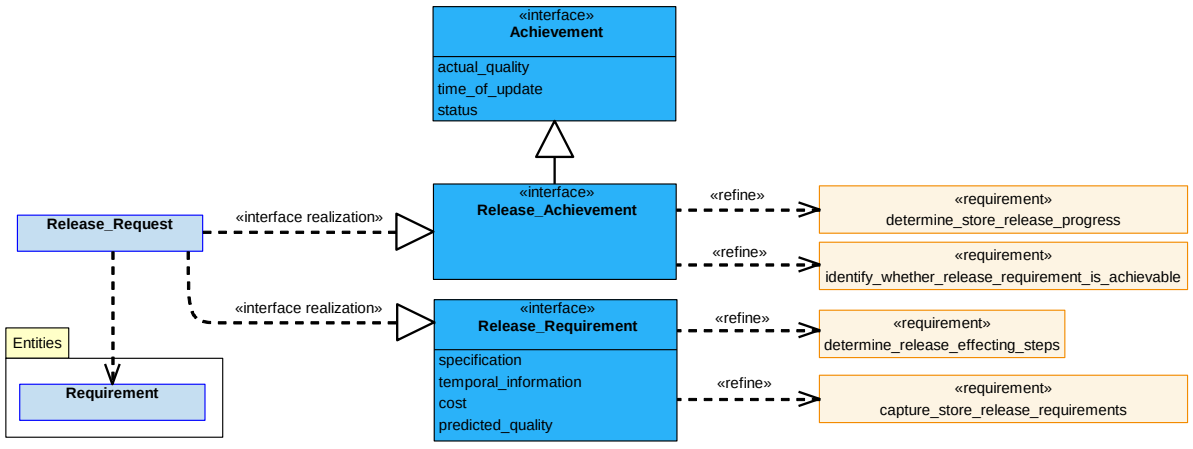

Figure 875: Release_Request Service Definition

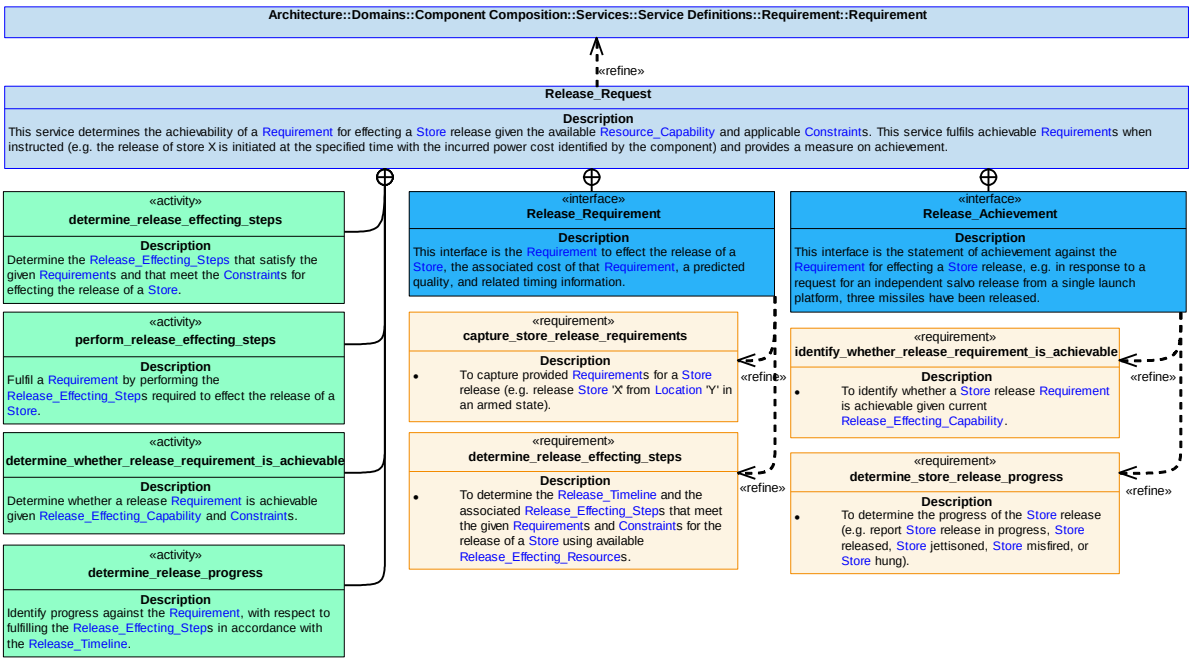

Figure 876: Release_Request Service Policy

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 an independent salvo release from a single launch platform, 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.

B.2.47.7.1.2 Resource_Dependency

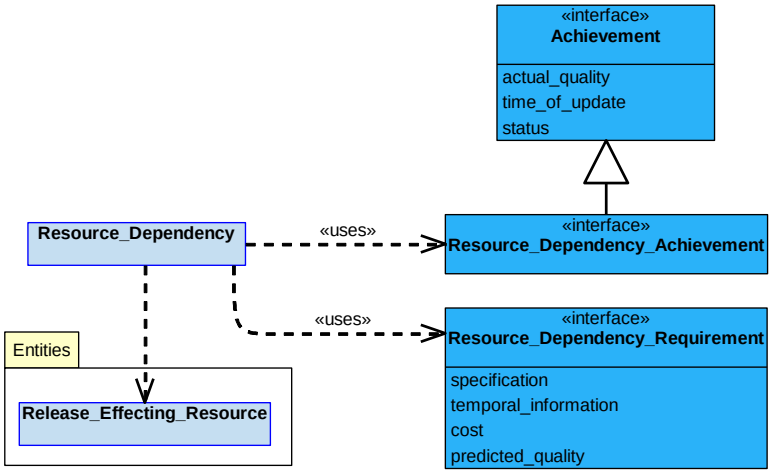

Figure 877: Resource_Dependency Service Definition

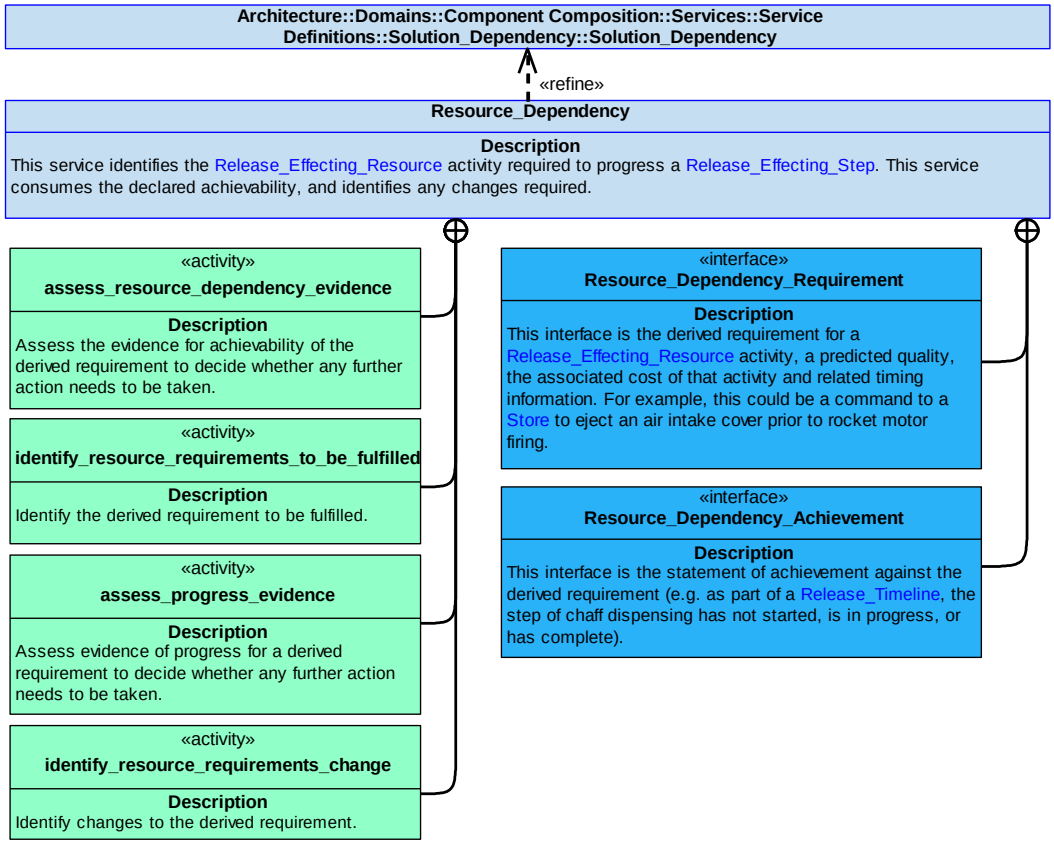

Figure 878: Resource_Dependency Service Policy

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 complete).

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.

B.2.47.7.1.3 Release_Precondition

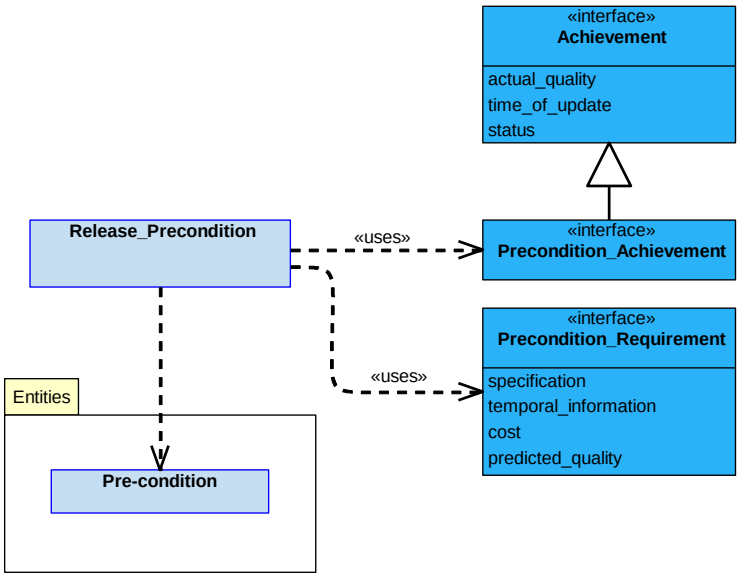

Figure 879: Release_Precondition Service Definition

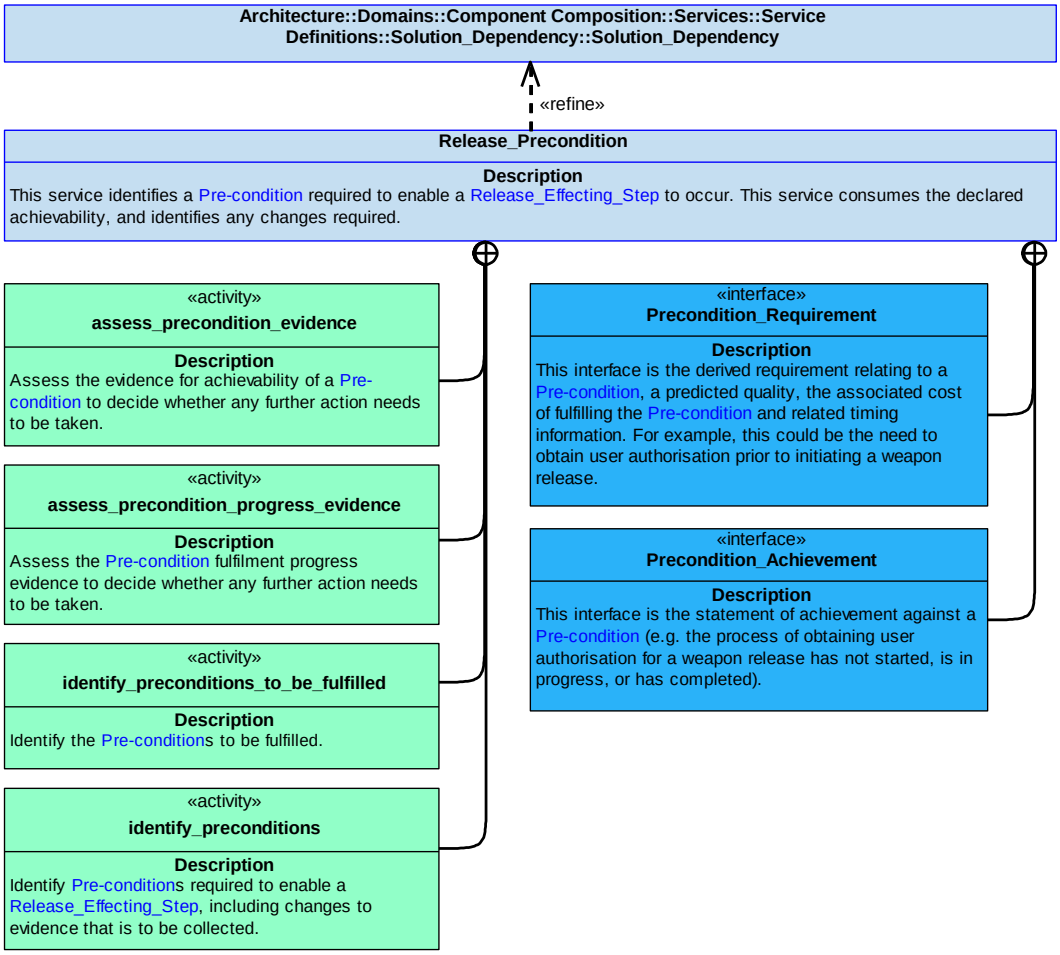

Figure 880: Release_Precondition Service Policy

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 obtain user authorisation prior to initiating a weapon 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 obtaining user authorisation for a weapon release 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.

B.2.47.7.1.4 Release_Information

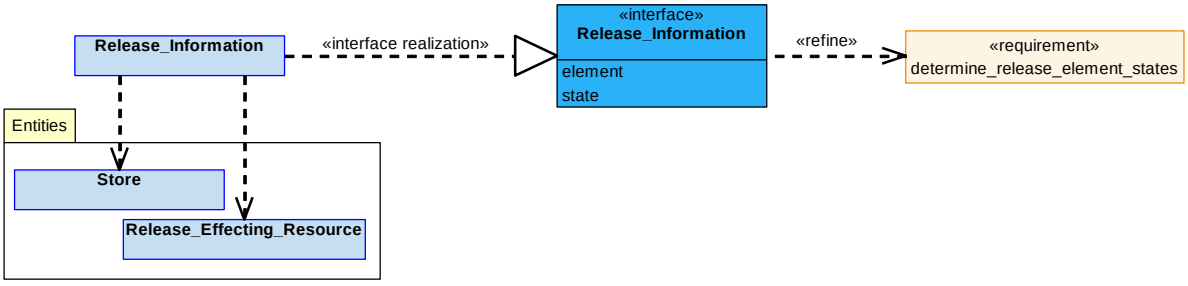

Figure 881: Release_Information Service Definition

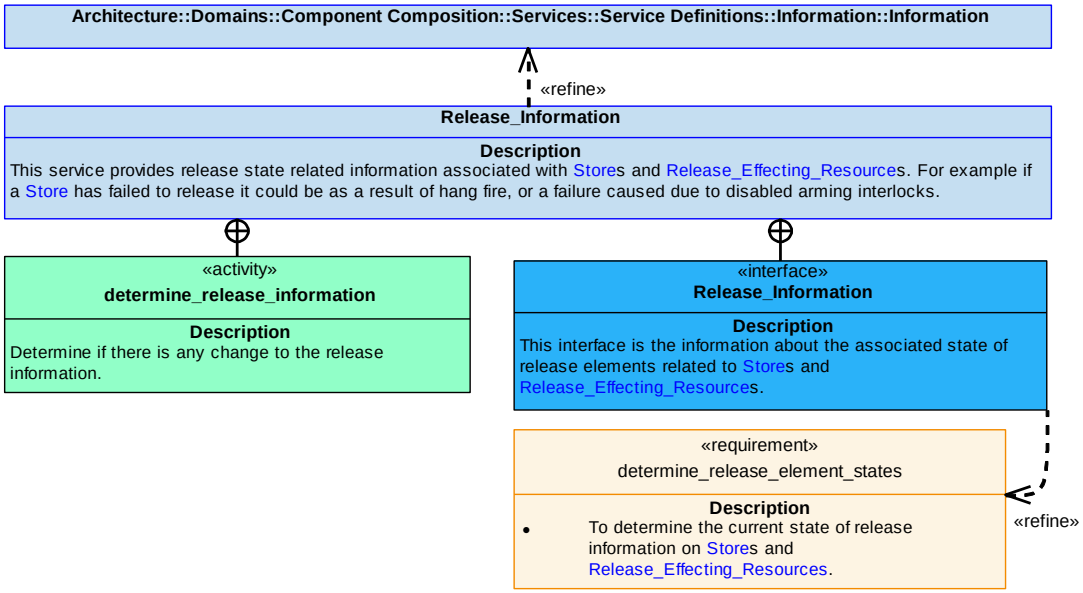

Figure 882: Release_Information Service Policy

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.

B.2.47.7.1.5 Store_Information

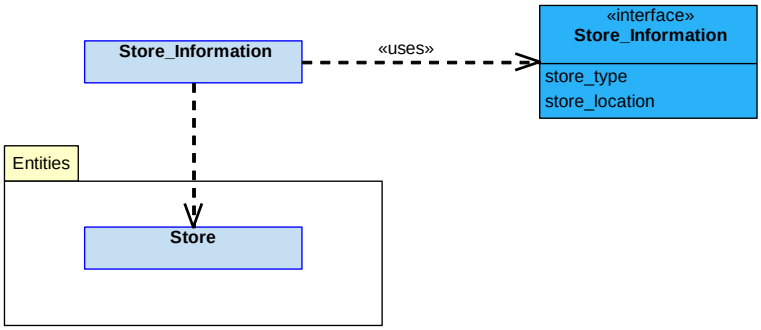

Figure 883: Store_Information Service Definition

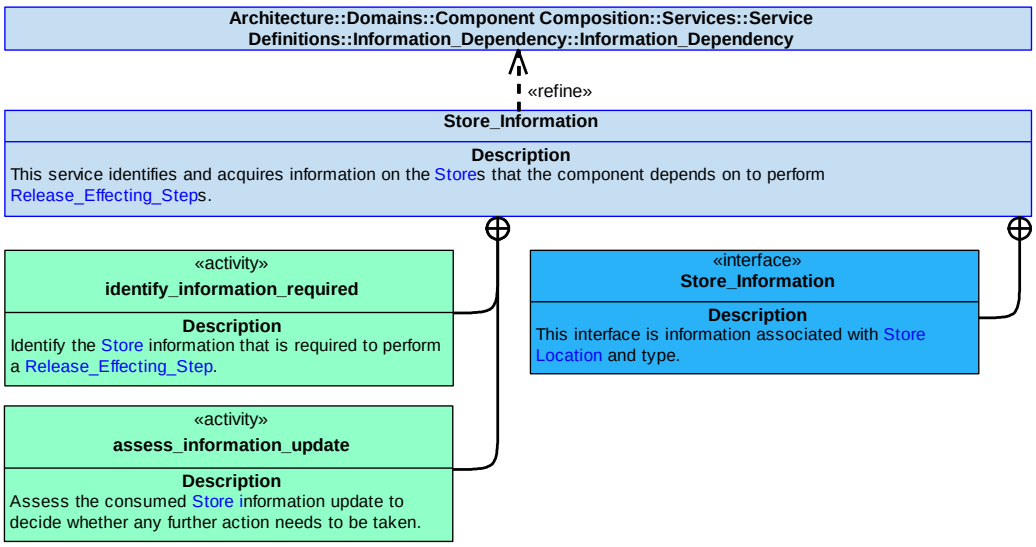

Figure 884: Store_Information Service Policy

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.

B.2.47.7.1.6 Constraint

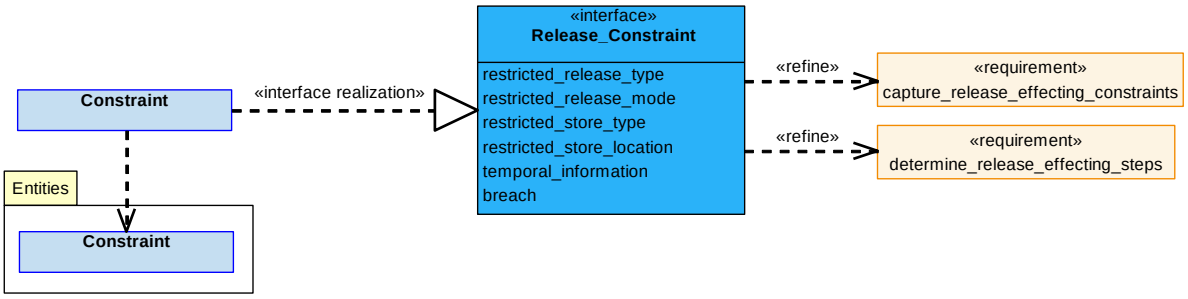

Figure 885: Constraint Service Definition

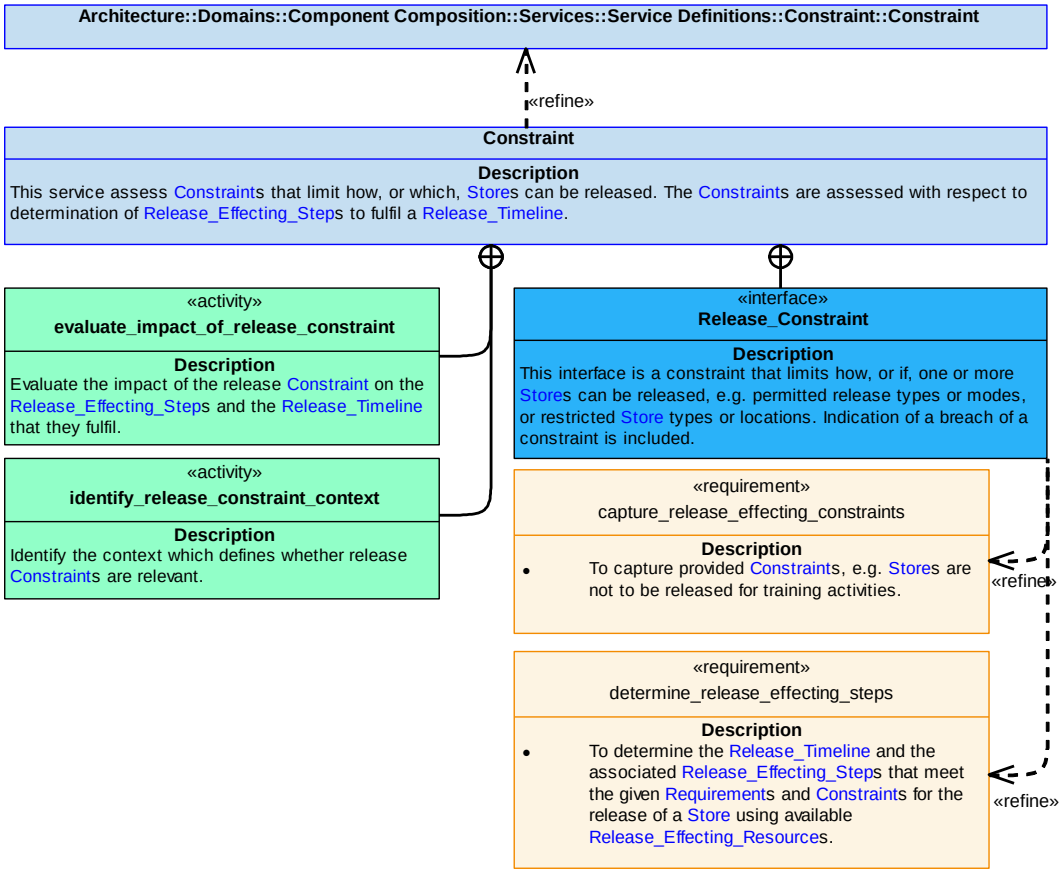

Figure 886: Constraint Service Policy

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.

B.2.47.7.1.7 Release_Capability

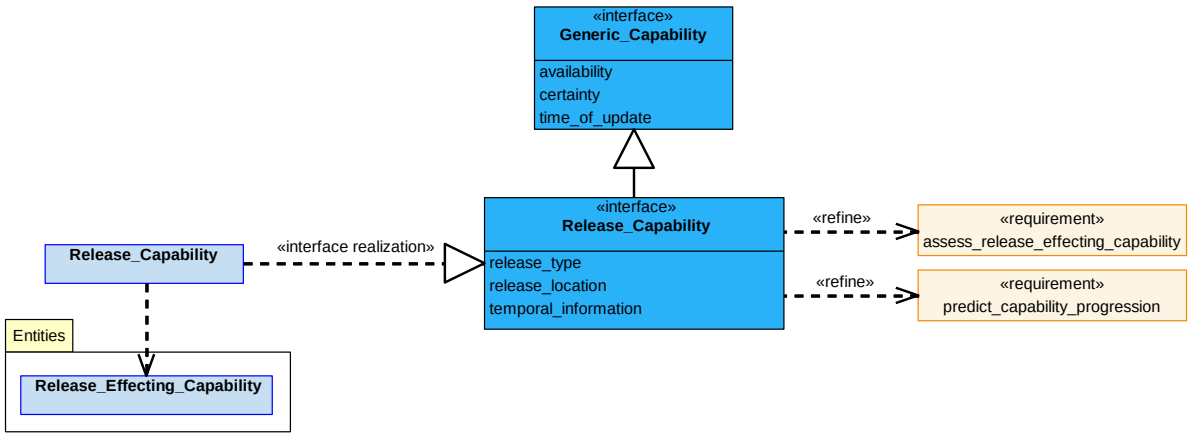

Figure 887: Release_Capability Service Definition

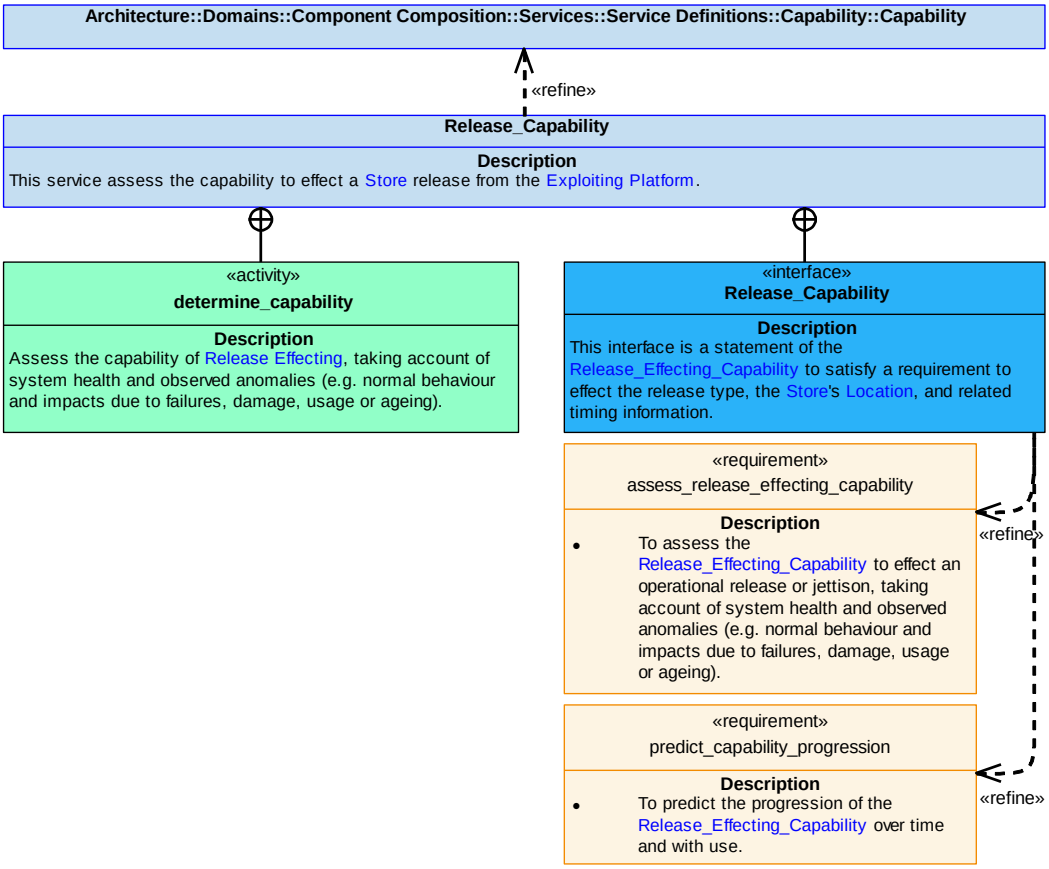

Figure 888: Release_Capability Service Policy

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).

B.2.47.7.1.8 Capability_Evidence

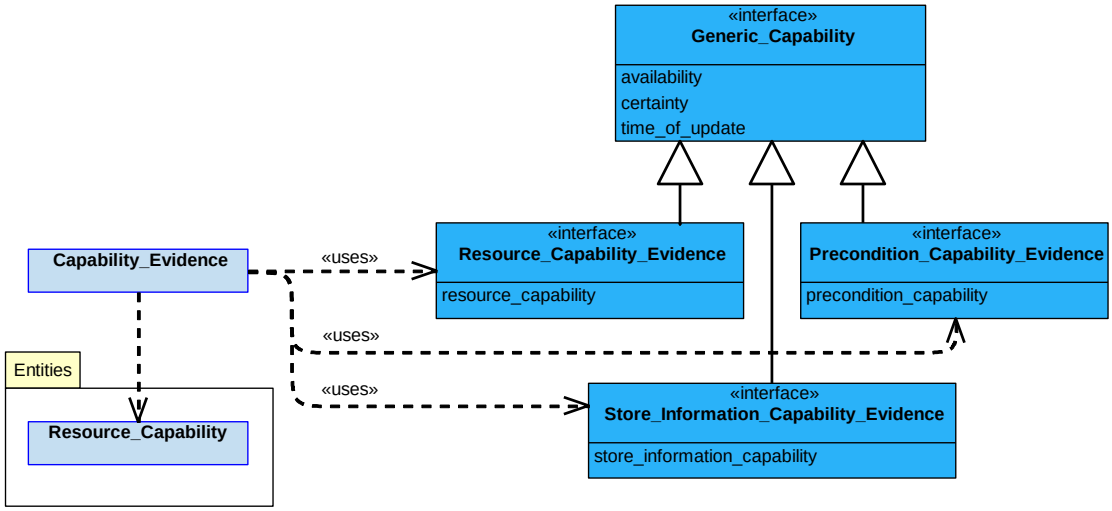

Figure 889: Capability_Evidence Service Definition

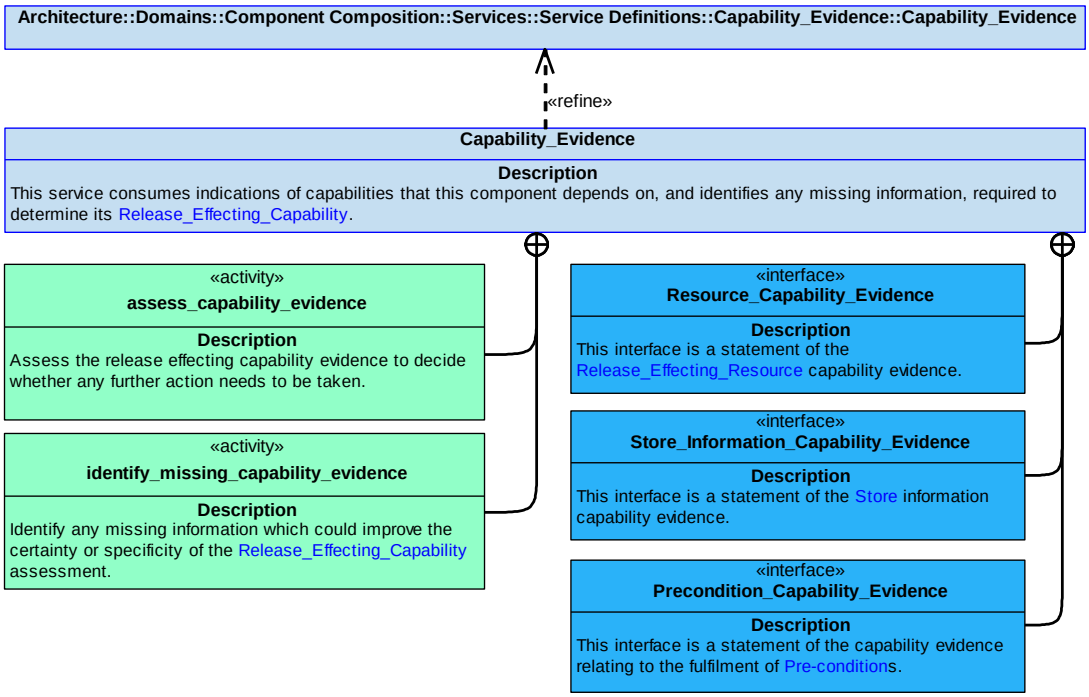

Figure 890: Capability_Evidence Service Policy

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.

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.

B.2.47.7.2 Service Dependencies

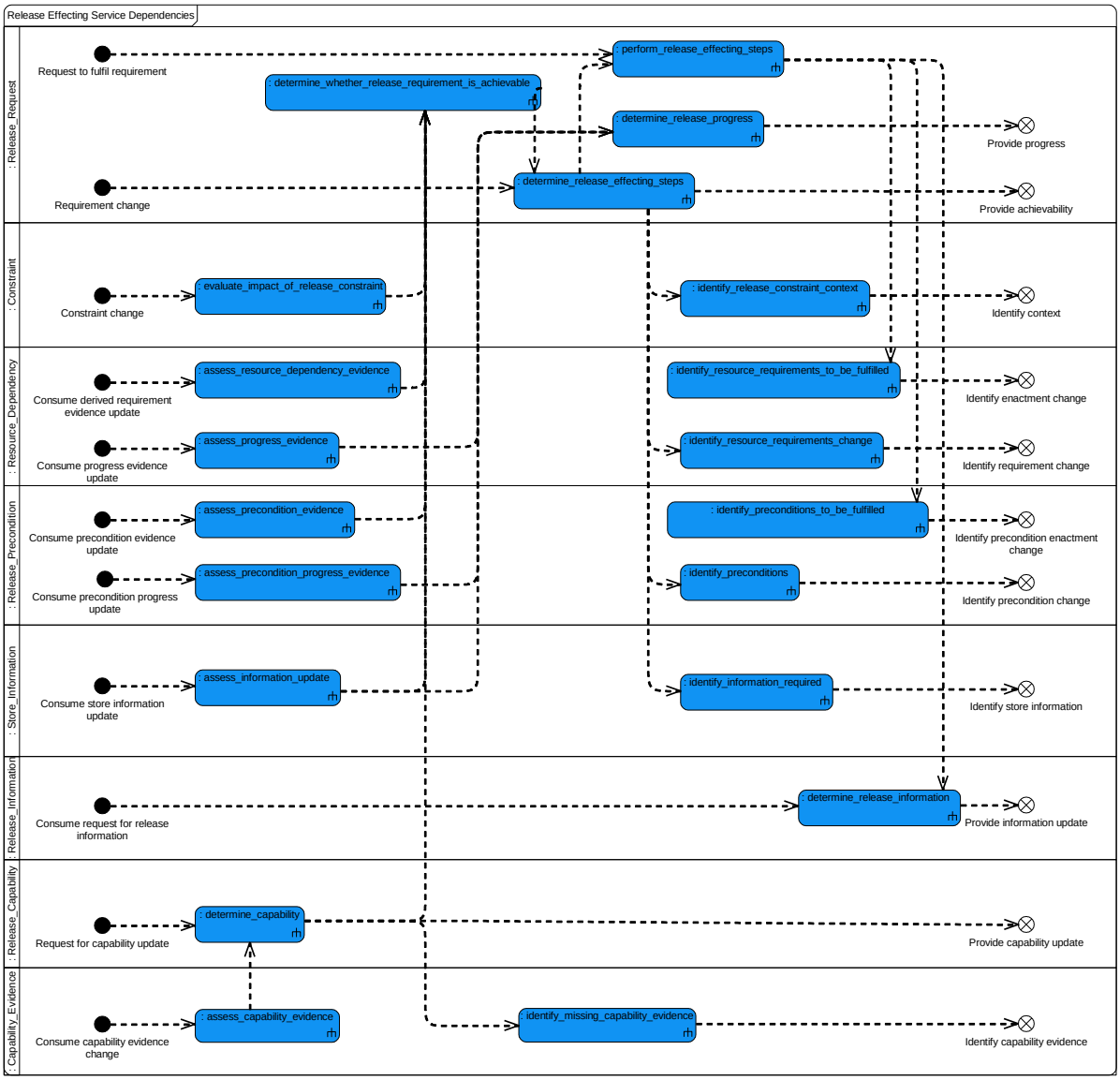

Figure 891: Release Effecting Service Dependencies

B.2.48 Resource Brokerage

B.2.48.1 Role

The role of Resource Brokerage is to allocate resources to meet current and future demands and to identify conflicts in resource allocation that prevent the demands from being met.

B.2.48.2 Overview

Control Architecture

Resource Brokerage is a service component as defined in the Control Architecture policy.

Standard Pattern of Use

Resource Brokerage receives a Request for the use of a Resource from a Resource user. Resource Brokerage determines the resulting Allocation based on the Scheme being used and reports the proposed Allocation to the Resource user. The Resource user accepts the Allocation, Resource Brokerage reserves the associated Resource and determines the updated Resource availability.

Examples of Use

Resource Brokerage will be used in any system which has Resources that can be dynamically allocated, for example:

- Ensuring that changes to a vehicle's planned flightpath can be accommodated with remaining fuel reserves.
- Use of a multi-function antenna for concurrent sensing or other tasks.

B.2.48.3 Service Summary

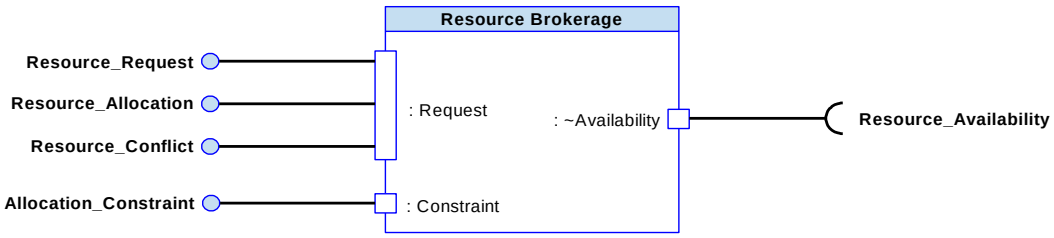

Figure 892: Resource Brokerage Service Summary

B.2.48.4 Responsibilities

capture_allocation_requirements

- To capture Requests for allocation of Resources.

capture_resource_dependencies

- To capture any Dependency between the use of different Resources.

maintain_request_traceability

- To maintain traceability of Allocations to their triggering Requests.

maintain_allocation_traceability

- To maintain traceability between Allocations of Resources due to a Dependency between them.

determine_allocations

- To determine Allocations using the appropriate Scheme in order to satisfy Requests.

identify_resource_availability

- To identify Resource availability, identify Conflicts, and determine the impact on Allocations.

manage_allocation_interruption

- To identify the interruption of an Allocation.

capture_allocation_constraints

- To capture provided Constraints, e.g. a particular requestor should not to be granted any Allocations.

B.2.48.5 Subject Matter Semantics

The subject matter of Resource Brokerage is Resource allocation.

Exclusions

The subject matter of Resource Brokerage does not include:

- The actual provision of Resources or how they are used in executing system actions.

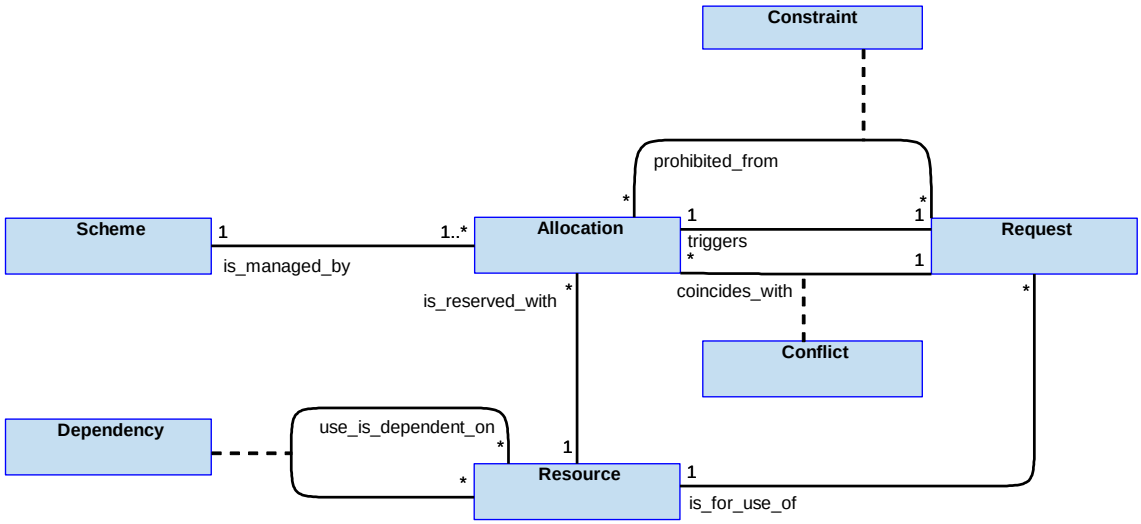

Figure 893: Resource Brokerage Semantics

B.2.48.5.1 Entities

Allocation

An amount of a Resource allocated as a result of a Request.

Conflict

An overlap between an existing Allocation and a new Request to use the same Resource at the same time.

Dependency

A reliance on the use of a Resource by another.

Request

A requirement for an Allocation of a Resource.

Resource

Something that is available to be used to execute an action and whose allocation requires management.

Scheme

A strategy for determining the Allocations of a Resource, e.g. scheduling time slots, provisioning interrupt periods or budgeting an available quantity.

Constraint

A prohibition on particular requestors from triggering Allocations. For example a constraint may apply if a resource user has been compromised due to a cyber attack.

B.2.48.6 Design Rationale

B.2.48.6.1 Assumptions

- Resource providers will provide this component with an up-to-date view of available Resources, allowing it to identify the total availability of Resources.
- Each Resource brokered by this component has a single Resource provider.

B.2.48.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Resource Brokerage:

- Data Driving - data driving is a recommended approach to designing this component as it is only concerned with the general characteristics of Resources that support their scheduling and is not concerned with the detailed behaviour or state of different types of Resource.
- Resource Management - This policy describes how the Resource Brokerage component is intended to be used to support the management of system Resources.

Note that the Resource Brokerage component's capability to broker Conflicts between Requests is not dependent on consumed capability evidence and does not evolve with time, and this component is not concerned with the provided capability of any particular subject matter area hence does not determine capability in the way described in the Capability Assessment policy.

Extensions

- This component may be supported by extensions which define the policy for managing, and the specific characteristics of, a particular category of Resource. For example Resources that can be consumed, shared or replenished.

Other Factors that were Taken into Account

- This component ensures that Resource providers do not need to manage a reservation timeline and are only concerned with the complexities of their subject matter.

Exploitation Considerations

- The Exploiter will be required to define the instances of this component required for an Exploiting Platform, and the assignment of Resources to these instances.
- Dependencies can be simultaneous (e.g. a door has to stay open whilst a sensor behind that door is being used) or sequential (e.g. a seeker head has to be cooled to a working temperature before it can be used).
- The requirement could include details of the scope of the resource to be used, e.g. the specific frequency band of an emitter.
- Requests can take the form of:
  - A requested amount, e.g. a quantity of fuel.
  - A requested period of time, e.g. a scheduled slot for the use of a resource.
  - A potential period of time, e.g. a period in which a resource may be required and which would interrupt other users of that resource.

B.2.48.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

Failure of this component could:

- Fail to allocate Resource when it is required. 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 involved systems.
- Incorrectly allocate Resource when it is already being used. This could cause the functionality dependant on the Resource to be unavailable. For example, allowing two high power consumers of electrical power to operate simultaneously may result in the power supply being overloaded and tripping out. Consequently, the functionality of all consumers of that power supply would be unavailable.
- Allocate Resource so it is used up, and so not available later in the flight when it may be required for, potentially, a safety critical function. For example, depleting 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.

B.2.48.6.4 Security Considerations

The indicative security classification is O-S.

Based on the assumption that this component will manage the system access to shared Resources without requiring any knowledge of performance, capability or mission objectives, the indicative security classification is considered to be O-S. Where types of data or aggregation of data may occur that might provide an insight to platform capability, this may lead to a more stringent classification being required. 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 resources will not be adequately managed and may not be provided to users at all.

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

- **Identifying Data Sources** with regards the demands for resources being only brokered if submitted by trusted users of the resource.
- **Maintaining Audit Records** of the resources brokered for accountability purposes**.**
- **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 the monitoring of the system resource status, and of requests for resources not normally used in the scope of the requester.

This component is considered unlikely to directly implement any security enforcing functions.

B.2.48.7 Services

B.2.48.7.1 Service Definitions

B.2.48.7.1.1 Request

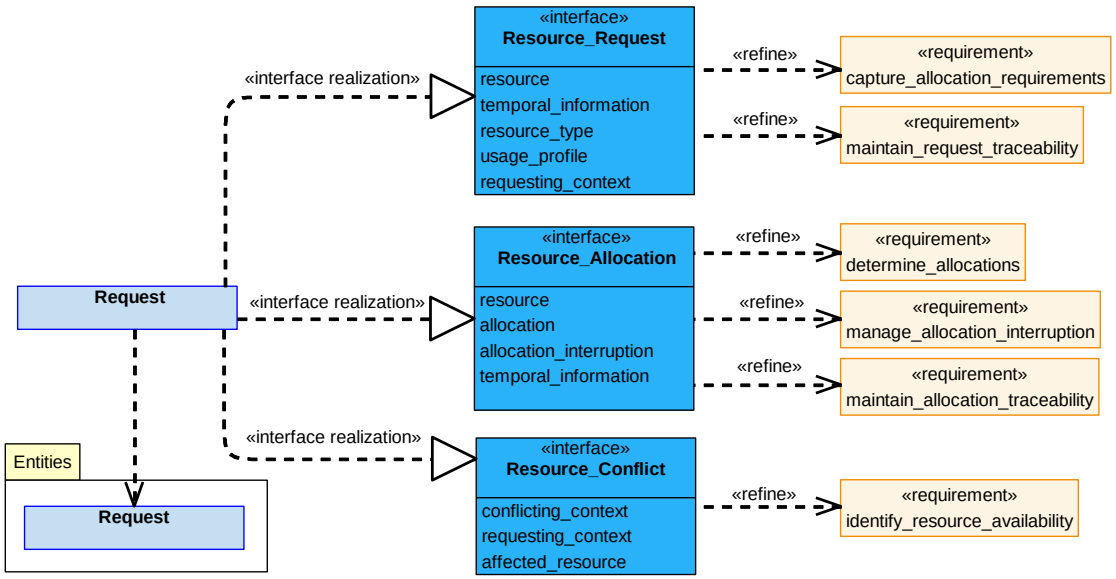

Figure 894: Request Service Definition

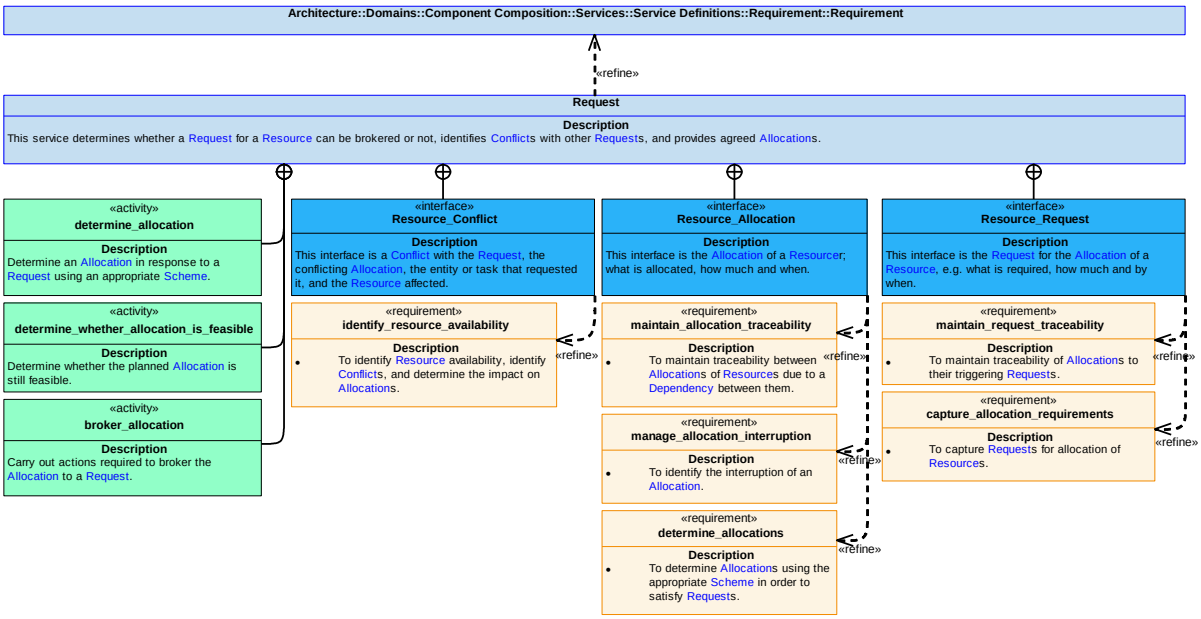

Figure 895: Request Service Policy

Request

This service determines whether a Request for a Resource can be brokered or not, identifies Conflicts with other Requests, and provides agreed Allocations.

Interfaces

Resource_Request

This interface is the Request for the Allocation of a Resource, e.g. what is required, how much and by when.

Attributes

resource	The Resource being requested.
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.
resource_type	The type of Resource, e.g. a consumed Resource (such as fuel), a shared Resource (such as cooling capacity) or an interruptible Resource (such as exclusive time using a multi-function array).
usage_profile	The quantity of Resource requested for use (e.g. a one-off amount or an increasing number of litres for a consumed Resource, a percentage of overall capacity of a shared Resource, or time block allocation for an interruptible Resource).
requesting_context	The tasking information that identifies the source or reason for the Request.

Resource_Allocation

This interface is the Allocation of a Resourcer; what is allocated, how much and when.

Attributes

resource	The Resource being allocated.
allocation	The allocated quantity to meet the usage profile (e.g. a number of litres, percentage of capacity or block of time).
allocation_interruption	A known or necessary interruption in the allocation of a Resource.
temporal_information	Information covering timing of the allocated Resource, such as start and end times.

Resource_Conflict

This interface is a Conflict with the Request, the conflicting Allocation, the entity or task that requested it, and the Resource affected.

Attributes

conflicting_context	The tasking information that identifies the source or reason for which an existing Allocation that conflicts with a Request, was raised.
requesting_context	The tasking information that identifies the source or reason for the Request.
affected_resource	The requested Resource affected by a conflicting Allocation.

Activities

determine_allocation

Determine an Allocation in response to a Request using an appropriate Scheme.

broker_allocation

Carry out actions required to broker the Allocation to a Request.

determine_whether_allocation_is_feasible

Determine whether the planned Allocation is still feasible.

B.2.48.7.1.2 Availability

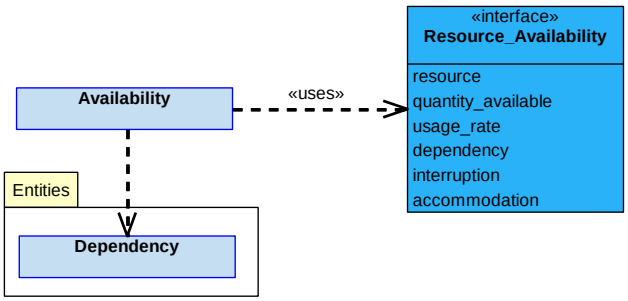

Figure 896: Availability Service Definition

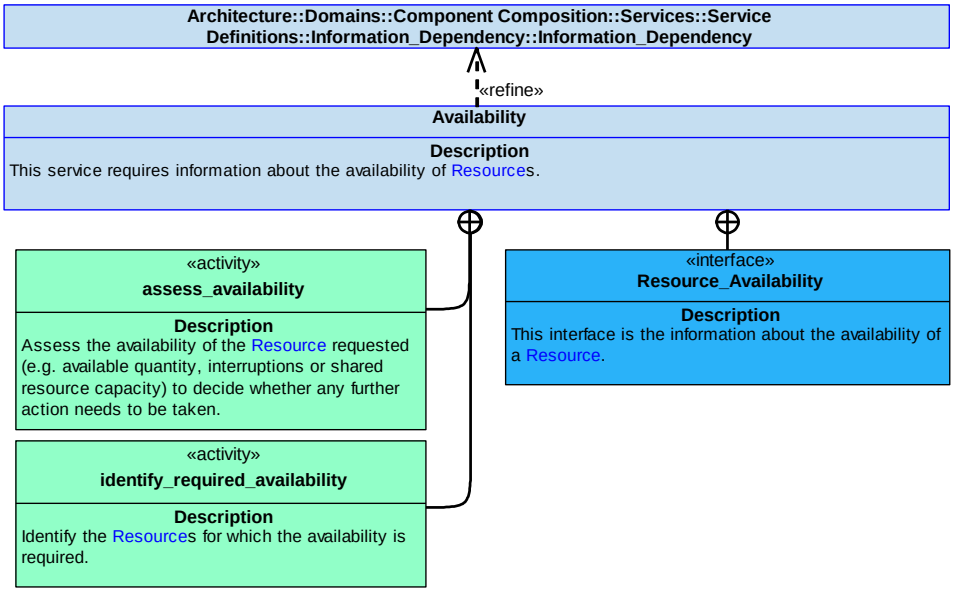

Figure 897: Availability Service Policy

Availability

This service requires information about the availability of Resources.

Interface

Resource_Availability

This interface is the information about the availability of a Resource.

Attributes

resource	The Resource in question.
quantity_available	The available quantity of a Resource.
usage_rate	The rate at which the Resource is being used.
dependency	A dependency placed upon the availability of the Resource.
interruption	A known interruption in the availability of a Resource.
accommodation	Whether an interruption can be accommodated.

Activities

assess_availability

Assess the availability of the Resource requested (e.g. available quantity, interruptions or shared resource capacity) to decide whether any further action needs to be taken.

identify_required_availability

Identify the Resources for which the availability is required.

B.2.48.7.1.3 Constraint

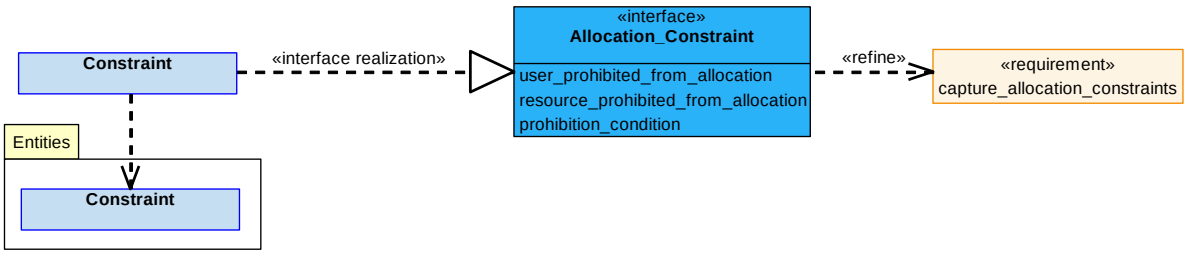

Figure 898: Constraint Service Definition

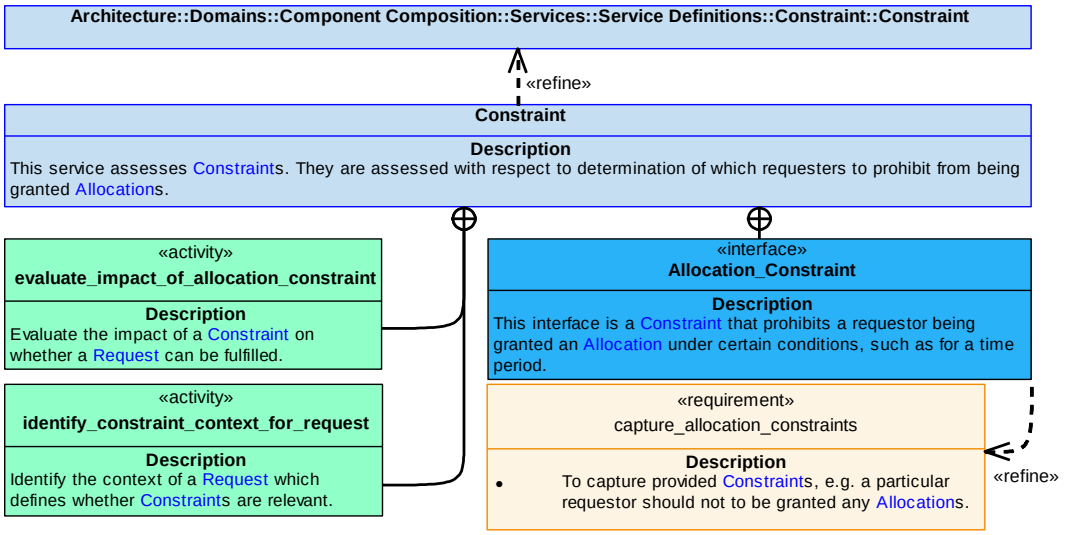

Figure 899: Constraint Service Policy

Constraint

This service assesses Constraints. They are assessed with respect to determination of which requesters to prohibit from being granted Allocations.

Interface

Allocation_Constraint

This interface is a Constraint that prohibits a requestor being granted an Allocation under certain conditions, such as for a time period.

Attributes

user_prohibited_from_allocation	The user, or type of user, that is prohibited from being allocated a Resource.
resource_prohibited_from_allocation	The Resource, or type of Resource, to which the restriction applies.
prohibition_condition	The conditions under which a Constraint is applicable, such as for a certain time period, or when a certain number of requests are received from any (or specific) requesters.

Activities

evaluate_impact_of_allocation_constraint

Evaluate the impact of a Constraint on whether a Request can be fulfilled.

identify_constraint_context_for_request

Identify the context of a Request which defines whether Constraints are relevant.

B.2.48.7.2 Service Dependencies

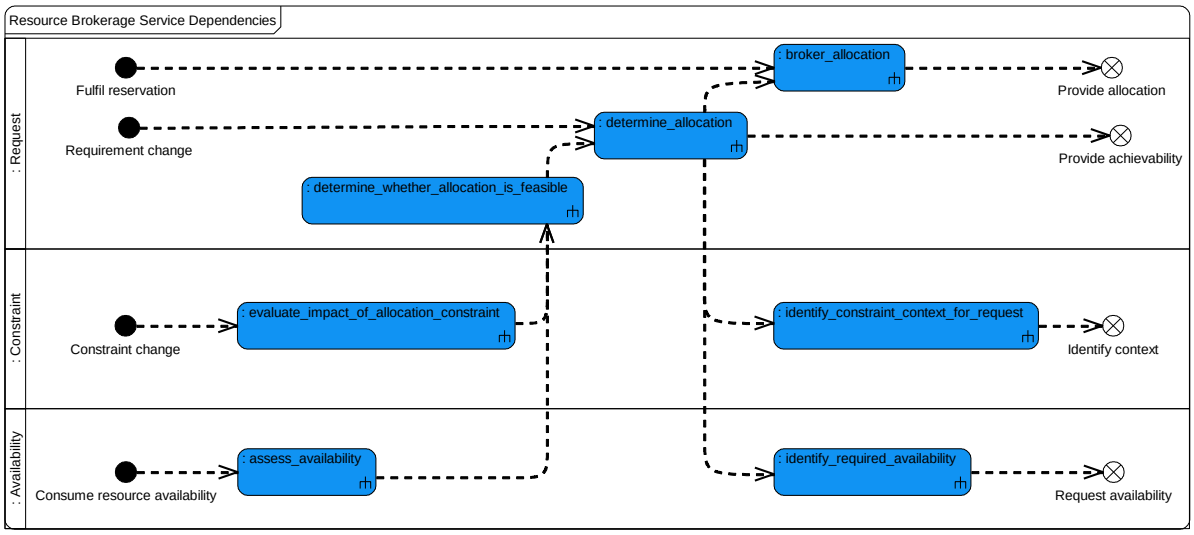

Figure 900: Resource Brokerage Service Dependencies

B.2.49 Routes

B.2.49.1 Role

The role of Routes is to determine and manage the execution of a route to satisfy positioning requirements for a vehicle.

B.2.49.2 Overview

Control Architecture

Routes is an action component as defined in the Control Architecture policy.

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.

B.2.49.3 Service Summary

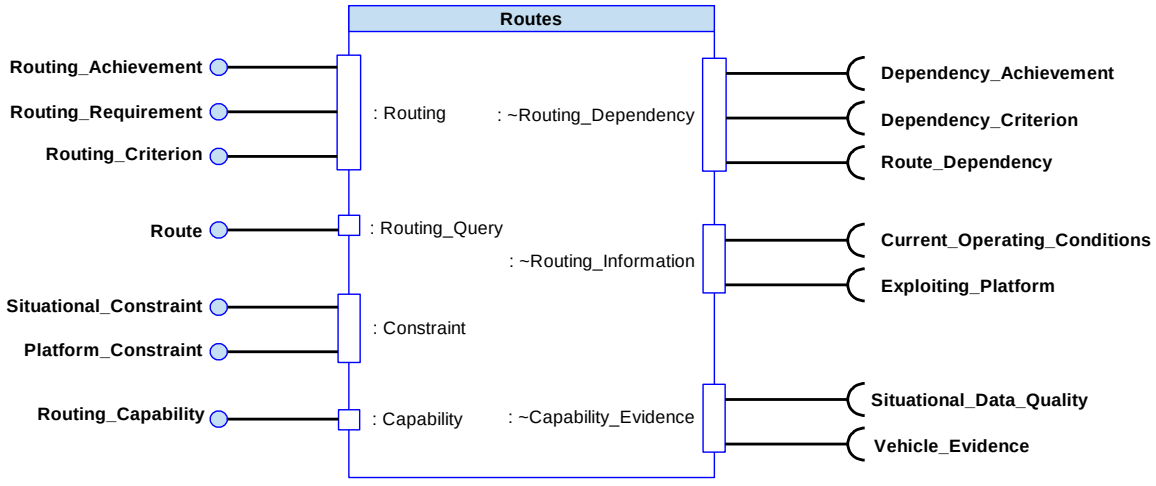

Figure 901: Routes Service Summary

B.2.49.4 Responsibilities

capture_positioning_requirements

- To capture given Positioning_Requirements.

capture_routing_constraints

- To capture given Routing_Constraints (e.g. weather volumes and threat volumes).

assess_routing_capability

- To determine the Routing_Capability, taking into account system health and observed anomalies.

predict_routing_capability

- To predict the progression of Routing_Capability over time and with use, taking account of system health and observed anomalies.

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.

collate_route_cost

- To collate the Cost (e.g. time or fuel use) for any generated Route against the provided Measurement_Criterion.

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.

capture_measurement_criteria

- To capture provided Measurement_Criterion (e.g. fuel Cost) for Routes.

identify_whether_requirement_remains_achievable

- To identify whether a Positioning_Requirement is still achievable given current or predicted Routing_Capability.

identify_missing_information

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

B.2.49.5 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.

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.

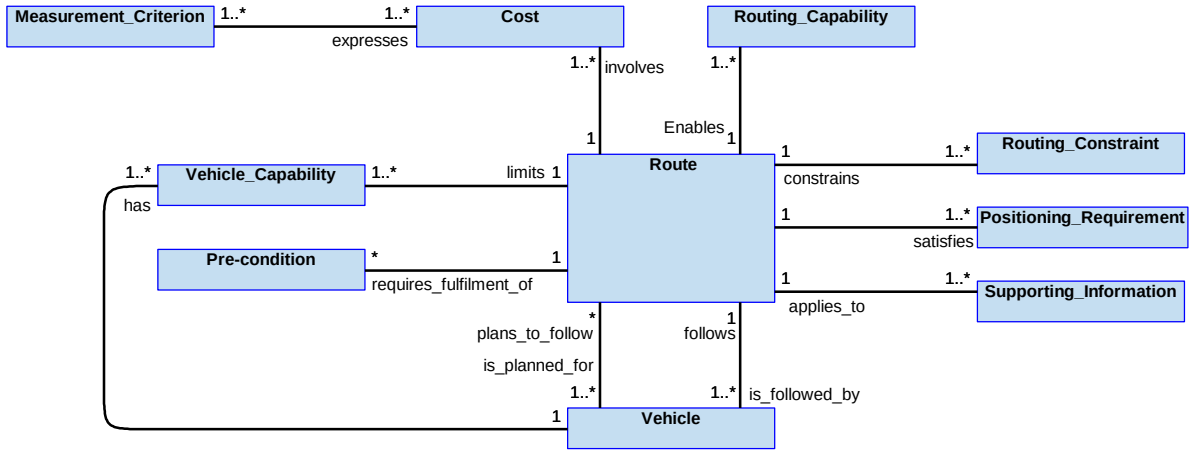

Figure 902: Routes Semantics

B.2.49.5.1 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. time, sequence, pattern and execution limit).

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

Constraints 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 Exploiting Platform or information about the operating environment.

Vehicle

A moveable object that requires a Route.

Vehicle_Capability

The capability of the Vehicle to execute Routes.

B.2.49.6 Design Rationale

B.2.49.6.1 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. attitude and altitude).

B.2.49.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy 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).

B.2.49.6.3 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).

B.2.49.6.4 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.

B.2.49.7 Services

B.2.49.7.1 Service Definitions

B.2.49.7.1.1 Routing

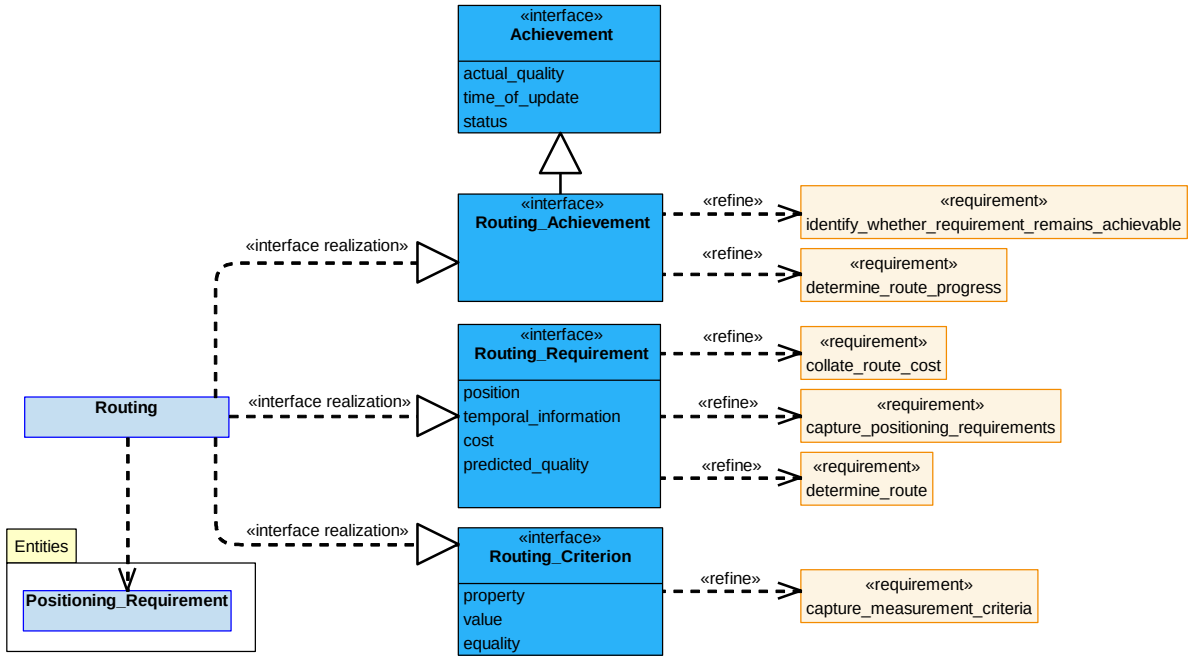

Figure 903: Routing Service Definition

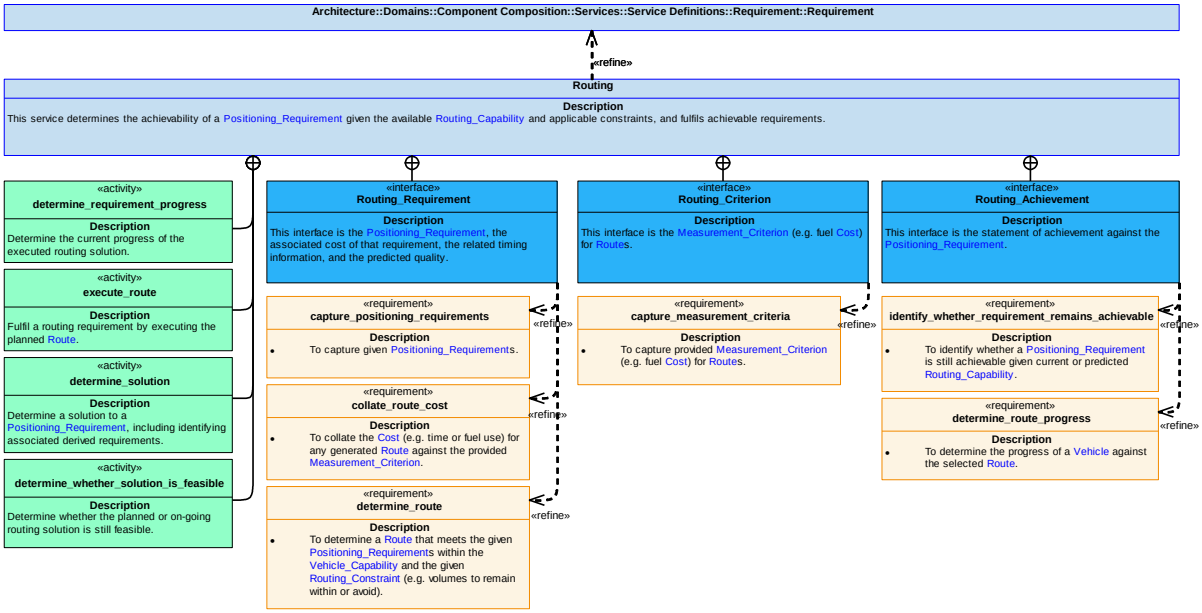

Figure 904: Routing Service Policy

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_requirement_is_achievable

Determine whether the planned or on-going Positioning_Requirement is still achievable.

B.2.49.7.1.2 Routing_Dependency

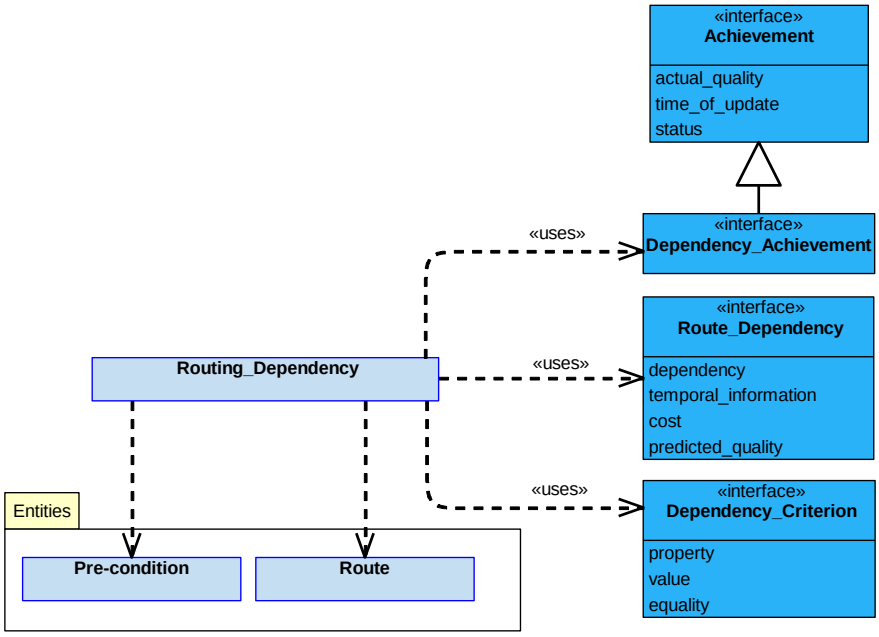

Figure 905: Routing_Dependency Service Definition

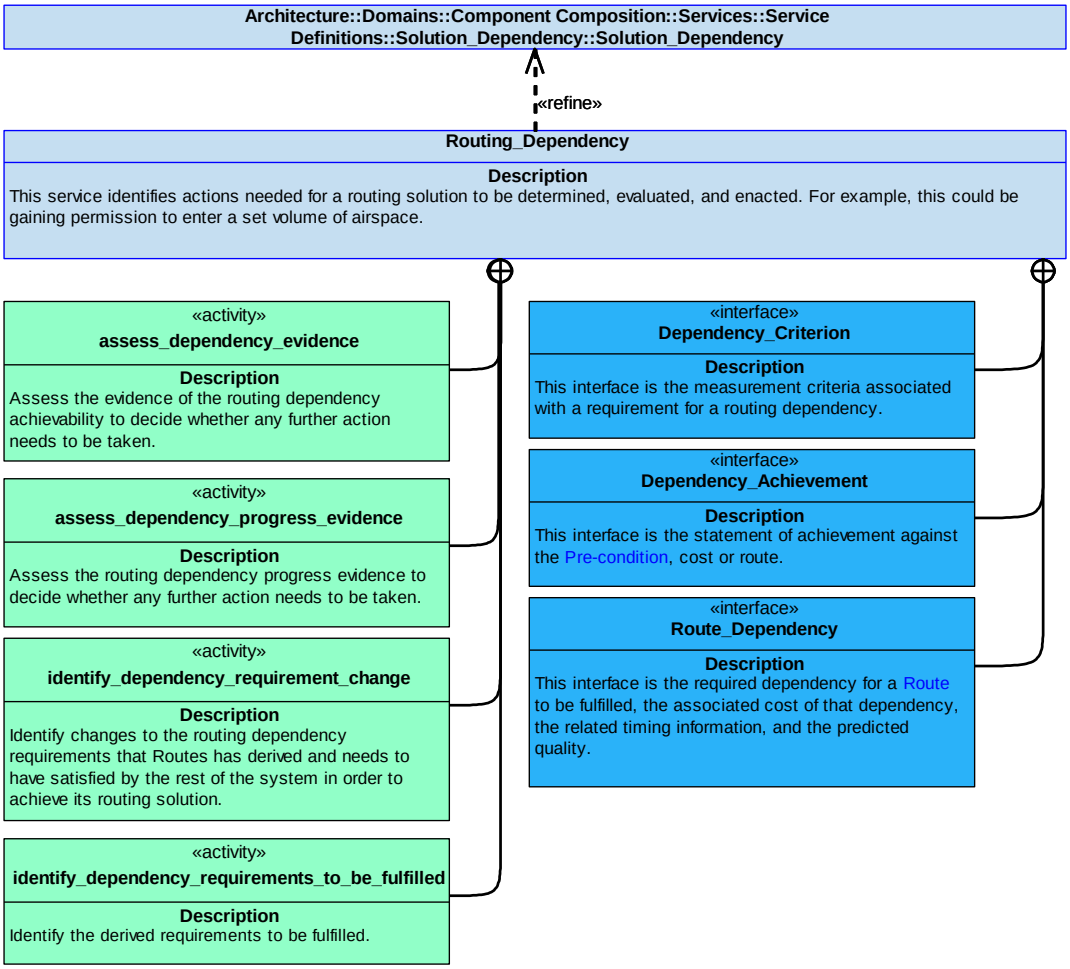

Figure 906: Routing_Dependency Service Policy

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 you have permission 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.

B.2.49.7.1.3 Routing_Query

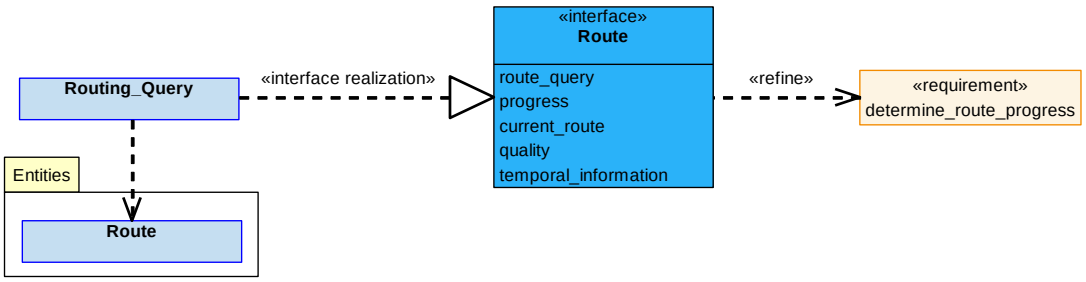

Figure 907: Routing_Query Service Definition

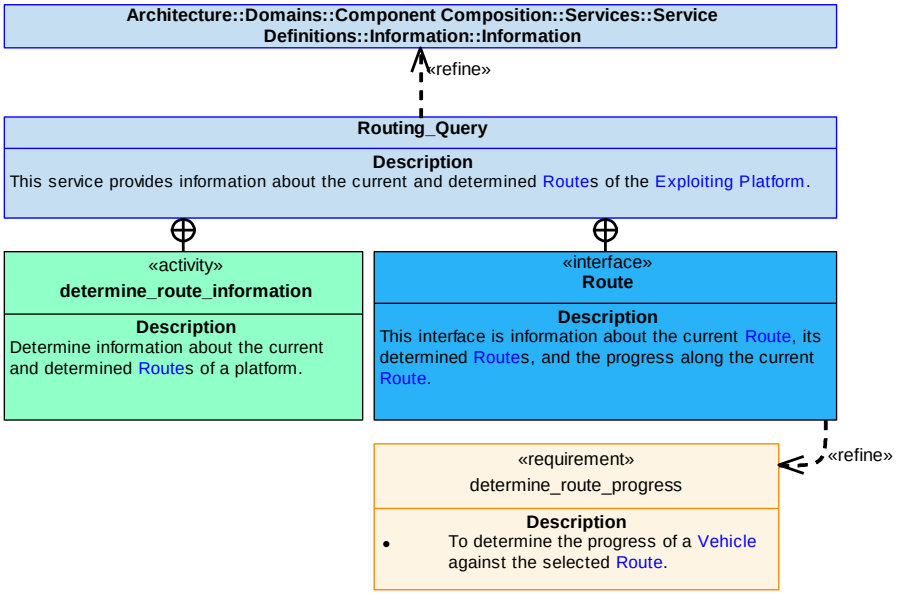

Figure 908: Routing_Query Service Policy

Routing_Query

This service provides information about the current and determined Routes of the Exploiting Platform.

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 an Exploiting Platform.
progress	The Exploiting Platforms progress along a Route.
current_route	The current Route of the Exploiting Platform.
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.

B.2.49.7.1.4 Routing_Information

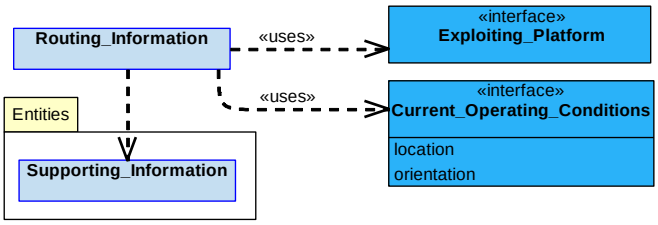

Figure 909: Routing_Information Service Definition

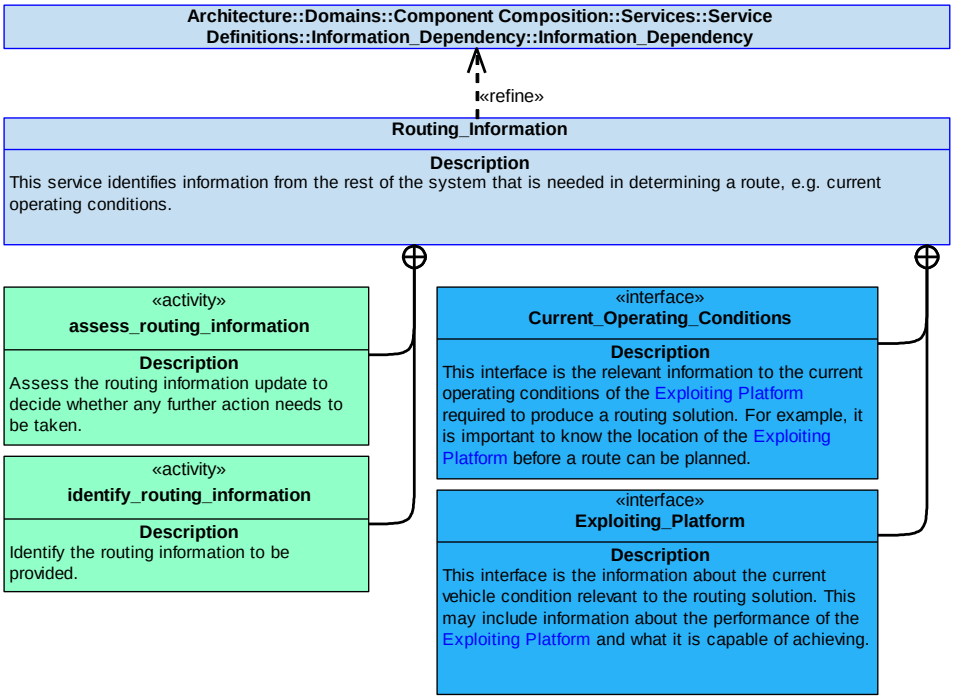

Figure 910: Routing_Information Service Policy

Routing_Information

This service identifies information from the rest of the system that is needed in determining a route, e.g. current operating conditions.

Interfaces

Exploiting_Platform

This interface is the information about the current vehicle condition relevant to the routing solution. This may include information about the performance of the Exploiting Platform and what it is capable of achieving.

Current_Operating_Conditions

This interface is the relevant information to the current operating conditions of the Exploiting Platform required to produce a routing solution. For example, it is important to know the location of the Exploiting Platform before a route can be planned.

Attributes

location	The current location of the Exploiting Platform.
orientation	The orientation of the Exploiting Platform.

Activities

assess_routing_information

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

identify_routing_information

Identify the routing information to be provided.

B.2.49.7.1.5 Constraint

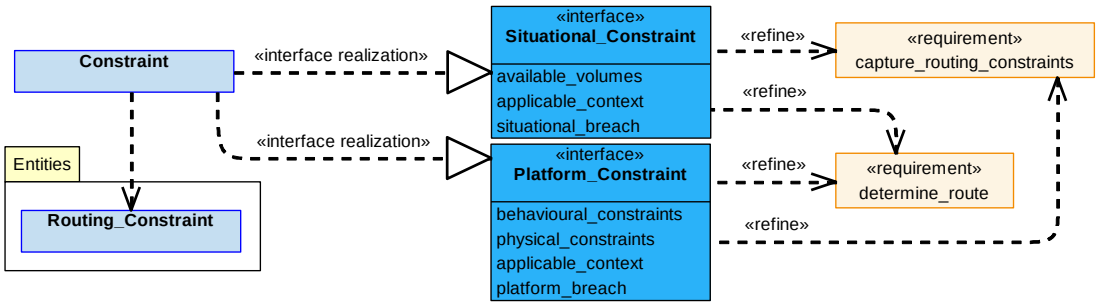

Figure 911: Constraint Service Definition

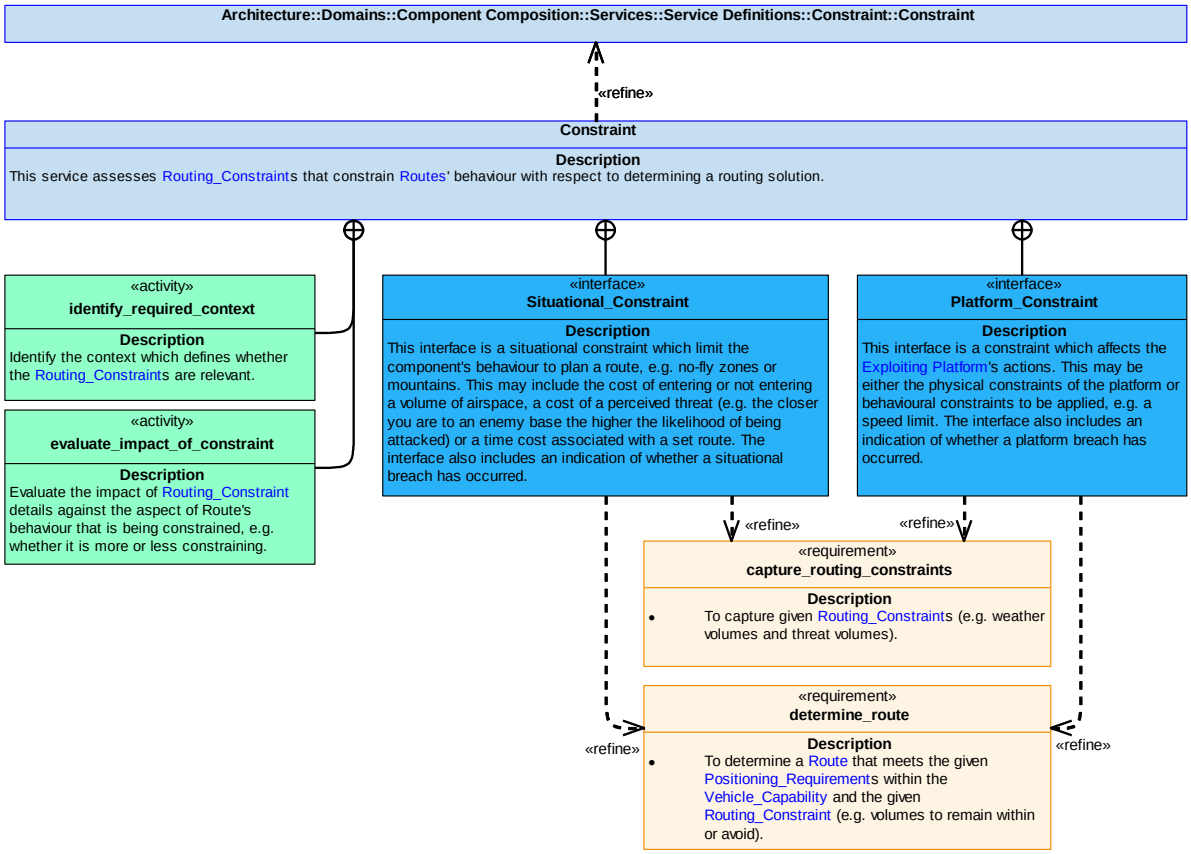

Figure 912: Constraint Service Policy

Constraint

This service assesses Routing_Constraints that constrain Routes' behaviour with respect to determining a routing solution.

Interfaces

Situational_Constraint

This interface is a situational constraint which limit the component's behaviour to plan a route, e.g. no-fly zones or mountains. This may include the cost of entering or not entering a volume of airspace, a cost of a perceived threat (e.g. the closer you are to an enemy base the higher the likelihood of being attacked) or a time cost associated with a set route. The interface also includes an indication of whether a situational breach has occurred.

Attributes

available_volumes	The available volumes which the Vehicle is allowed to use. For example, areas of clear airspace with no limitations, for example, no traffic or no physical obstacles such as land.
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 a constraint which affects the Exploiting Platform's actions. This may be either the physical constraints of the platform or behavioural constraints to be applied, e.g. a speed limit. The interface also includes an indication of whether a platform breach has occurred.

Attributes

behavioural_constraints	Limits that must be followed, such as minimum and maximum altitudes and speeds.
physical_constraints	Constraints applied by the Exploiting Platform such as fuel capacity or maximum cruising speeds.
applicable_context	The context in which the constraint is applicable.
platform_breach	A statement that the platform 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 Route's behaviour that is being constrained, e.g. whether it is more or less constraining.

B.2.49.7.1.6 Capability

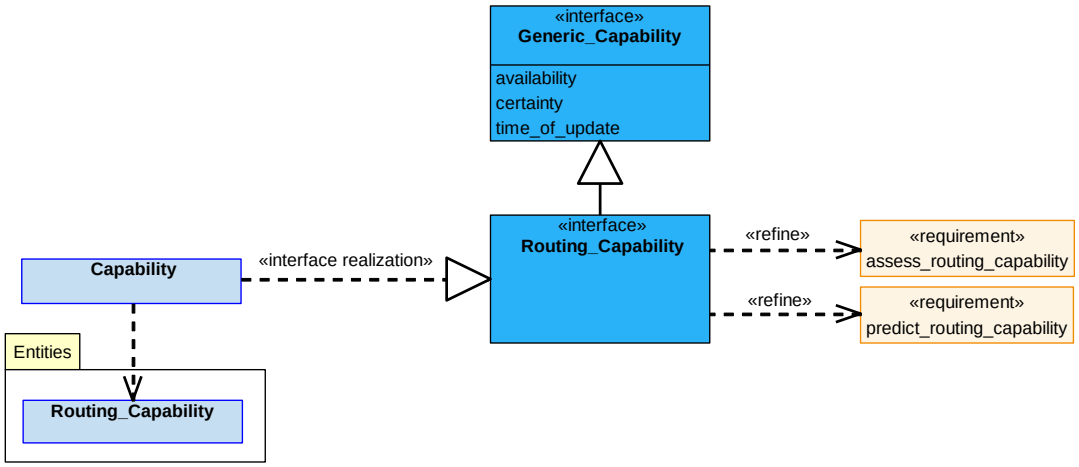

Figure 913: Capability Service Definition

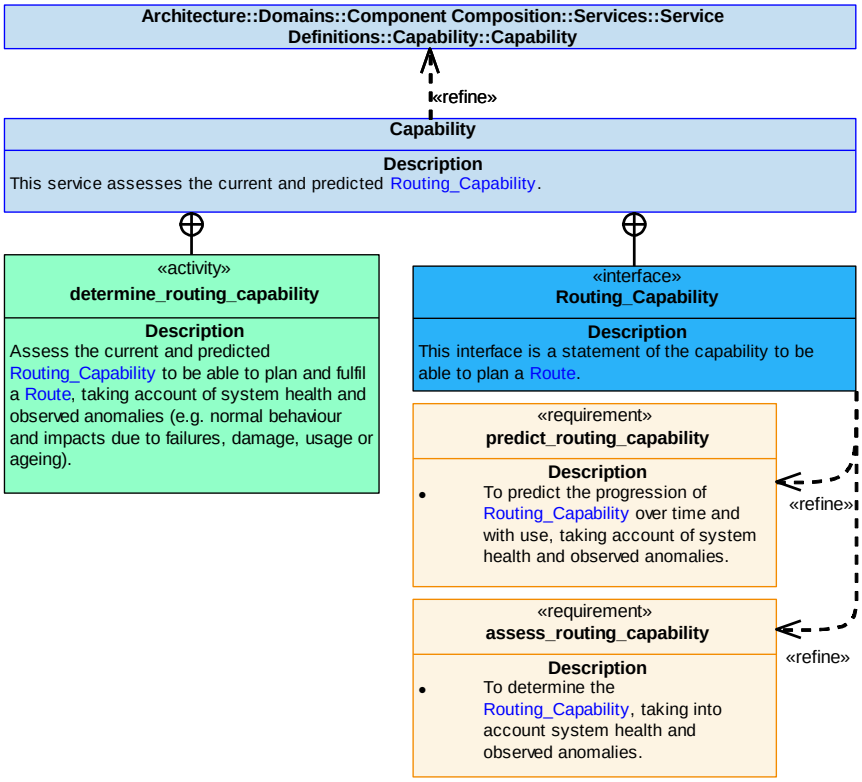

Figure 914: Capability Service Policy

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).

B.2.49.7.1.7 Capability_Evidence

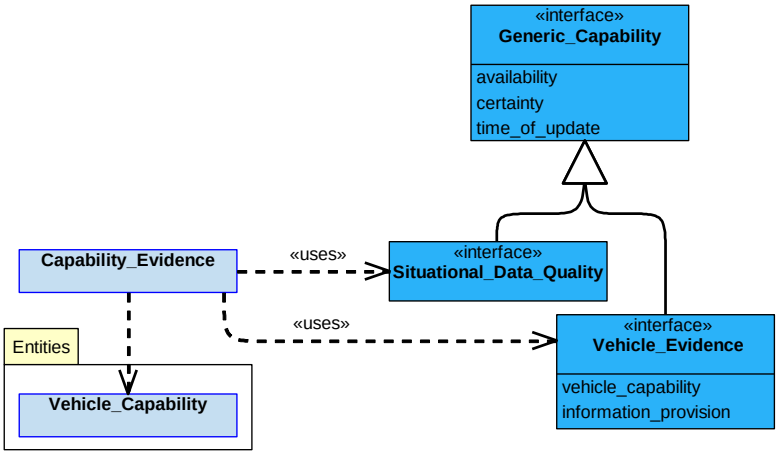

Figure 915: Capability_Evidence Service Definition

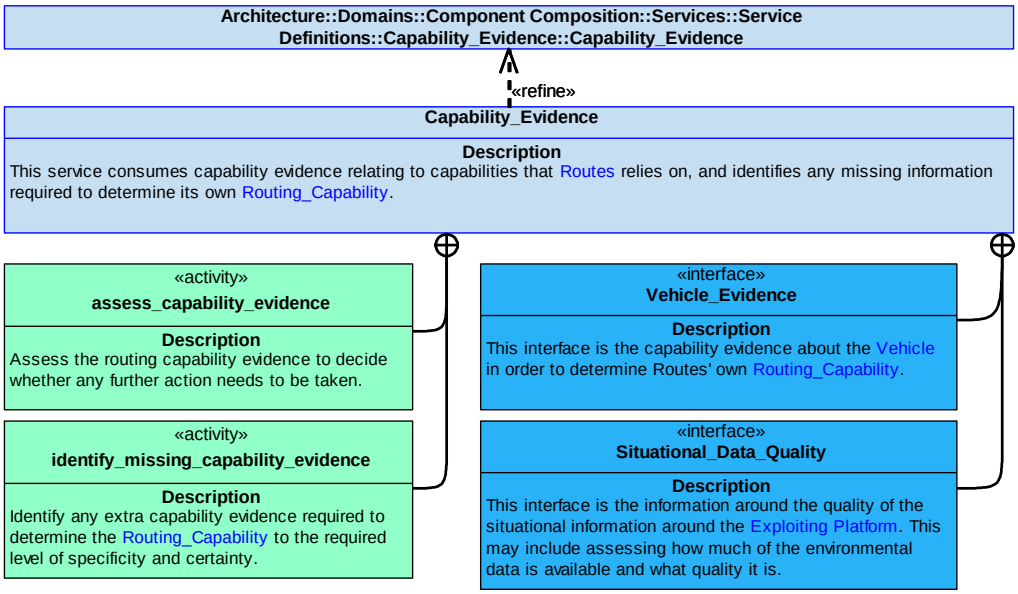

Figure 916: Capability_Evidence Service Policy

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 in order to determine Routes' own Routing_Capability.

Attributes

vehicle_capability	An indication of what the Exploiting Platform is capable of, e.g. whether it is capable of remaining within the altitude limits.
information_provision	An indication of the ability to supply vehicle capability information.

Situational_Data_Quality

This interface is the information around the quality of the situational information around the Exploiting Platform. This may include assessing how much of the environmental data is available and what quality it is.

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.

B.2.49.7.2 Service Dependencies

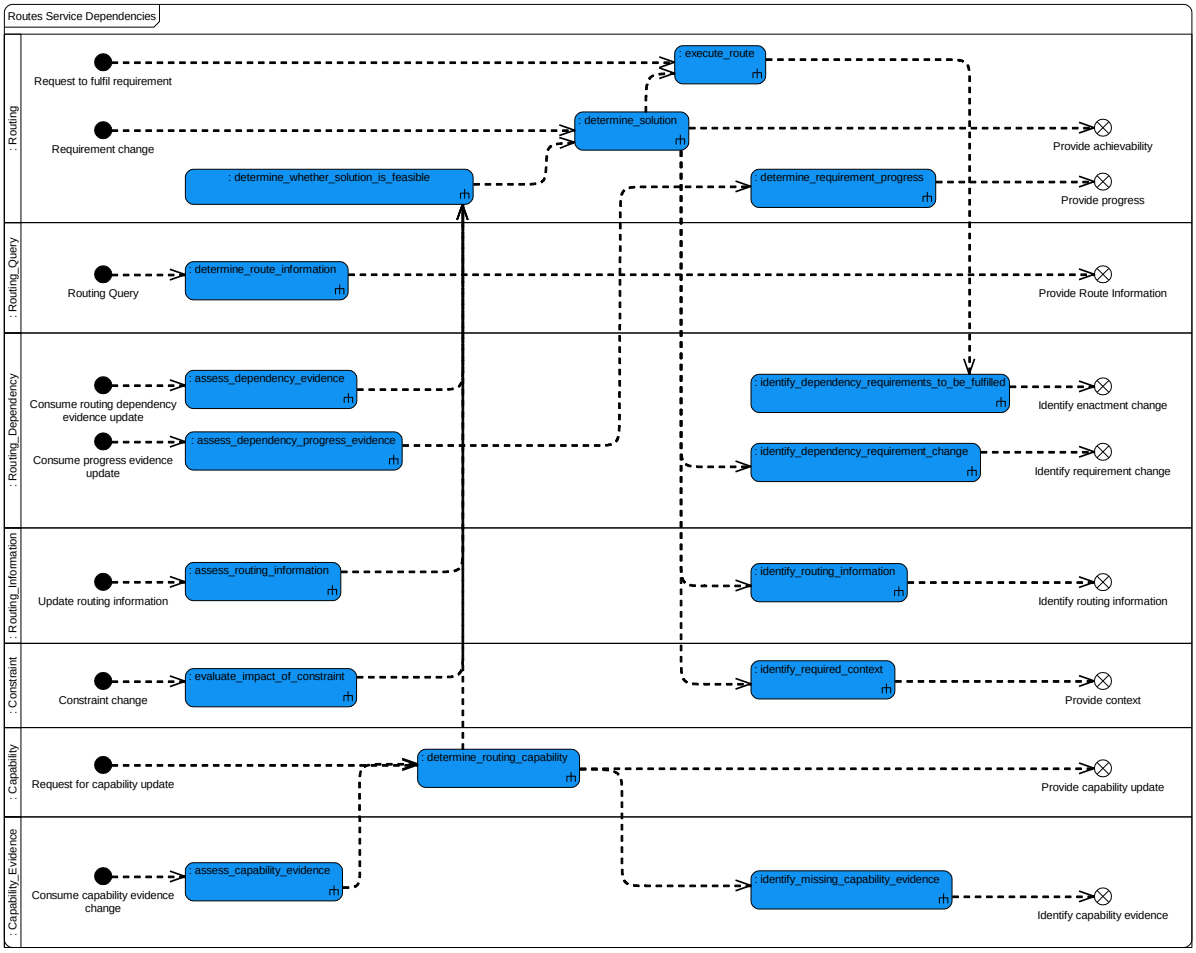

Figure 917: Routes Service Dependencies

B.2.50 Semantic Translation

B.2.50.1 Role

The role of Semantic Translation is to translate between data semantics of systems.

B.2.50.2 Overview

Control Architecture

Semantic Translation is a service component as defined in the Control Architecture policy.

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 hand-over in a five-way handover, whereas another System uses a STANAG 4586 based three-way handover with no commander role).

B.2.50.3 Service Summary

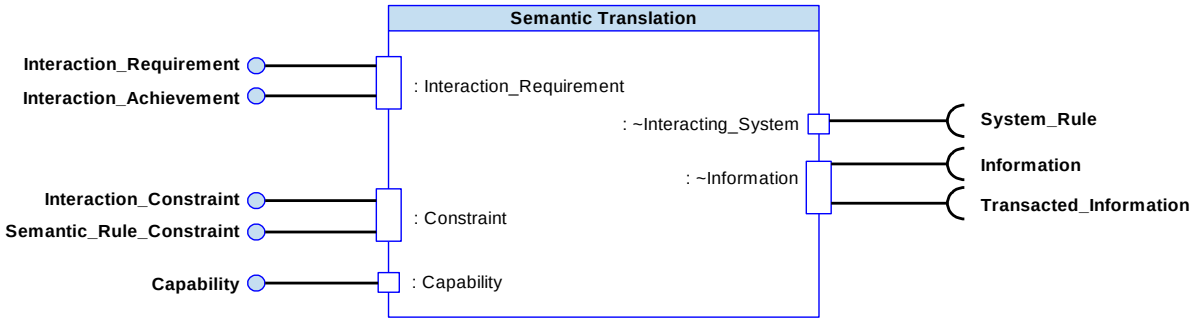

Figure 918: Semantic Translation Service Summary

B.2.50.4 Responsibilities

capture_interaction_requirements

- To capture provided requirements for Interactions 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_transaction

- To determine how to meet the given requirements for an Interaction between Systems.

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).

capture_interaction_constraints

- To capture provided constraints on Interactions and application of Semantic_Rules.

predict_capability_progression

- To predict the progression of Semantic Translation Capability over time and with use.

determine_if_interaction_remains_achievable

- To determine if an Interaction requirement remains achievable given current Capability and Constraints.

B.2.50.5 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.

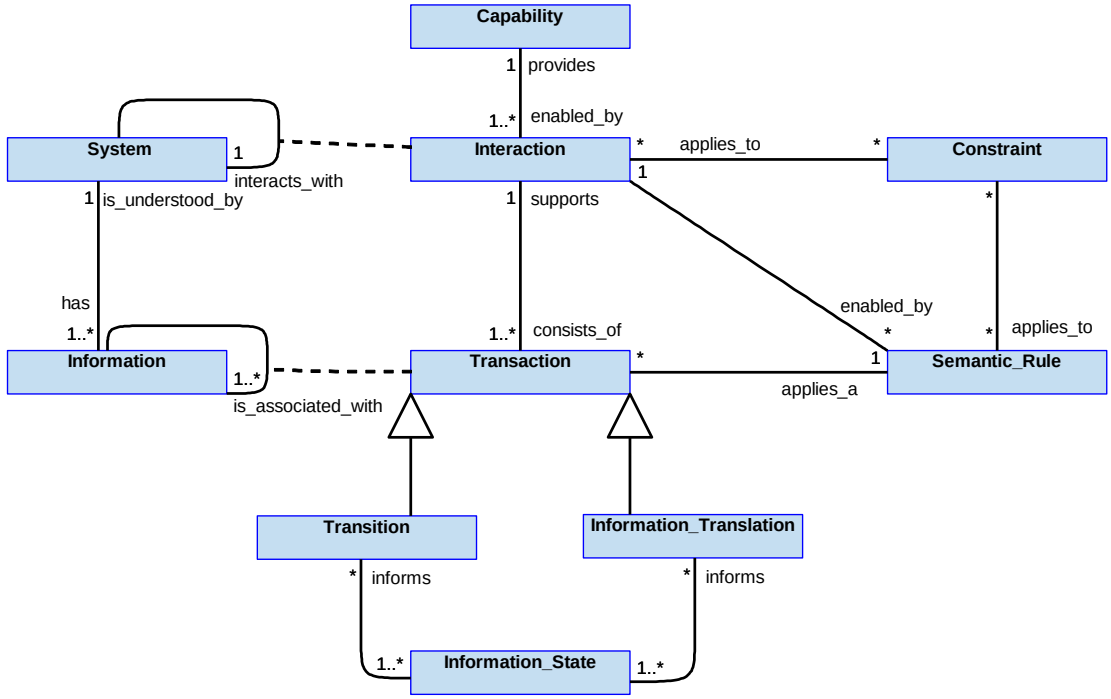

Figure 919: Semantic Translation Semantics

B.2.50.5.1 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 activity 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.

B.2.50.6 Design Rationale

B.2.50.6.1 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.

B.2.50.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Semantic Translation:

- Data Exchange - This component is highly involved in the exchange of information.
- Use of Communications - Whilst this component is not specifically part of the communications capability, it is 'communications aware'.
- Recording and Logging - The component will perform data logging.

Note that the Semantic Translation component's capability to perform translations is entirely dependent on the availability of appropriate Semantic_Rules, and does not evolve with time. This component is not concerned with the provided capability of any particular subject matter area hence does not consume capability evidence or predict capability progression in the way described in the Capability Assessment policy.

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.

B.2.50.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- Whilst this component may enact the semantic translation between many different systems, this safety analysis has considered the interface to a complex weapon, using a MIL-STD-1760 Ref. [15] interface.
- Failure of this component could cause safety critical electrical outputs (28V DC2 and release consent) or databus commands to be activated when not required. This may, for example, cause a weapons motor to fire or enable weapon arming when it is not required. These could result in catastrophic consequences.
- Whilst the Interlocks component may be used to prevent inappropriate activation of electrical discretes independently of this component, for databus commands any protection from Interlocks may be enacted through this component. Therefore, no credit is taken for any protection that the Interlocks component may be able to provide in this analysis.

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

B.2.50.6.4 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.

The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about security of data exchange.

B.2.50.7 Services

B.2.50.7.1 Service Definitions

B.2.50.7.1.1 Interaction_Requirement

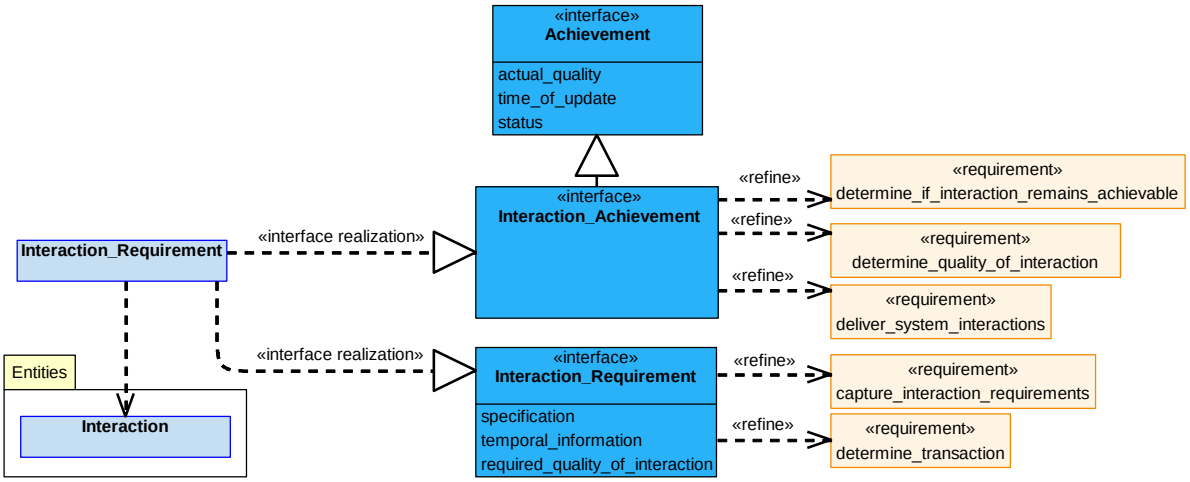

Figure 920: Interaction_Requirement Service Definition

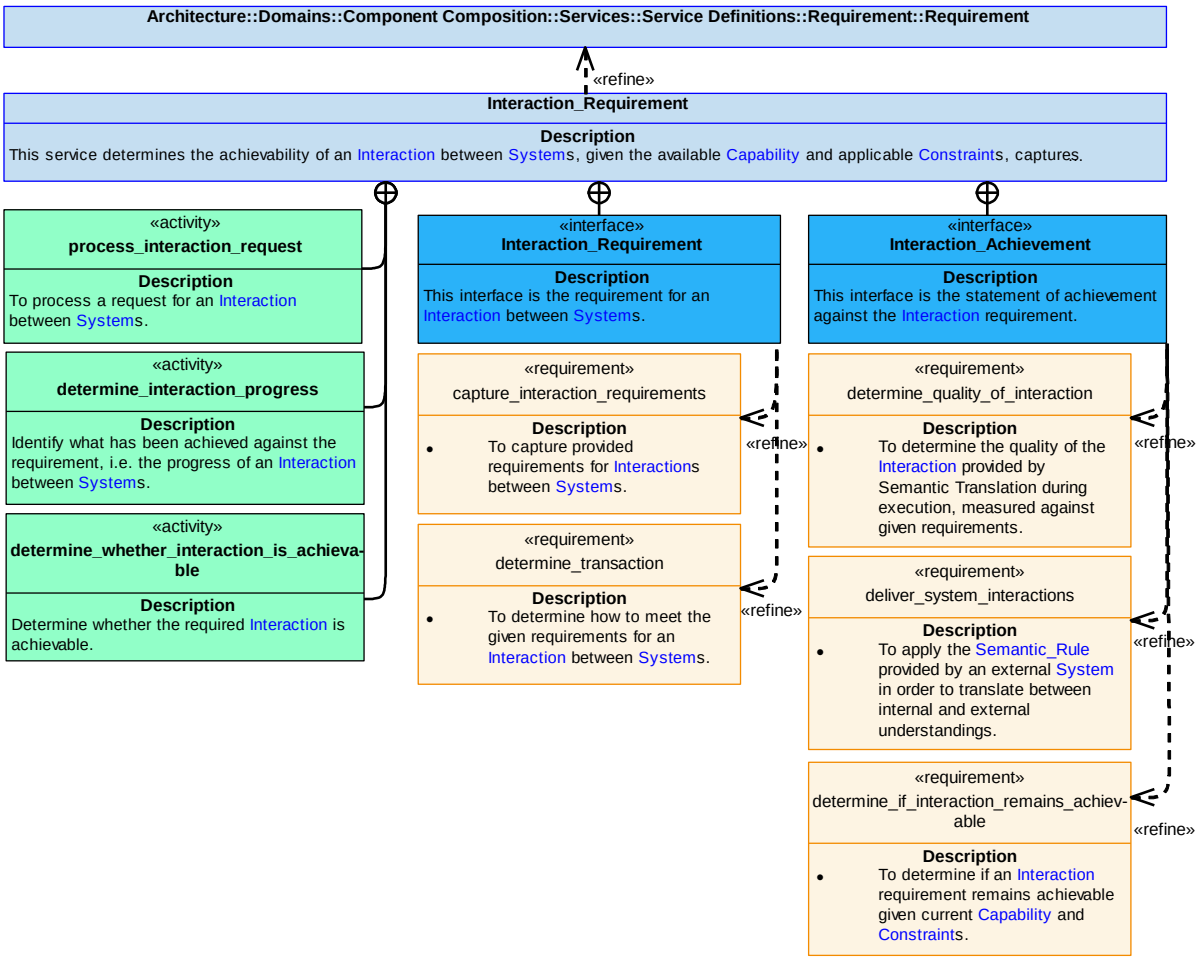

Figure 921: Interaction_Requirement Service Policy

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.

B.2.50.7.1.2 Interacting_System

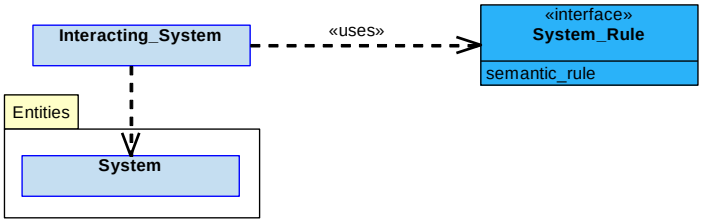

Figure 922: Interacting_System Service Definition

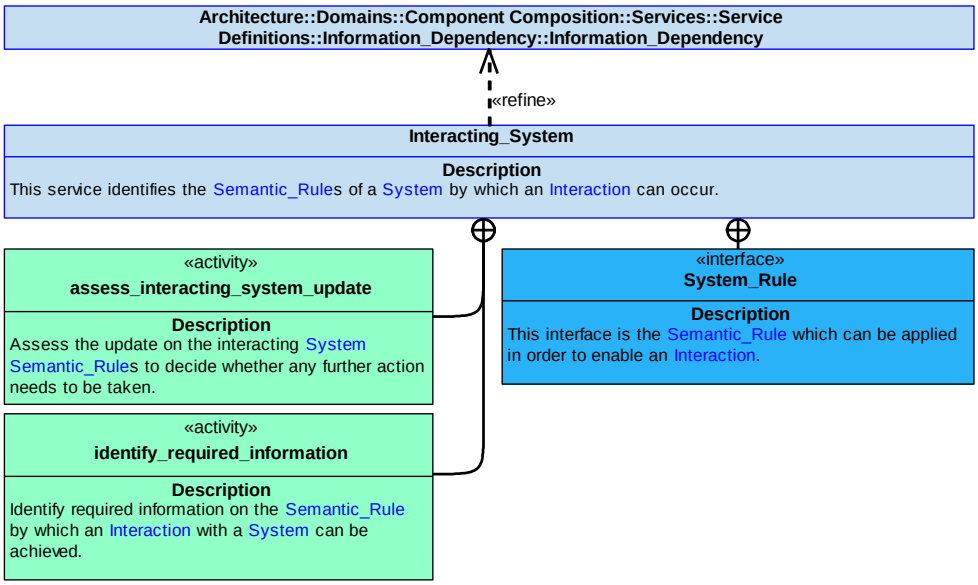

Figure 923: Interacting_System Service Policy

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.

B.2.50.7.1.3 Information

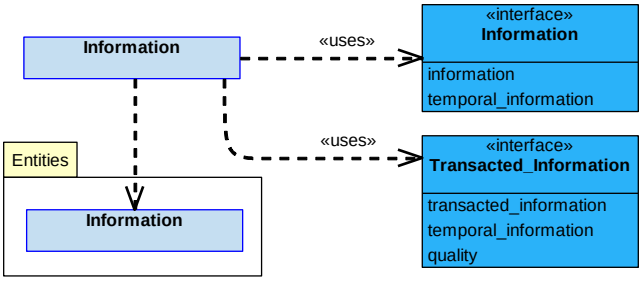

Figure 924: Information Service Definition

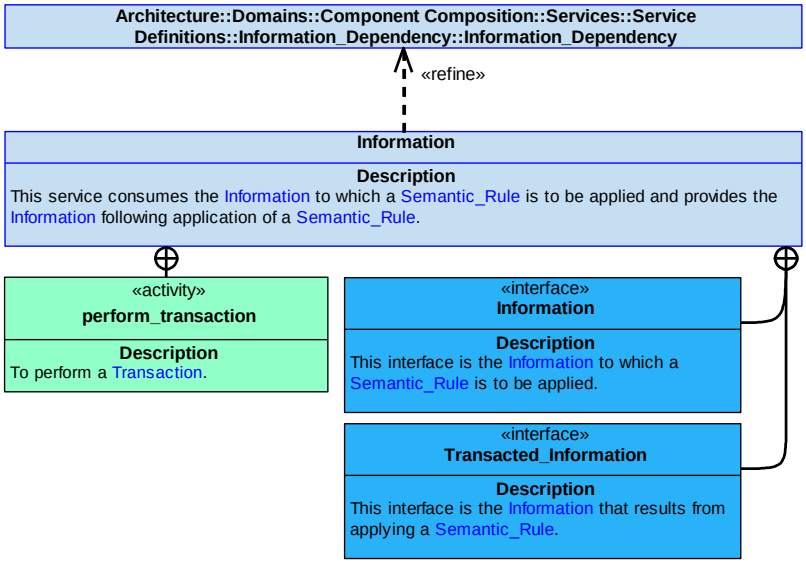

Figure 925: Information Service Policy

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.

B.2.50.7.1.4 Constraint

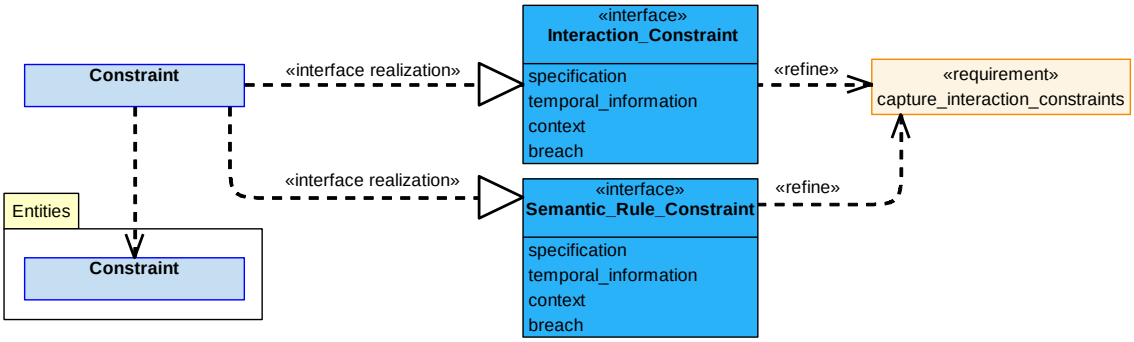

Figure 926: Constraint Service Definition

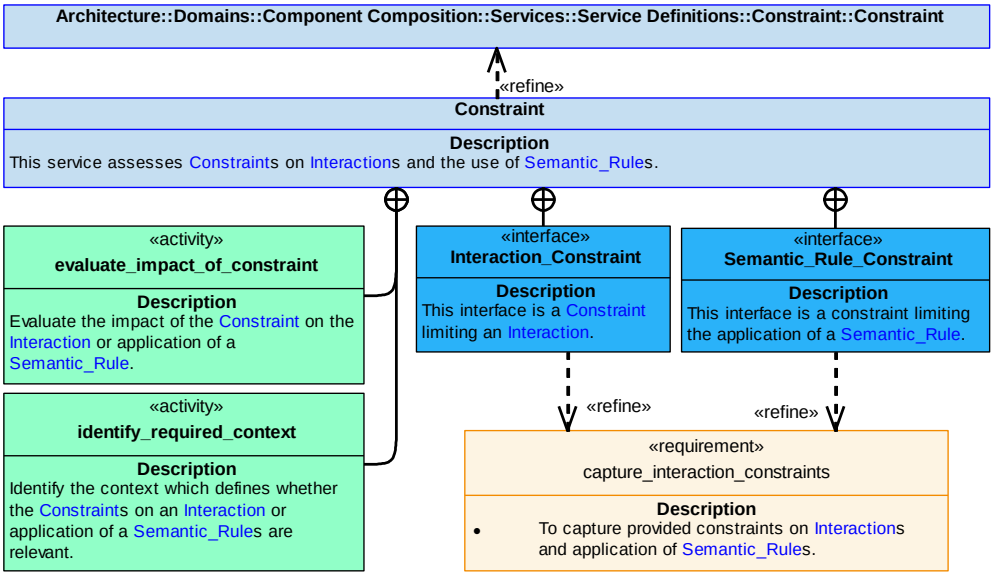

Figure 927: Constraint Service Policy

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.

B.2.50.7.1.5 Capability

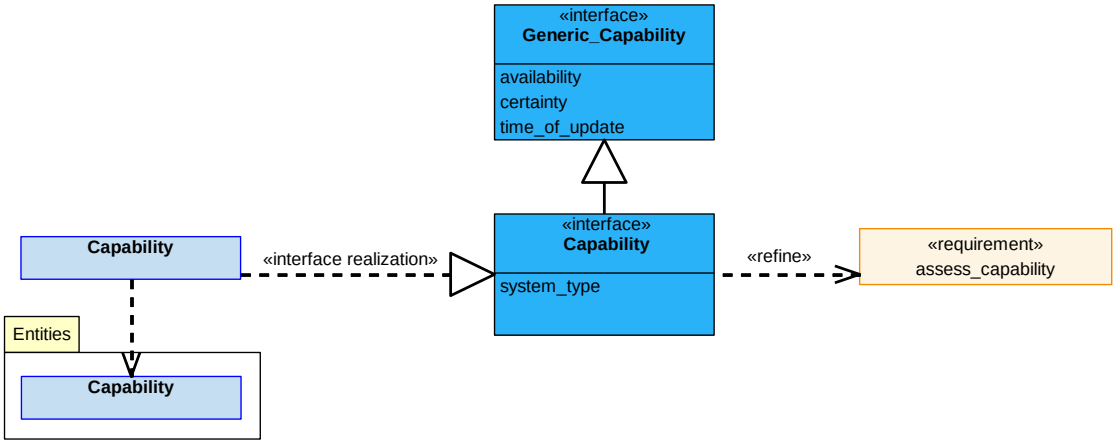

Figure 928: Capability Service Definition

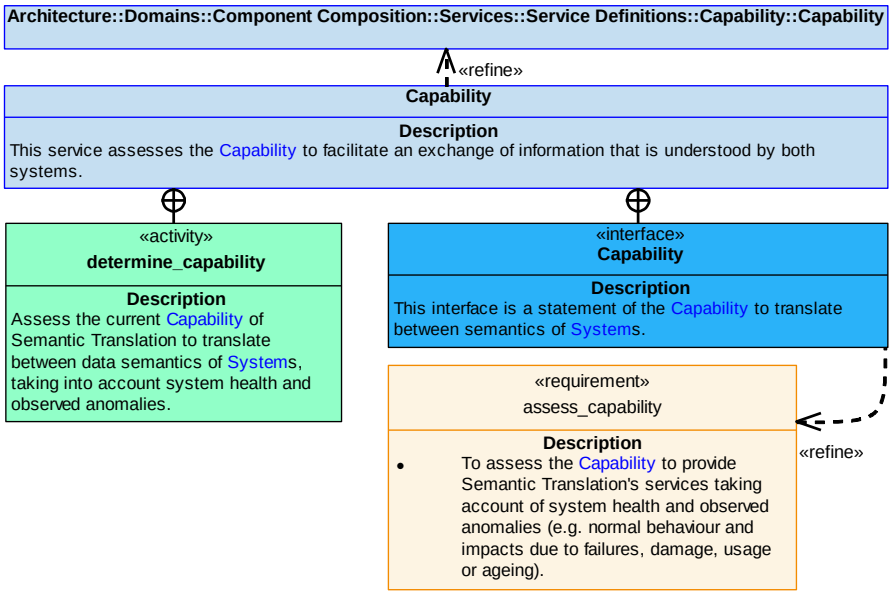

Figure 929: Capability Service Policy

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.

B.2.50.7.2 Service Dependencies

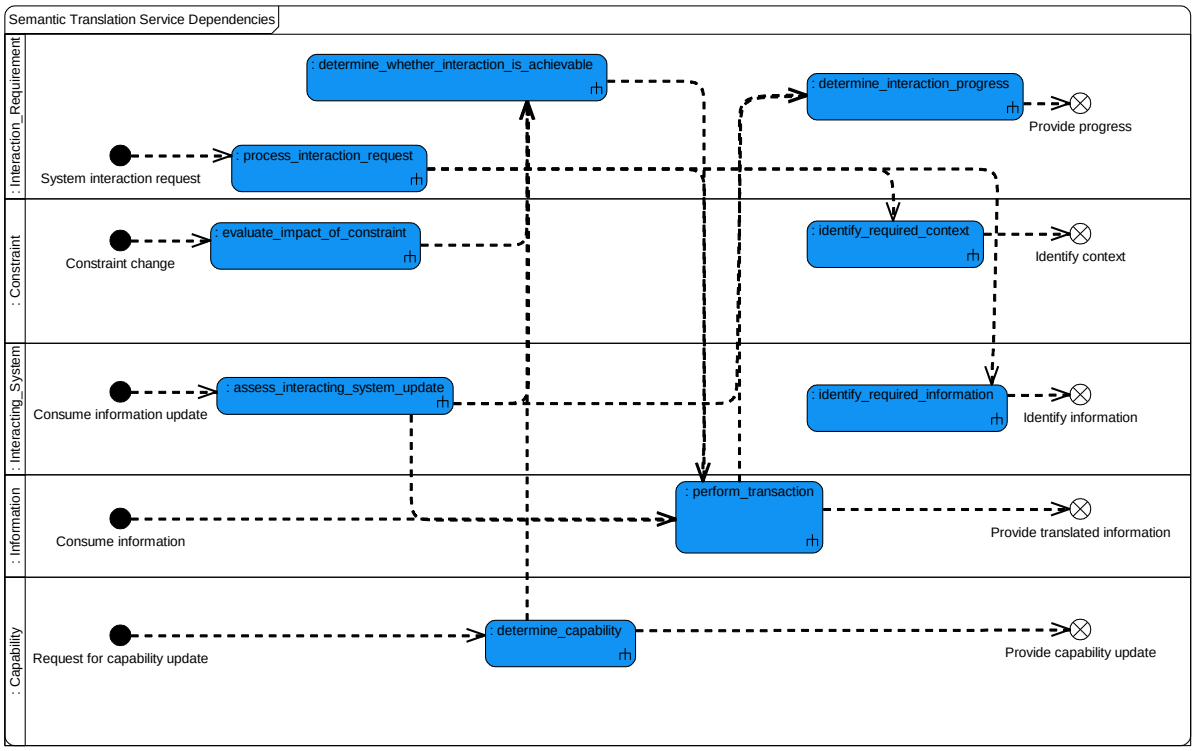

Figure 930: Semantic Translation Service Dependencies

B.2.51 Sensing

B.2.51.1 Role

The role of Sensing is to perform sensing actions by using resources.

B.2.51.2 Overview

Control Architecture

Sensing is an action component as defined in the Control Architecture policy.

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.

B.2.51.3 Service Summary

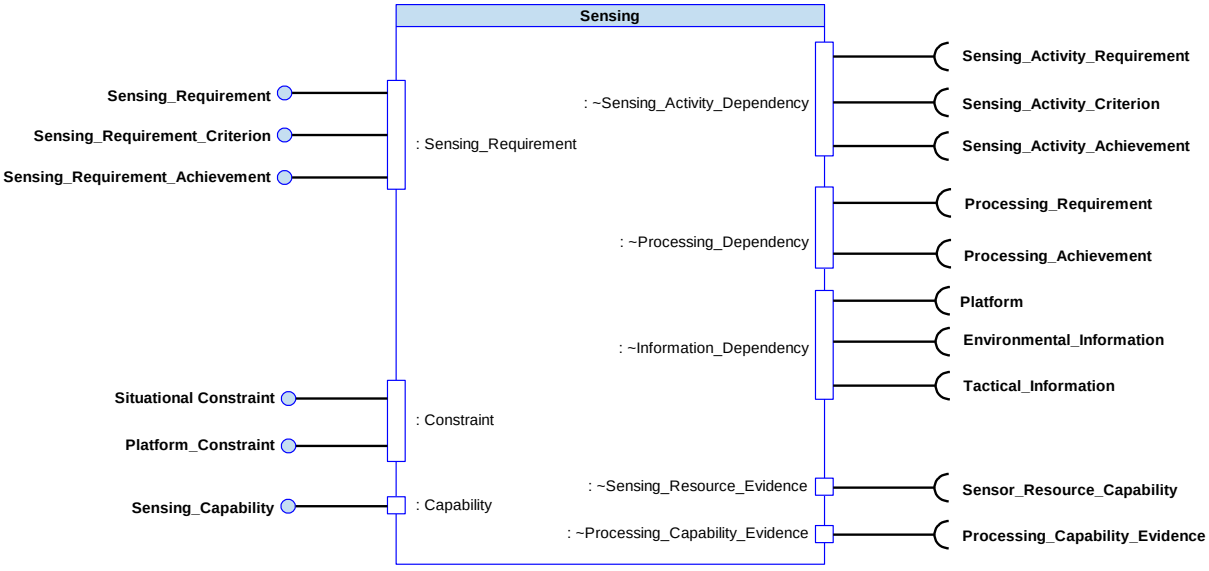

Figure 931: Sensing Service Summary

B.2.51.4 Responsibilities

capture_sensing_action_requirements

- To capture given Sensing_Requirements (e.g. target criteria, location and time).

capture_sensing_action_constraints

- To capture given sensing Constraints (e.g. spatial restrictions or EMCON restrictions on active sensing).

capture_sensing_action_measurement_criteria

- To capture given Measurement_Criterion/criteria for sensing output (e.g. recon image quality).

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.

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.

identify_progress_of_sensing_solution

- To identify the progress of a Sensing_Solution against the Sensing_Requirement.

assess_sensing_action_capability

- To determine the available Sensing_Capability provided by installed sensing resources, taking into account anomalies and sensor health.

predict_sensing_action_capability_progression

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

identify_whether_requirement_remains_achievable

- To identify if a Sensing_Solution in progress remains achievable given current resources.

identify_required_capability_information

- To identify missing resource information that is required for assessing Sensing_Capability.

B.2.51.5 Subject Matter Semantics

The subject matter of Sensing is the Sensing_Resources that can be used to measure properties of the environment.

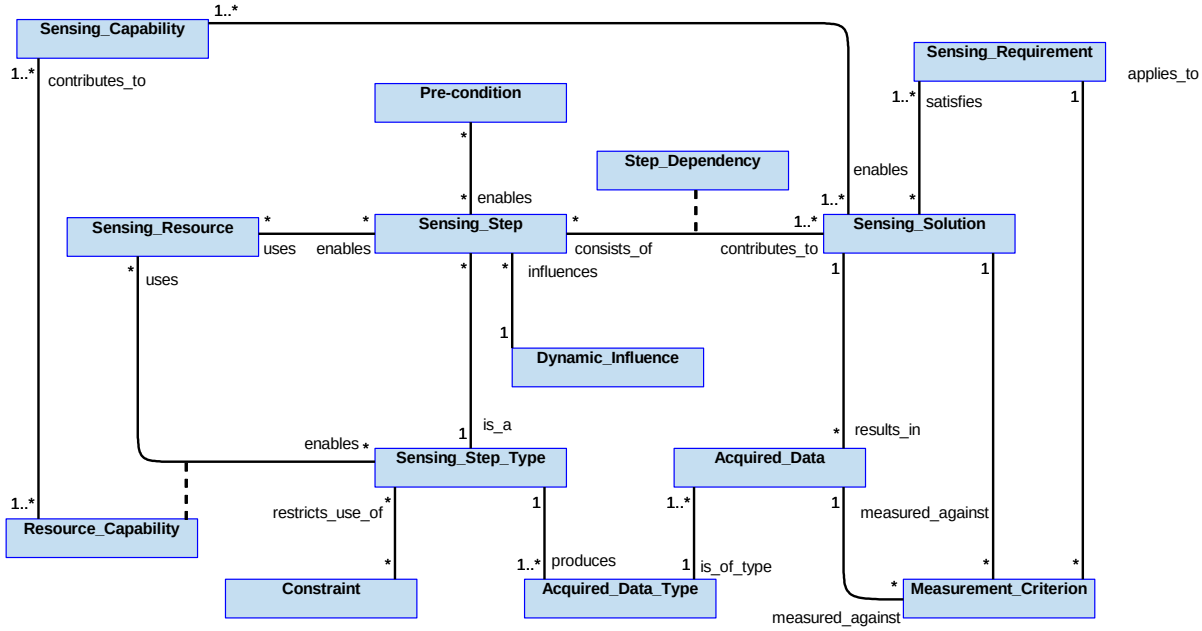

Figure 932: Sensing Semantics

B.2.51.5.1 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.

B.2.51.6 Design Rationale

B.2.51.6.1 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).

B.2.51.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy.
- 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 policy.
- 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.

B.2.51.6.3 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. In accordance with UK MoD direction (see Safety Analysis policy) this drives a DAL B indicative IDAL.

B.2.51.6.4 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.

B.2.51.7 Services

B.2.51.7.1 Service Definitions

B.2.51.7.1.1 Sensing_Requirement

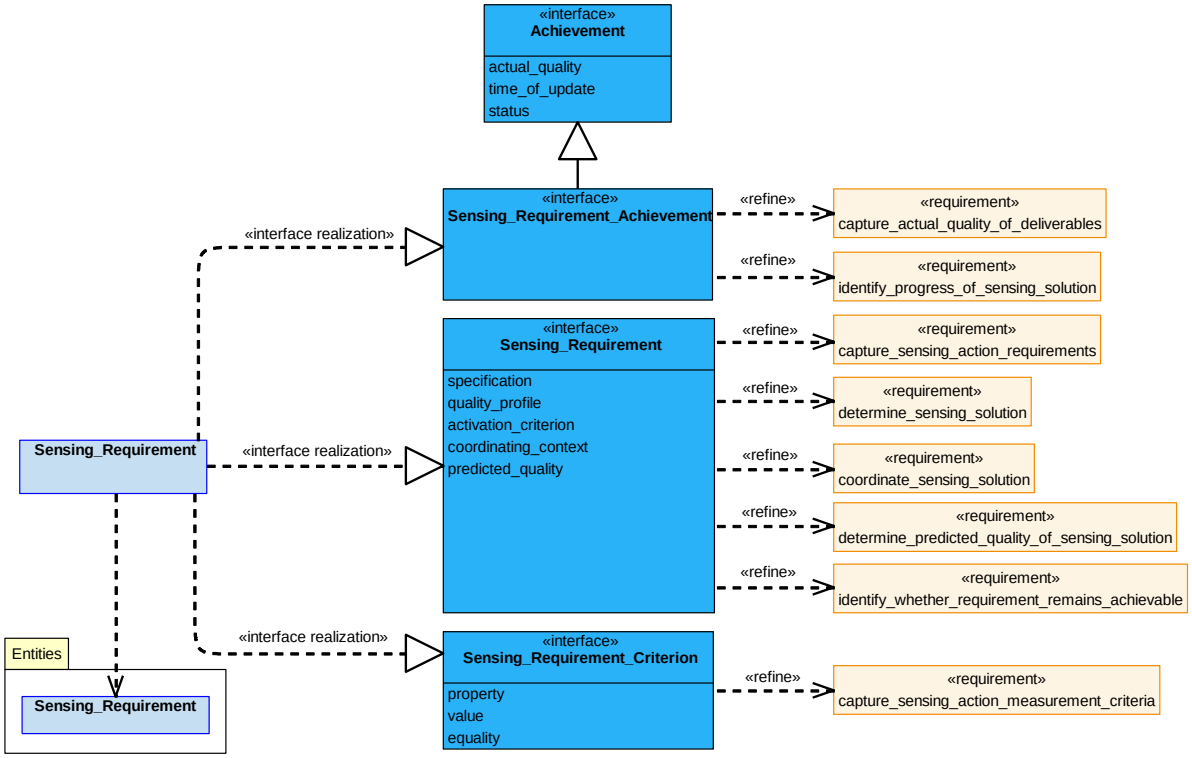

Figure 933: Sensing_Requirement Service Definition

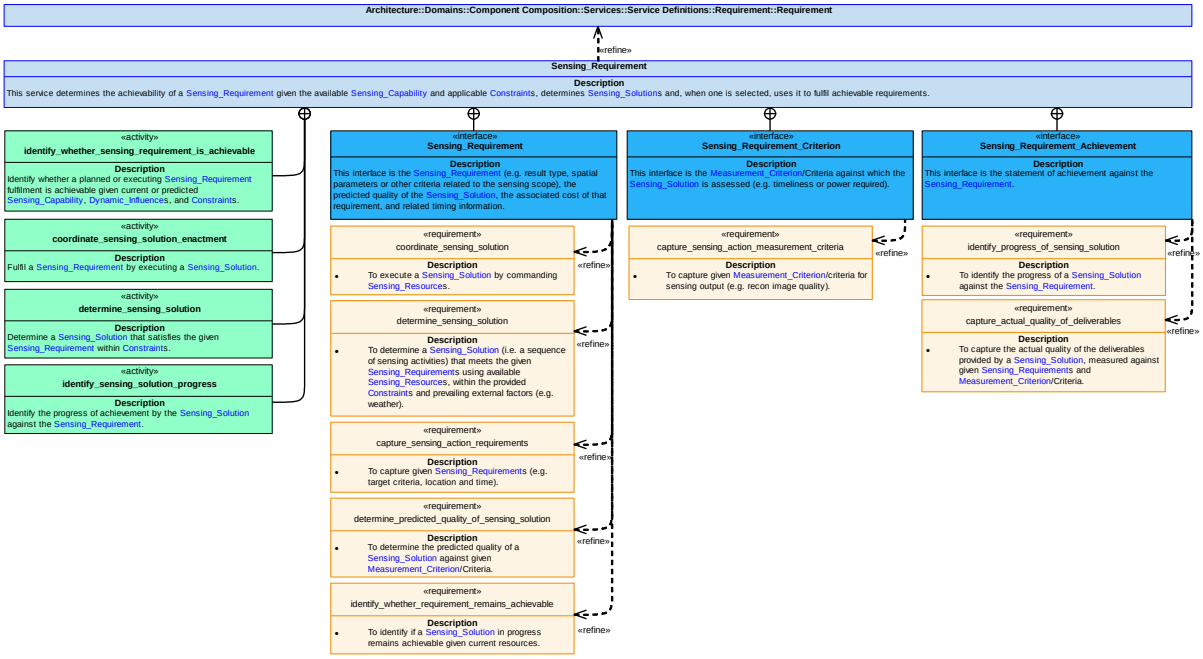

Figure 934: Sensing_Requirement Service Policy

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.

B.2.51.7.1.2 Sensing_Activity_Dependency

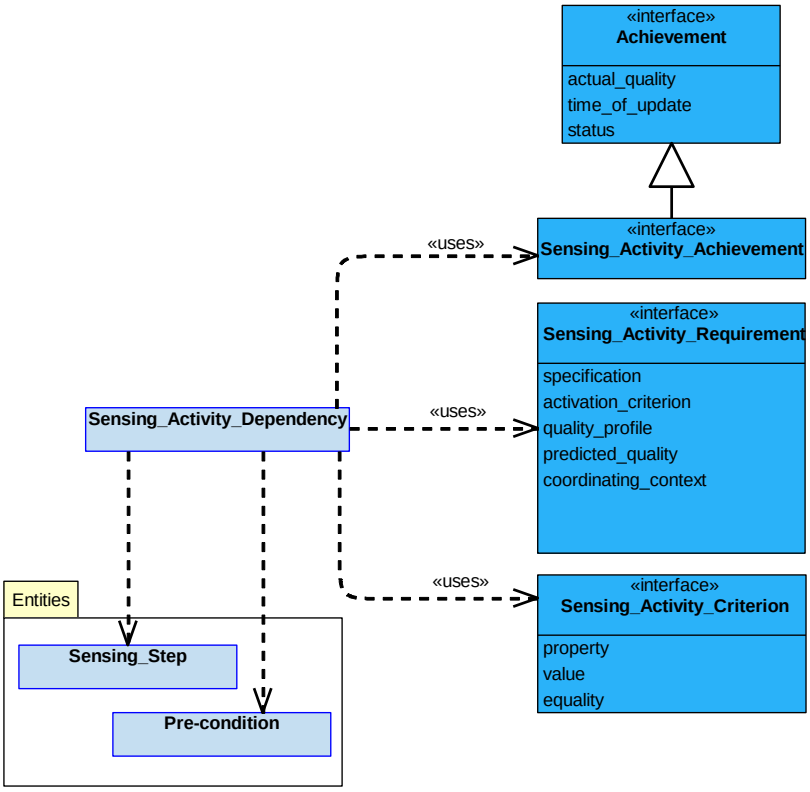

Figure 935: Sensing_Activity_Dependency Service Definition

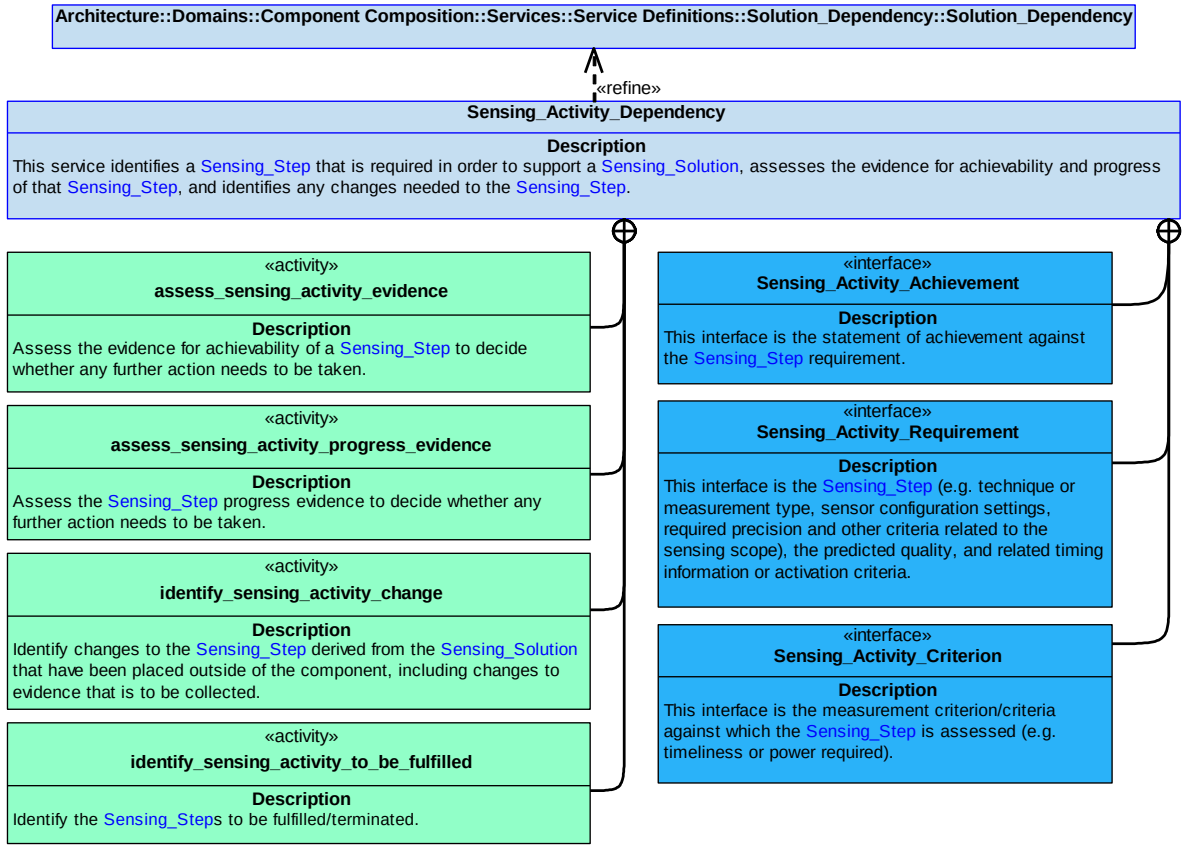

Figure 936: Sensing_Activity_Dependency Service Policy

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 activity 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.

B.2.51.7.1.3 Processing_Dependency

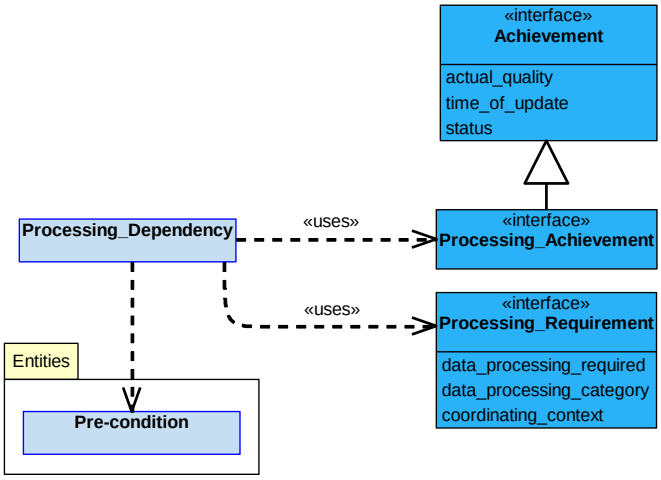

Figure 937: Processing_Dependency Service Definition

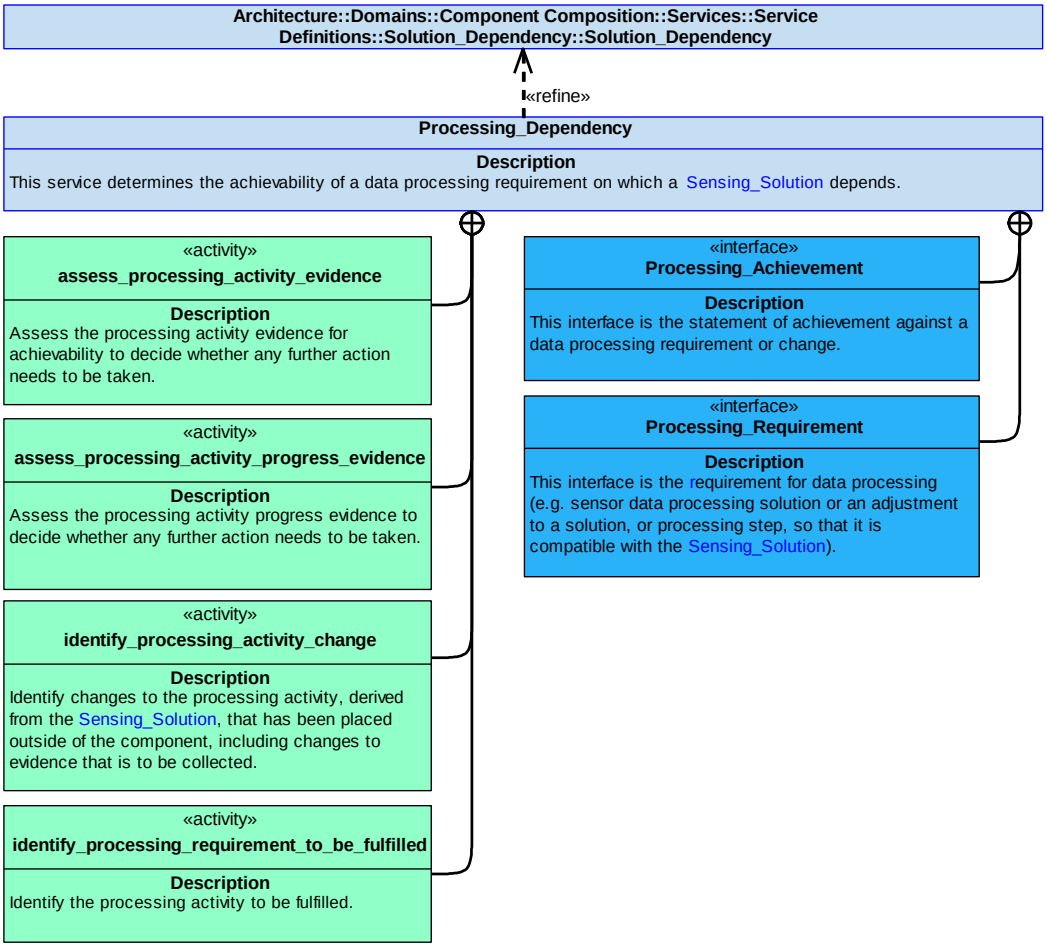

Figure 938: Processing_Dependency Service Policy

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.

B.2.51.7.1.4 Information_Dependency

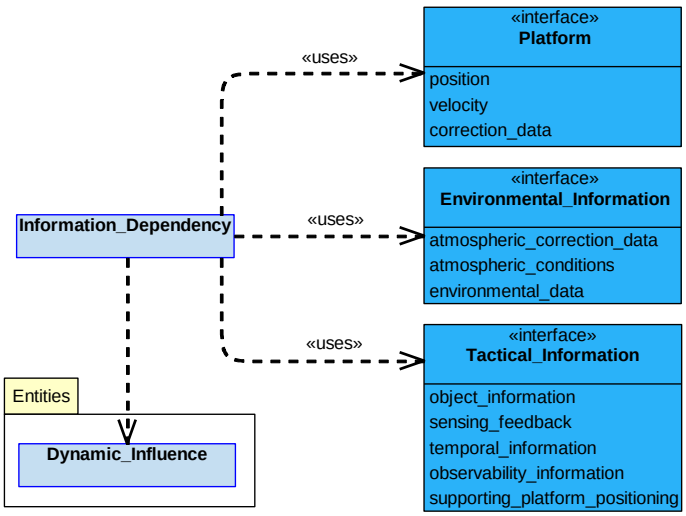

Figure 939: Information_Dependency Service Definition

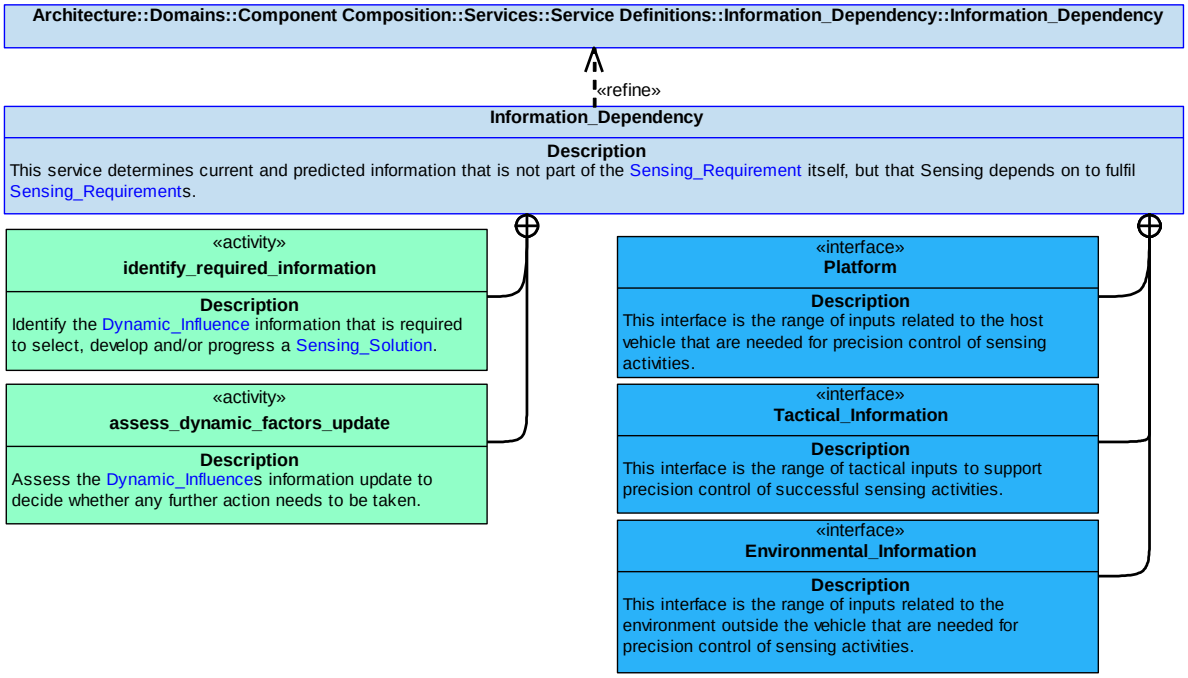

Figure 940: Information_Dependency Service Policy

Information_Dependency

This service determines current and predicted information that is not part of the Sensing_Requirement itself, but that Sensing depends on to fulfil Sensing_Requirements.

Interfaces

Environmental_Information

This interface is the range of inputs related to the environment outside the vehicle that are needed for precision control of sensing activities.

Attributes

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.

Platform

This interface is the range of inputs related to the host vehicle that are needed for precision control of sensing activities.

Attributes

position	Current and predicted location and orientation of the vehicle(s) that will be used to perform Sensing_Steps.
velocity	Velocity of the vehicle(s) that will be used to perform Sensing_Steps.
correction_data	Spatial correction data for sensors affected by elastic deformation, caused by dynamic forces acting on parts of the vehicle's frame. For example, sensors mounted on wing tips or other vehicle extremities.

Tactical_Information

This interface is the range of tactical inputs to support precision control of successful 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 synchronization of sensing techniques.
observability_information	Dynamic and interactive low observability factors affecting the use of active sensing techniques.
supporting_platform_positioning	Positioning and kinematics of vehicles involved in multi-vehicle sensing techniques.

Activities

assess_dynamic_factors_update

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

identify_required_information

Identify the Dynamic_Influence information that is required to select, develop and/or progress a Sensing_Solution.

B.2.51.7.1.5 Constraint

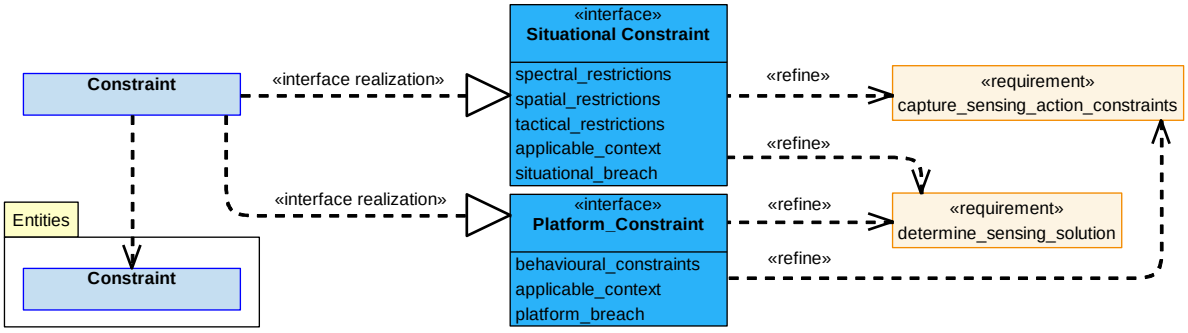

Figure 941: Constraint Service Definition

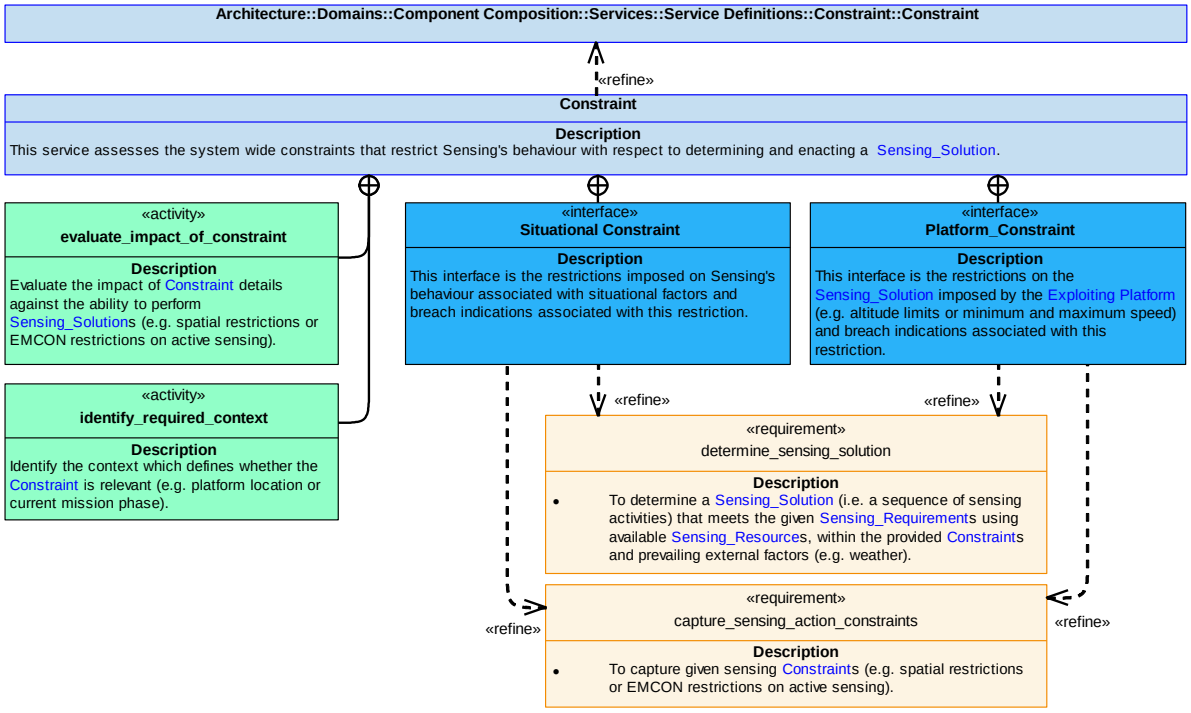

Figure 942: Constraint Service Policy

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).

B.2.51.7.1.6 Capability

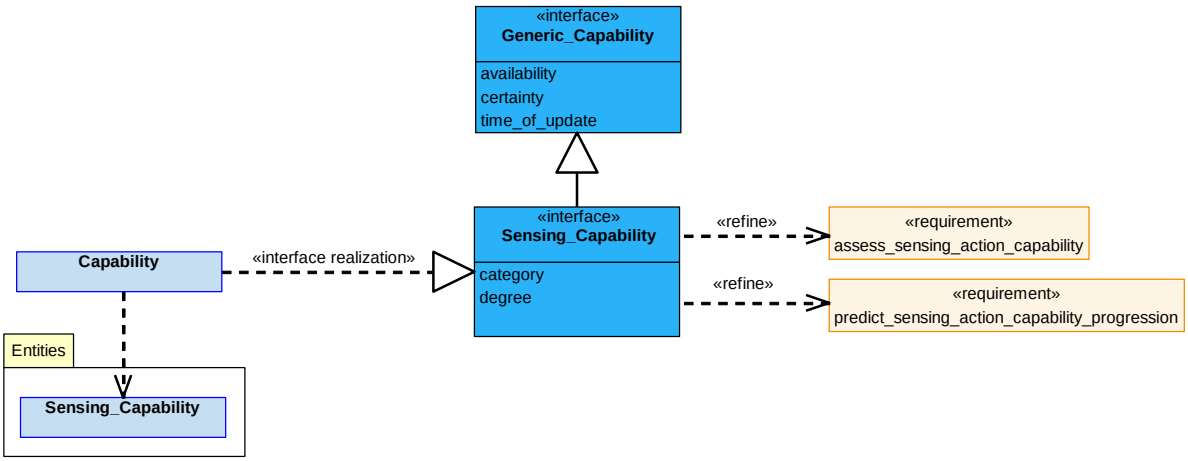

Figure 943: Capability Service Definition

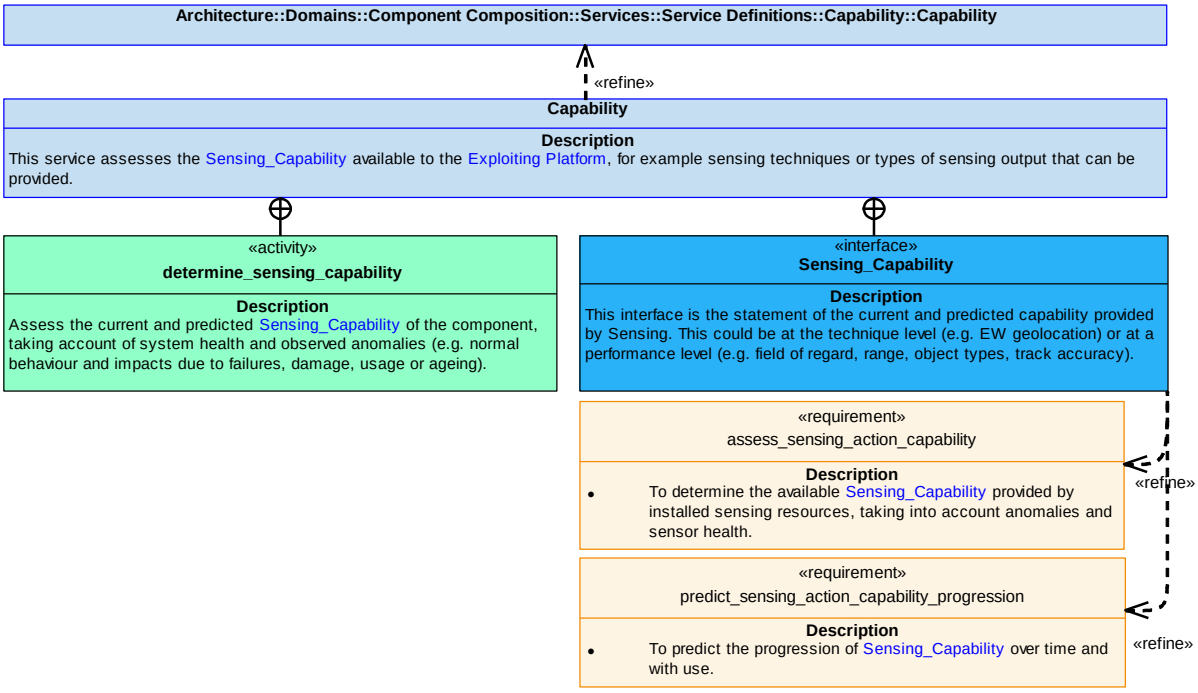

Figure 944: Capability Service Policy

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, 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).

B.2.51.7.1.7 Sensing_Resource_Evidence

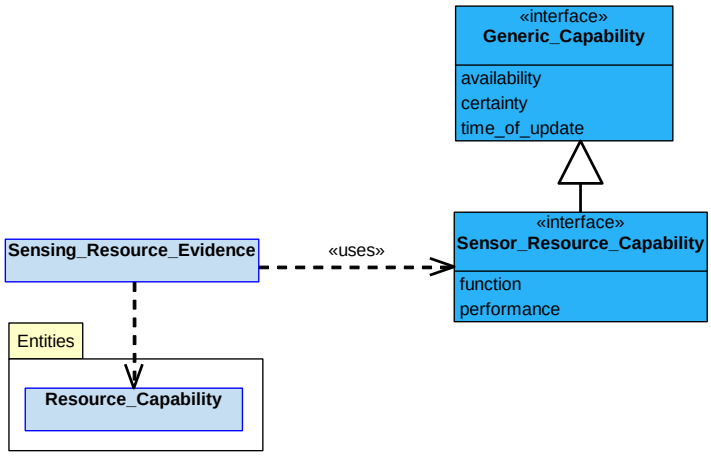

Figure 945: Sensing_Resource_Evidence Service Definition

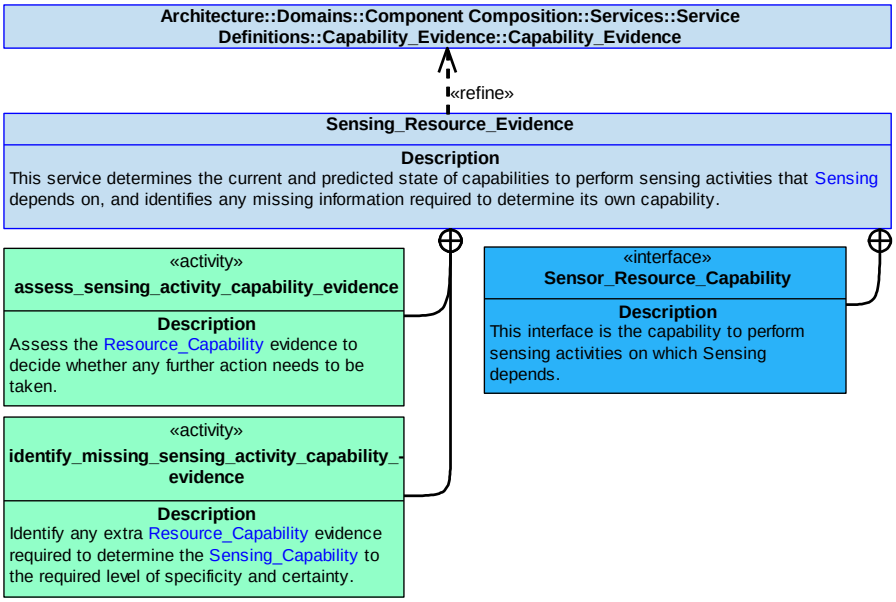

Figure 946: Sensing_Resource_Evidence Service Policy

Sensing_Resource_Evidence

This service determines the current and predicted state of capabilities to perform sensing activities that Sensing depends on, and identifies any missing information required to determine its own capability.

Interface

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, and the function would be operated.
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, minimum detectable signal.

Activities

assess_sensing_activity_capability_evidence

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

identify_missing_sensing_activity_capability_evidence

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

B.2.51.7.1.8 Processing_Capability_Evidence

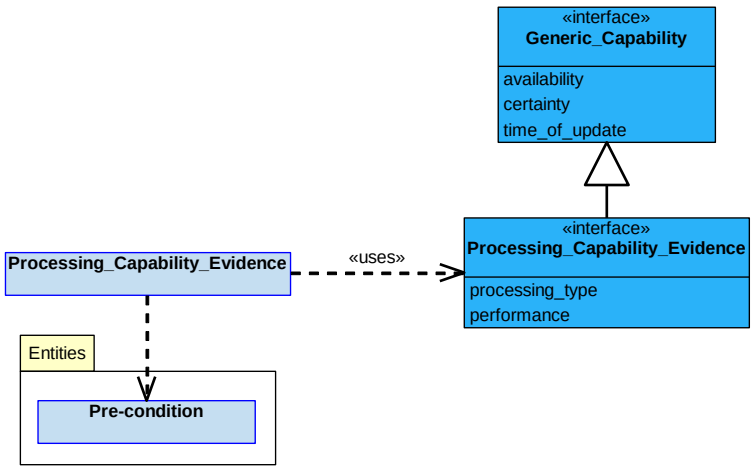

Figure 947: Processing_Capability_Evidence Service Definition

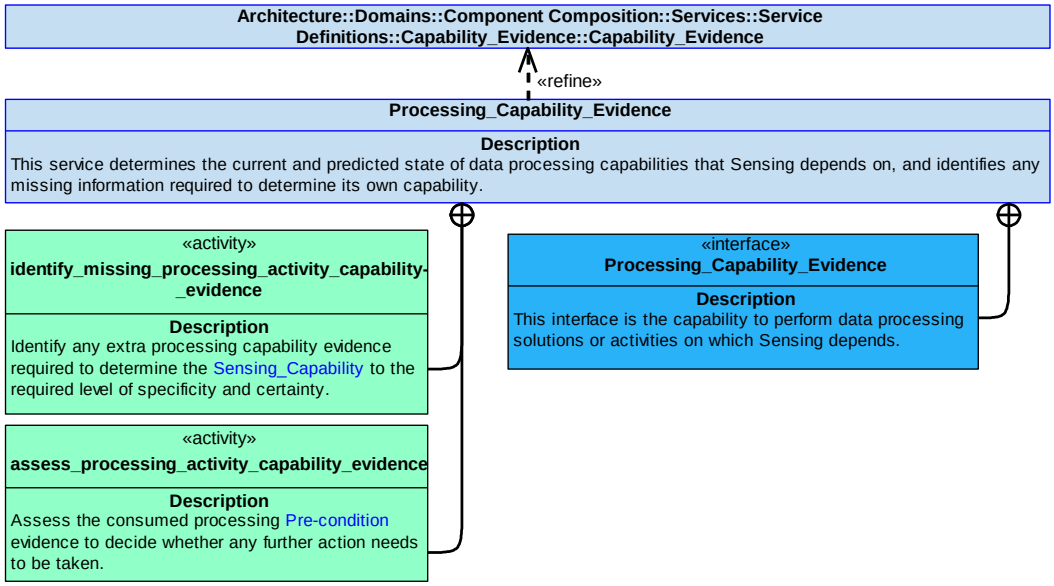

Figure 948: Processing_Capability_Evidence Service Policy

Processing_Capability_Evidence

This service determines the current and predicted state of data processing capabilities that Sensing depends on, and identifies any missing information required to determine its own capability.

Interface

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_processing_activity_capability_evidence

Assess the consumed processing Pre-condition evidence to decide whether any further action needs to be taken.

identify_missing_processing_activity_capability_evidence

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

B.2.51.7.2 Service Dependencies

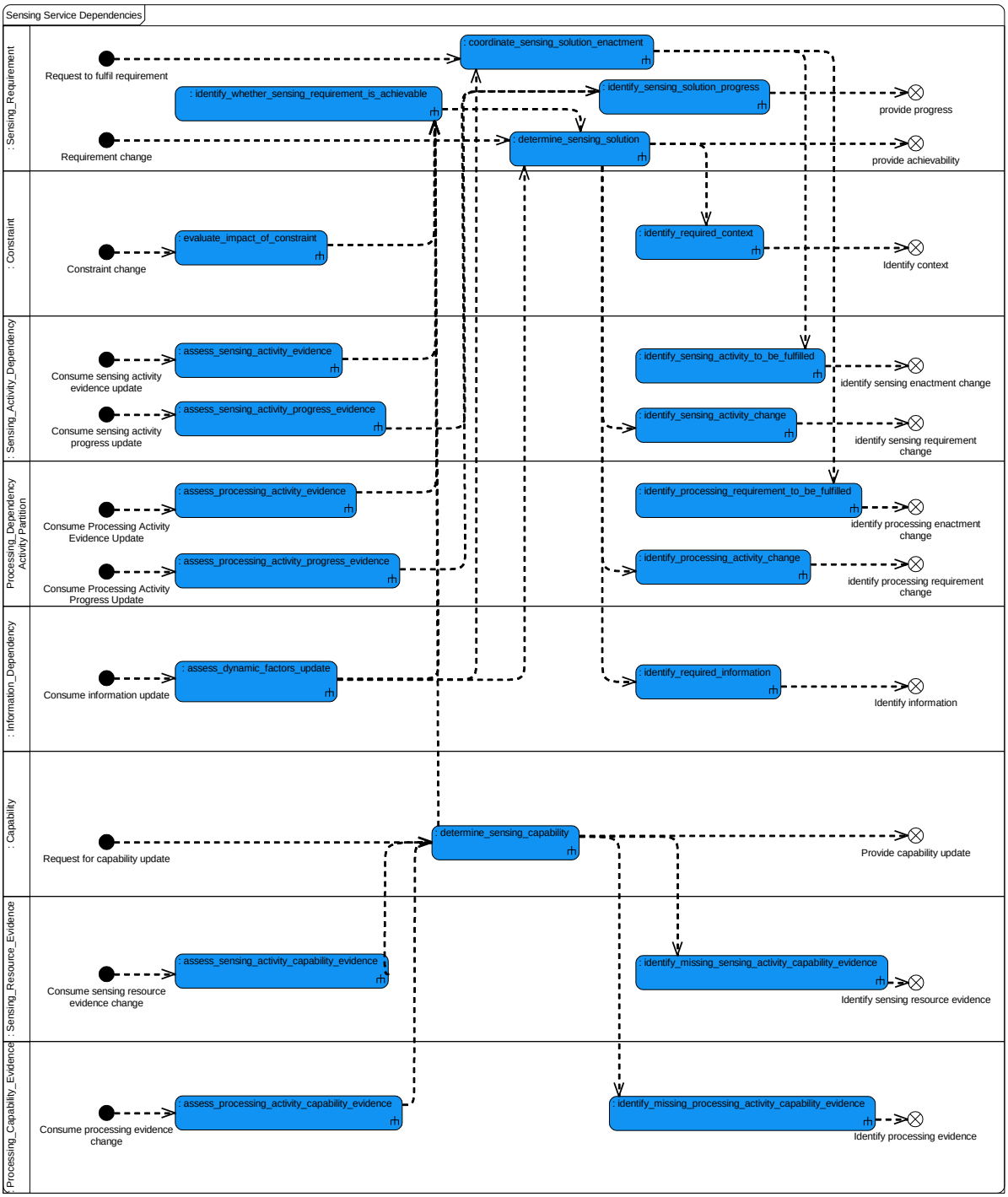

Figure 949: Sensing Service Dependencies

B.2.52 Sensor Data Interpretation

B.2.52.1 Role

The role of Sensor Data Interpretation is to coordinate the extraction of meaning from data produced by sensors.

B.2.52.2 Overview

Control Architecture

Sensor Data Interpretation is an action component as defined in the Control Architecture policy.

Standard Pattern of Use

When a Requirement is received, Sensor Data Interpretation will determine a Data_Interpretation_Solution consisting of one Data_Processing_Activity or several, each with associated Pre-conditions (e.g. a pre-condition might be the timely availability of Sensor_Data on which to perform a Data_Processing_Activity). The Data_Interpretation_Solution is then enacted, which will involve coordinated control of Interpretation_Resources. The Data_Interpretation_Solution will produce Interpreted_Data from the Interpretation_Resources.

Examples of Use

Sensor Data Interpretation is required where coordinated planning and execution of Data_Processing_Activity may be necessary. For example:

- To coordinate development of Interpreted_Data that detects, recognizes, or identifies objects from Sensor_Data for the purpose of reconnaissance.
- To coordinate development of Interpreted_Data that locates objects from Sensor_Data in order that they can be targeted with ordnance.

B.2.52.3 Service Summary

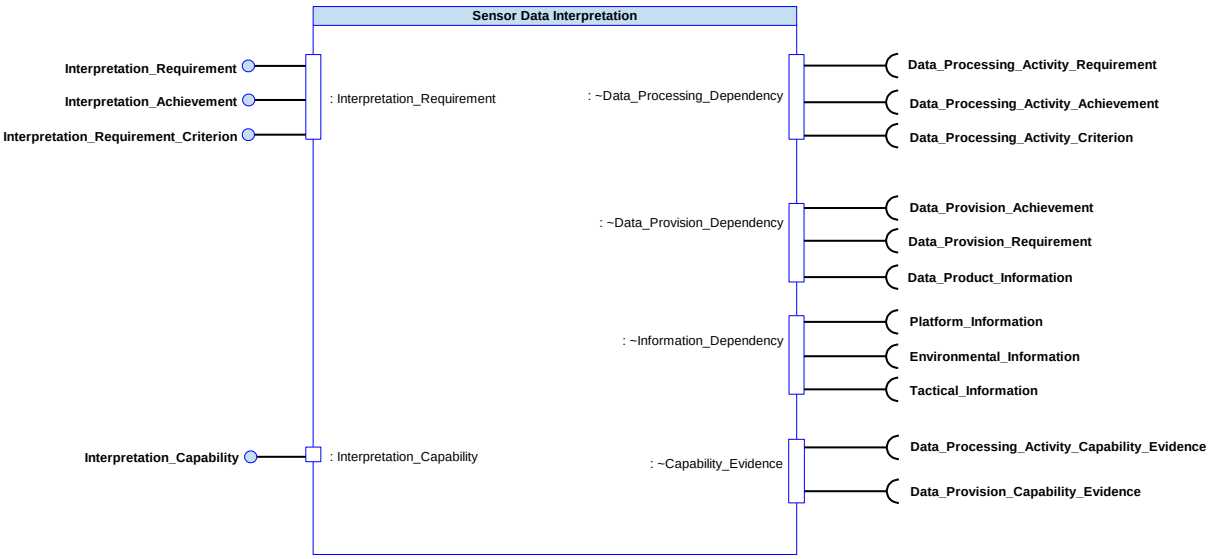

Figure 950: Sensor Data Interpretation Service Summary

B.2.52.4 Responsibilities

assess_capability

- To assess the Activity_Capability that can be applied to sensor data, taking account of observed anomalies (e.g. erroneous outputs of Data_Processing_Activity).

capture_measurement_criteria

- To capture the method for measuring a Data_Interpretation_Solution (e.g. by comparison with a required confidence in the result).

capture_requirements

- To capture provided Requirements for Interpreted_Data (e.g. determine the position of objects of a particular type within a region).

coordinate_solution

- To coordinate the execution of a Data_Interpretation_Solution.

determine_quality_of_deliverables

- To determine the quality of the Interpreted_Data, measured against given Requirements and Measurement_Criterion.

determine_predicted_quality_of_solution

- To determine the quality of a proposed Data_Interpretation_Solution against given Measurement_Criterion.

determine_solution

- To determine a coordinated and combined sequence of Data_Processing_Activity to use as the Data_Interpretation_Solution to an interpretation Requirement.

identify_preconditions

- To identify Pre-conditions required to support the Data_Interpretation_Solution or an Data_Processing_Activity in the Data_Interpretation_Solution.

identify_progress_of_solution

- To identify the progress of a Data_Interpretation_Solution against the Requirements.

identify_if_requirement_remains_achievable

- To identify if a Requirement remains achievable given current Interpretation_Resources.

predict_capability_progression

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

identify_missing_information

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

B.2.52.5 Subject Matter Semantics

The subject matter of Sensor Data Interpretation is the resources that can be used to interpret data from sensors.

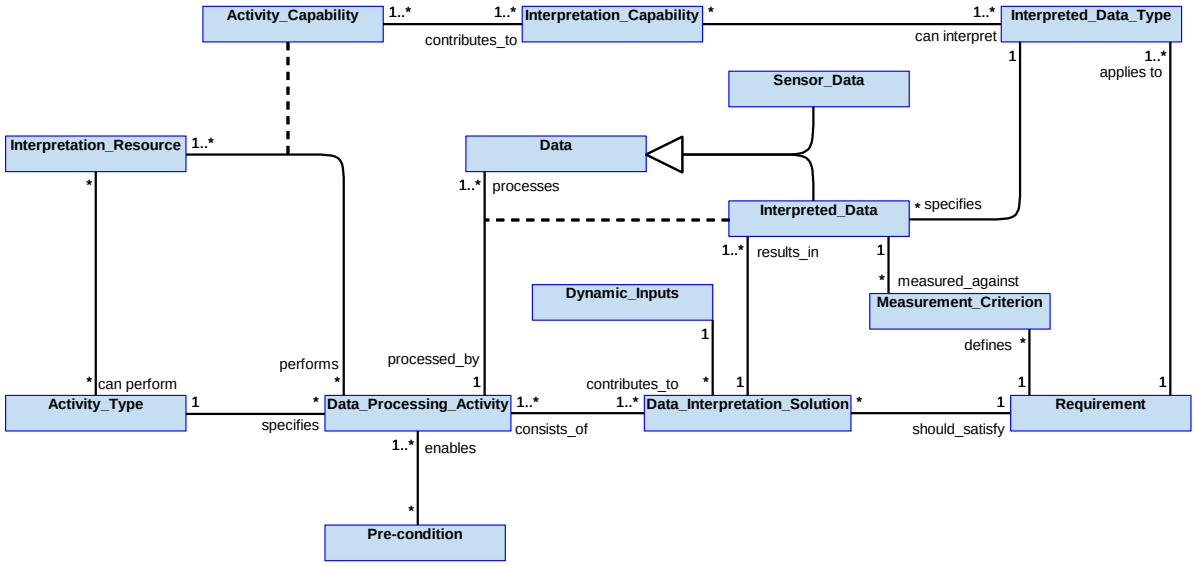

Figure 951: Sensor Data Interpretation Subject Matter Diagram

B.2.52.5.1 Entities

Activity_Capability

The range of Data_Processing_Activity that can be performed by an Interpretation_Resource on different kinds of Data.

Activity_Type

A type of activity that can be performed.

Data

Any data that can be subject to interpretation, this can include data which has already been interpreted and is subject to further interpretation.

Data_Interpretation_Solution

A planned combination of Data_Processing_Activity that fulfils an interpretation Requirement (e.g. a number of different Interpretation_Resources applied to Sensor_Data in a particular combination so as to derive and enhance actionable information, for example locating of enemy tanks in a region from ground image data).

Data_Processing_Activity

An algorithmic process provided by an Interpretation_Resource upon which Sensor Data Interpretation depends.

Interpretation_Capability

The ability of this component to provide sensor data interpretation actions.

Interpretation_Resource

A resource capable of providing a transformative process on some Sensor_Data. This represents a combination of processing power with an algorithm that is able to extract 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

The product of executing a Data_Interpretation_Solution on Sensor_Data (e.g. a change of state, confidence level, filtering or enhancement of attributes, detection or identification of objects).

Interpreted_Data_Type

The range of possible result data types (e.g. the required Interpreted_Data could be object detections, confidence levels, enhanced attributes, human readable recon images).

Measurement_Criterion

A criterion for measuring the quality of required Interpreted_Data (e.g. speed, confidence, precision or accuracy of the Interpreted_Data produced). The cost of a particular Data_Interpretation_Solution can also be a measurement criterion, e.g. processing power required.

Pre-condition

A condition upon which a Data_Interpretation_Solution is dependent or co-dependent (e.g. suitable Sensor_Data has been, or is being, captured or processing power is available).

Requirement

A requirement to achieve a result by planning and executing a Data_Interpretation_Solution (e.g. to locate enemy tanks in a region of territory, or provide human readable recon images enhanced from SAR data).

Sensor_Data

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 originating from multiple sources.

Dynamic_Inputs

Platform, environmental or tactical information that influences the planning or enactment of Data_Interpretation_Solutions; for example, atmospheric conditions affecting EW wave propagation, weather features occluding targets of sensing activity, vehicle speed and position that determine sensor field of view, rate of change and direction of movement.

B.2.52.6 Design Rationale

B.2.52.6.1 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.
- Additional capabilities of new Interpretation_Resources or configurations installed on an Exploiting Platform can be added easily, which can cater for changes in Activity_Capability between missions.

B.2.52.6.2 Design Considerations

Directly Applicable Policies

These policies 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 Activity_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. This facilitates reusability, exploitability and maintainability.
- Recording and Logging - Logging operations and record retention will be performed in accordance with this policy.
- 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_Processing_Activity should be applied to Sensor_Data to achieve the required Interpreted_Data output) could be separated out using an extension point on the Sensor Data Interpretation component. This contributes towards Sensor Data Interpretation and its extensions being configurable, resilient against obsolescence and exploitable.

Exploitation Considerations

- Sensor Data Interpretation is responsible for coordinating Interpretation_Resources, but not for capturing their inputs (i.e. it sets up Interpretation_Resources to act on data products that have been captured, digitised and pre-processed into Sensor_Data, but it does not capture or generate its own Sensor_Data). However, Sensor Data Interpretation can take into account the availability and quality of Sensor_Data to adapt Data_Interpretation_Solutions during their planning and execution to optimise effectiveness.
- The effective deployment of complex Data_Interpretation_Solutions may require their planning and execution to be tightly coordinated with the component(s) responsible for coordinating the provision of the Sensor_Data, taking into account highly dynamic changes with very low latency. This may require the use of coordination points and feedback loops between those components and Sensor Data Interpretation, so that the sensor data capture can be dynamically adapted to provide Sensor_Data that is more efficiently processed by Sensor Data Interpretation into Interpreted_Data.
- Implementations of this component may determine selection and sequencing of processes into Data_Interpretation_Solutions either:
  - Dynamically - utilising current performance and limitation data about each process, along with current Sensor_Data inputs.
  - Statically - utilising performance and limitation data previously recorded against a fixed solution.

- Sensor Data Interpretation may select a solution based on available sensor data or a solution based on obtainable sensor data. In the latter case it will raise a pre-condition against that solution for the system to obtain that type and quality of data. Note: Sensor Data Interpretation does not know the source of the Sensor_Data. Sensor_Data may have been provided by other components in the deployment or by external systems.
- Sensor Data Interpretation is able to coordinate Interpretation_Resources to extract meaning from stored Sensor_Data.

B.2.52.6.3 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.

B.2.52.6.4 Security Considerations

The indicative security classification is SNEO.

This component is responsible for managing the interpretation of available Sensor_Data, 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.

B.2.52.7 Services

B.2.52.7.1 Service Definitions

B.2.52.7.1.1 Interpretation_Requirement

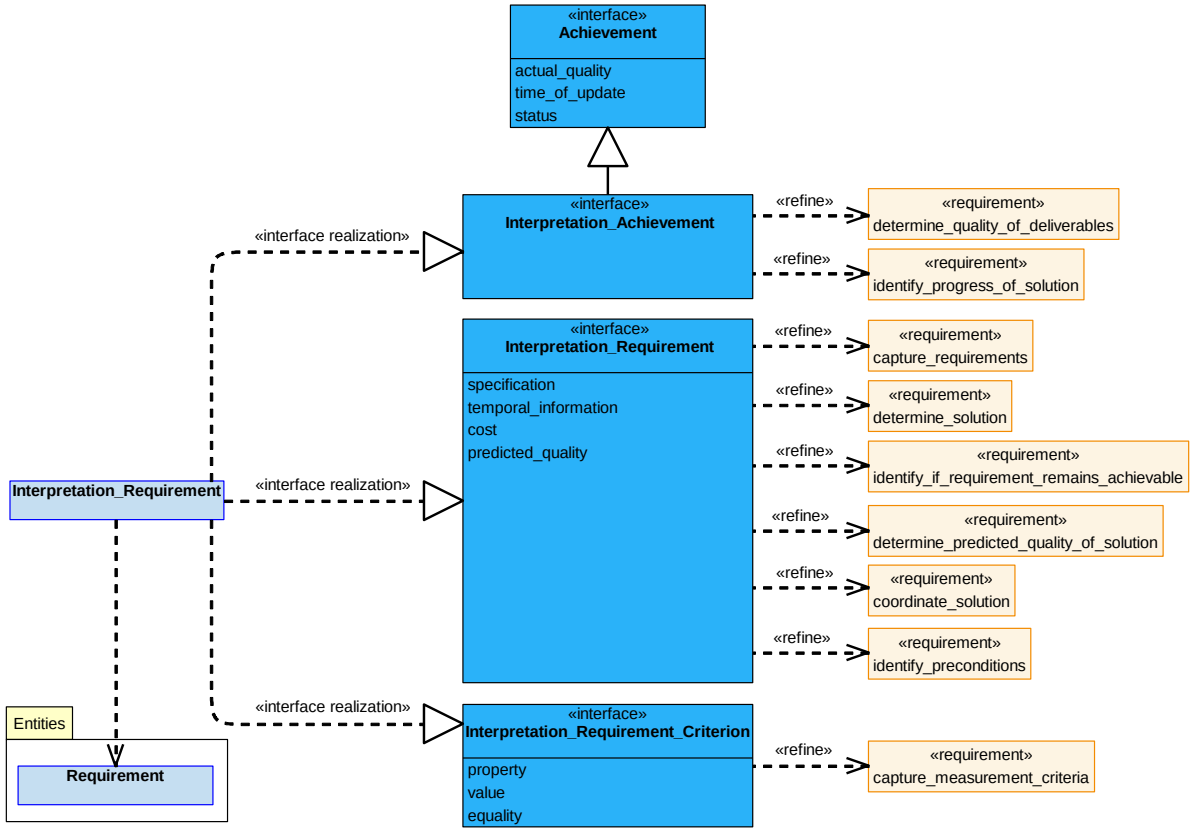

Figure 952: Interpretation_Requirement Service Definition

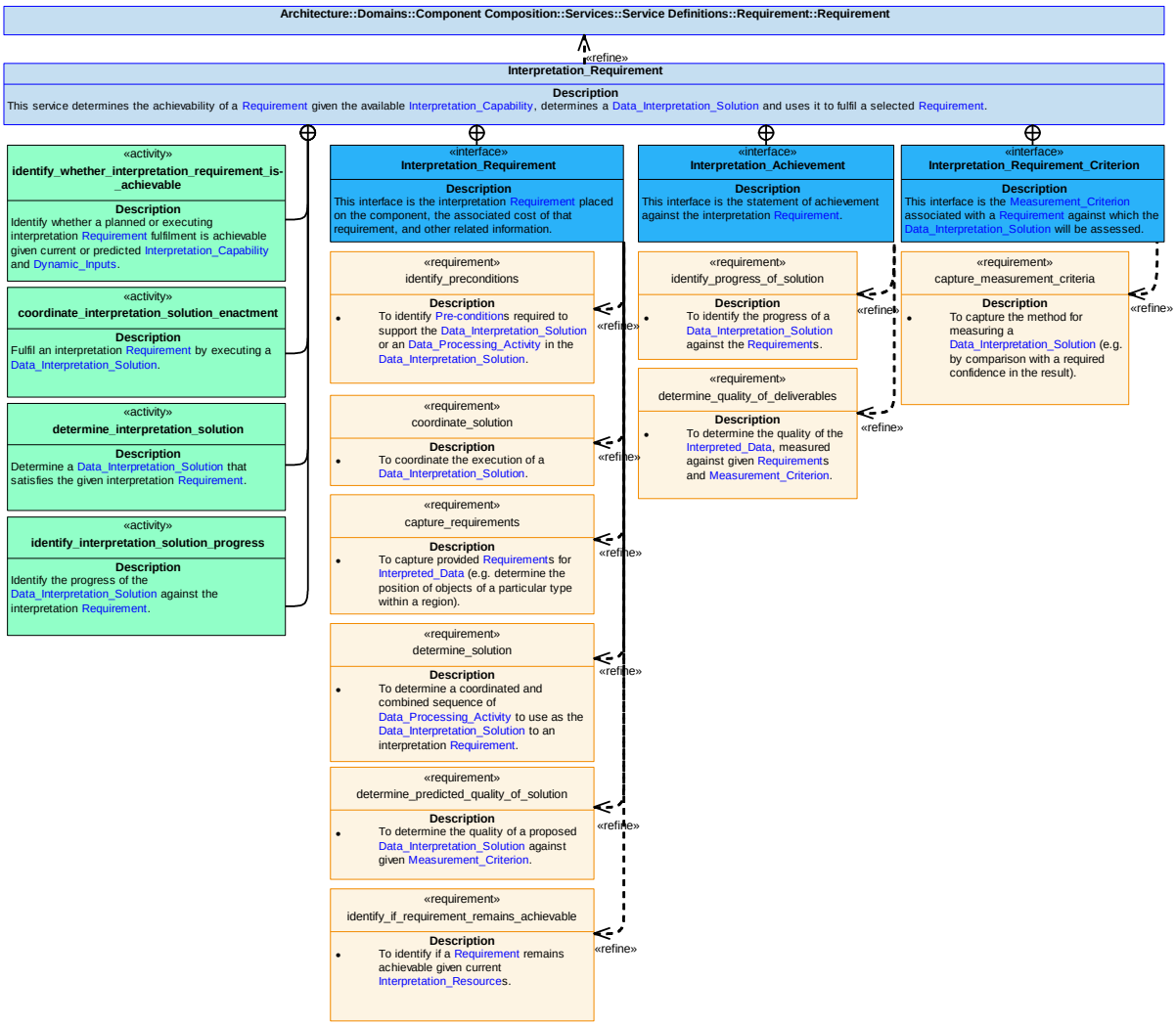

Figure 953: Interpretation_Requirement Service Policy

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 Dynamic_Inputs.

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.

B.2.52.7.1.2 Data_Processing_Dependency

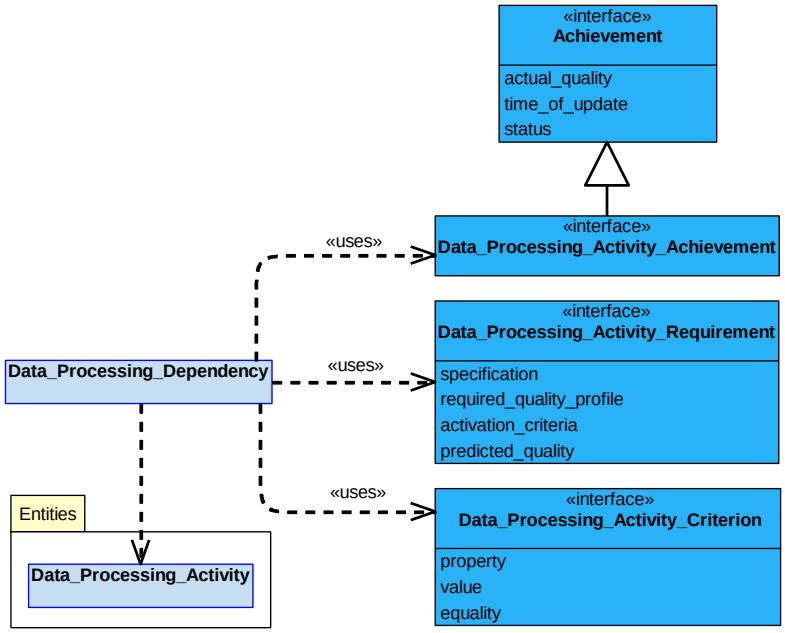

Figure 954: Data_Processing_Dependency Service Definition

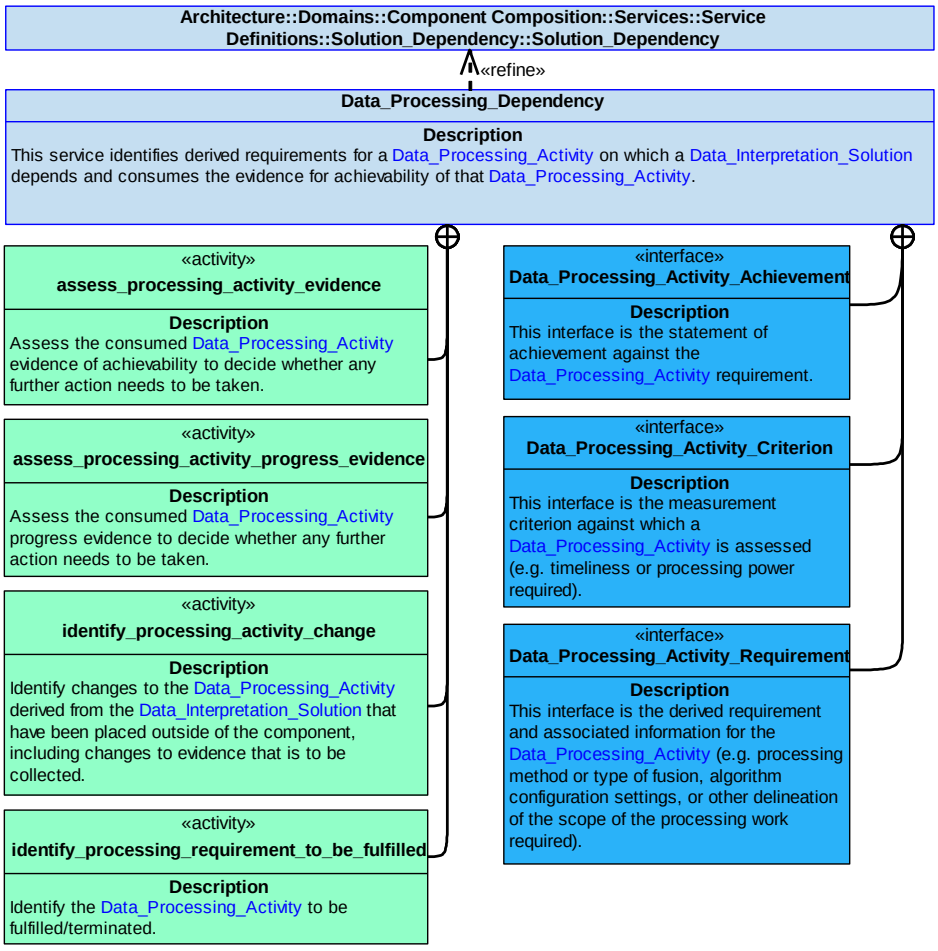

Figure 955: Data_Processing_Dependency Service Policy

Data_Processing_Dependency

This service identifies derived requirements for a Data_Processing_Activity on which a Data_Interpretation_Solution depends and consumes the evidence for achievability of that Data_Processing_Activity.

Interfaces

Data_Processing_Activity_Achievement

This interface is the statement of achievement against the Data_Processing_Activity requirement.

Data_Processing_Activity_Criterion

This interface is the measurement criterion against which a Data_Processing_Activity 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_Processing_Activity (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_Processing_Activity technique needed, for example, object detections, signal processing, image sharpening or fusion algorithm.
required_quality_profile	Precision and accuracy required for the Data_Processing_Activity for example, specific confidence levels, resolution of output, granularity of object detections.
activation_criteria	Trigger for initiating and ceasing the Data_Processing_Activity, 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_Processing_Activity.

Activities

assess_processing_activity_evidence

Assess the consumed Data_Processing_Activity evidence of achievability to decide whether any further action needs to be taken.

assess_processing_activity_progress_evidence

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

identify_processing_activity_change

Identify changes to the Data_Processing_Activity 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_Processing_Activity to be fulfilled/terminated.

B.2.52.7.1.3 Data_Provision_Dependency

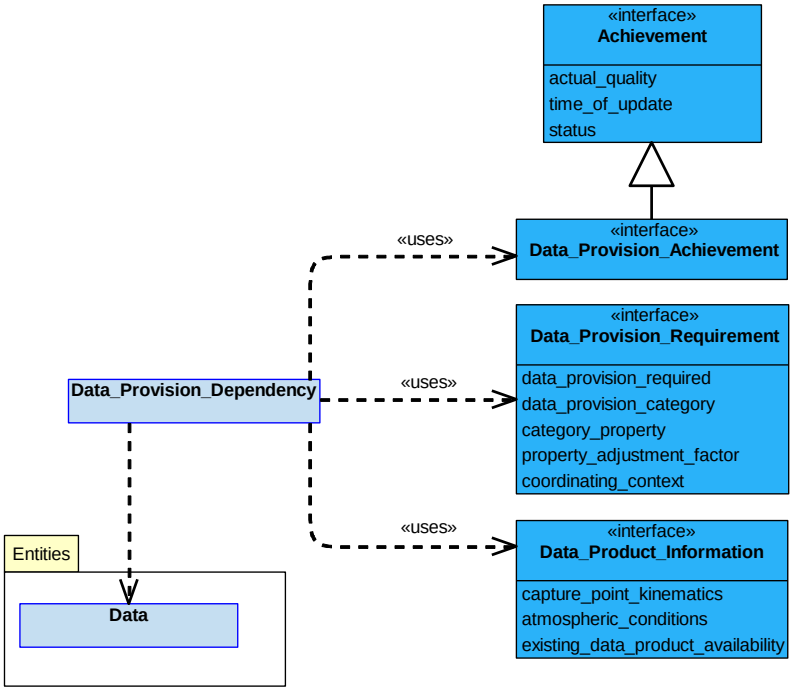

Figure 956: Data_Provision_Dependency Service Definition

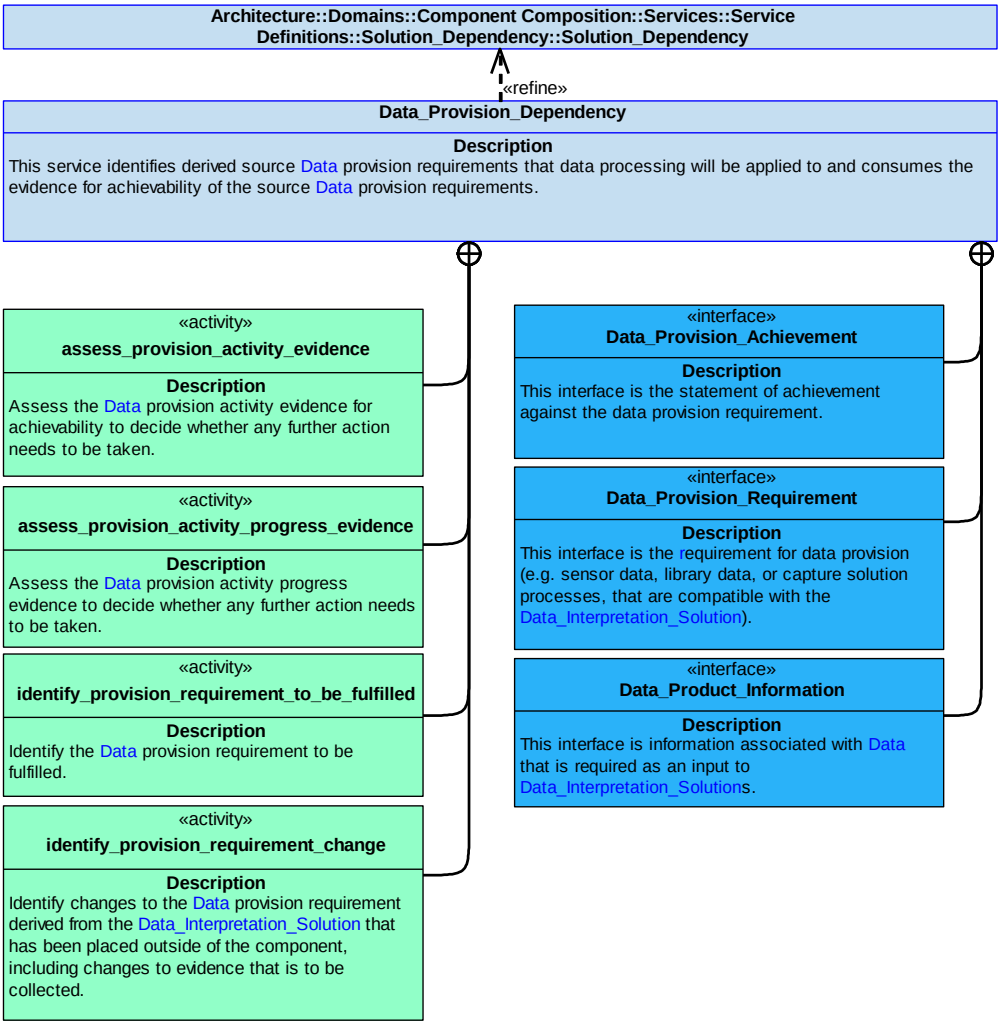

Figure 957: Data_Provision_Dependency Service Policy

Data_Provision_Dependency

This service identifies derived source Data provision requirements that data processing will be applied to and consumes the evidence for achievability of the source Data 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 provision (e.g. sensor data, library data, or capture solution processes, that are compatible with the Data_Interpretation_Solution).

Attributes

data_provision_required	A required source data provision, e.g. a type of Sensor_Data 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 capture.
category_property	Aspect of data_provision_category that would benefit from adjustment of control inputs.
property_adjustment_factor	Quantitative degree of adjustment or value to be applied for data provision category_property.
coordinating_context	Identification and character of coordination points with other actions; for example, where this component is interpreting data to be used as part of a larger task that includes co-dependent data capture action(s), the identification of the other actions involved might be needed, along with the nature of their interaction as it pertains to this Data_Processing_Activity or Data_Interpretation_Solution.

Data_Product_Information

This interface is information associated with Data that is required as an input to Data_Interpretation_Solutions.

Attributes

capture_point_kinematics	Spatial location, orientation and velocity of platforms and sensors at the time of the existing Data input capture.
atmospheric_conditions	Atmospheric factors affecting interpretation, through which existing Data products were captured.
existing_data_product_availability	Availability of existing Data that can be used as inputs to a Data_Interpretation_Solution.

Activities

assess_provision_activity_evidence

Assess the Data provision activity evidence for achievability to decide whether any further action needs to be taken.

assess_provision_activity_progress_evidence

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

identify_provision_requirement_change

Identify changes to the Data 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 provision requirement to be fulfilled.

B.2.52.7.1.4 Information_Dependency

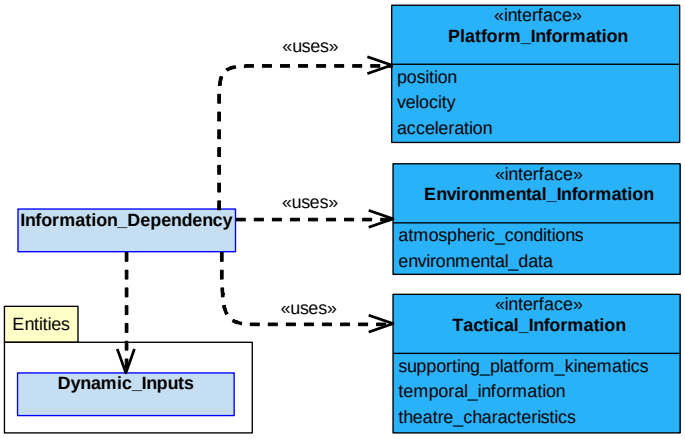

Figure 958: Information_Dependency Service Definition

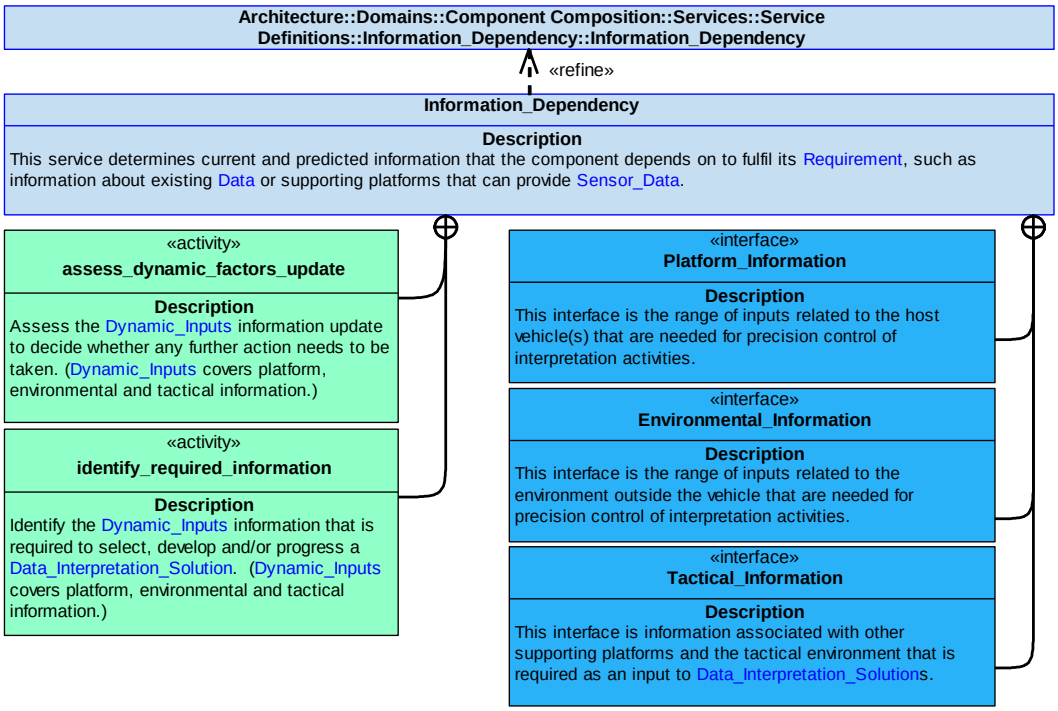

Figure 959: Information_Dependency Service Policy

Information_Dependency

This service determines current and predicted information that the component depends on to fulfil its Requirement, such as information about existing Data or supporting platforms that can provide Sensor_Data.

Interfaces

Tactical_Information

This interface is information associated with other supporting platforms and the tactical environment that is required as an input to Data_Interpretation_Solutions.

Attributes

supporting_platform_kinematics	Location, orientation, velocity and acceleration of other platform(s) providing Sensor_Data.
temporal_information	Timing of supporting platform availability for new Sensor_Data capture and low latency synchronization of sensing product processing techniques.
theatre_characteristics	Known tactical environment characteristics that may be important in Data_Interpretation_Solution planning, such as filtering out a background of non-target objects like civilian vehicles.

Environmental_Information

This interface is the range of inputs related to the environment outside the vehicle that are needed for precision control of interpretation activities.

Attributes

atmospheric_conditions	Current and predicted weather conditions and features.
environmental_data	Information describing surfaces and features in the environment.

Platform_Information

This interface is the range of inputs related to the host vehicle(s) that are needed for precision control of interpretation activities.

Attributes

position	Current and predicted platform location and orientation.
velocity	Speed and direction of the platform.
acceleration	Acceleration of the platform.

Activities

assess_dynamic_factors_update

Assess the Dynamic_Inputs information update to decide whether any further action needs to be taken. (Dynamic_Inputs covers platform, environmental and tactical information.)

identify_required_information

Identify the Dynamic_Inputs information that is required to select, develop and/or progress a Data_Interpretation_Solution. (Dynamic_Inputs covers platform, environmental and tactical information.)

B.2.52.7.1.5 Interpretation_Capability

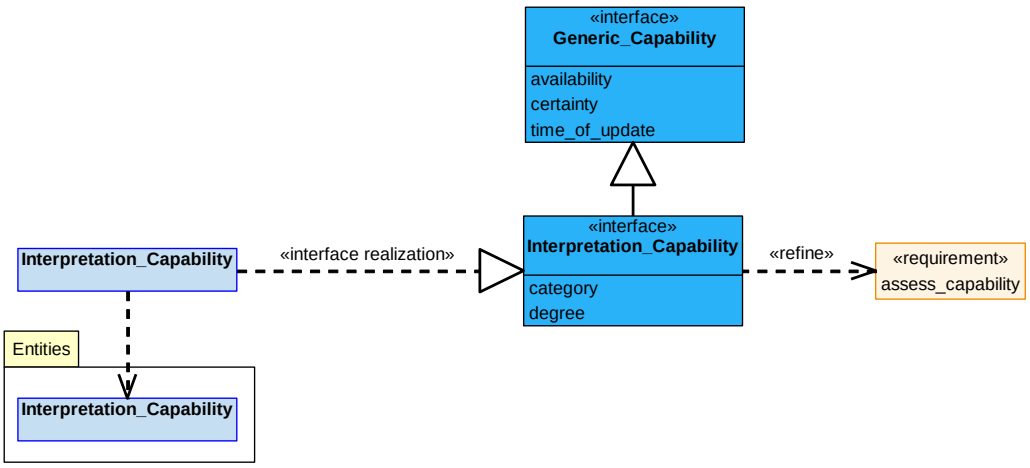

Figure 960: Interpretation_Capability Service Definition

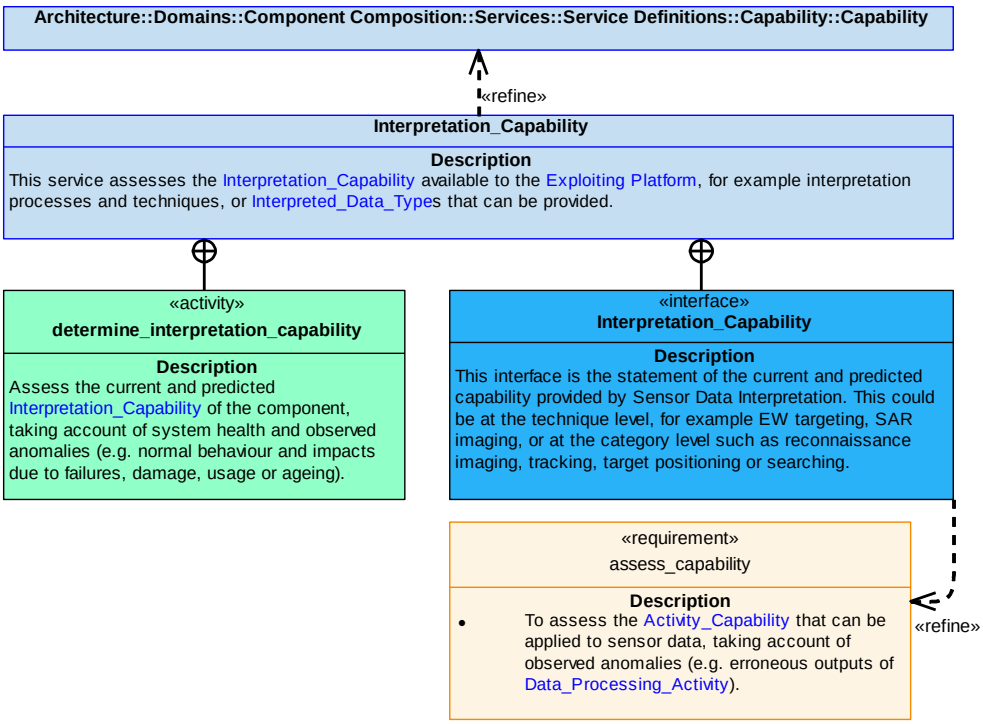

Figure 961: Interpretation_Capability Service Policy

Interpretation_Capability

This service assesses the Interpretation_Capability available to the Exploiting Platform, for example interpretation processes and techniques, or Interpreted_Data_Types 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 or category of Interpretation_Capability that can be provided.
degree	The level of performance or effectiveness that can be achieved for a given type 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).

B.2.52.7.1.6 Capability_Evidence

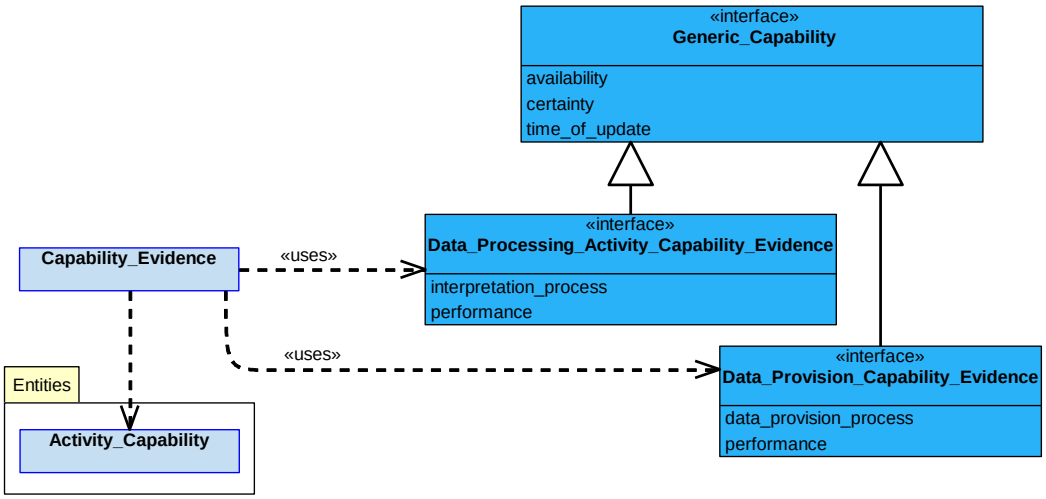

Figure 962: Capability_Evidence Service Definition

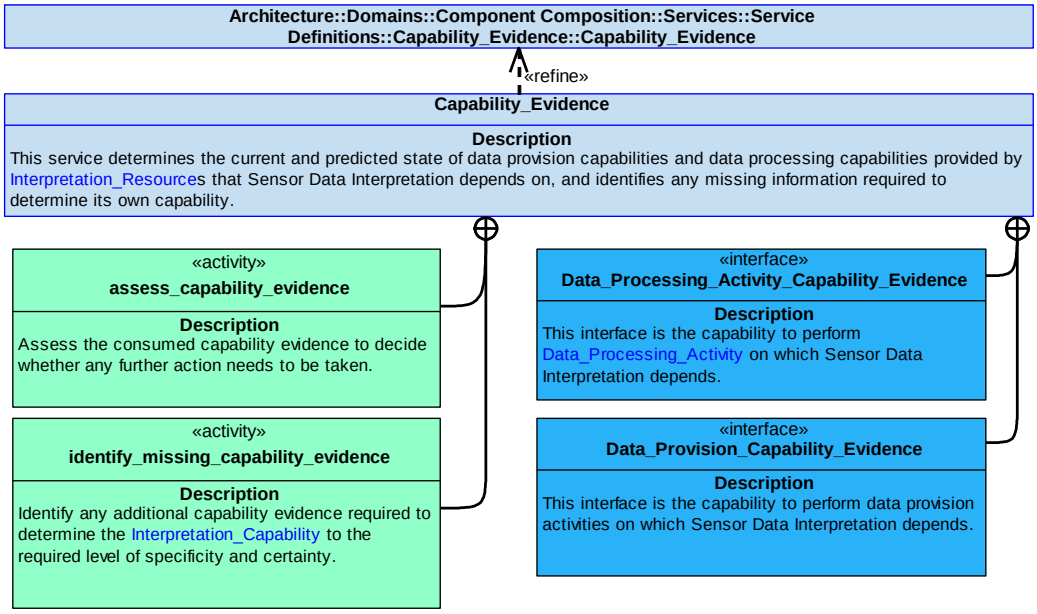

Figure 963: Capability_Evidence Service Policy

Capability_Evidence

This service determines the current and predicted state of data provision capabilities and data processing capabilities provided by Interpretation_Resources 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 Data_Processing_Activity on which Sensor Data Interpretation depends.

Attributes

interpretation_process	The specific processing algorithm or technique. 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_process, 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 data provision technique or process that delivers this capability, including the inputs required for its use. For example, capability of information gathering (e.g. image gathering, electronic surveillance gathering).
performance	The level or degree of capability available for this particular data provision Pre-condition, taking into account the data provision resources installed.

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.

B.2.52.7.2 Service Dependencies

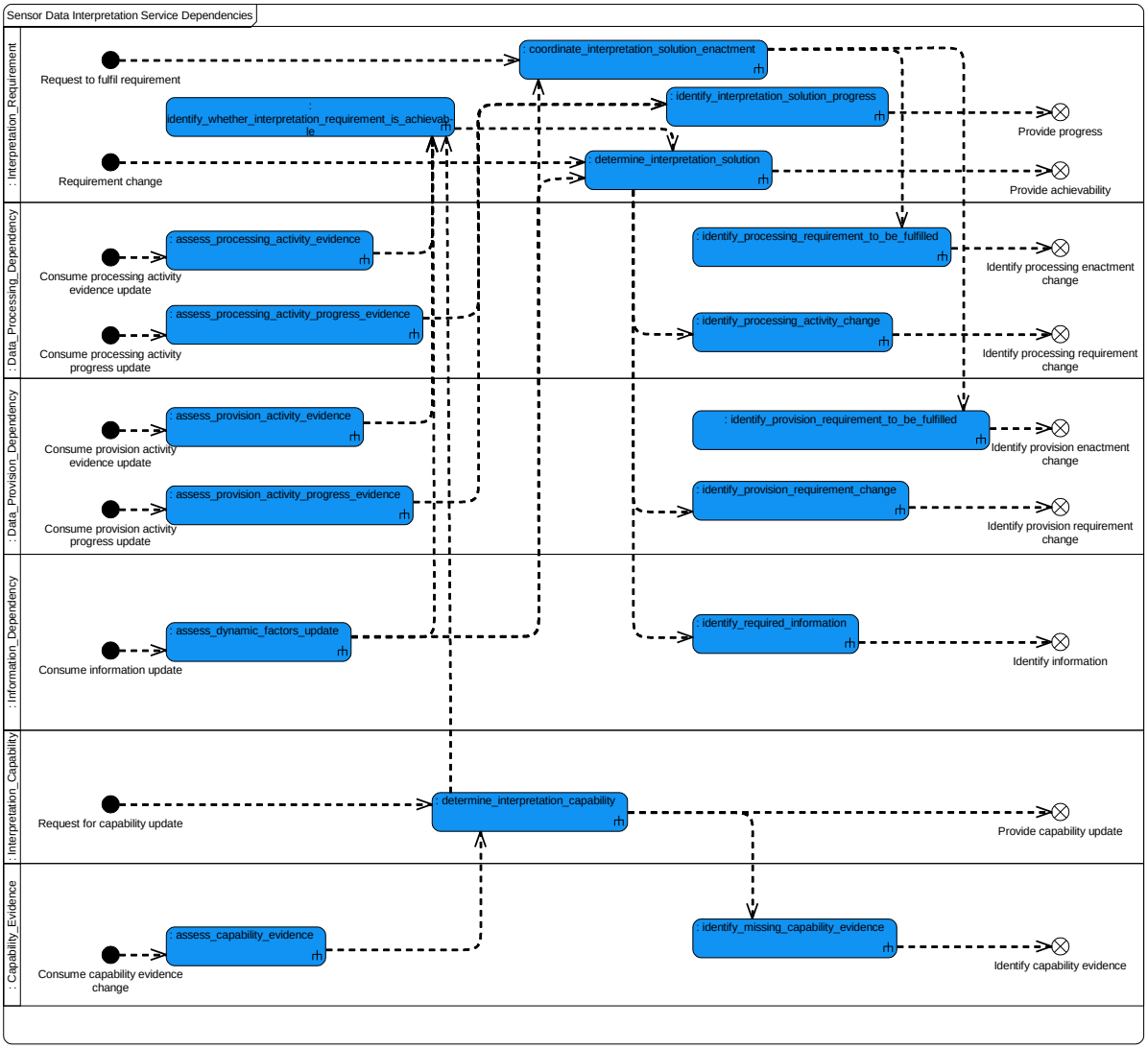

Figure 964: Sensor Data Interpretation Service Dependencies

B.2.53 Sensor Products

B.2.53.1 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.

B.2.53.2 Overview

Control Architecture

Sensor Products is a service component as defined in the Control Architecture policy.

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).

B.2.53.3 Service Summary

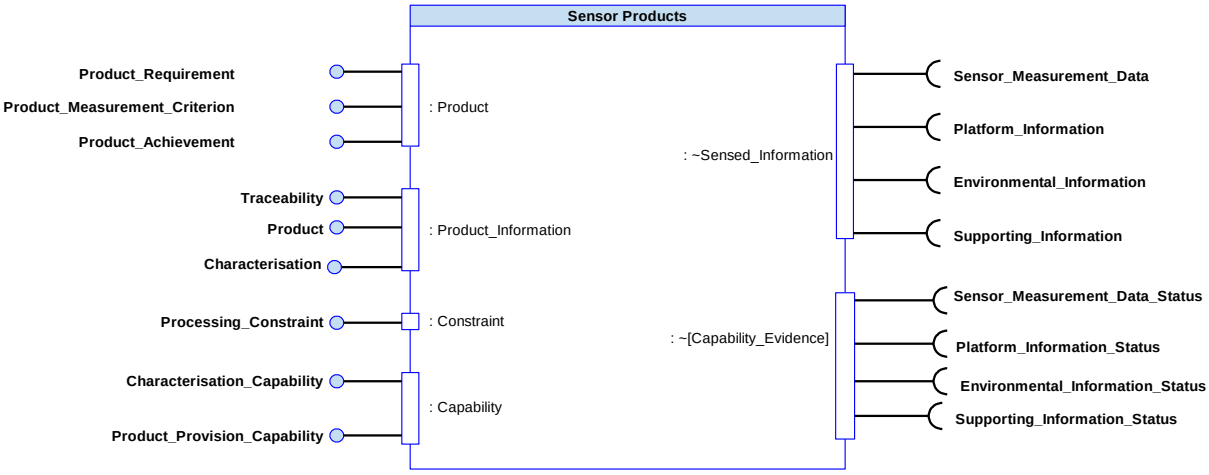

Figure 965: Sensor Products Service Summary

B.2.53.4 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_solution

- To determine a solution which meets the Requirements and satisfies the Constraints for Sensor_Product provision, manipulation, or analysis.

determine_solution_quality

- To determine the Quality of a solution against the Measurement_Criterion.

capture_sensor_products

- To capture Sensor_Products.

capture_acquisition_characteristics

- To capture the acquisition characteristics of Sensor_Products (e.g. time of capture, spatial region captured, or spectral frequency captured).

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).

execute_solution

- To provide, manipulate, or analyse Sensor_Products based on determined solutions.

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.

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).

predict_capability_progression

- To predict the progression of Sensor Products Capability over time and with use.

determine_if_requirement_is_achievable

- To determine if a Sensor_Product requirement is achievable given current Capability and Constraints.

B.2.53.5 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.

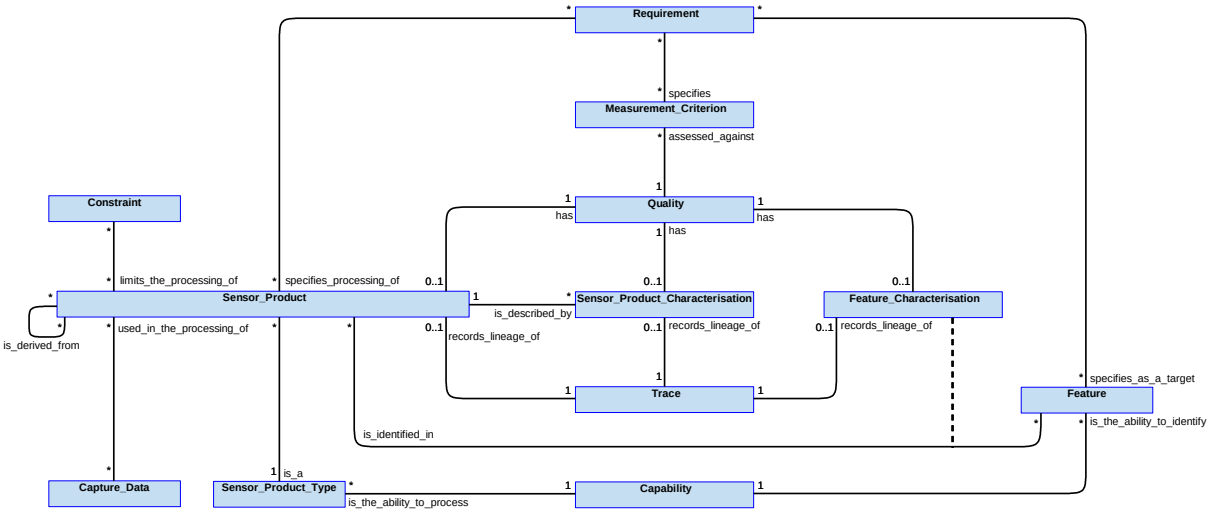

Figure 966: Sensor Products Semantics

B.2.53.5.1 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.

B.2.53.6 Design Rationale

B.2.53.6.1 Assumptions

None.

B.2.53.6.2 Design Considerations

Directly Applicable Policies

These policies 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**.**

B.2.53.6.3 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. In accordance with UK MoD direction (see Safety Analysis policy) this drives a DAL B indicative IDAL.

B.2.53.6.4 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.

B.2.53.7 Services

B.2.53.7.1 Service Definitions

B.2.53.7.1.1 Product

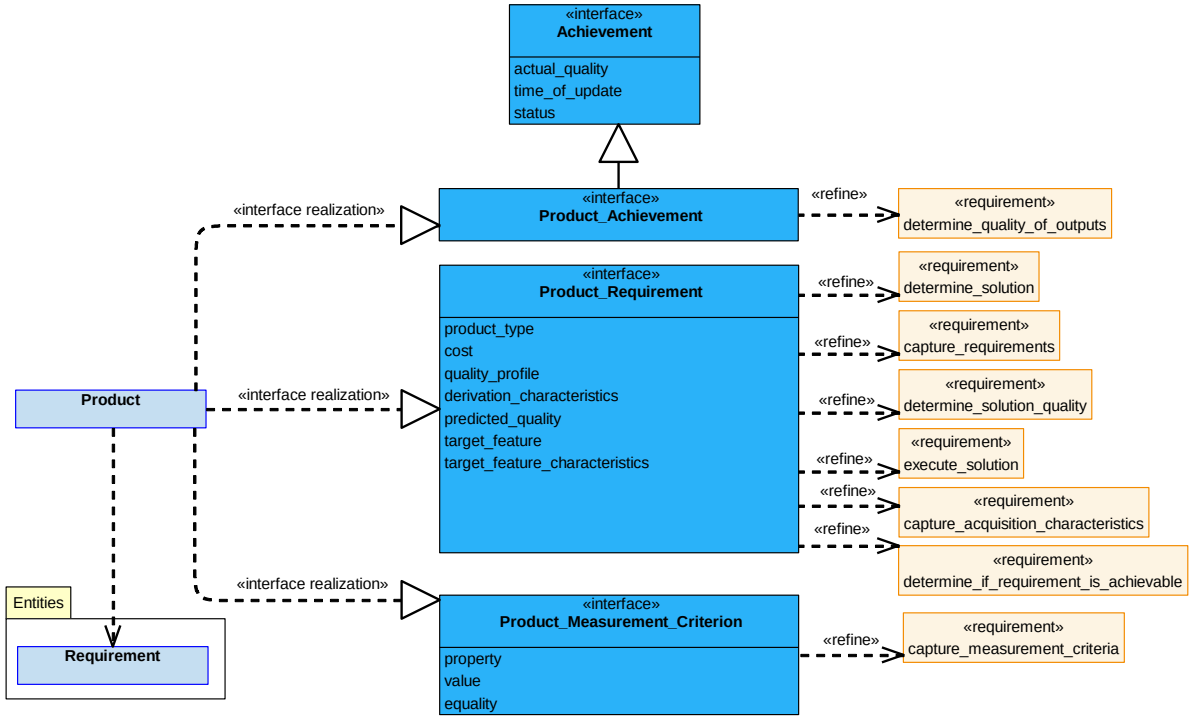

Figure 967: Product Service Definition

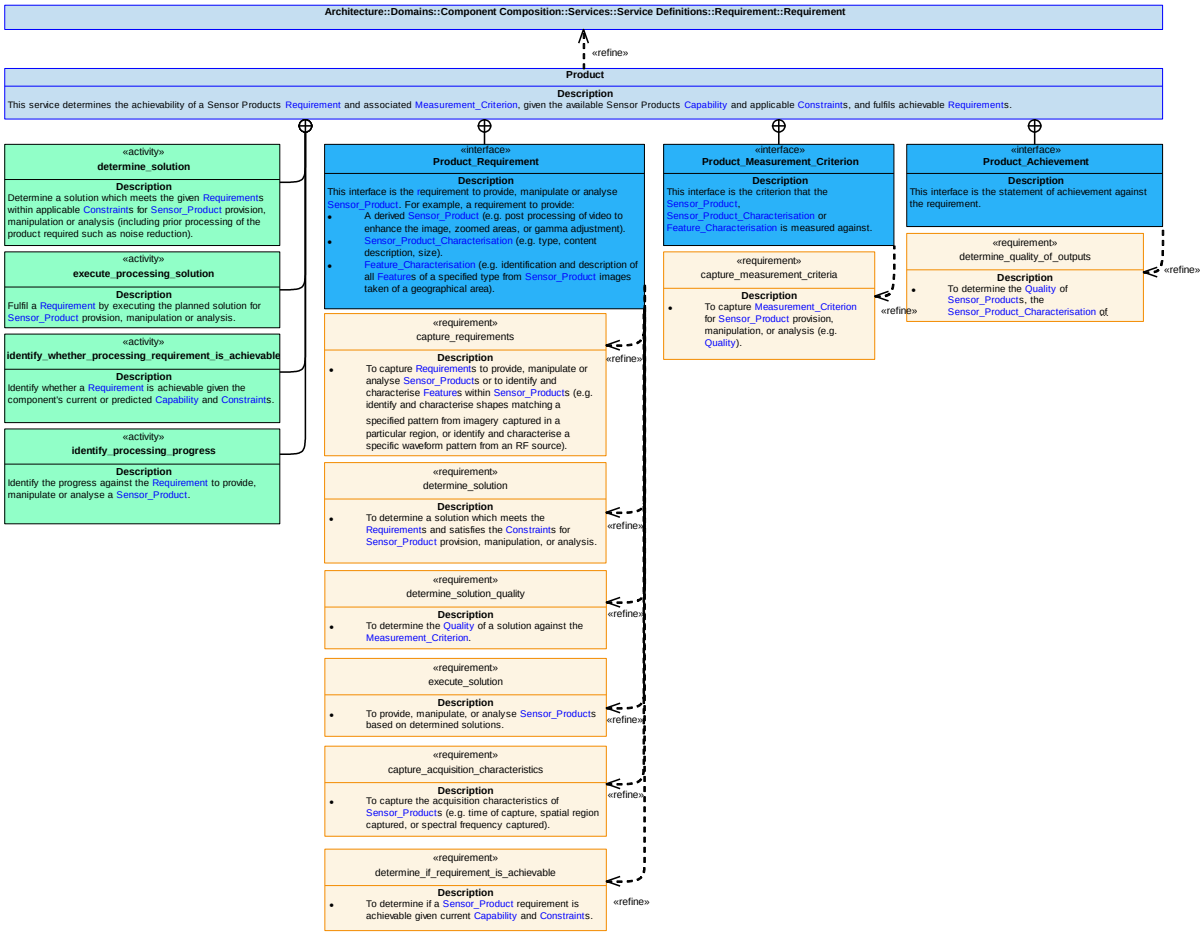

Figure 968: Product Service Policy

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.

B.2.53.7.1.2 Product_Information

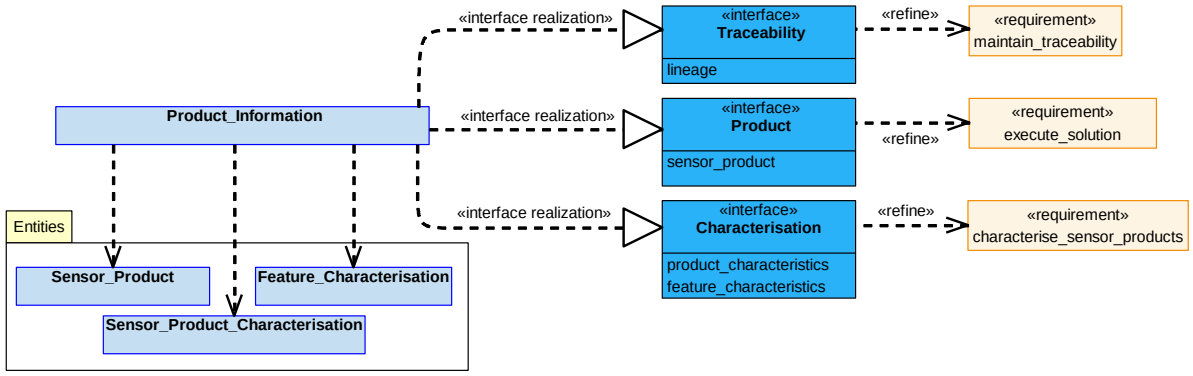

Figure 969: Product Information Service Definition

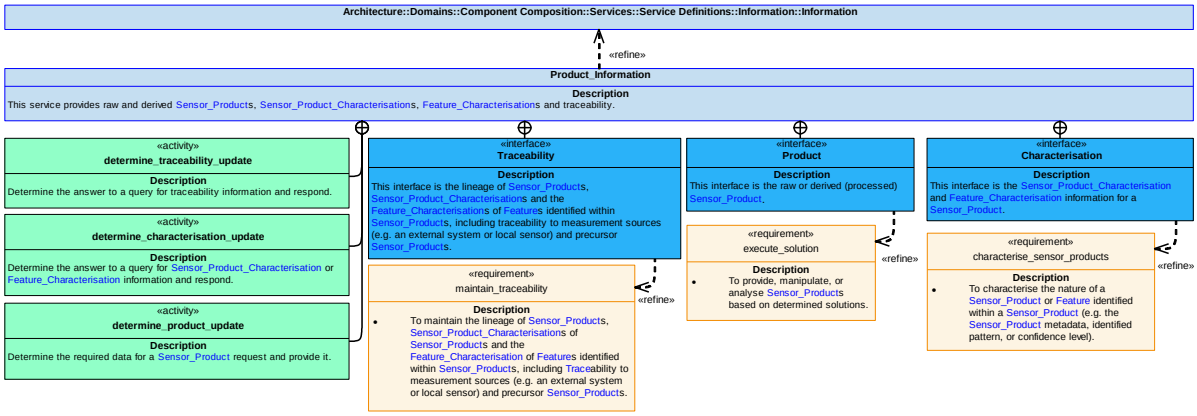

Figure 970: Product Information Service Policy

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.

B.2.53.7.1.3 Sensed_Information

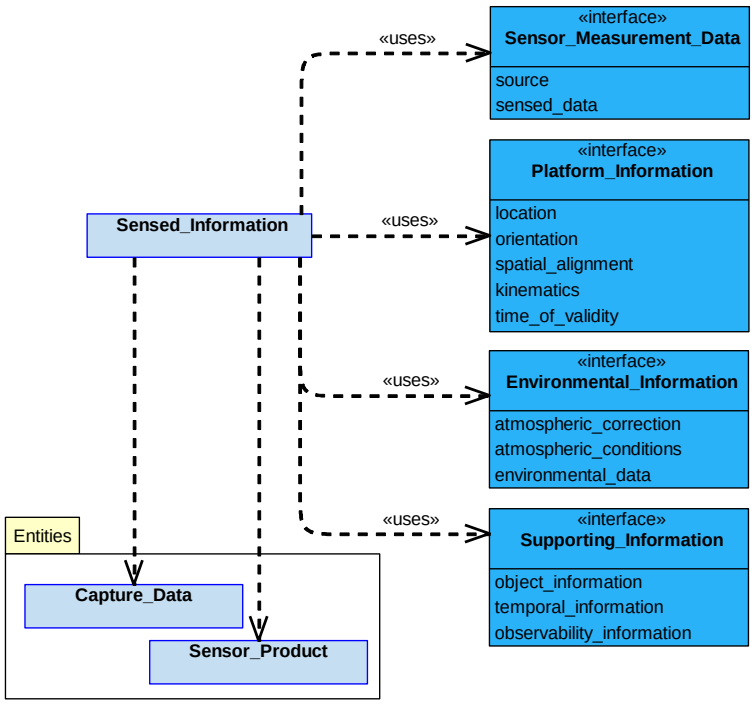

Figure 971: Sensed_Information Service Definition

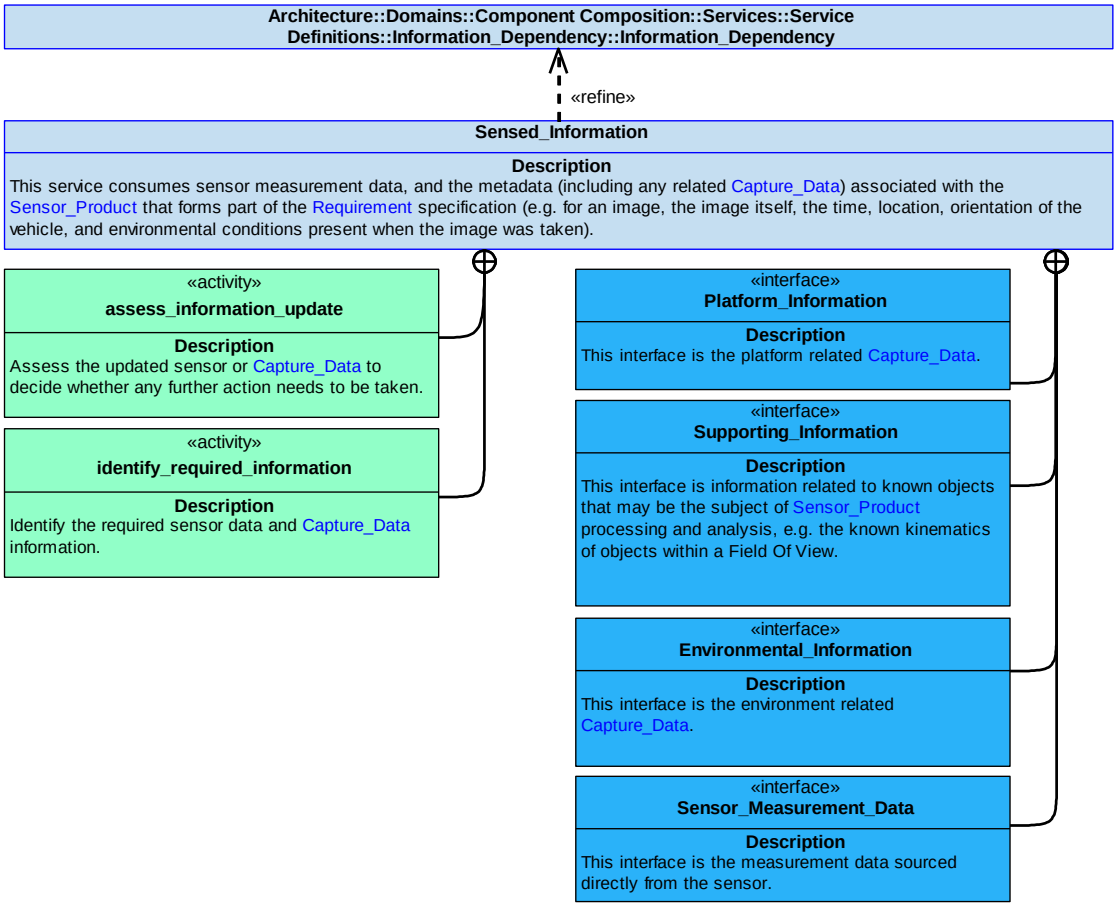

Figure 972: Sensed_Information Service Policy

Sensed_Information

This service consumes sensor measurement data, and the metadata (including any related Capture_Data) associated with the Sensor_Product that forms part of the Requirement specification (e.g. for an image, the image itself, the time, location, orientation of the vehicle, and environmental conditions present when the image was taken).

Interfaces

Platform_Information

This interface is the 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.

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).

Supporting_Information

This interface is 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.

Attributes

object_information	Kinetic information about sensing target criteria, location and movement, or that of surrounding objects. Knowledge of an objects motion at the time of Sensor_Product capture can influence subsequent processing activities.
temporal_information	Timing information required for low latency synchronization of sensing techniques.
observability_information	Interactive low observability factors affecting the use of active sensing techniques.

Sensor_Measurement_Data

This interface is the measurement data sourced directly from the sensor.

Attributes

source	The source of the sensed data, e.g. a specific sensor and its location (which could be another platform).
sensed_data	The data sourced from the sensor itself including associated metadata (e.g. radar record, image, video recording or EW record).

Activities

assess_information_update

Assess the updated sensor or Capture_Data to decide whether any further action needs to be taken.

identify_required_information

Identify the required sensor data and Capture_Data information.

B.2.53.7.1.4 Constraint

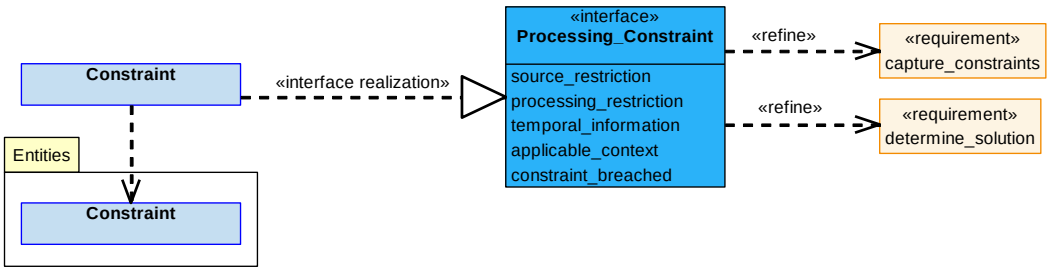

Figure 973: Constraint Service Definition

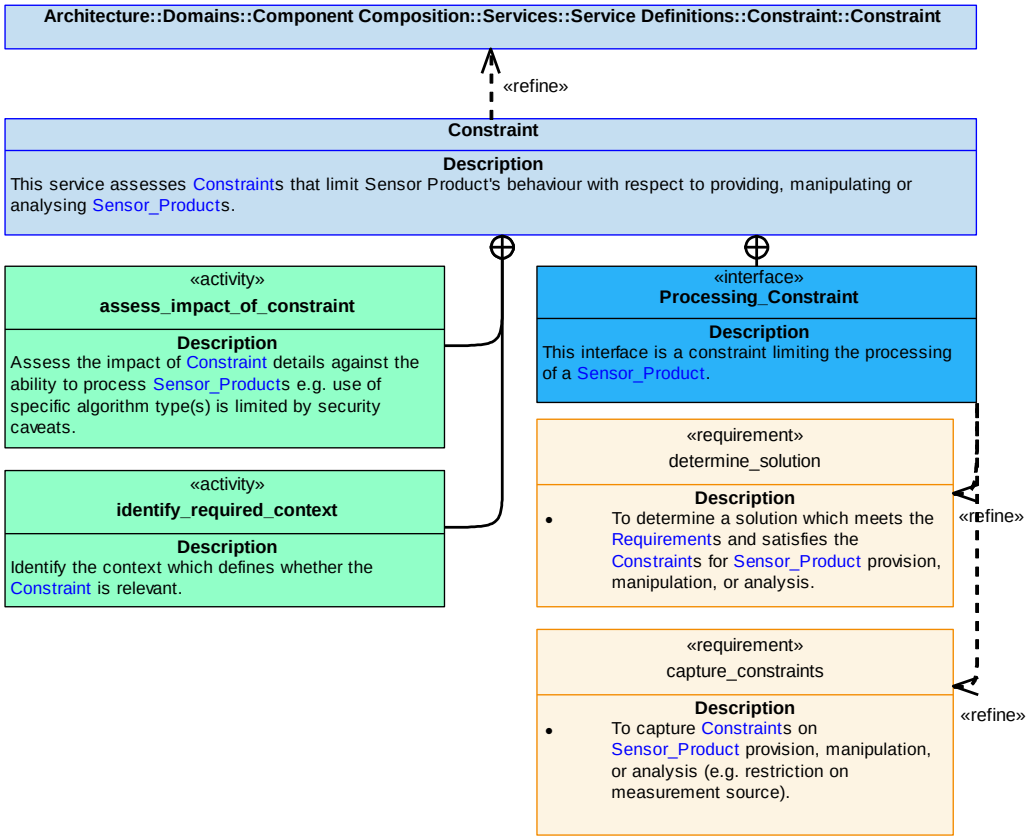

Figure 974: Constraint Service Policy

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.

B.2.53.7.1.5 Capability

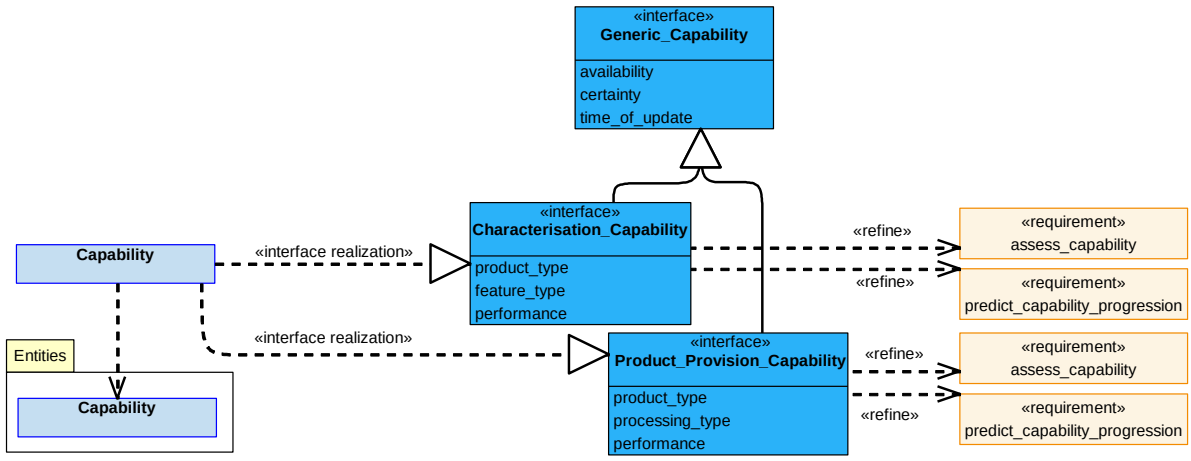

Figure 975: Capability Service Definition

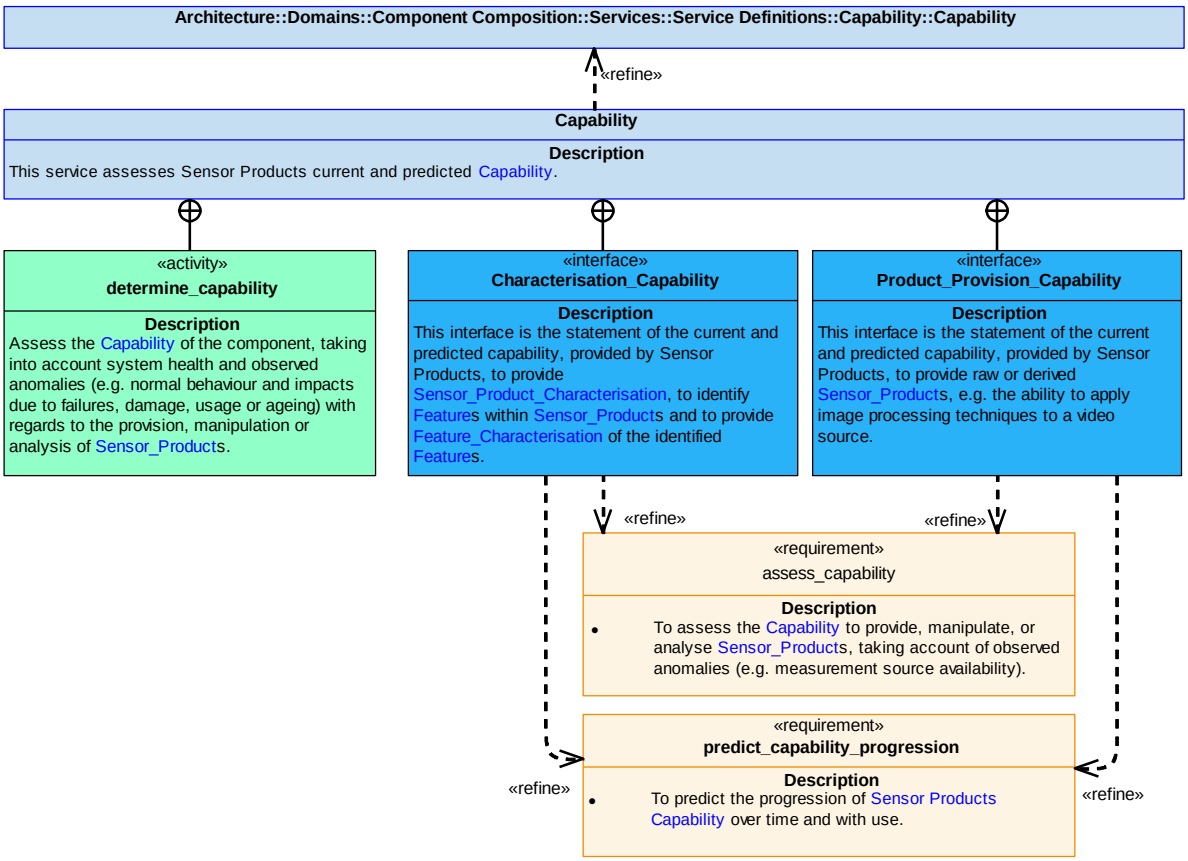

Figure 976: Capability Service Policy

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.

B.2.53.7.1.6 Capability_Evidence

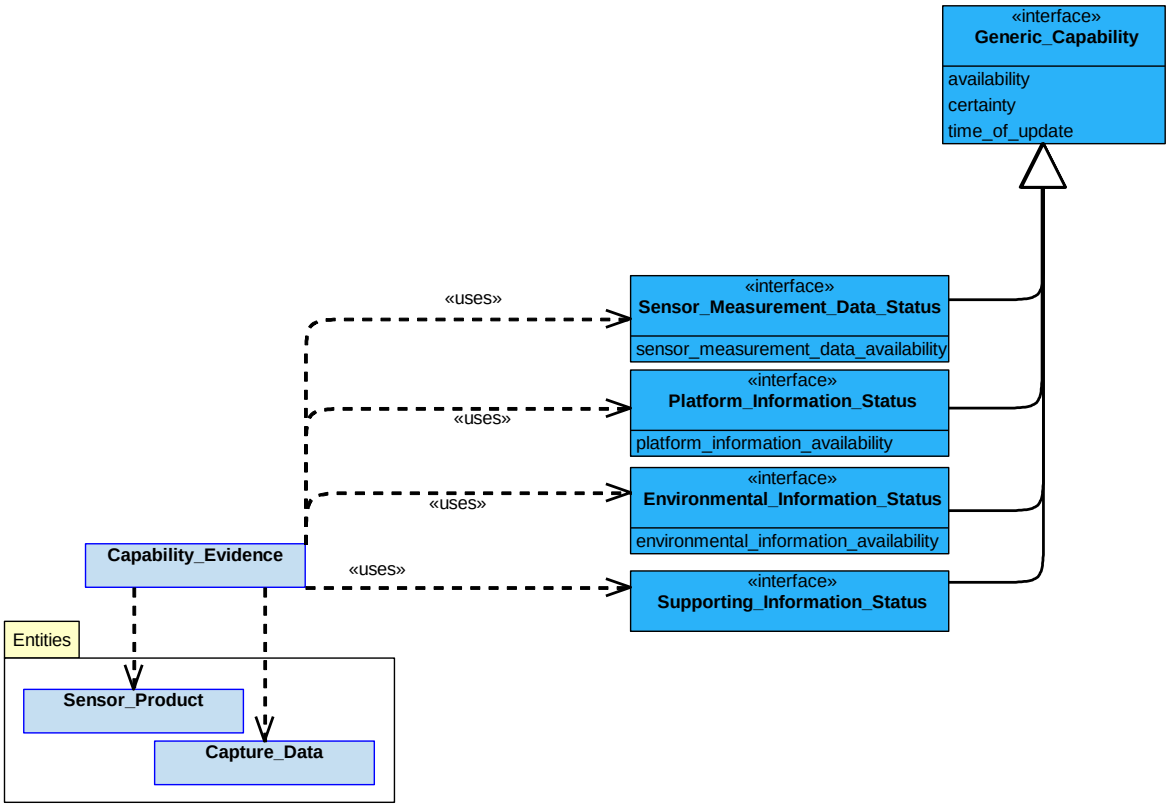

Figure 977: Capability_Evidence Service Definition

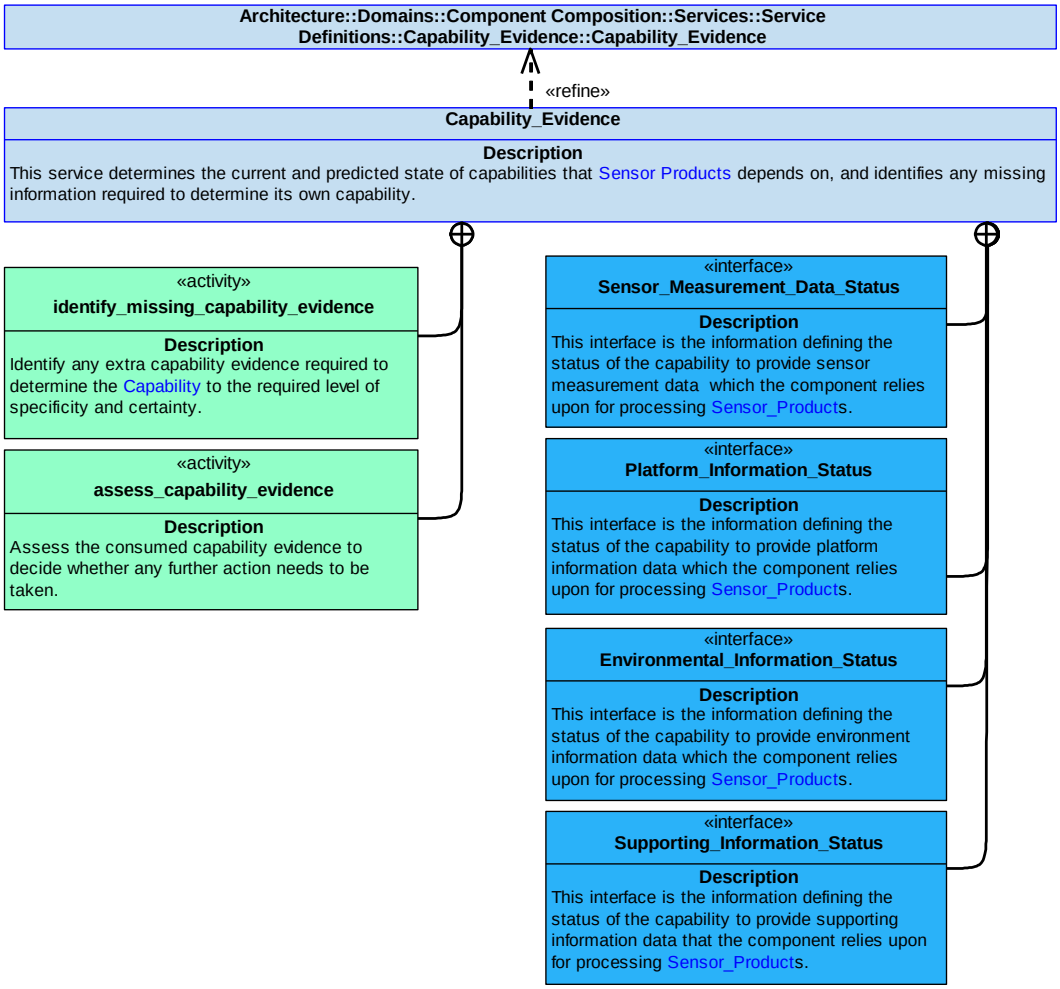

Figure 978: Capability_Evidence Service Policy

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.

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.

Platform_Information_Status

This interface is the information defining the status of the capability to provide platform information data which the component relies upon for processing Sensor_Products.

Attribute

platform_information_availability	The availability of 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.

B.2.53.7.2 Service Dependencies

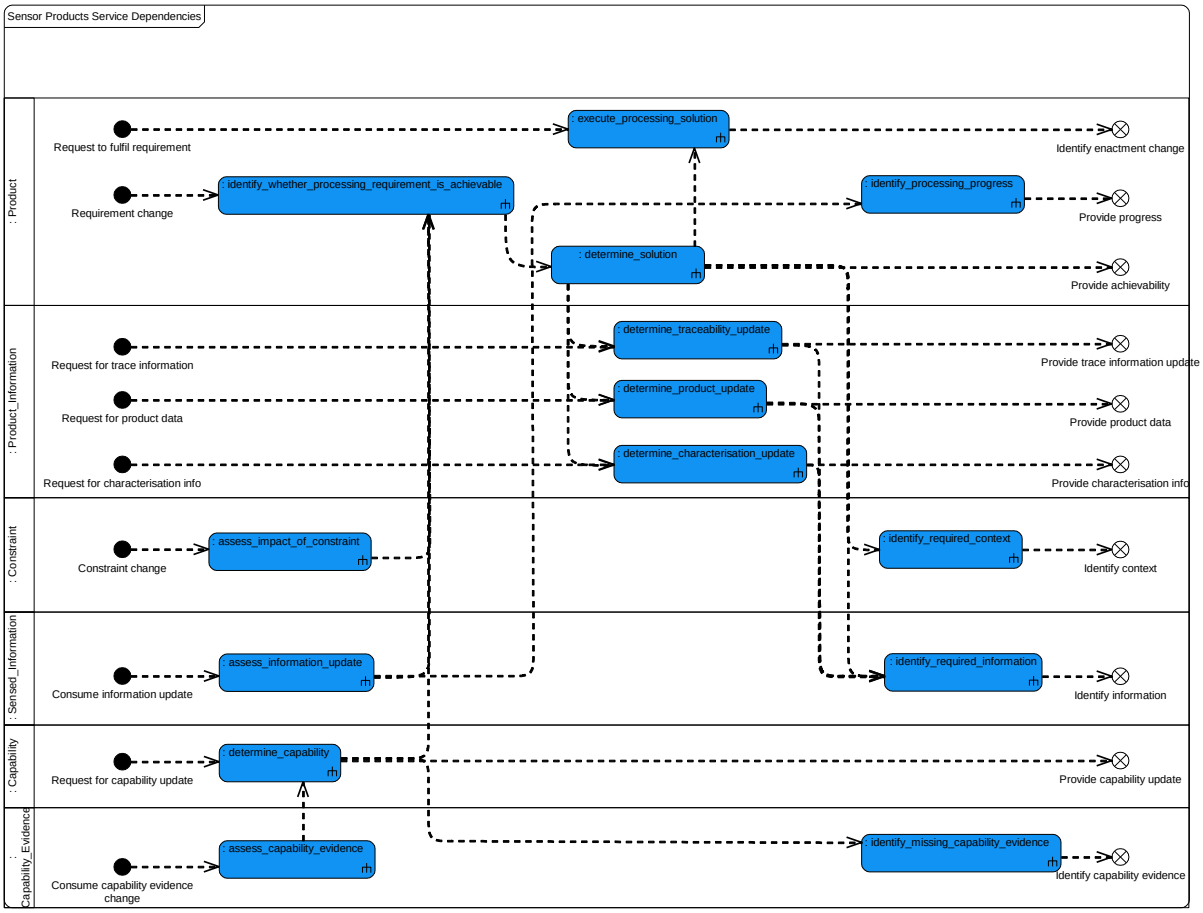

Figure 979: Sensor Products Service Dependencies

B.2.54 Sensors

B.2.54.1 Role

The role of Sensors is to provide an interface to obtain measurements from sensors.

B.2.54.2 Overview

Sensors is a resource component as defined in the Control Architecture policy.

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.

B.2.54.3 Service Summary

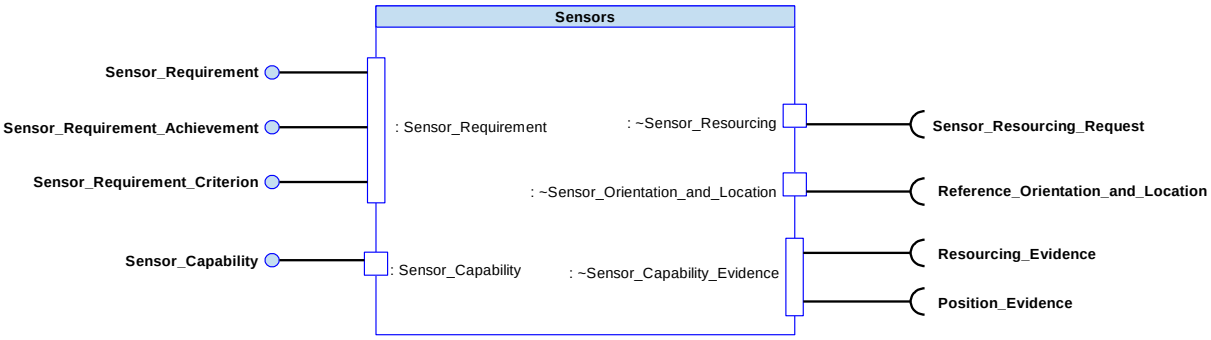

Figure 980: Sensors Service Summary

B.2.54.4 Responsibilities

capture_requirements_for_sensor_resources

- To capture provided requirements (e.g. turn on/off or obtain a measurement) for use of Sensor_Resources.

control_use_of_sensor

- To control the use of Sensor_Resources to obtain a measurement.

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).

predict_capability_progression

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

capture_sensor_data

- To collect data from a Sensor_Resource (e.g. temperature from a thermometer or location of object in an area).

determine_sensor_solution

- To determine a solution for use of Sensor_Resources that will meet given Requirements.

capture_measurement_criteria

- To capture Measurement_Criterion for a Sensor_Measurement.

update_resource_usages

- To update the status of the sensors usage of the platform's resources.

provide_sensor_data_feedback

- To provide the wider system with feedback data.

identify_missing_information

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

identify_progress

- To identify progress against a Requirement.

determine_if_solution_remains_feasible

- To determine if a planned or ongoing Sensor_Solution remains feasible given current Capability.

B.2.54.5 Subject Matter Semantics

The subject matter of Sensors is measurements from Sensor_Resources.

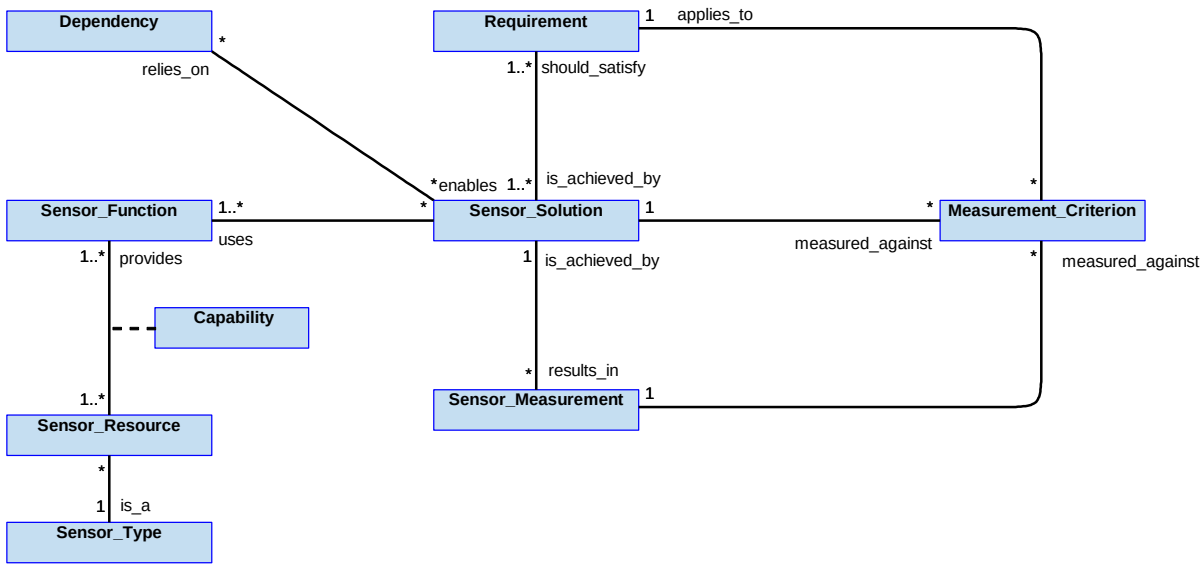

Figure 981: Sensors Semantics

B.2.54.5.1 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).

B.2.54.6 Design Rationale

B.2.54.6.1 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 policy.
- A Sensor_Resource provides a measurement of a physical characteristic (e.g. EM spectrum, acoustics or optical).

B.2.54.6.2 Design Considerations

Directly Applicable Policies

These policies 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 direct (i.e. not via a service interface), therefore will 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 policy.)
- 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.

B.2.54.6.3 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.

B.2.54.6.4 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.

B.2.54.7 Services

B.2.54.7.1 Service Definitions

B.2.54.7.1.1 Sensor_Requirement

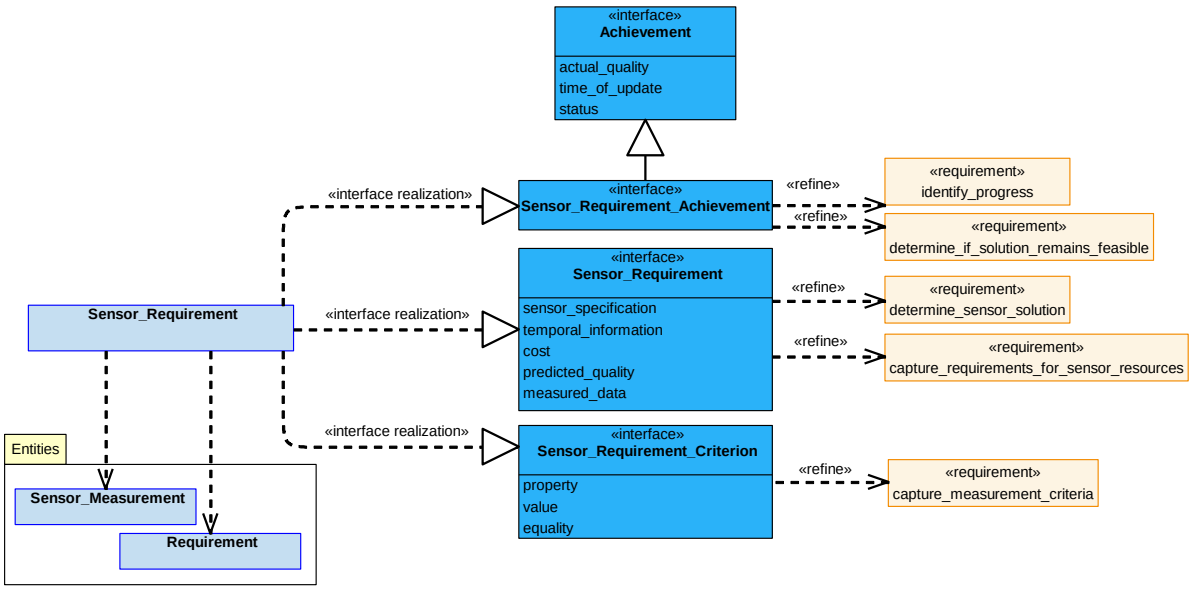

Figure 982: Sensor_Requirement Service Definition

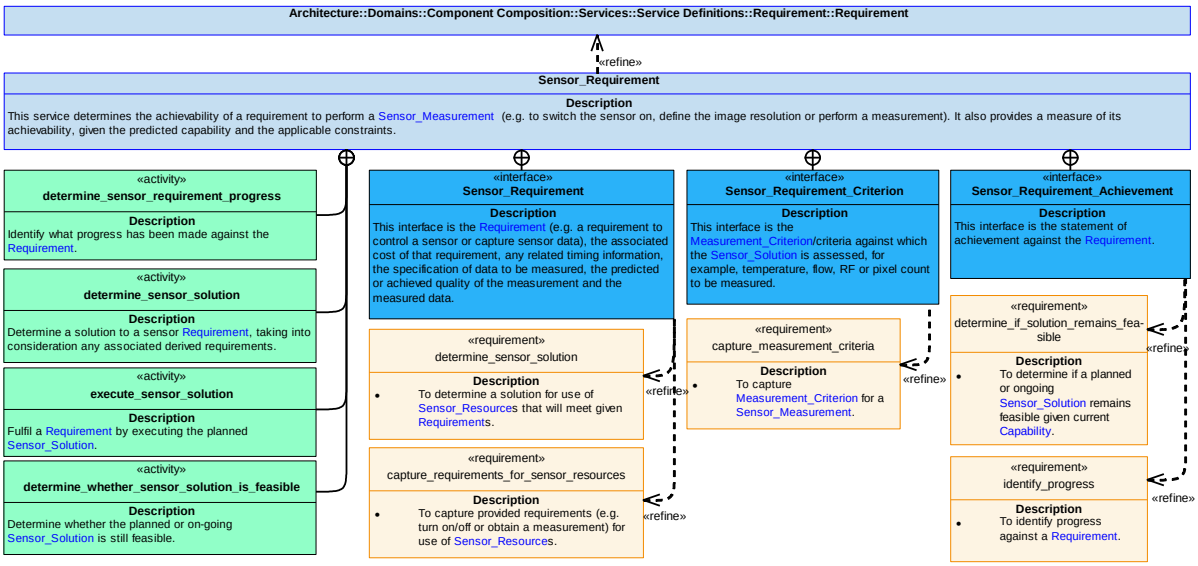

Figure 983: Sensor_Requirement Service Policy

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.

B.2.54.7.1.2 Sensor_Resourcing

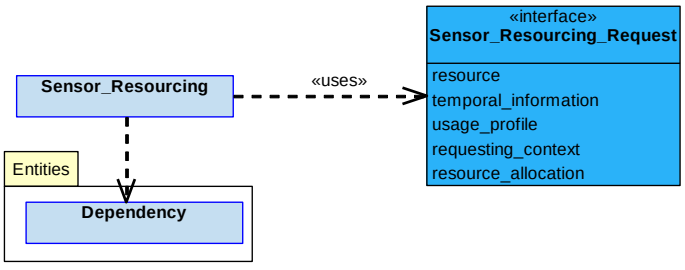

Figure 984: Sensor_Resourcing Service Definition

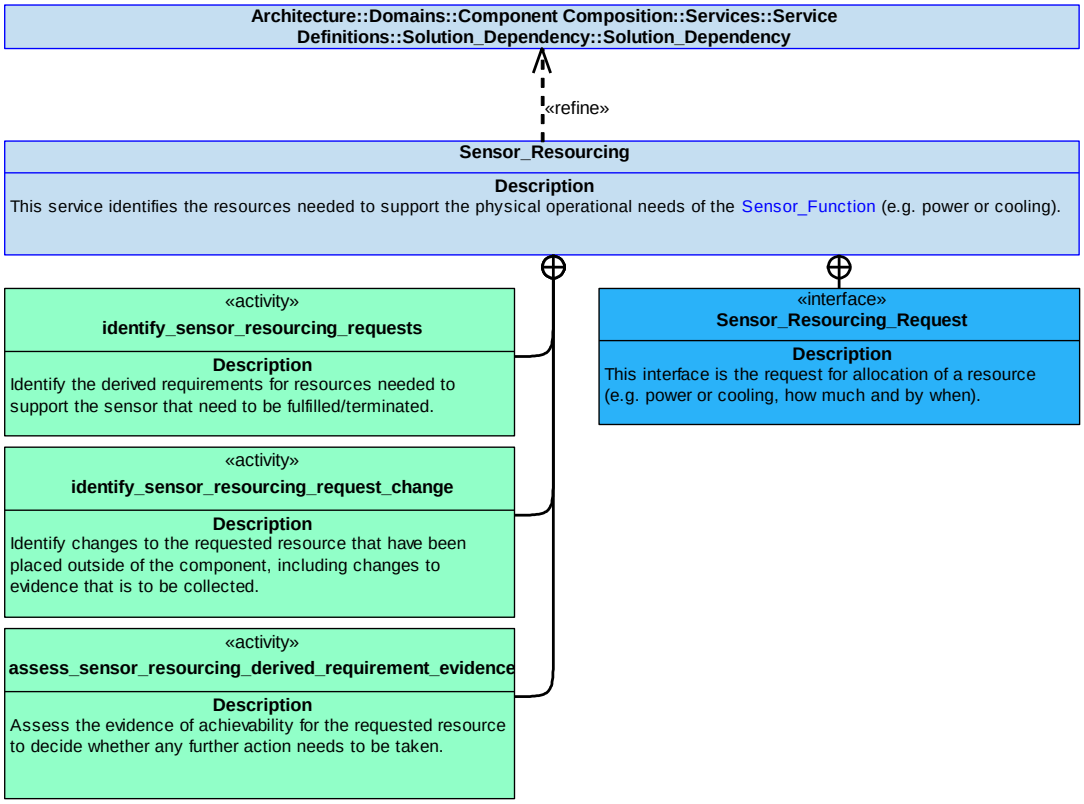

Figure 985: Sensor_Resourcing Service Policy

Sensor_Resourcing

This service identifies the resources needed to support the physical operational needs of the Sensor_Function (e.g. power or cooling).

Interface

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.

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.

B.2.54.7.1.3 Sensor_Orientation_and_Location

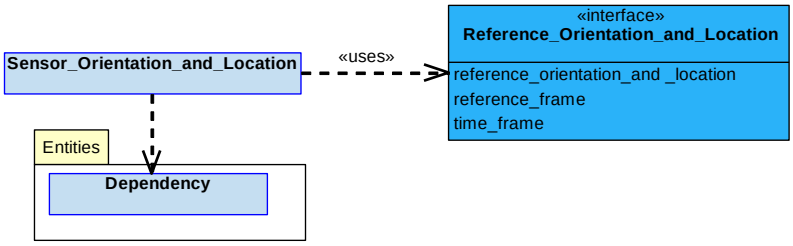

Figure 986: Sensor_Orientation_and_Location Service Definition

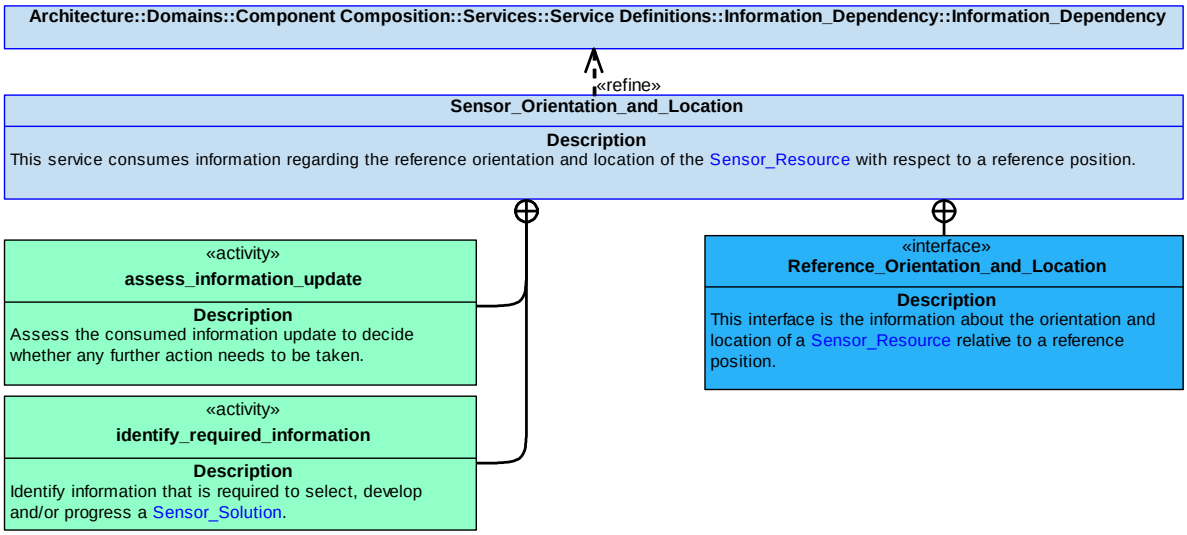

Figure 987: Sensor_Orientation_and_Location Service Policy

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.

B.2.54.7.1.4 Sensor_Capability

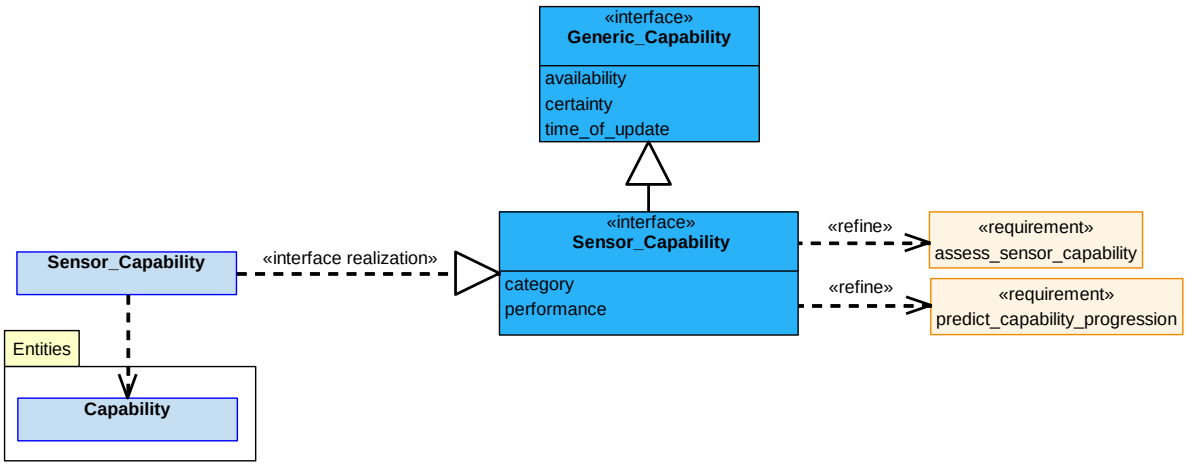

Figure 988: Sensor_Capability Service Definition

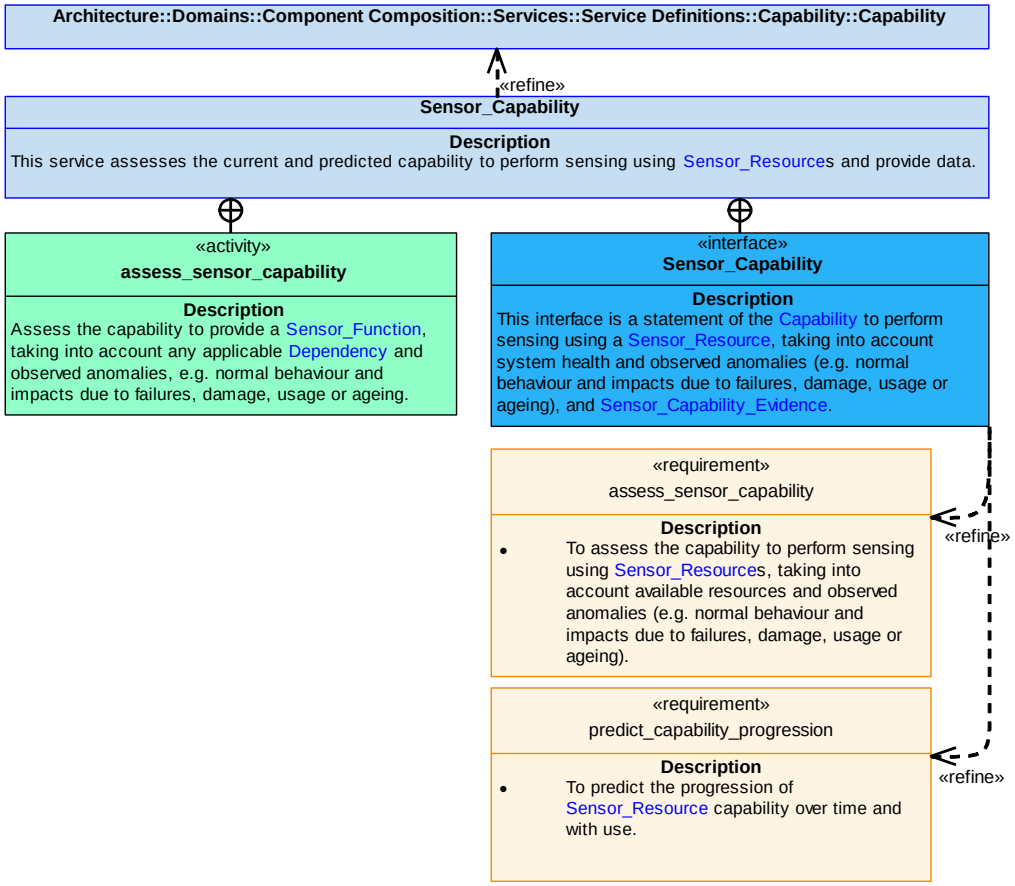

Figure 989: Sensor_Capability Service Policy

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), and Sensor_Capability_Evidence.

Attributes

category	The type of Sensor_Function that is being provided (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.

B.2.54.7.1.5 Sensor_Capability_Evidence

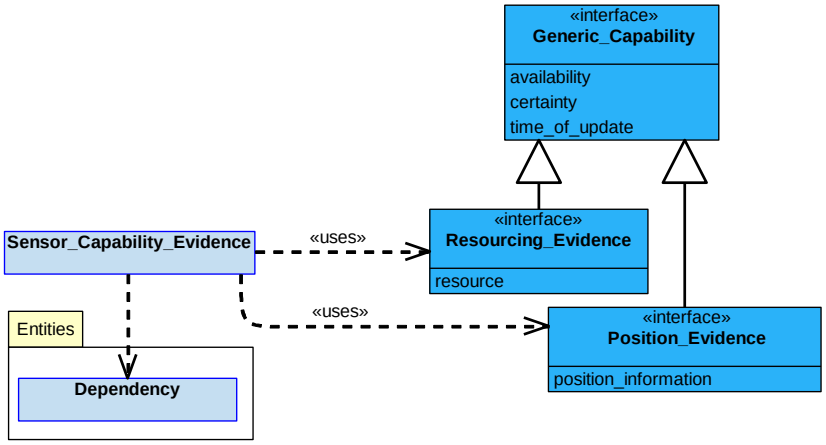

Figure 990: Sensor_Capability_Evidence Service Definition

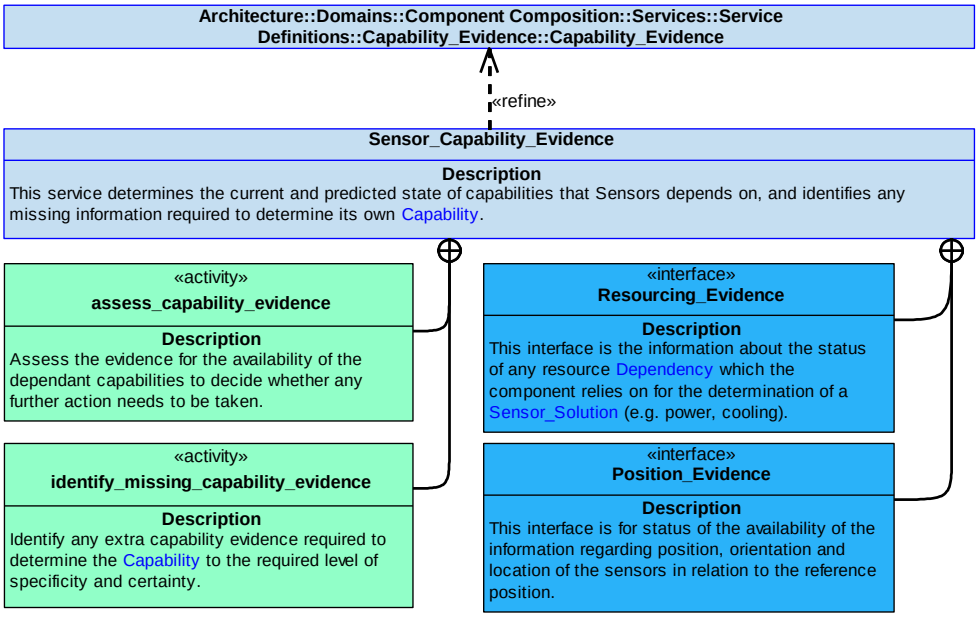

Figure 991: Sensor_Capability_Evidence Service Policy

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, 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.

B.2.54.7.2 Service Dependencies

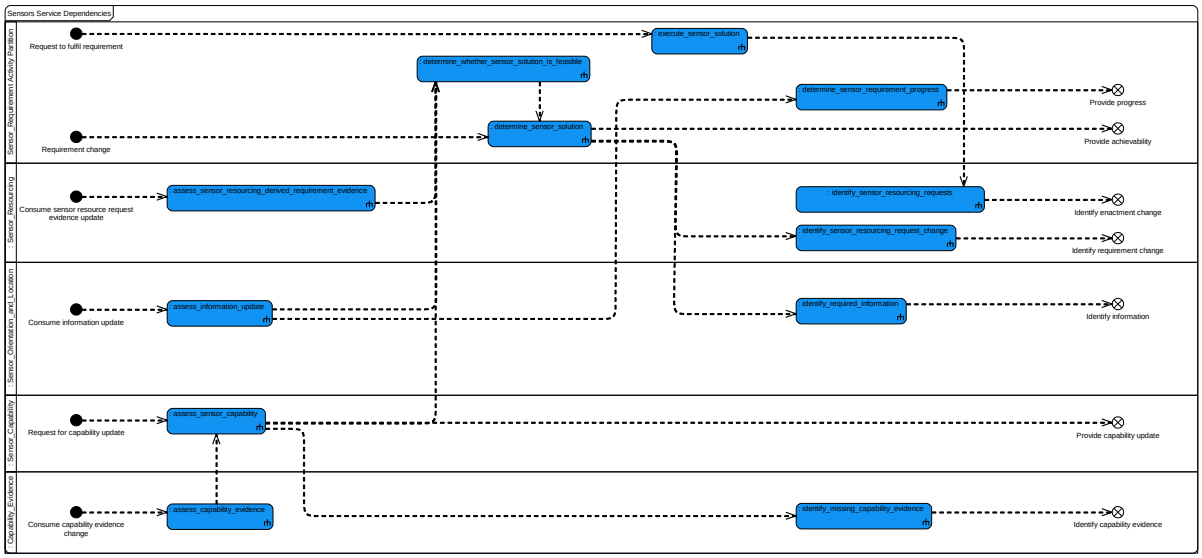

Figure 992: Sensors Service Dependencies

B.2.55 Signature

B.2.55.1 Role

The role of Signature is to calculate or estimate the signature of an object(s), whether for itself or an external entity.

B.2.55.2 Overview

Control Architecture

Signature is a service component as defined in the Control Architecture policy.

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.

B.2.55.3 Service Summary

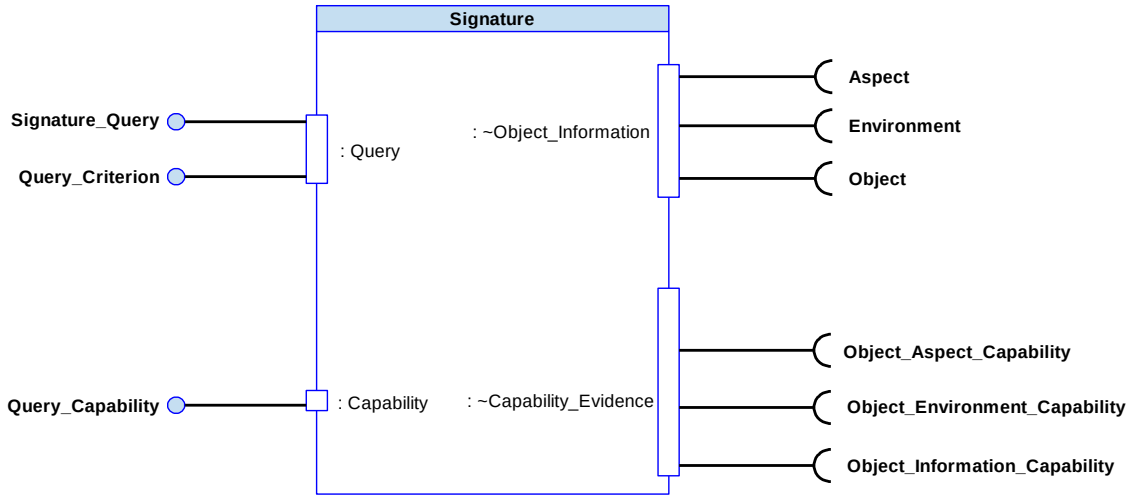

Figure 993: Signature Service Summary

B.2.55.4 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.

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.

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.

predict_capability_progression

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

determine_configuration

- To determine a State that satisfies a required Signature level.

B.2.55.5 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.

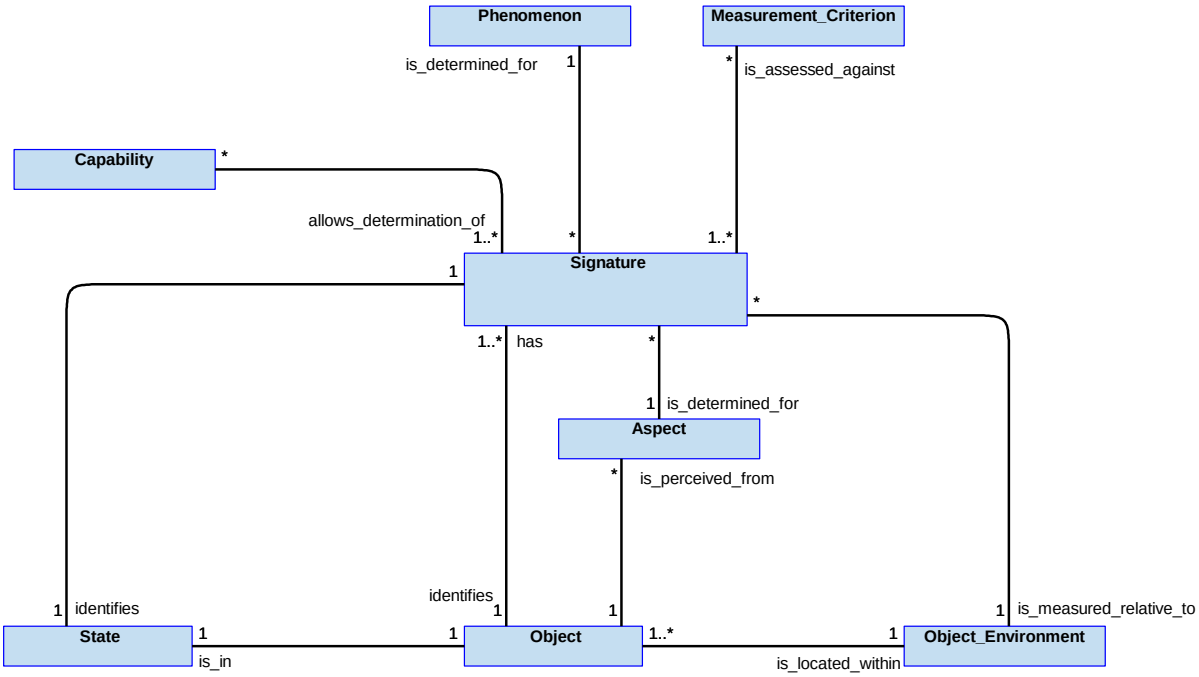

Figure 994: Signature Semantics

B.2.55.5.1 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).

B.2.55.6 Design Rationale

B.2.55.6.1 Assumptions

- The observability of objects will be determined by another component.

B.2.55.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.55.6.3 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.

B.2.55.6.4 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.

B.2.55.7 Services

B.2.55.7.1 Service Definitions

B.2.55.7.1.1 Query

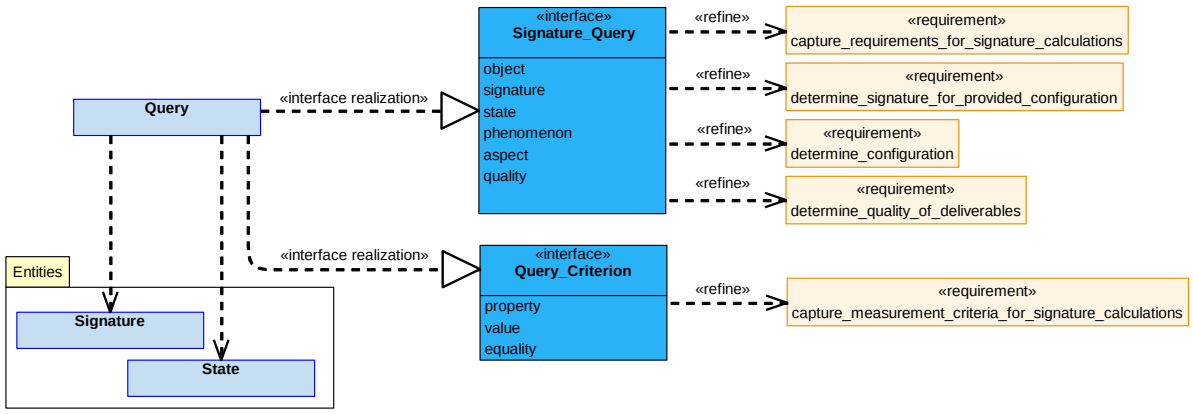

Figure 995: Query Service Definition

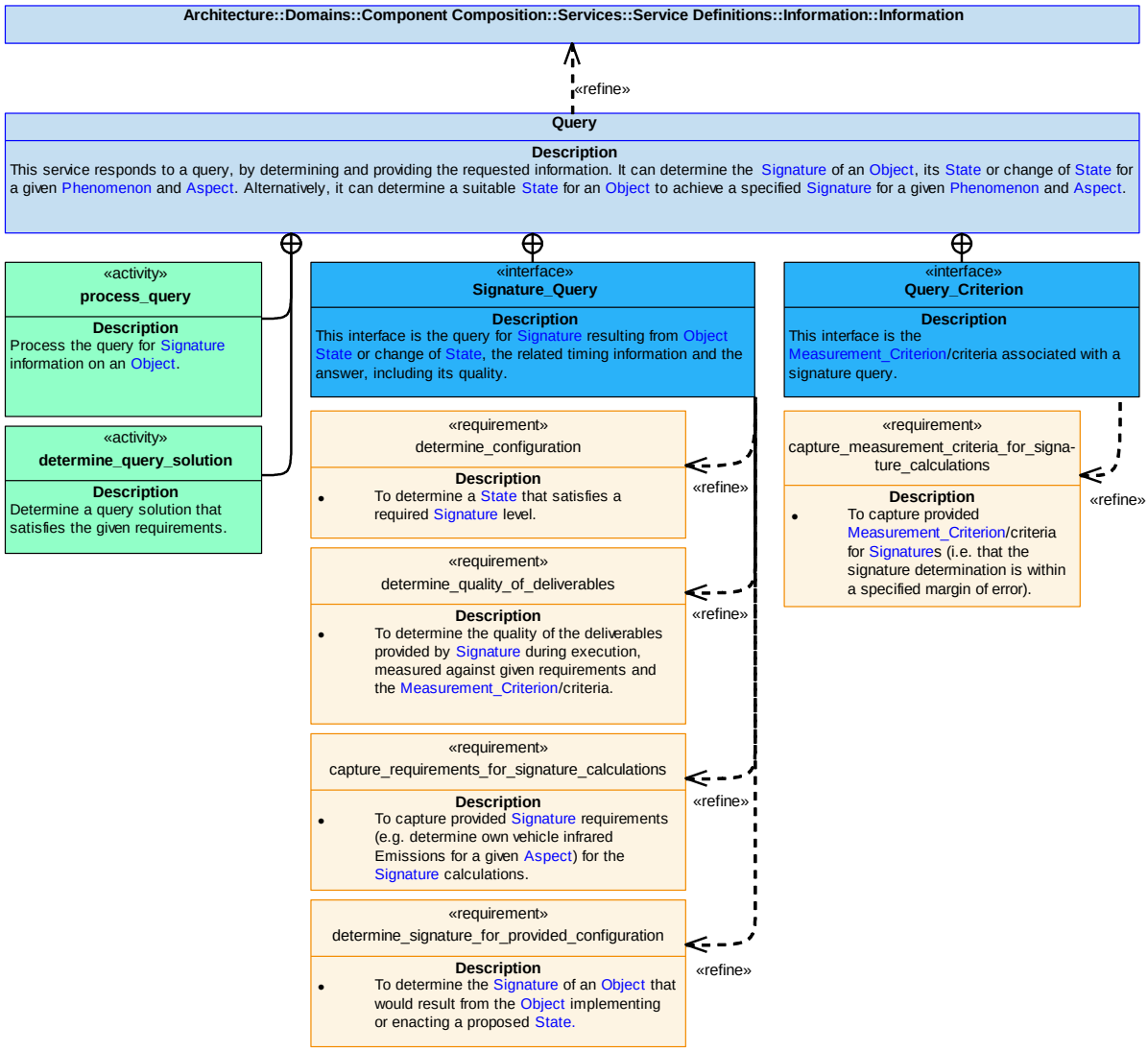

Figure 996: Query Service Policy

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.

B.2.55.7.1.2 Object_Information

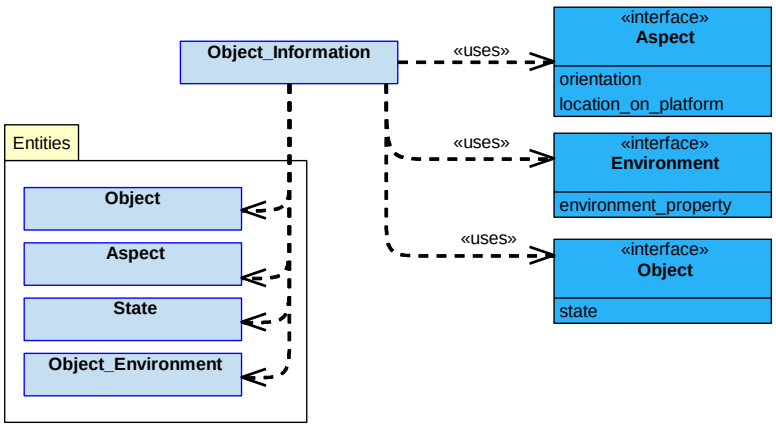

Figure 997: Object_Information Service Definition

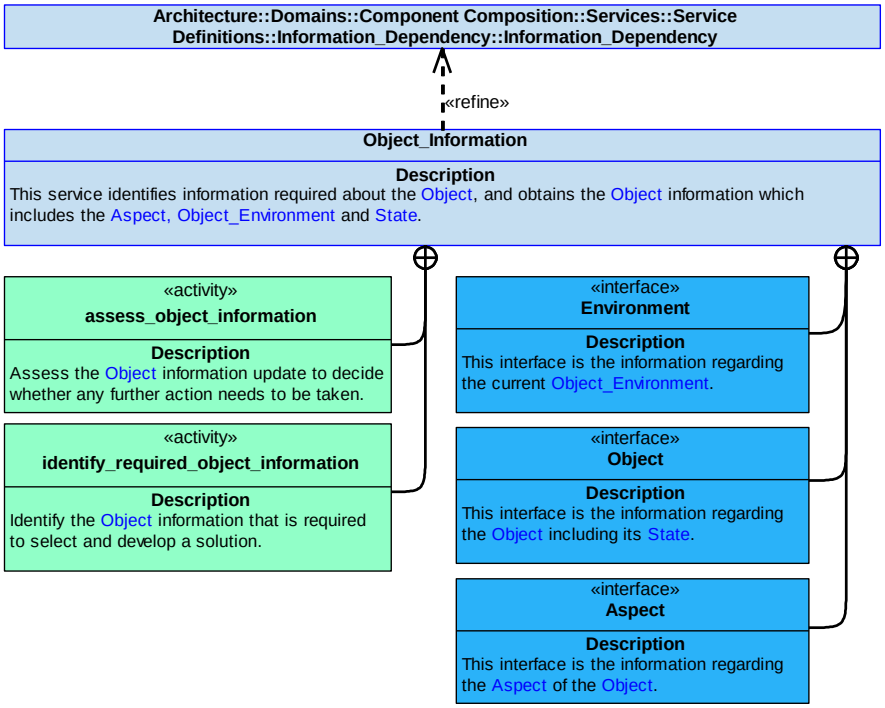

Figure 998: Object Information Service Policy

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.

B.2.55.7.1.3 Capability

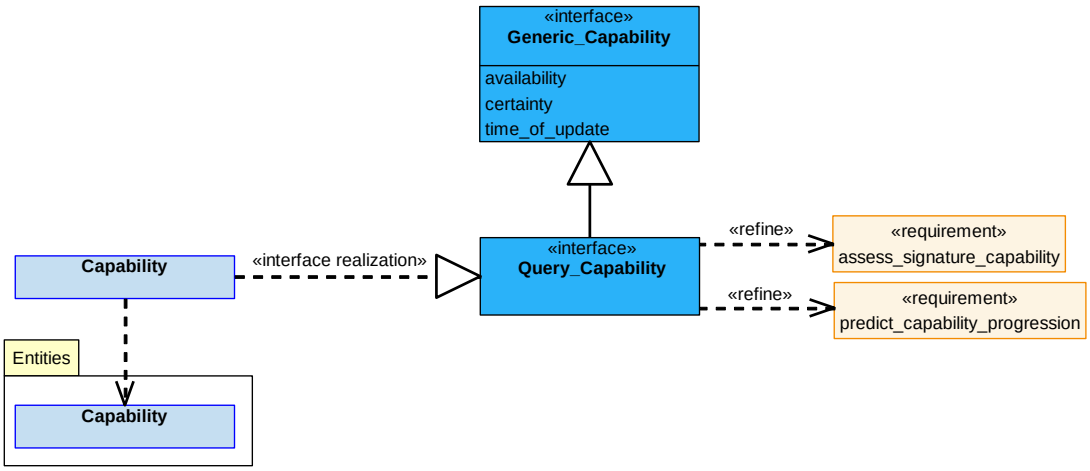

Figure 999: Capability Service Definition

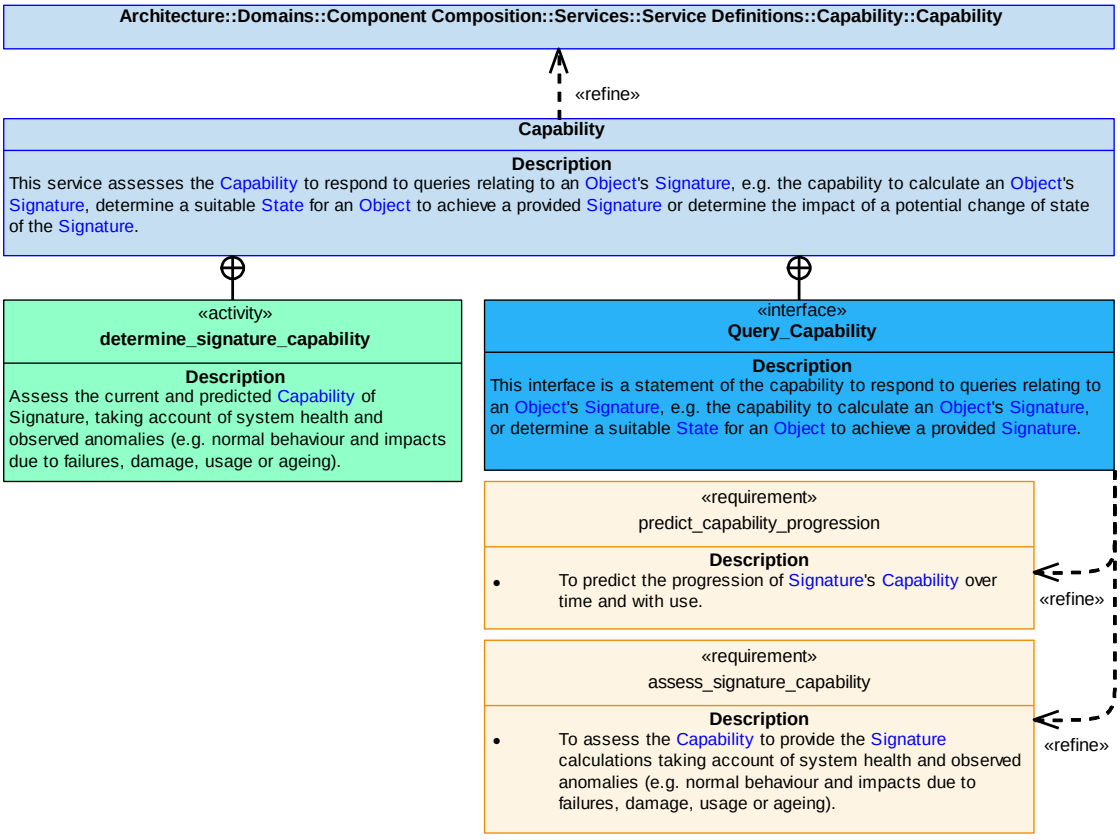

Figure 1000: Capability Service Policy

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).

B.2.55.7.1.4 Capability_Evidence

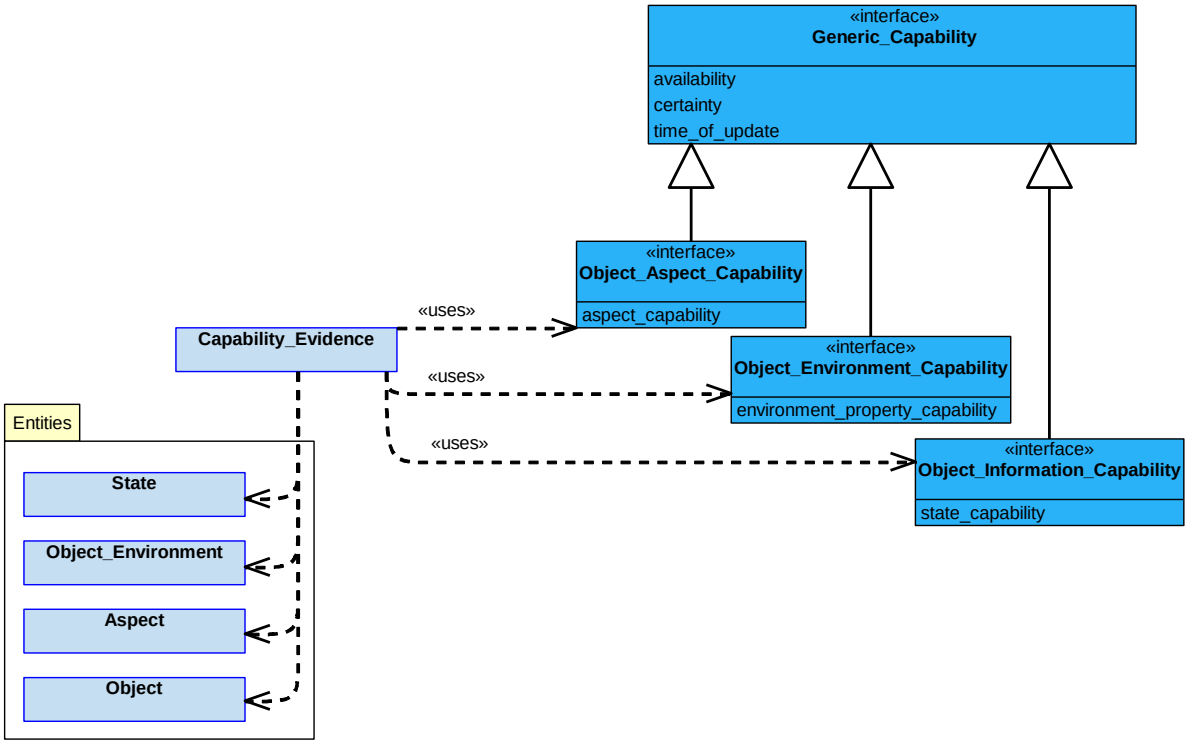

Figure 1001: Capability_Evidence Service Definition

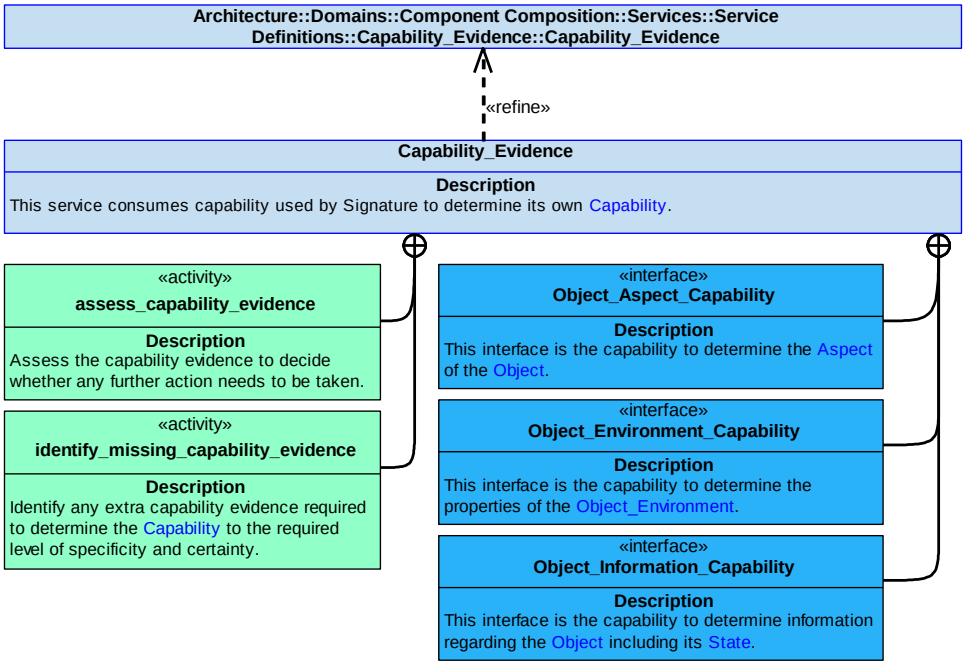

Figure 1002: Capability Evidence Service Policy

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.

B.2.55.7.2 Service Dependencies

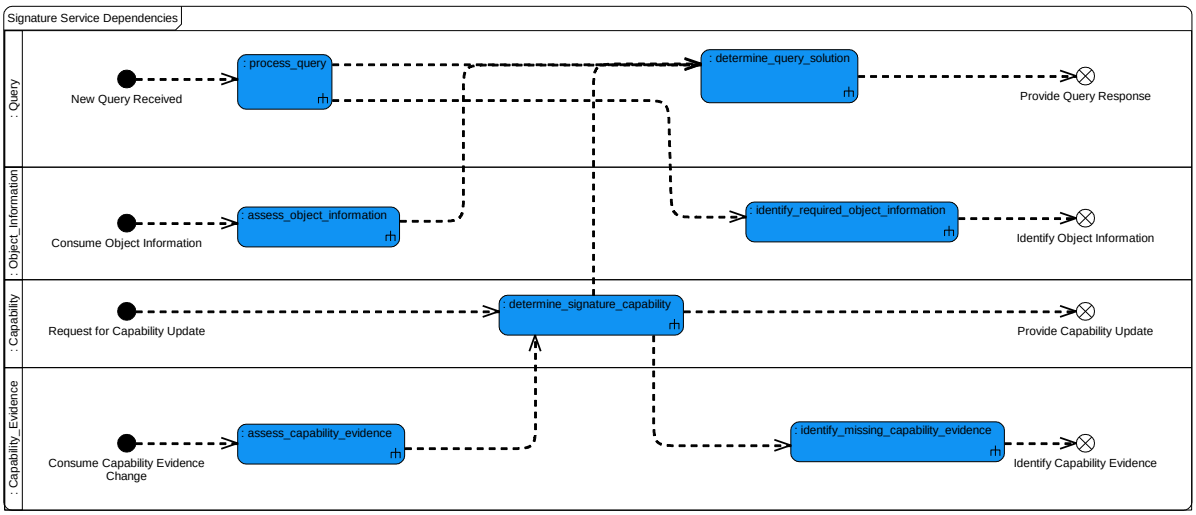

Figure 1003: Signature Service Dependencies

B.2.56 Spatial Correction

B.2.56.1 Role

The role of Spatial Correction is to determine the correction to the spatial relationship between different positions to account for physical effects.

B.2.56.2 Overview

Control Architecture

Spatial Correction is a service component as defined in the Control Architecture policy.

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.

B.2.56.3 Service Summary

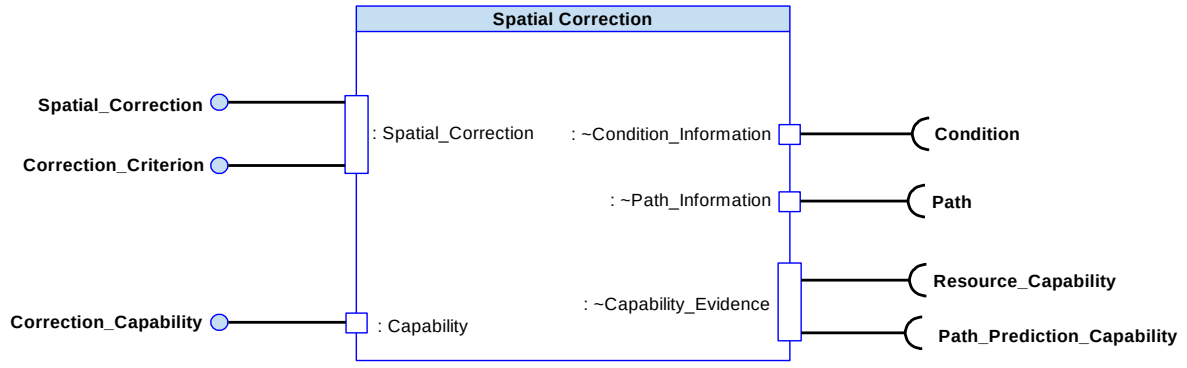

Figure 1004: Spatial Correction Service Summary

B.2.56.4 Responsibilities

capture_requirements_for_spatial_correction

- To capture provided Requirements for calculating Spatial_Corrections.

determine_spatial_correction

- To determine the Spatial_Corrections resulting from a Physical_Effect.

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).

predict_capability_progression

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

identify_missing_information

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

capture_measurement_criteria

- To capture given Measurement_Criterion.

determine_quality_of_spatial_correction

- To determine the quality of the Spatial_Correction against the Measurement_Criterion.

B.2.56.5 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).

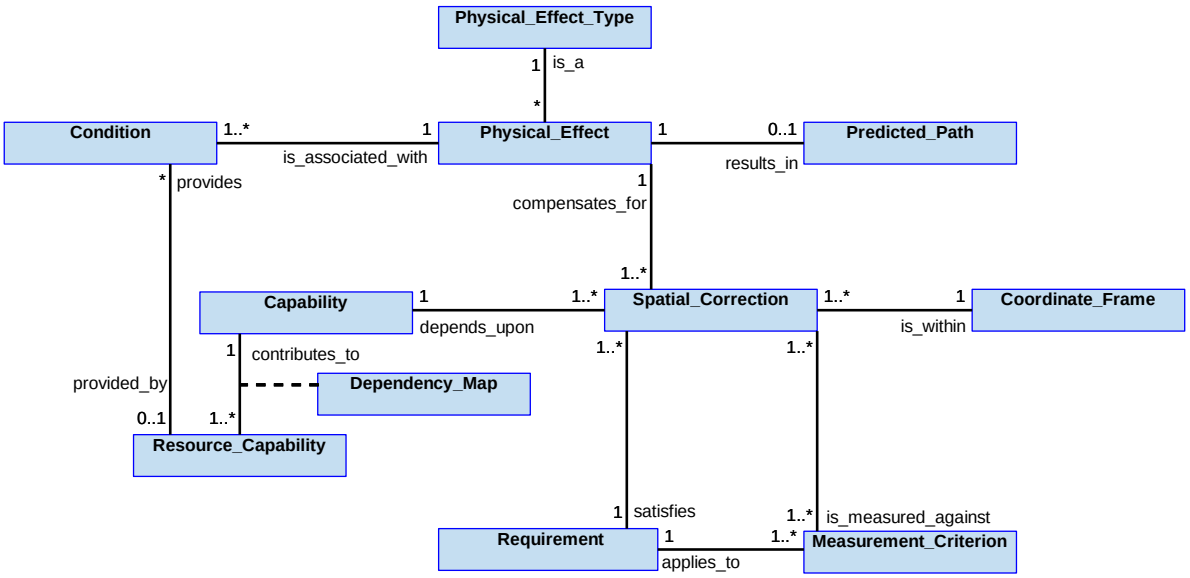

Figure 1005: Spatial Correction Semantics

B.2.56.5.1 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.

B.2.56.6 Design Rationale

B.2.56.6.1 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.

B.2.56.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.56.6.3 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 be 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.

B.2.56.6.4 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.

B.2.56.7 Services

B.2.56.7.1 Service Definitions

B.2.56.7.1.1 Spatial_Correction

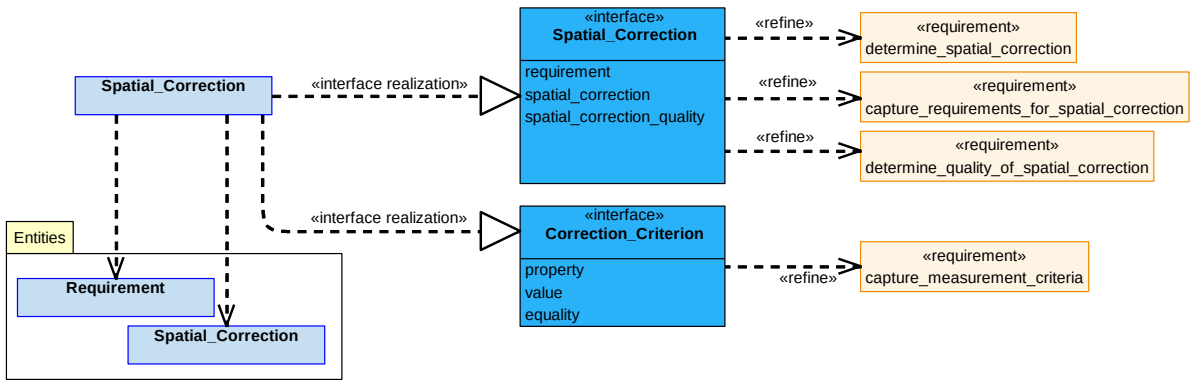

Figure 1006: Spatial_Correction Service Definition

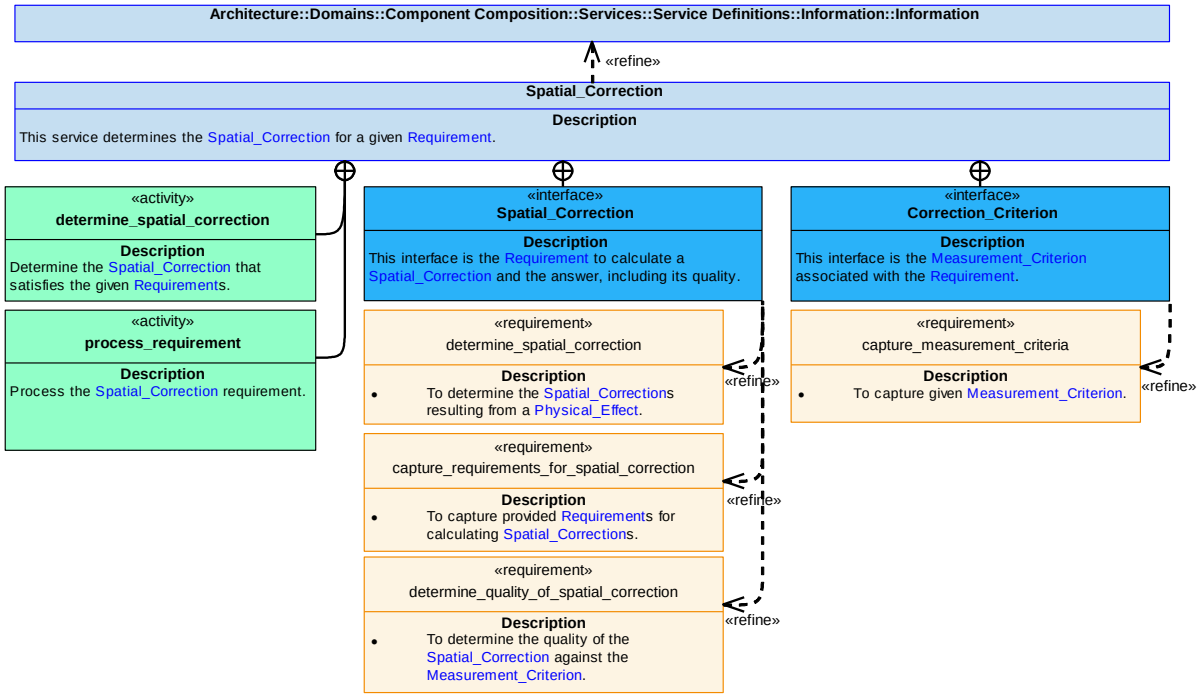

Figure 1007: Spatial_Correction Service Policy

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.

B.2.56.7.1.2 Condition_Information

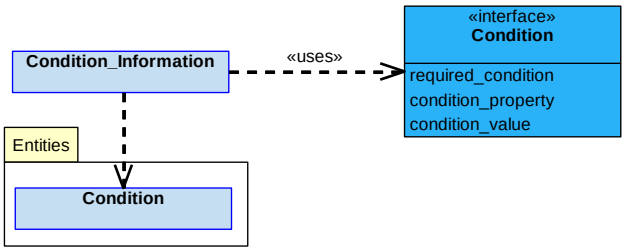

Figure 1008: Condition_Information Service Definition

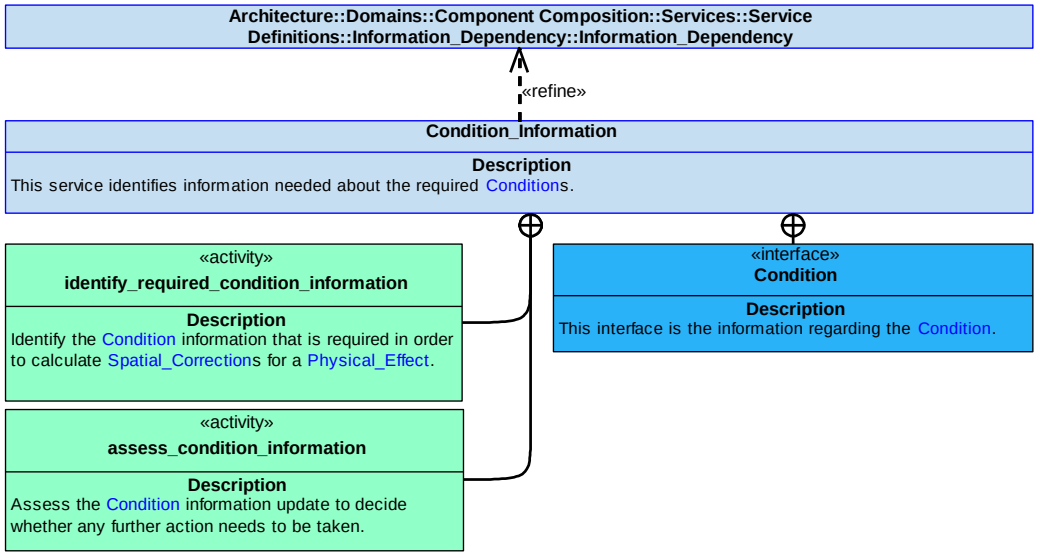

Figure 1009: Condition_Information Service Policy

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.

B.2.56.7.1.3 Path_Information

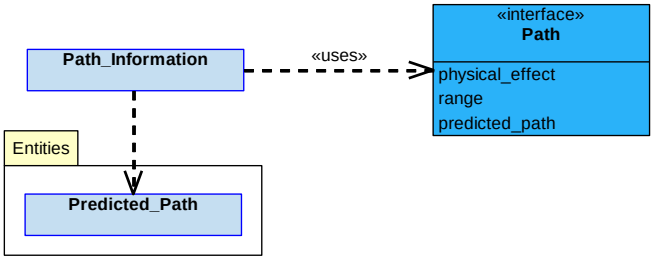

Figure 1010: Path_Information Service Definition

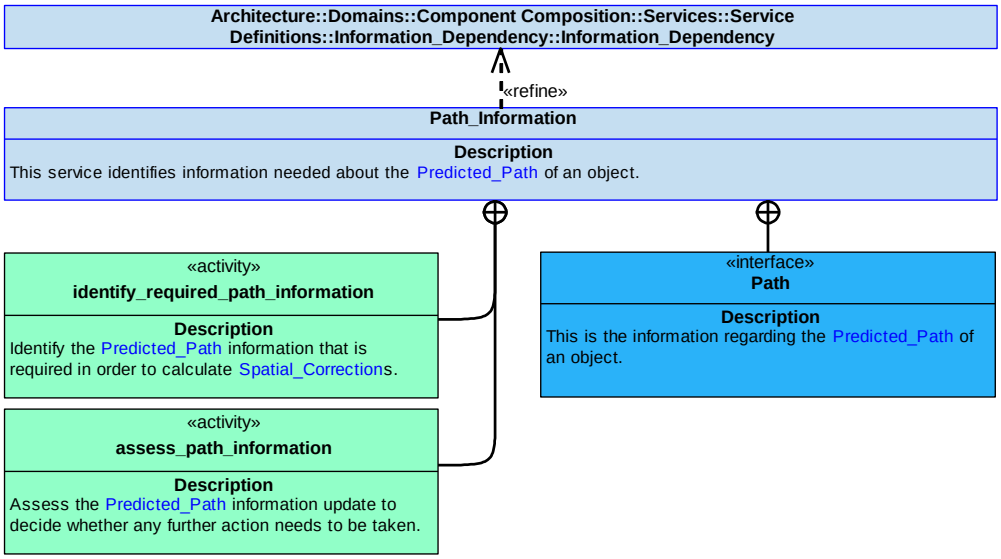

Figure 1011: Path_Information Service Policy

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.

B.2.56.7.1.4 Capability

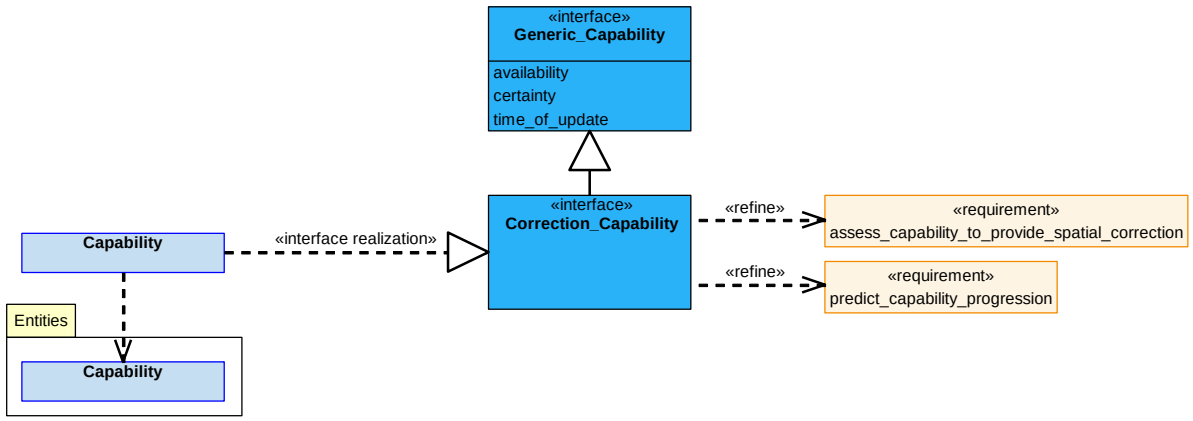

Figure 1012: Capability Service Definition

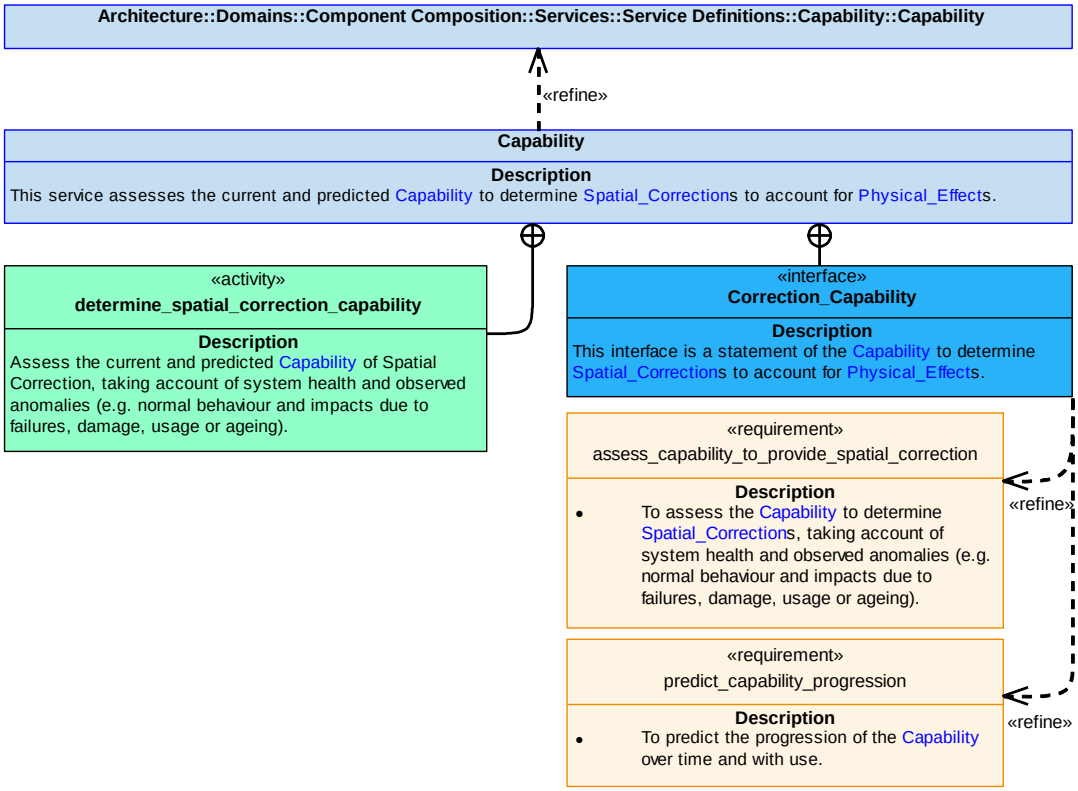

Figure 1013: Capability Service Policy

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).

B.2.56.7.1.5 Capability_Evidence

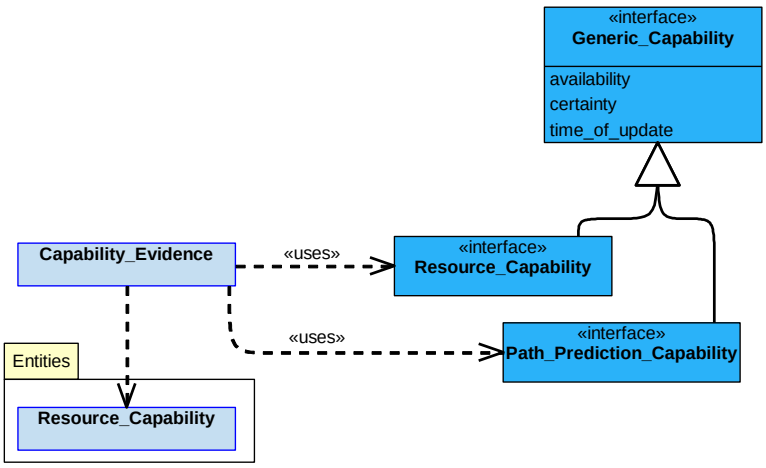

Figure 1014: Capability_Evidence Service Definition

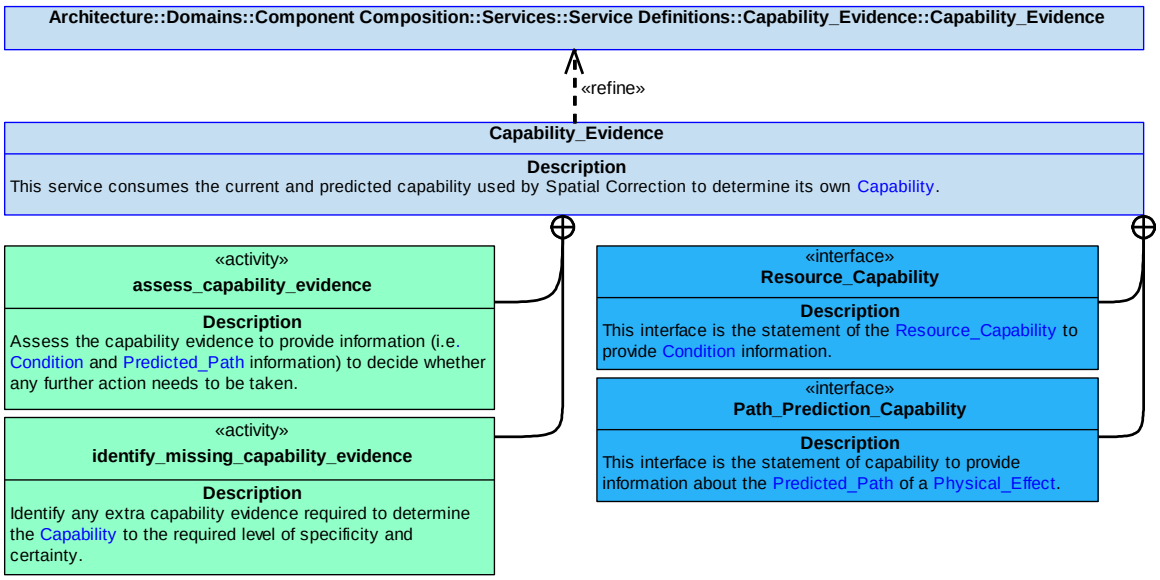

Figure 1015: Capability_Evidence Service Policy

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.

Activities

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.

B.2.56.7.2 Service Dependencies

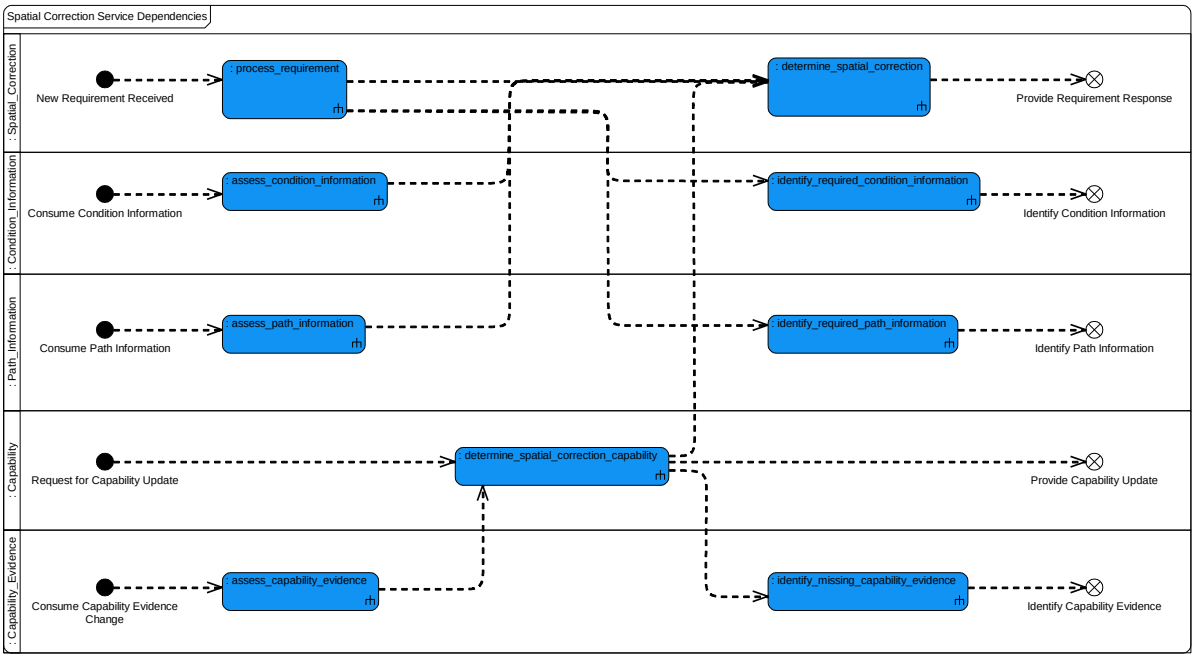

Figure 1016: Spatial Correction Service Dependencies

B.2.57 Spectrum

B.2.57.1 Role

The role of Spectrum is to allocate spectrum to meet current and future demands, and to identify and resolve conflicts for spectrum use (e.g. electromagnetic and acoustic).

B.2.57.2 Overview

Control Architecture

Spectrum is a service component as defined in the Control Architecture policy.

Standard Pattern of Use

Spectrum is used to coordinate the use of spectrum resources (e.g. 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 maximum power level). Spectrum will manage the allocation based upon existing constraints and current usage of the spectrum by physical hardware.

Examples of Use

- Prevent multiple devices using the same electromagnetic frequency.
- Prevent interference between equipment using frequencies which are liable to interact.

B.2.57.3 Service Summary

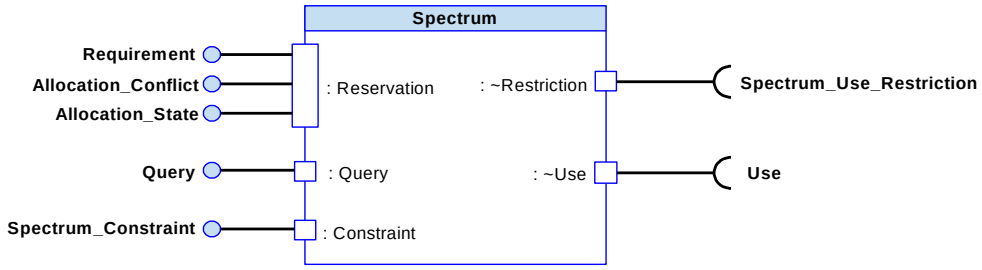

Figure 1017: Spectrum Service Summary

B.2.57.4 Responsibilities

capture_requirements

- To capture Requirements to use parts of a Spectrum.

capture_constraints

- To capture provided Spectrum_Restrictions (e.g. EMCON).

determine_available_spectrum

- To determine parts of a Spectrum available for use.

determine_spectrum_in_use

- To determine Spectrum_Element_Allocation currently in use, or planned to be used.

determine_permitted_power_level

- To determine permitted transmitted power level for a particular Spectrum_Element_Allocation.

determine_solutions_to_conflicts

- To determine a resolution to spectrum use request conflicts.

deliver_spectrum_service

- To manage Spectrum_Element_Allocation for emitters and receivers to meet the interoperability requirements within the constraints.

identify_solution_remains_achievable

- To identify whether a Spectrum_Element_Allocation currently in progress remains feasible given current resources.

B.2.57.5 Subject Matter Semantics

The subject matter of Spectrum is the use of the electro-magnetic and acoustic spectrum.

Exclusions

The subject matter of Spectrum does not include:

- Ensuring that spectrum users stick to their allotted frequency ranges.
- Observability between platforms.
- Spatial deconfliction relative to a movable object or direction control (e.g. keeping pointed at a satellite).

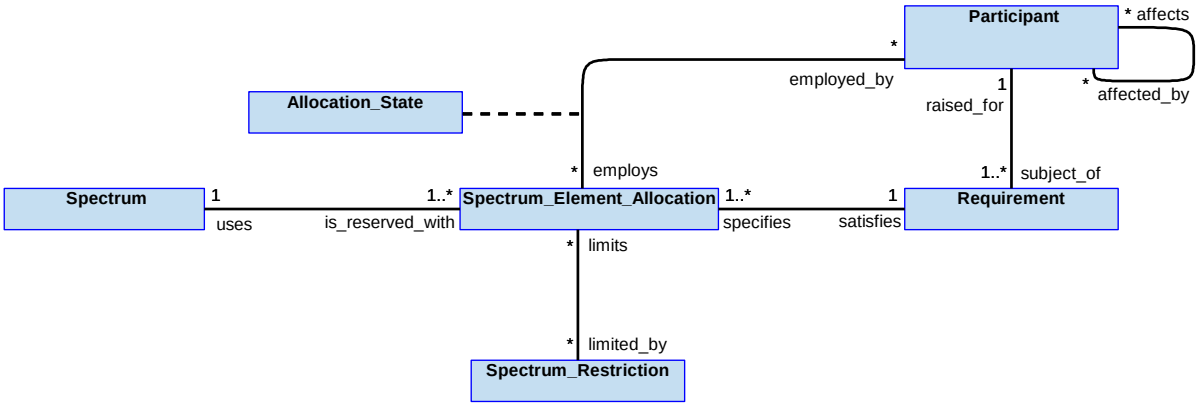

Figure 1018: Spectrum Semantics

B.2.57.5.1 Entities

Participant

A unique recipient or generator of energy. Note: a participant may represent a use of spectrum that is outside the control of this component.

Requirement

The need for use of spectrum. This may include a start and stop time, a direction, an assigned priority in the case of conflict resolution, a frequency range and whether it is a requirement for reception or transmission.

Spectrum_Element_Allocation

The properties that relate to an allocation of the spectrum. This may notionally consist of power, direction, frequency range and time period.

Spectrum

A portion of the spectrum that is being monitored and/or managed (electromagnetic or acoustic).

Spectrum_Restriction

A constraint, preference or allocation rule that limits the allocation of certain parts of the spectrum.

Allocation_State

The Participant's relationship to the Spectrum_Element_Allocation. For example, the allocation can be reserved or allocated to a Participant; it may be in-use, independent of whether it has been allocated or not. It may also be blocked by a higher priority task that is using the same allocation.

B.2.57.6 Design Rationale

B.2.57.6.1 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.
- This component is not responsible for preventing high power transmitters causing catastrophic accidents (e.g. if operated in the wrong circumstances such as near ground crew or another air vehicle).
- Not all uses of Spectrum will be under the control of this component.

B.2.57.6.2 Design Considerations

Directly Applicable Policies

These policies 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 the spectrum resources in alignment with this policy.

Note that the Spectrum component's capability is fixed at design time and is not dependent on consumed capability evidence and does not evolve with time, and this component is not concerned with the provided capability of any particular subject matter area hence does not determine capability in the way described in the Capability Assessment policy.

Extensions

- Implementations of Spectrum could follow the Component Extensions policy, to accommodate a variety of emitters and sensors profiles and allocation algorithms.

Exploitation Considerations

- The approach to providing reservations needs to be considered, for example a notification could be provided when the window for use for a reservation opens (or closes), allowing a participant to claim it, or if it is not required release it so it can be used by a lower priority task.
- The Query service allows for a potential user of a spectrum to ask about a frequency range availability, to tailor its expectations before making an electro-magnetic or acoustic allocation request.
- Directionality is relative to the subject of analysis (e.g. node, antenna, or set of antenna).
- The component will cover a defined Spectrum, and it's use would be in accordance with that, and therefore capability will be binary.

B.2.57.6.3 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.

B.2.57.6.4 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 conflict resolutions 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 electro-magnetic and acoustic spectrum, for example from EMCON.

B.2.57.7 Services

B.2.57.7.1 Service Definitions

B.2.57.7.1.1 Reservation

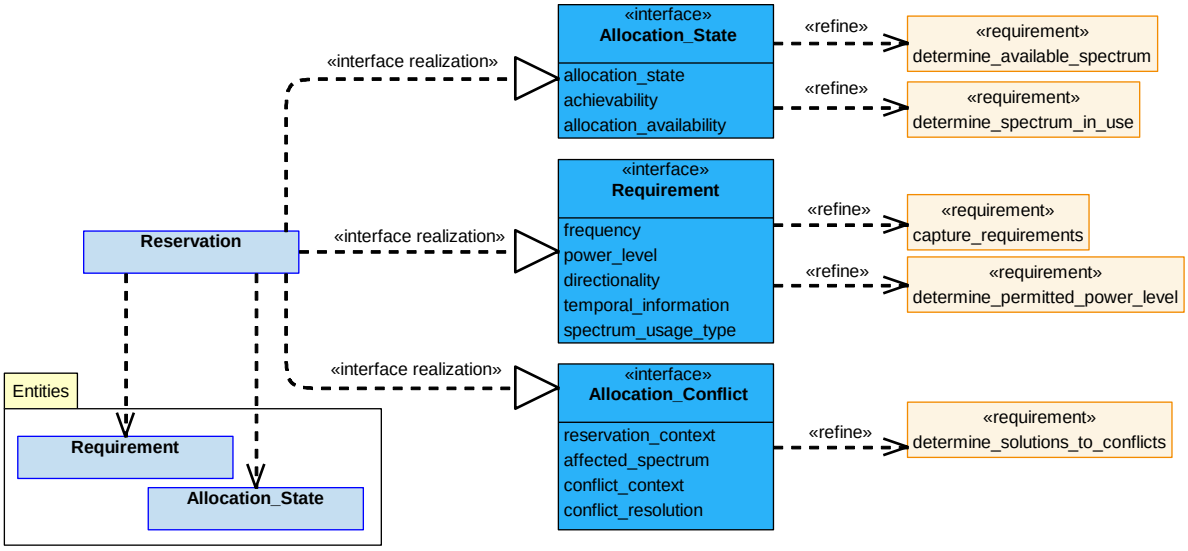

Figure 1019: Reservation Service Definition

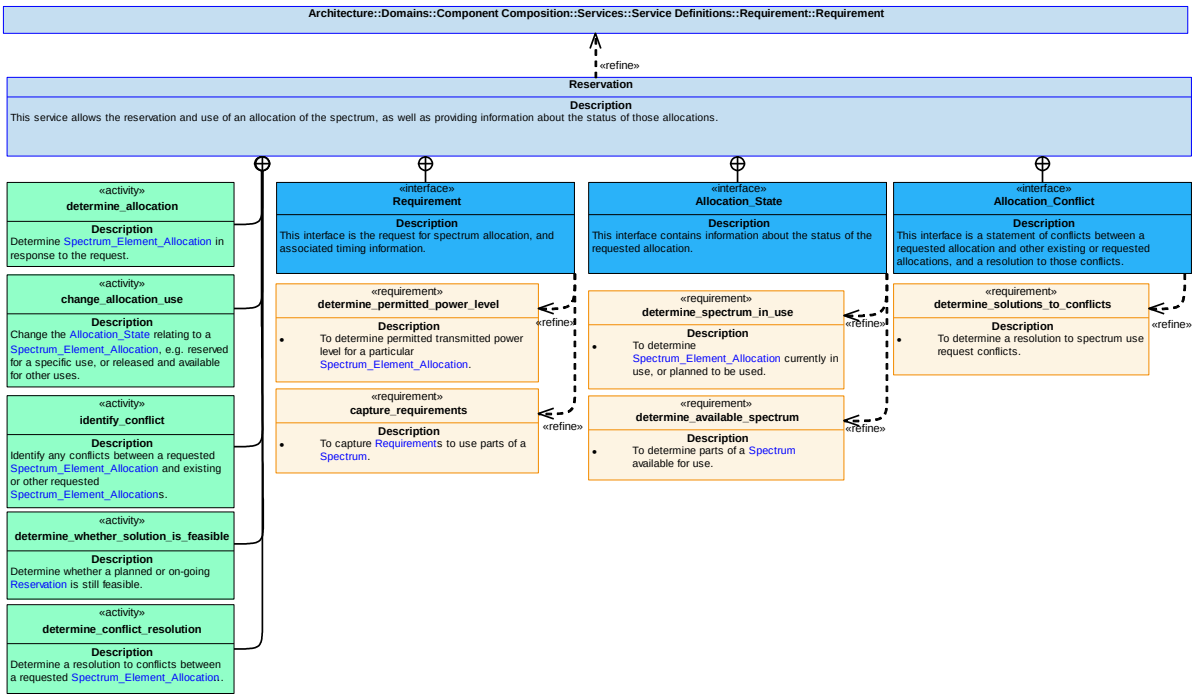

Figure 1020: Reservation Service Policy

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	This attribute describes the spectrum of interest (i.e. frequency, frequency range and tolerance).
power_level	This attribute describes the preferred power level.
directionality	This attribute describes 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 might include segments of a requested time window that must not be interrupted, etc.
spectrum_usage_type	This attribute describes 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	This attribute captures the current state of the allocation within its own lifecycle (e.g. raised, requested, acknowledged, allocated, rejected, claimed or released).
achievability	This attribute captures the achievability of a particular requirement (e.g. cannot find a resolution). The response can include conditions/constraints that are non-binary allowing for operationally acceptable consequences for the requester.
allocation_availability	This attributes describes 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.

Allocation_Conflict

This interface is a statement of conflicts between a requested allocation and other existing or requested allocations, and a resolution to those conflicts.

Attributes

reservation_context	The tasking information that identifies the source or reason for the Requirement.
affected_spectrum	The requested Spectrum_Element_Allocation affected by a conflicting Requirement.
conflict_context	This attribute identifies the Requirement source or reason for a Spectrum_Element_Allocation conflict.
conflict_resolution	The determined resolution to a conflict.

Activities

determine_allocation

Determine Spectrum_Element_Allocation in response to the request.

change_allocation_use

Change the Allocation_State relating to a Spectrum_Element_Allocation, e.g. reserved for a specific use, or released and available for other uses.

identify_conflict

Identify any conflicts between a requested Spectrum_Element_Allocation and existing or other requested Spectrum_Element_Allocations.

determine_whether_solution_is_feasible

Determine whether a planned or on-going Reservation is still feasible.

determine_conflict_resolution

Determine a resolution to conflicts between a requested Spectrum_Element_Allocation and existing or other requested Spectrum_Element_Allocations.

B.2.57.7.1.2 Restriction

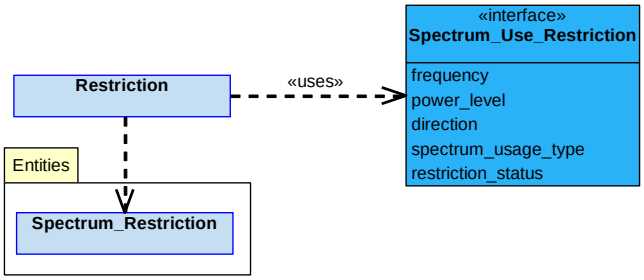

Figure 1021: Restriction Service Definition

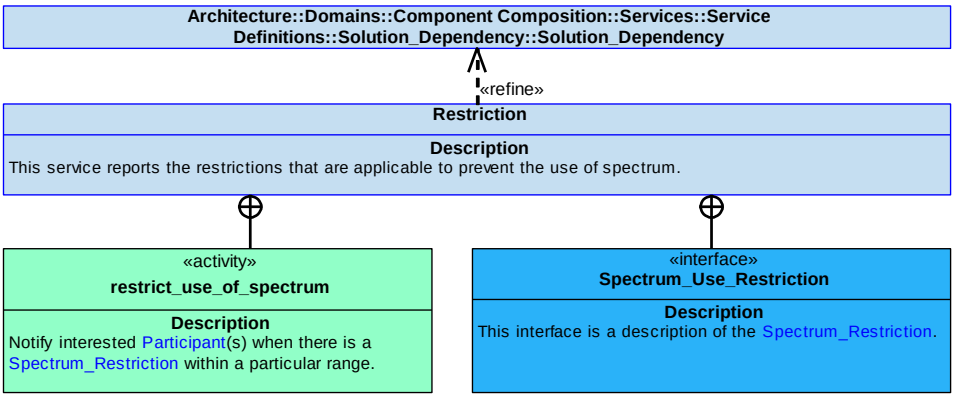

Figure 1022: Restriction Service Policy

Restriction

This service reports the restrictions that are applicable to prevent the use of spectrum.

Interface

Spectrum_Use_Restriction

This interface is a description of the Spectrum_Restriction.

Attributes

frequency	The spectrum restriction of interest (e.g. frequency and frequency range).
power_level	The restriction on the usable power level or levels.
direction	The restriction on direction relative to the platform, i.e. portside 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_status	The status of a Spectrum_Restriction, such as whether it is currently applicable and when the restriction is lifted.

Activity

restrict_use_of_spectrum

Notify interested Participant(s) when there is a Spectrum_Restriction within a particular range.

B.2.57.7.1.3 Query

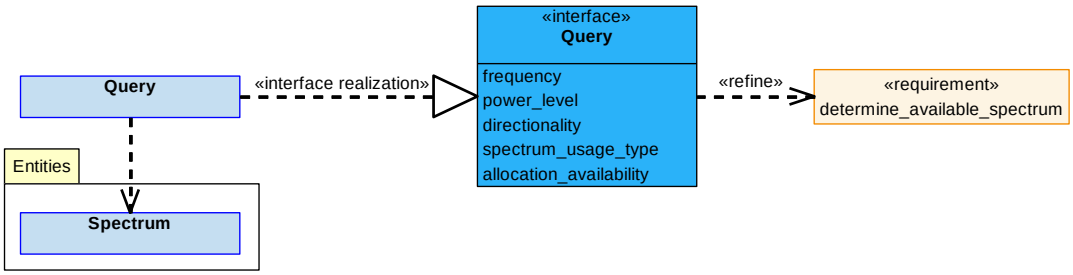

Figure 1023: Query Service Definition

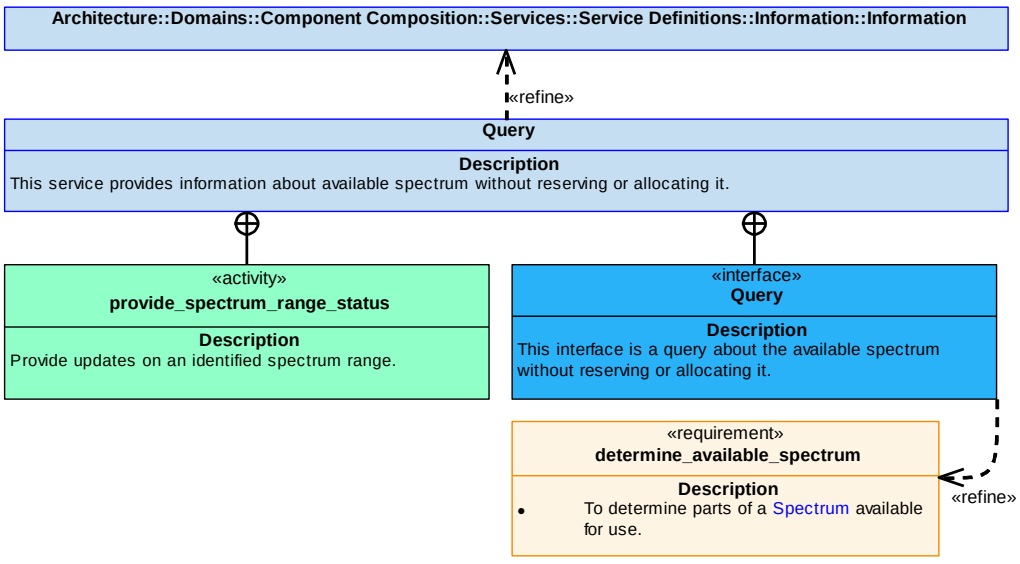

Figure 1024: Query Service Policy

Query

This service provides information about available spectrum without reserving or allocating it.

Interface

Query

This interface is a query about the available spectrum without reserving or allocating it.

Attributes

frequency	This attribute describes the spectrum of interest (i.e. frequency, frequency range and tolerance).
power_level	This attribute describes the power level.
directionality	This attribute describes the direction and spread. For example a tight beam pattern, in a specified direction from the source.
spectrum_usage_type	This attribute describes if the usage is for transmitting, receiving or both.
allocation_availability	This attributes describes 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.

Activity

provide_spectrum_range_status

Provide updates on an identified spectrum range.

B.2.57.7.1.4 Use

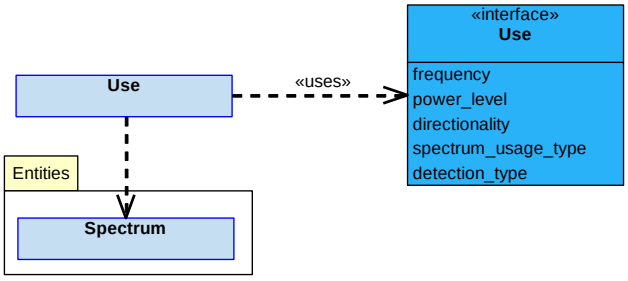

Figure 1025: Use Service Definition

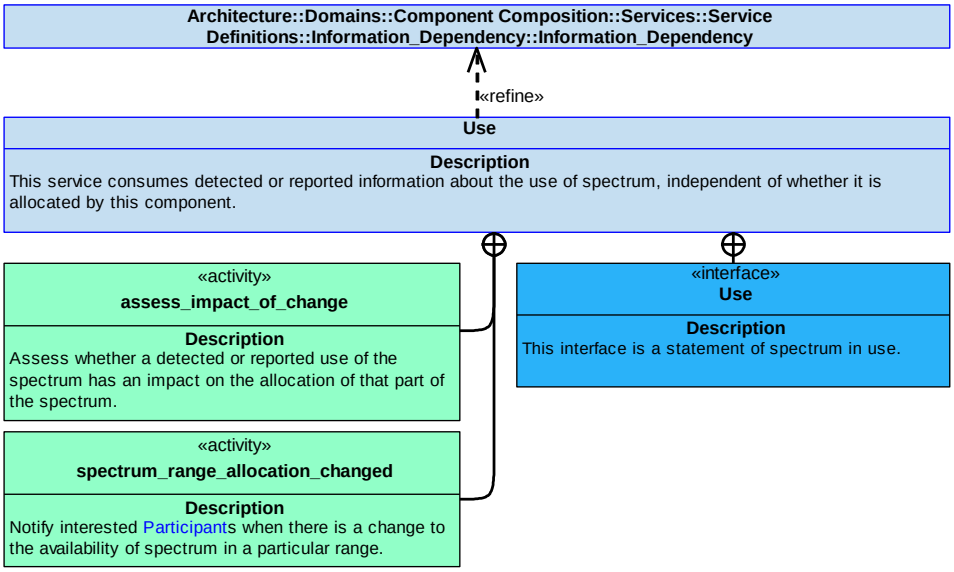

Figure 1026: Use Service Policy

Use

This service consumes detected or reported information about the use of spectrum, independent of whether it is allocated by this component.

Interface

Use

This interface is a statement of spectrum in use.

Attributes

frequency	This attribute describes the spectrum of interest (i.e. frequency, frequency range and tolerance).
power_level	This attribute describes the power level.
directionality	This attribute describes the direction and spread. For example a tight beam pattern, in a specified direction from the source.
spectrum_usage_type	This attribute describes if the usage is for transmitting, receiving or both.
detection_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).

Activities

assess_impact_of_change

Assess whether a detected or reported use of the spectrum has an impact on the allocation of that part of the spectrum.

spectrum_range_allocation_changed

Notify interested Participants when there is a change to the availability of spectrum in a particular range.

B.2.57.7.1.5 Constraint

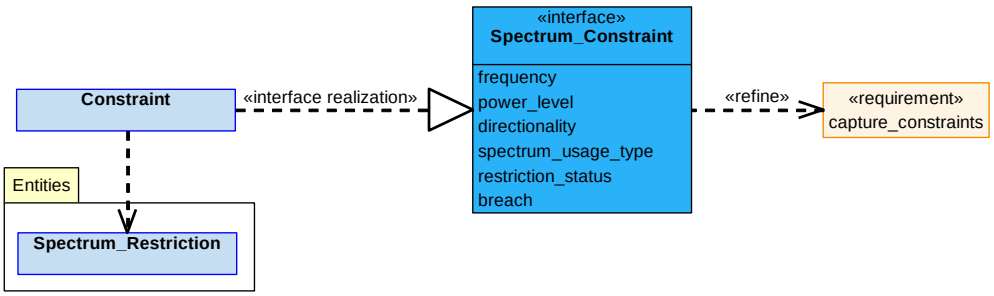

Figure 1027: Constraint Service Definition

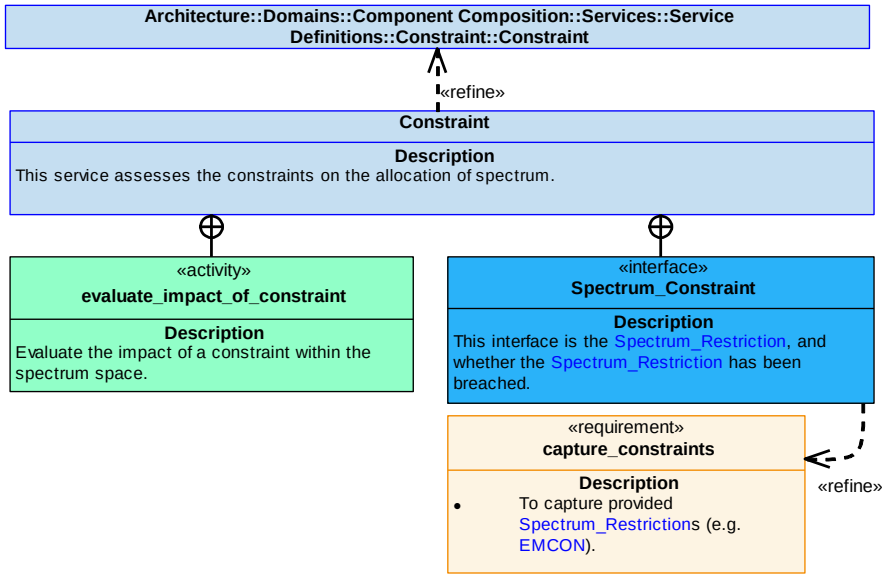

Figure 1028: Constraint Service Policy

Constraint

This service assesses the constraints on the allocation of spectrum.

Interface

Spectrum_Constraint

This interface is the Spectrum_Restriction, and whether the Spectrum_Restriction has been breached.

Attributes

frequency	This attribute describes the spectrum of interest (i.e. frequency, frequency range and tolerance).
power_level	This attribute describes the power level.
directionality	This attribute describes the constrained direction and spread. E.g. in a specific direction only.
spectrum_usage_type	This attribute describes if the usage is for transmitting, receiving or both.
restriction_status	This attribute captures the status of a Spectrum_Restriction (e.g. whether it is currently applicable and how long it will be applicable for).
breach	A statement that the constraint has been breached.

Activity

evaluate_impact_of_constraint

Evaluate the impact of a constraint within the spectrum space.

B.2.57.7.2 Service Dependencies

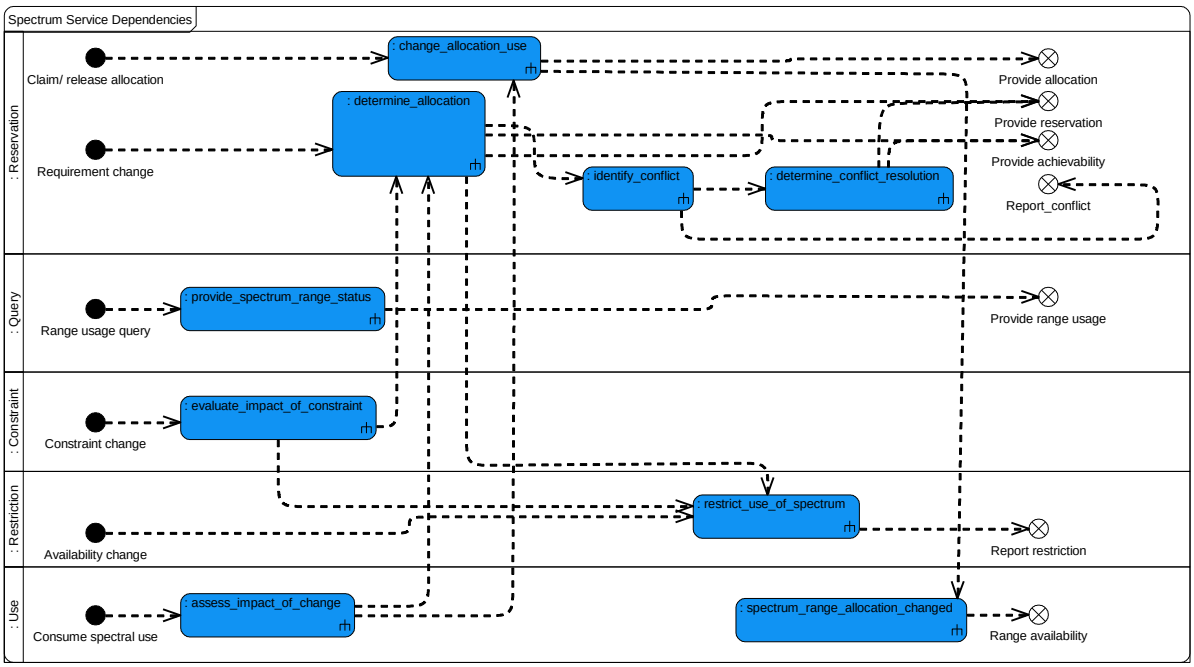

Figure 1029: Spectrum Service Dependencies

B.2.58 Storage

B.2.58.1 Role

The role of Storage is to manage the storage infrastructure.

B.2.58.2 Overview

Control Architecture

Storage is a service component as defined in the Control Architecture policy.

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.

B.2.58.3 Service Summary

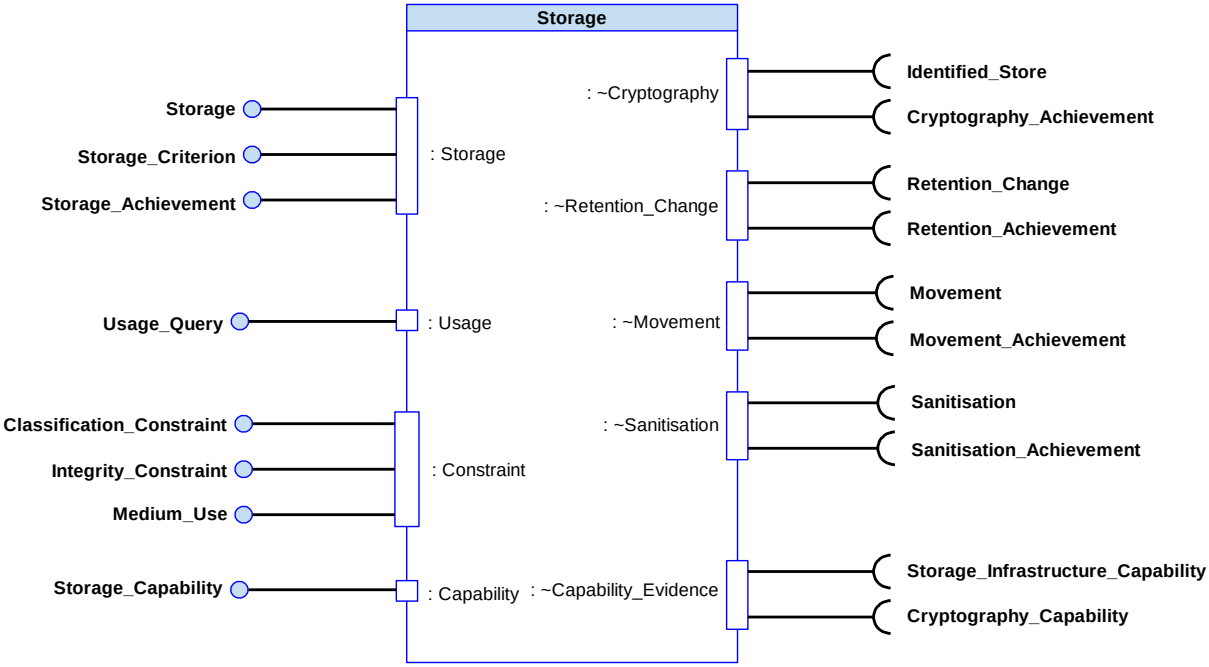

Figure 1030: Storage Service Summary

B.2.58.4 Responsibilities

capture_requirements_for_storage

- To capture provided Requirement(s) (e.g. desired effect or output, timing or balancing) for storage.

capture_storage_constraints

- To capture Constraint(s) on the manner in which data is to be stored (e.g. security, accessibility or redundancy).

capture_requirements_for_sanitisation

- To capture provided Requirement(s) for sanitisation.

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).

initiate_sanitisation_of_storage_space

- To initiate the sanitisation of a Storage_Space.

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.

deliver_storage_solution

- To deliver a Storage_Solution to meet the Requirement(s) within the Constraint(s).

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.

capture_measurement_criteria_for_storage

- To capture given Measure_of_Achievement.

identify_progress

- To identify the progress against the Requirement.

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.

B.2.58.5 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.

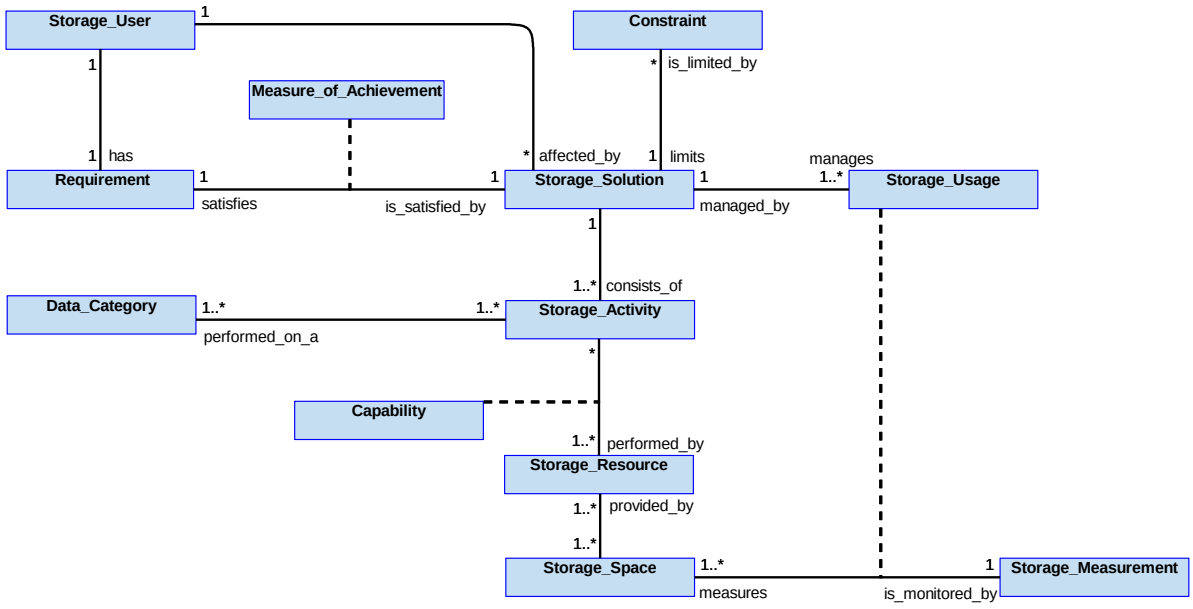

Figure 1031: Storage Semantics

B.2.58.5.1 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.

B.2.58.6 Design Rationale

B.2.58.6.1 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.

B.2.58.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.58.6.3 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.

B.2.58.6.4 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 policy 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 policy for further details.

Note: The Security Guidance for PYRAMID Exploiters, Ref. [60], provides additional information about the implementation of secure storage.

B.2.58.7 Services

B.2.58.7.1 Service Definitions

B.2.58.7.1.1 Storage

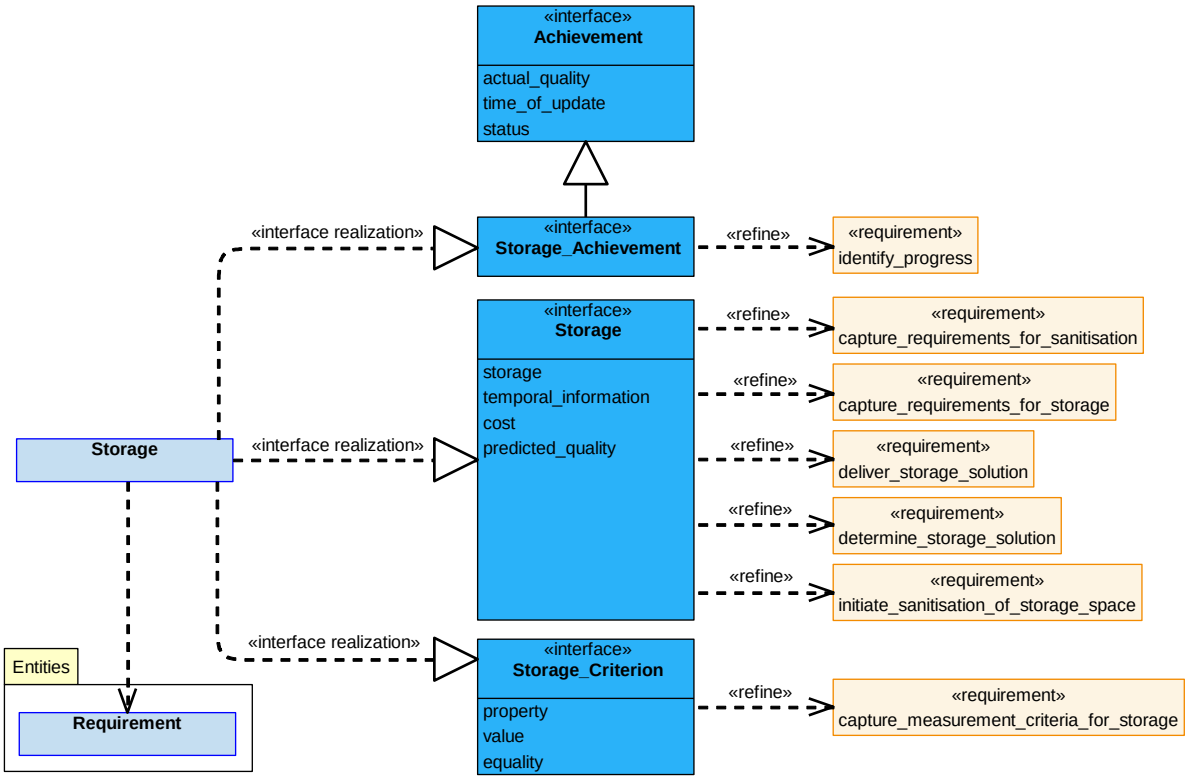

Figure 1032: Storage Service Definition

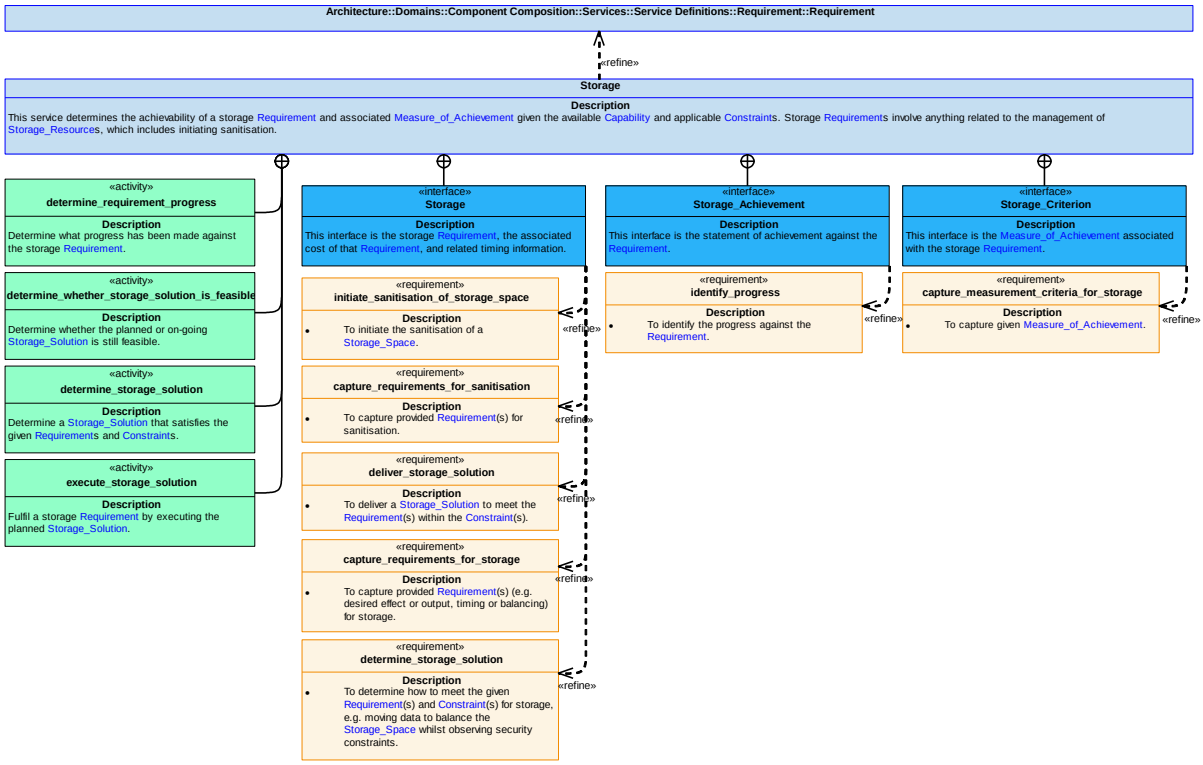

Figure 1033: Storage Service Policy

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.

B.2.58.7.1.2 Cryptography

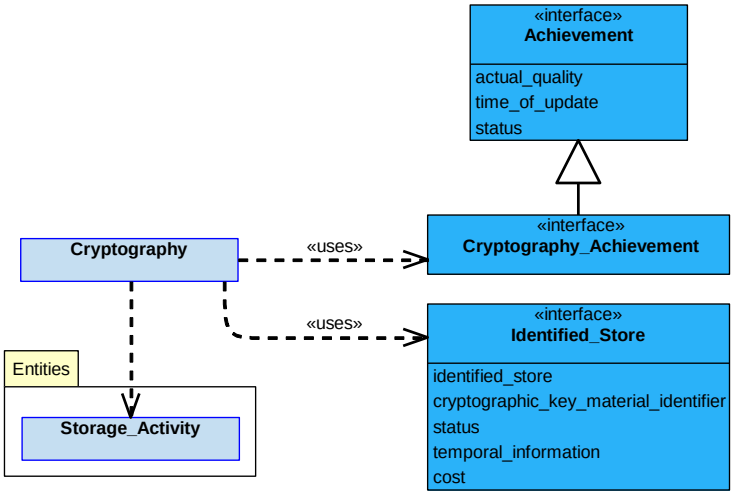

Figure 1034: Cryptography Service Definition

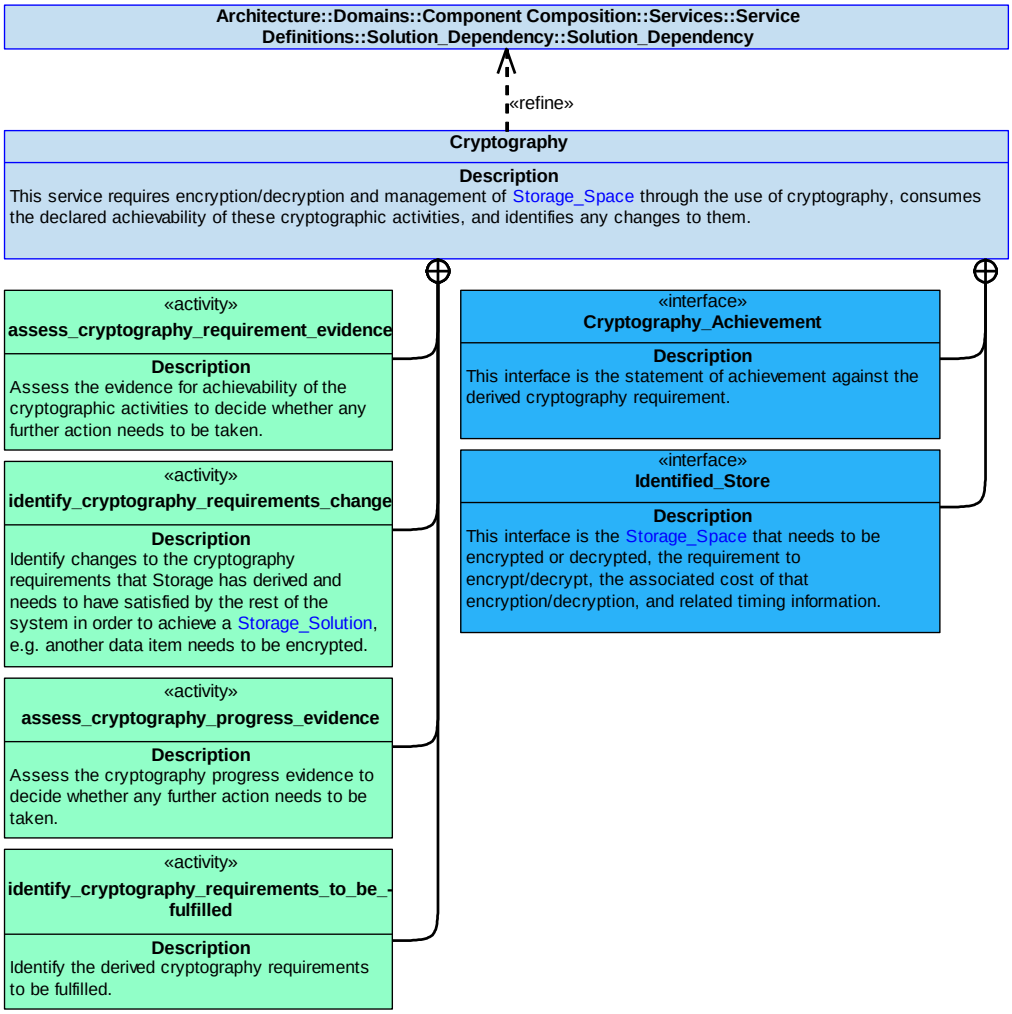

Figure 1035: Cryptography Service Policy

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.

B.2.58.7.1.3 Retention_Change

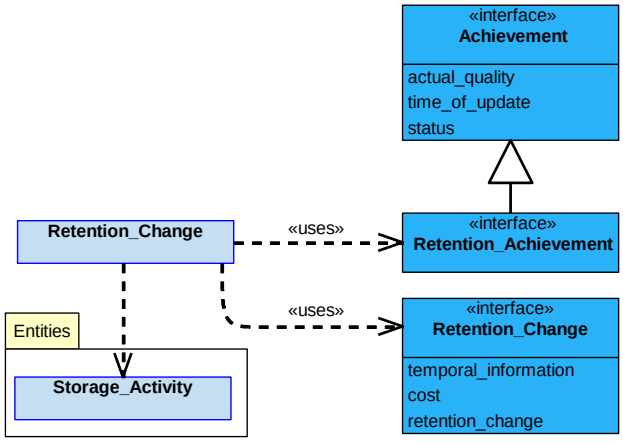

Figure 1036: Retention_Change Service Definition

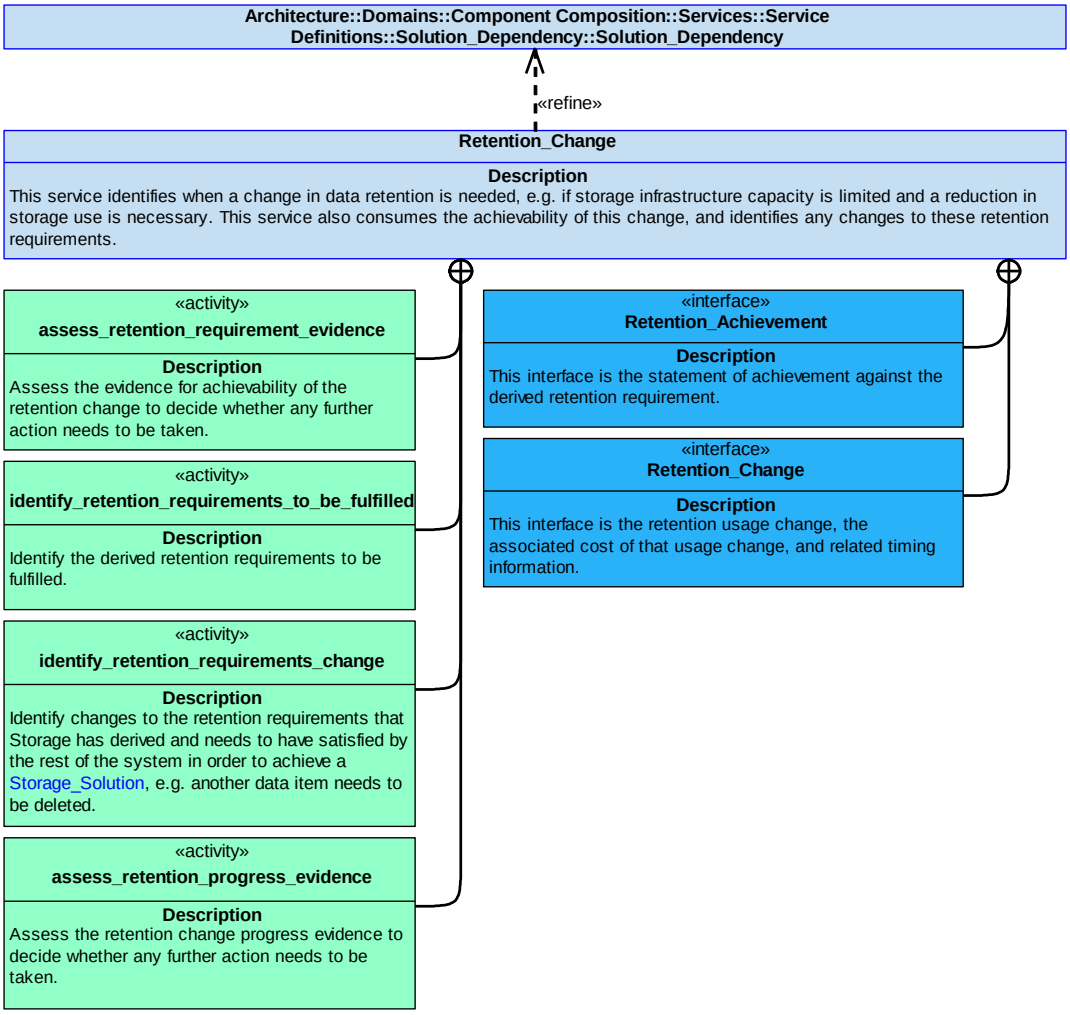

Figure 1037: Retention_Change Service Policy

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.

B.2.58.7.1.4 Movement

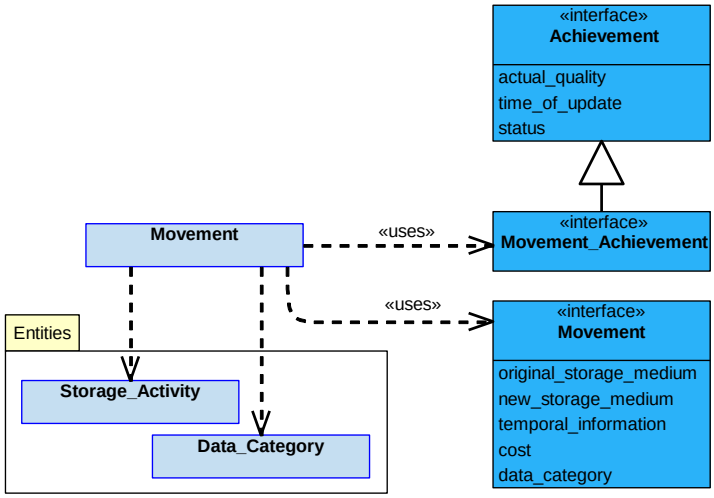

Figure 1038: Movement Service Definition

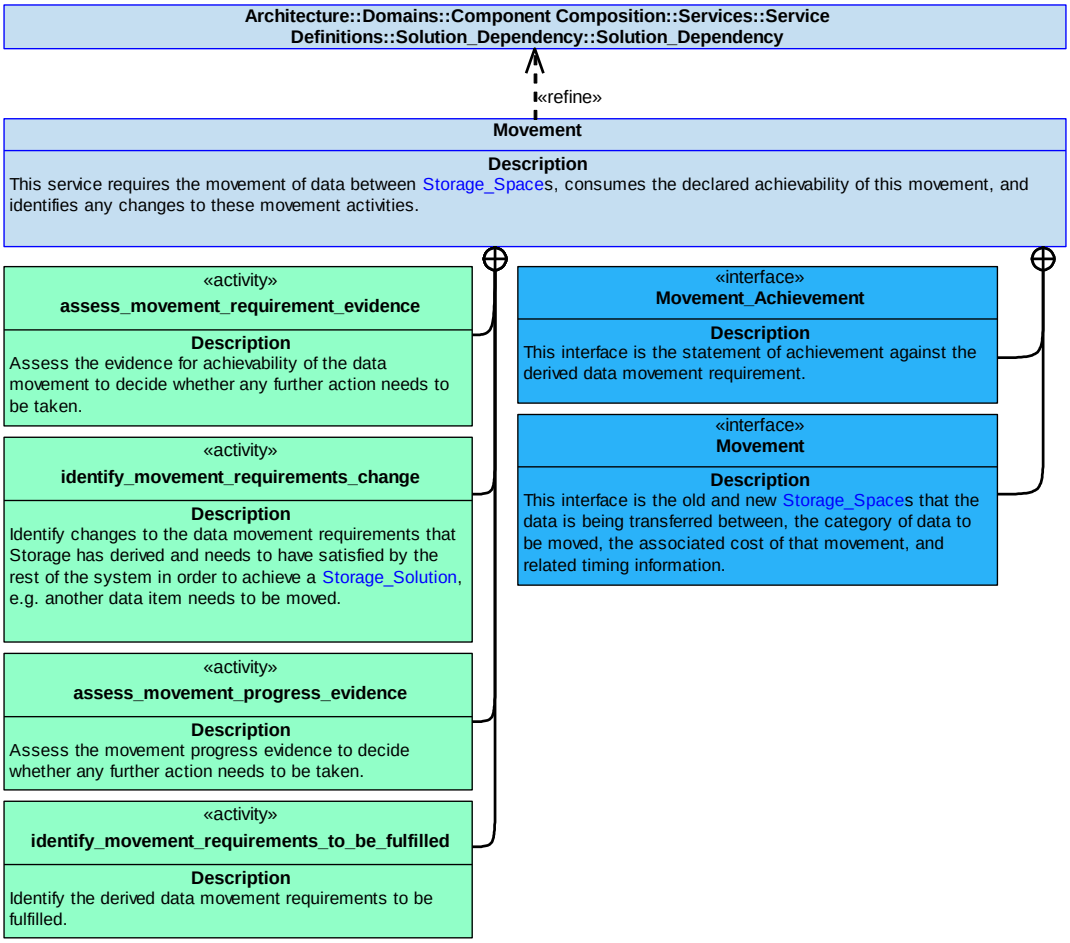

Figure 1039: Movement Service Policy

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.

B.2.58.7.1.5 Sanitisation

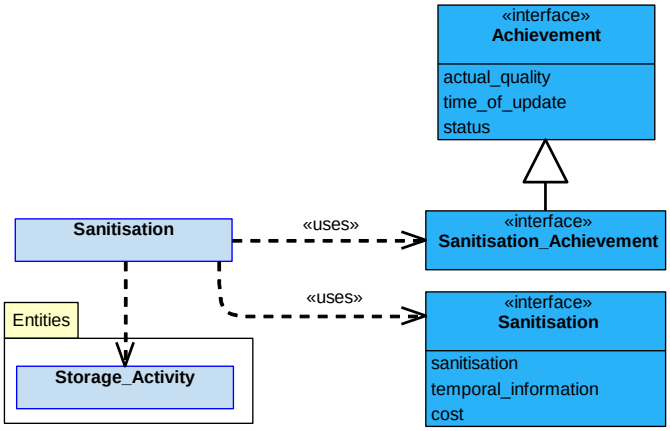

Figure 1040: Sanitisation Service Definition

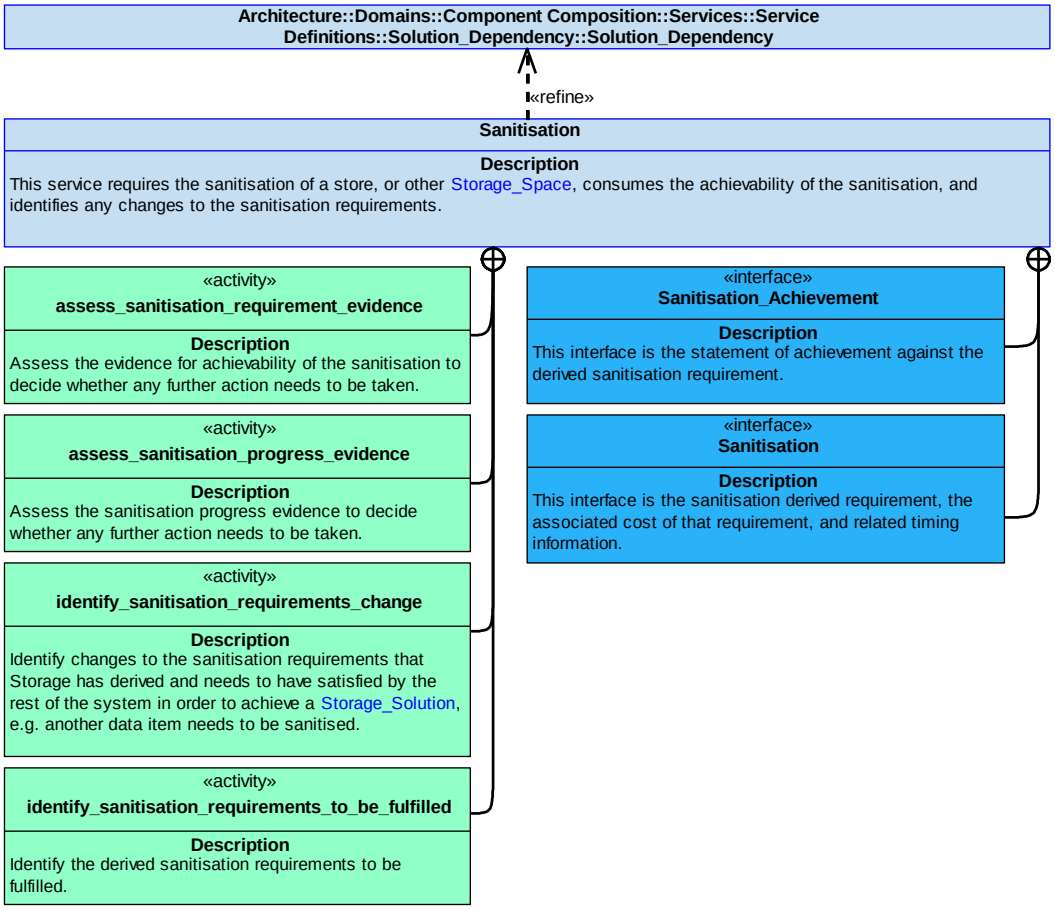

Figure 1041: Sanitisation Service Policy

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.

B.2.58.7.1.6 Usage

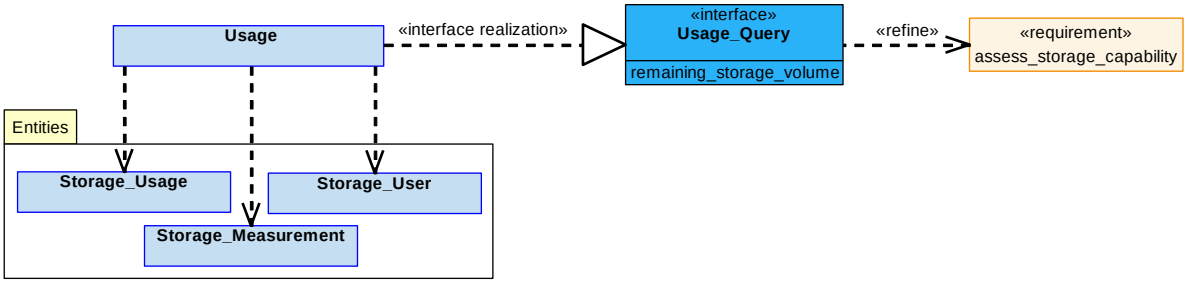

Figure 1042: Usage Service Definition

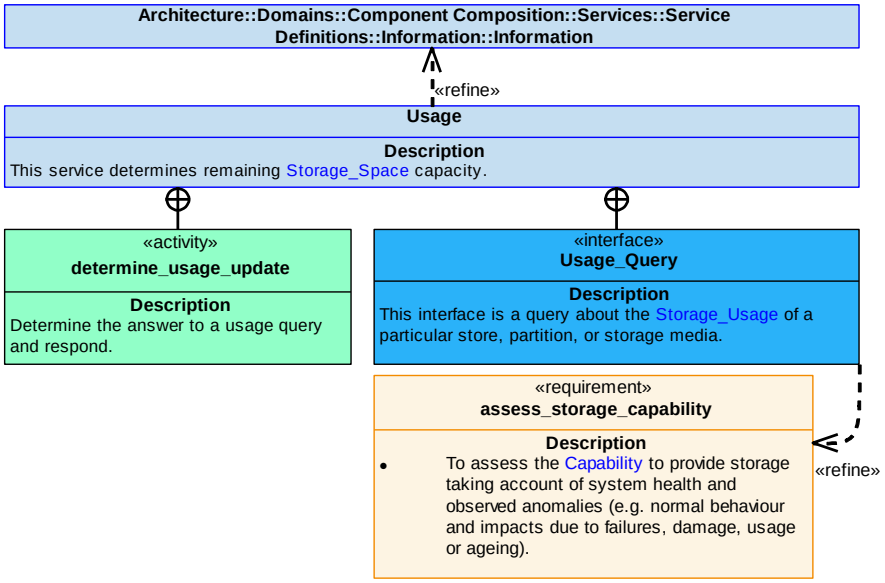

Figure 1043: Usage Service Policy

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.

B.2.58.7.1.7 Constraint

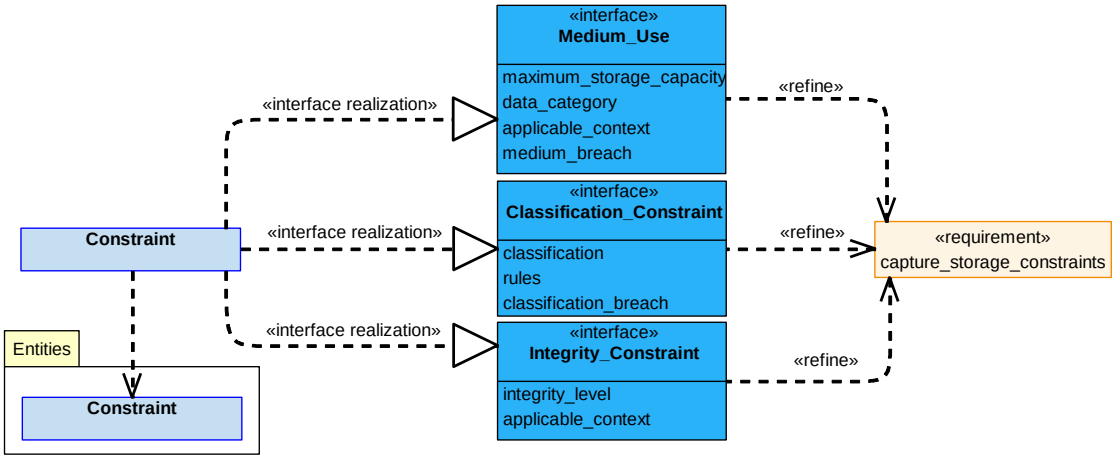

Figure 1044: Constraint Service Definition

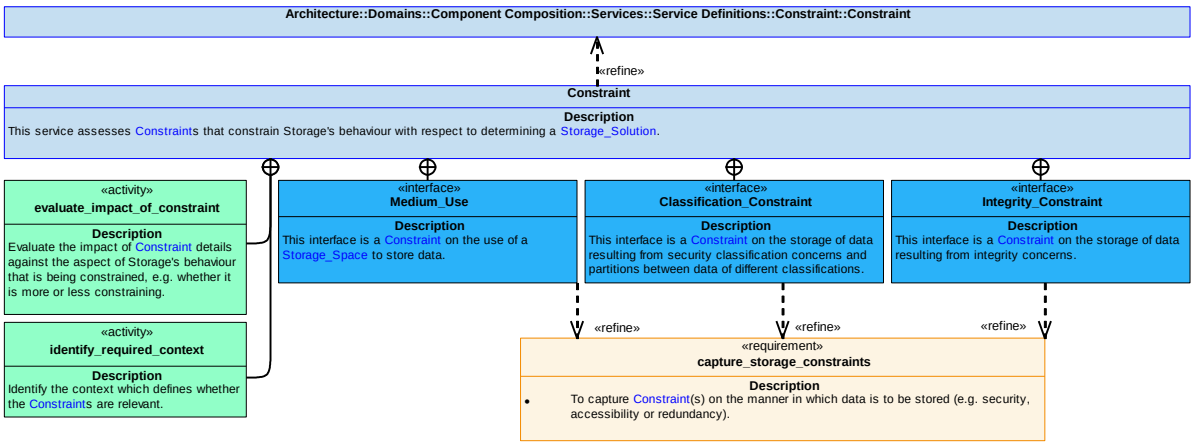

Figure 1045: Constraint Service Policy

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.

B.2.58.7.1.8 Capability

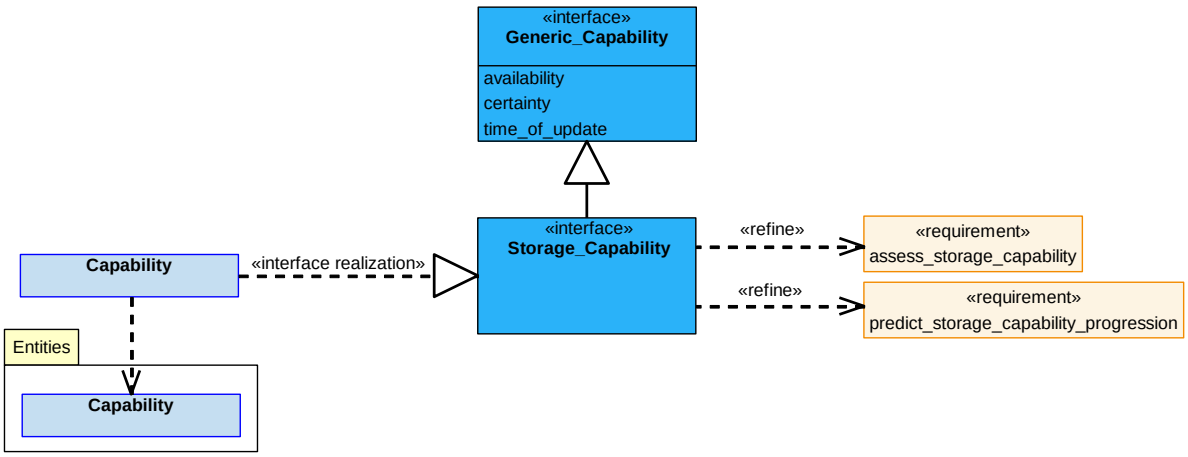

Figure 1046: Capability Service Definition

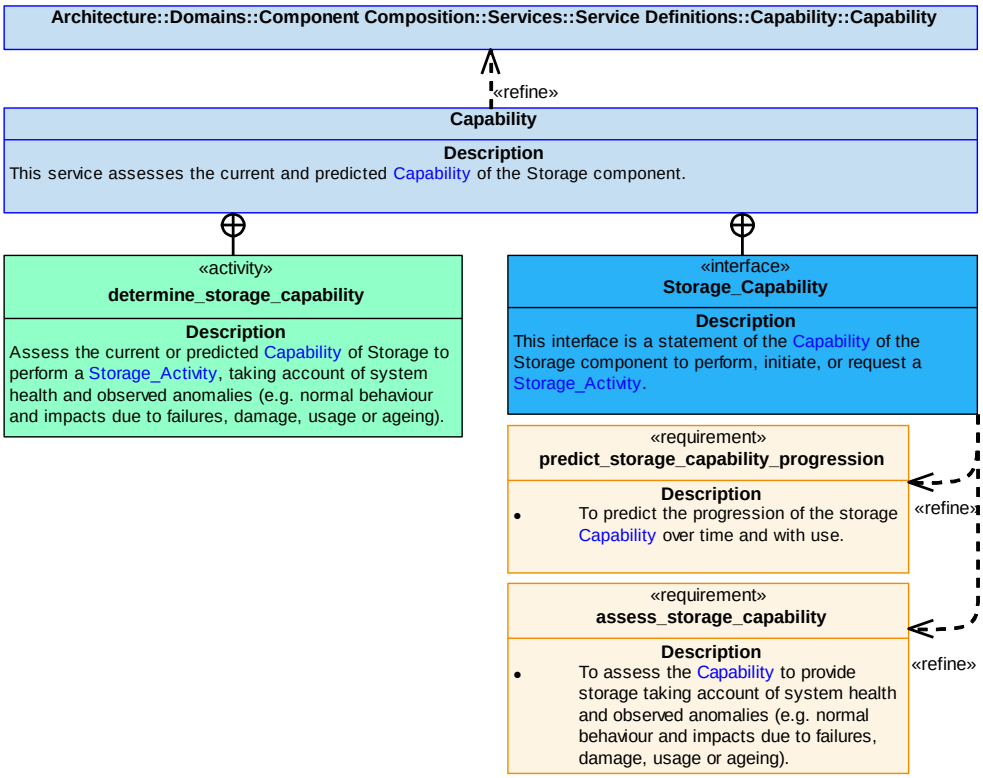

Figure 1047: Capability Service Policy

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).

B.2.58.7.1.9 Capability_Evidence

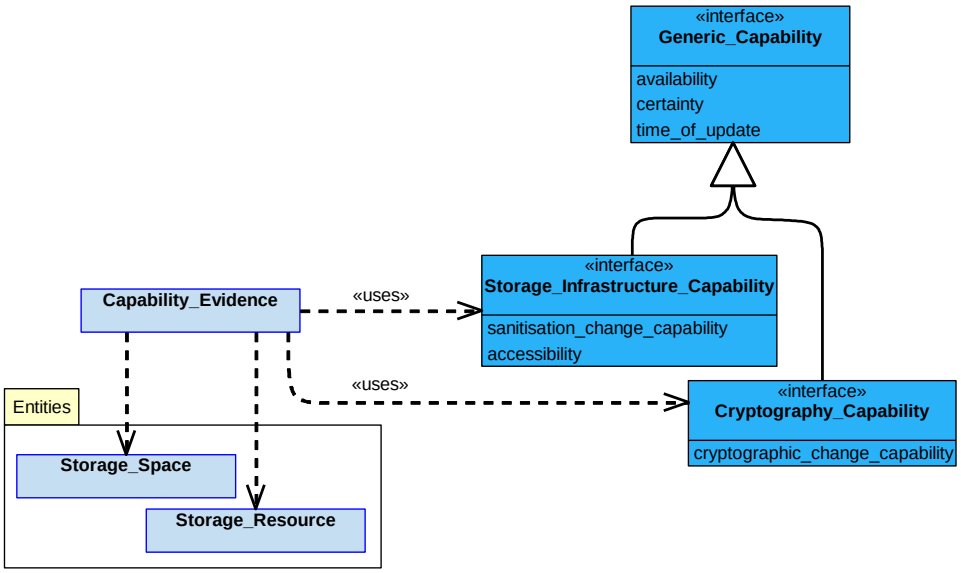

Figure 1048: Capability_Evidence Service Definition

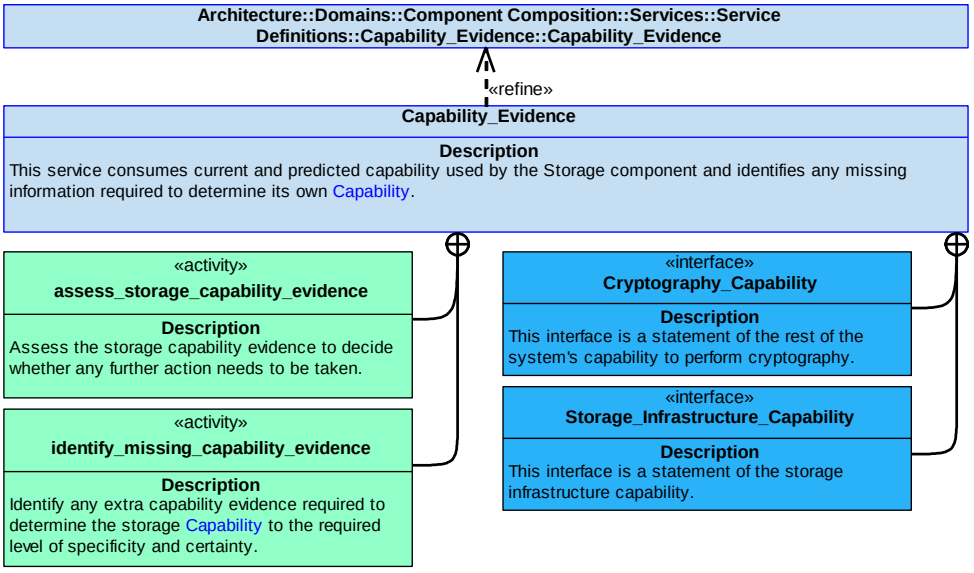

Figure 1049: Capability_Evidence Service Policy

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.

B.2.58.7.2 Service Dependencies

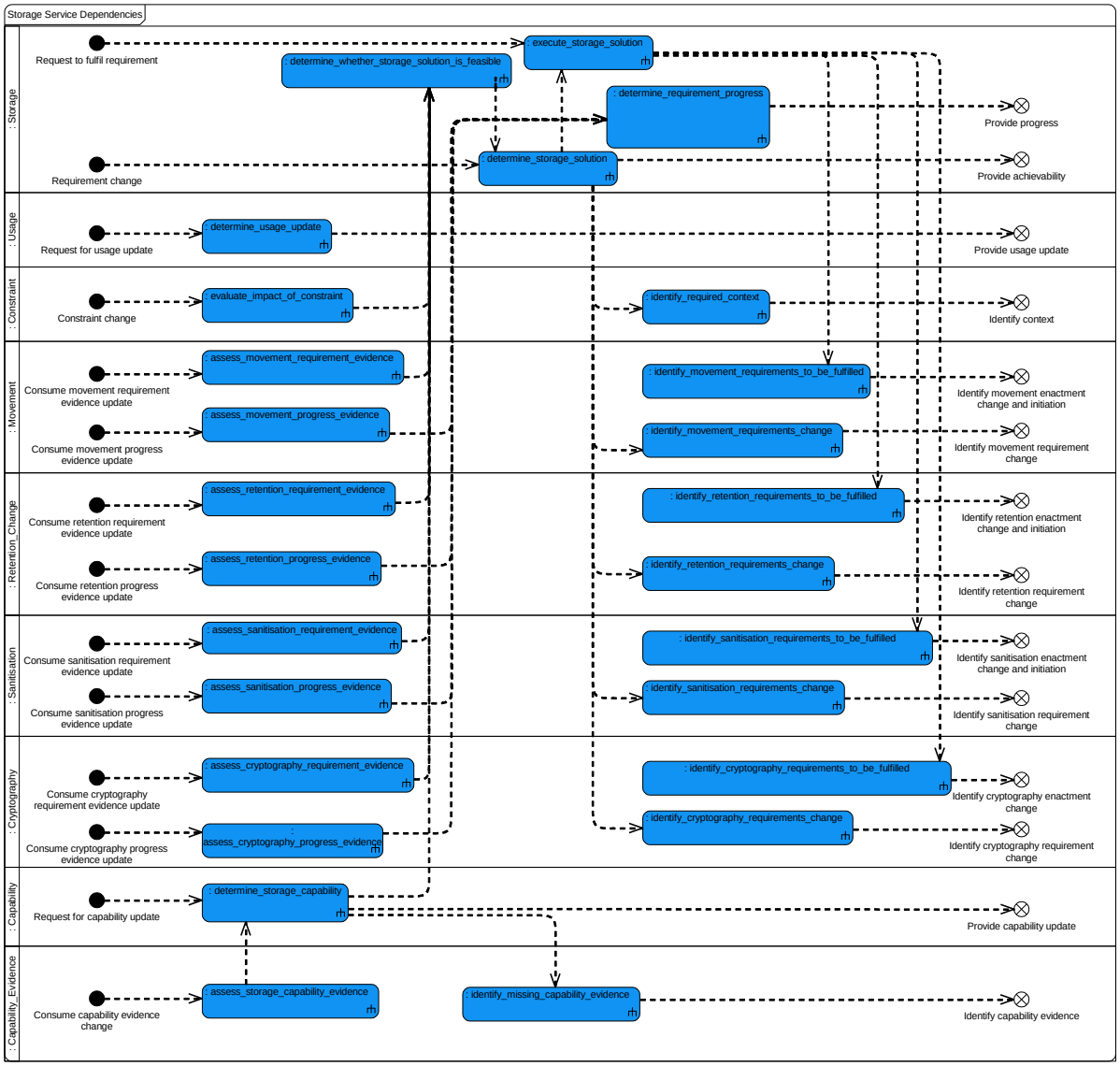

Figure 1050: Storage Service Dependencies

B.2.59 Stores Release

B.2.59.1 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.

B.2.59.2 Overview

Control Architecture

Stores Release is an action component as defined in the Control Architecture policy.

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.

B.2.59.3 Service Summary

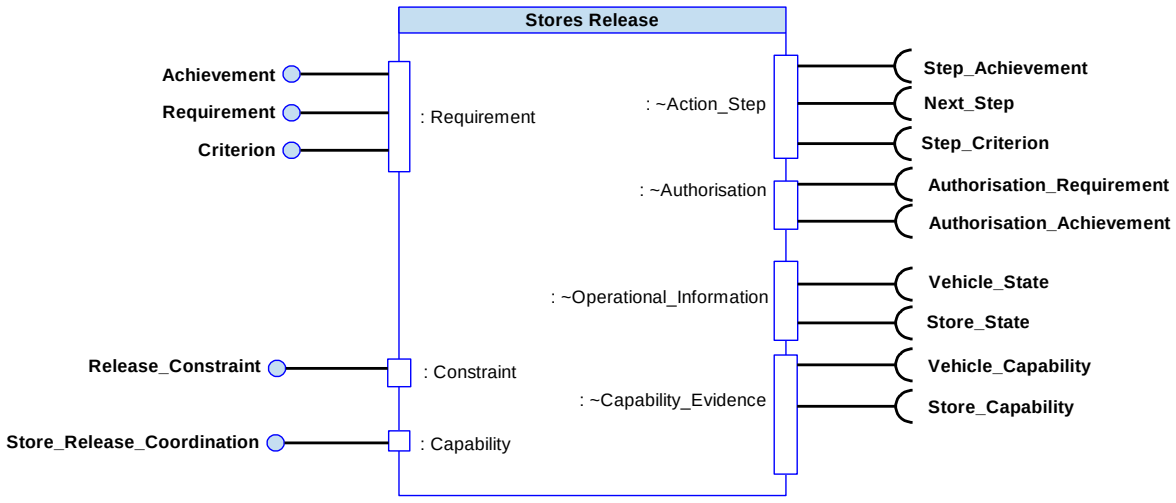

Figure 1051: Stores Release Service Summary

B.2.59.4 Responsibilities

capture_release_package_requirement

- To capture the Requirements for the release of Stores.

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).

determine_release_packages

- To determine the Release_Packages (i.e. packages that do not compromise air vehicle safety) in response to a Requirement.

determine_stores_releasability

- To determine which Stores are releasable.

determine_release_schedule

- To determine the Release_Schedule for Stores (when part of a Release_Package) with respect to provided Requirements and Constraints.

determine_store_release_progress

- To determine the progress against the Release_Schedule (e.g. report whether Stores have hung or have left the platform).

execute_release_solution

- To coordinate the execution of a Release_Schedule.

predict_stores_release_capability_progression

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

identify_whether_solution_is_feasible

- To identify whether a Release_Schedule in progress remains feasible given current resources.

identify_missing_information

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

capture_provided_constraints

- To capture provided Constraints for stores release.

capture_measurement_criteria

- To capture provided Measurement_Criterion which a Release_Schedule will be measured against.

determine_quality_of_solution

- To determine the quality of a proposed Release_Schedule against given Requirements.

identify_pre-conditions

- To identify Pre-conditions required to support the Release_Schedule.

B.2.59.5 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.

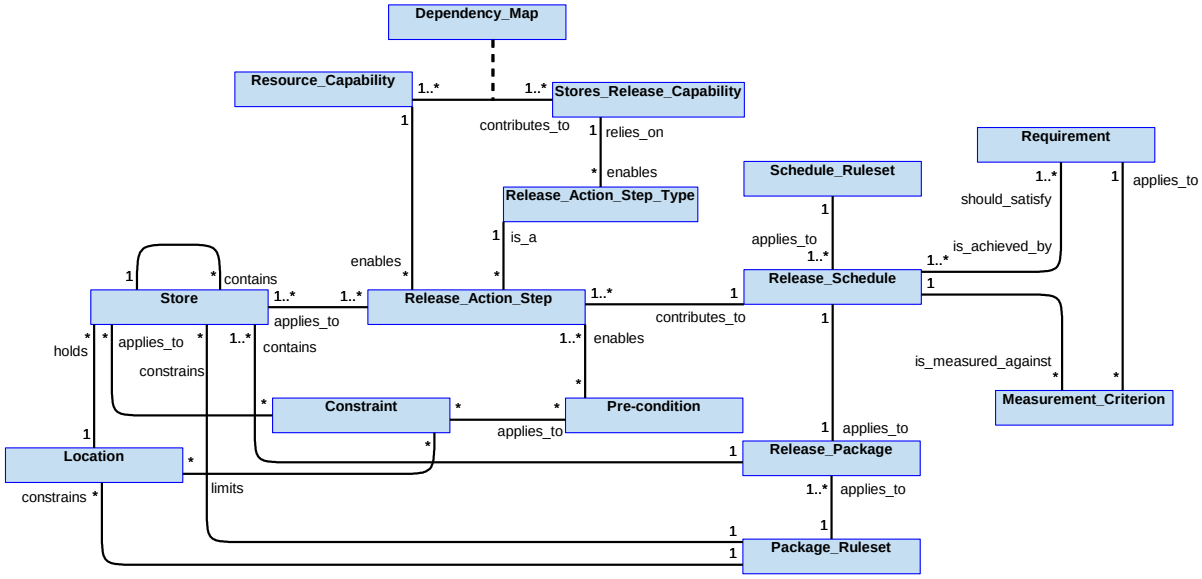

Figure 1052: Stores Release Semantics

B.2.59.5.1 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, the MASS is live or authorisation is granted).

Release_Action_Step_Type

The kinds of activity 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 to 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.

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 activity that, when performed, achieves (or partially achieves) the release of a Store from the Exploiting Platform.

B.2.59.6 Design Rationale

B.2.59.6.1 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.

B.2.59.6.2 Design Considerations

Directly Applicable Policies

These policies 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, utilising data driving for the Schedule_Ruleset and Package_Ruleset would provide benefits with regard to the Key User Requirements (KURs).

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.

B.2.59.6.3 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.

B.2.59.6.4 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.

B.2.59.7 Services

B.2.59.7.1 Service Definitions

B.2.59.7.1.1 Requirement

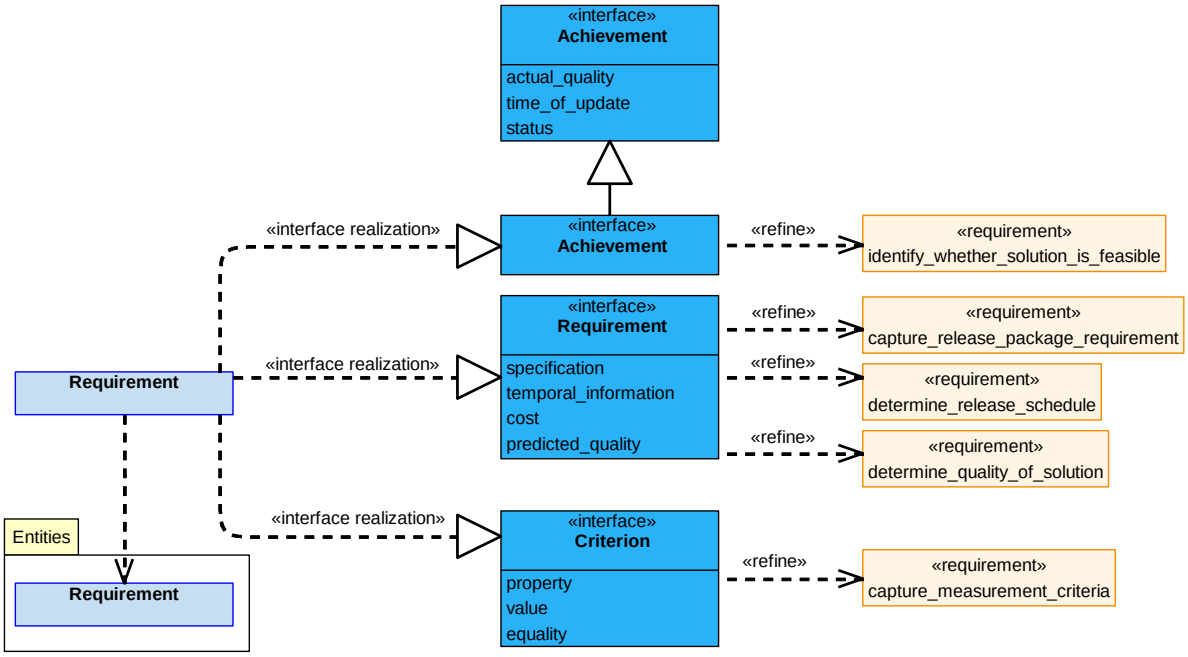

Figure 1053: Requirement Service Definition

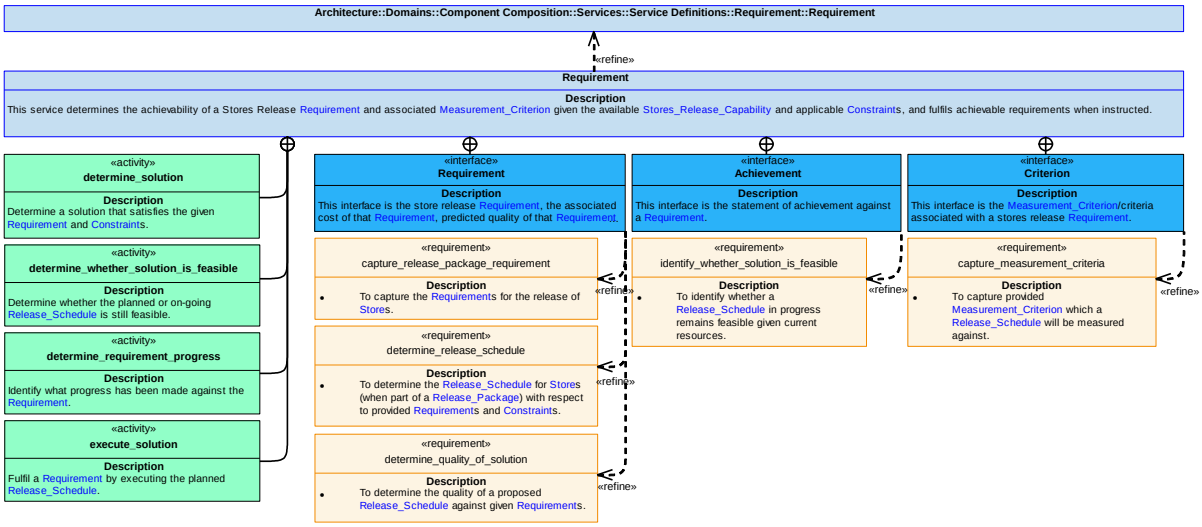

Figure 1054: Requirement Service Policy

Requirement

This service determines the achievability of a Stores 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.

B.2.59.7.1.2 Action_Step

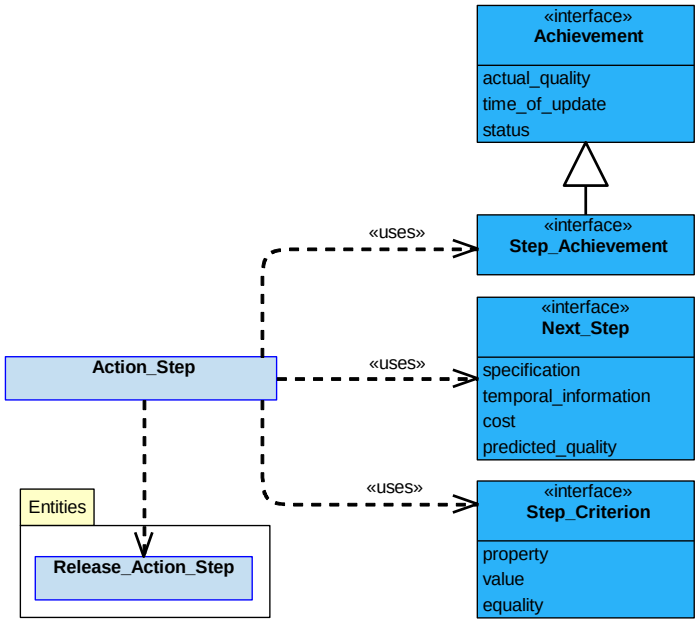

Figure 1055: Action_Step Service Definition

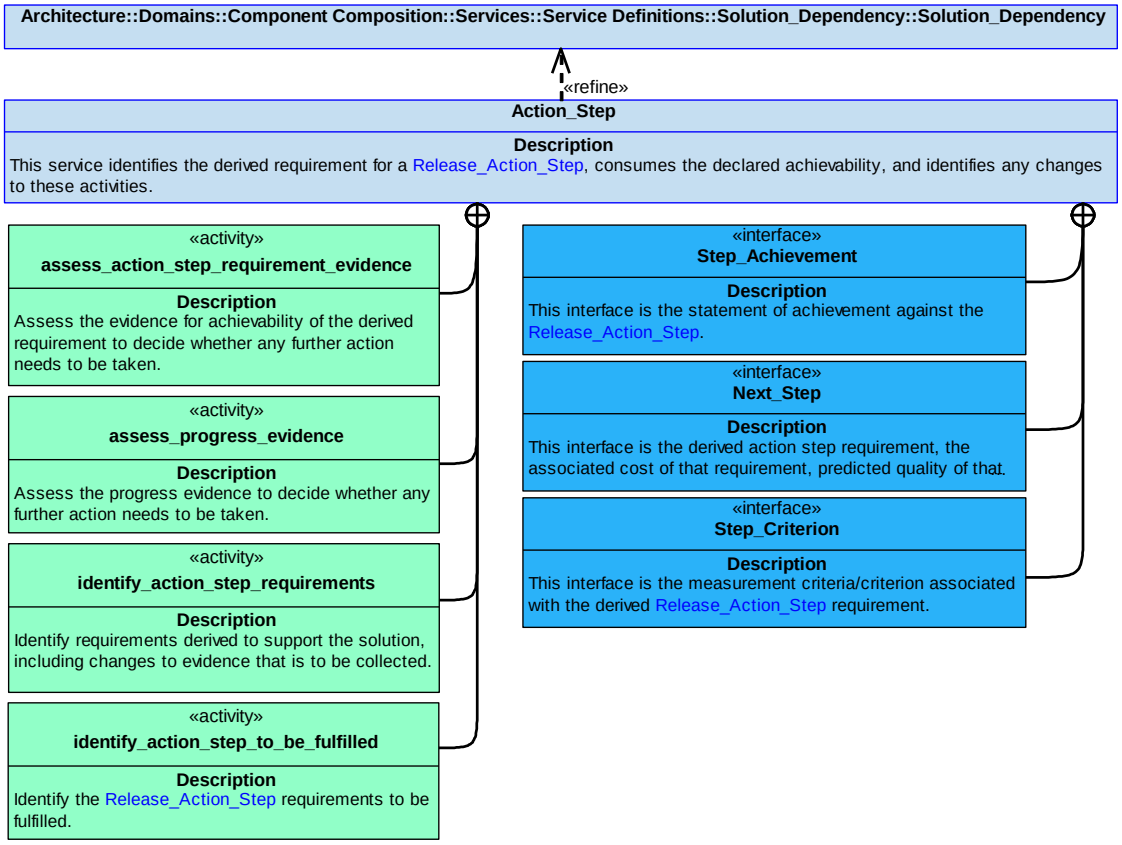

Figure 1056: Action_Step Service Policy

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 , for example, e.g. resources used, 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. mass of a package.
value	The measured value of the property, e.g. 500 lb.
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.

B.2.59.7.1.3 Authorisation

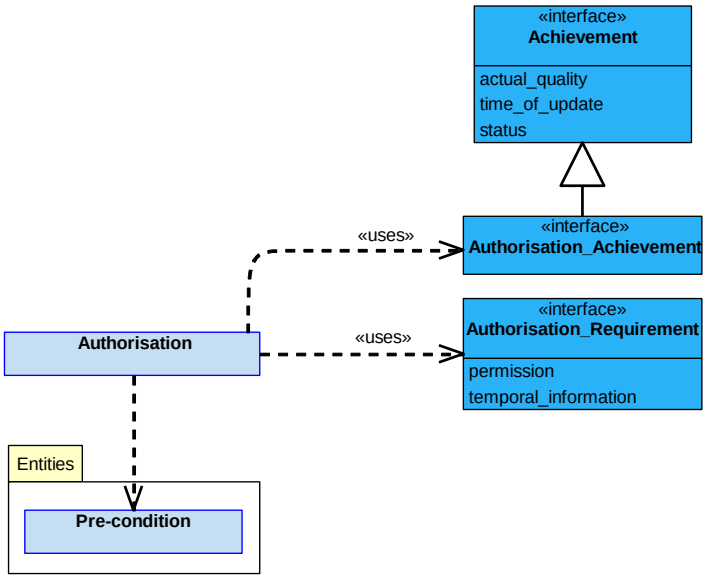

Figure 1057: Authorisation Service Definition

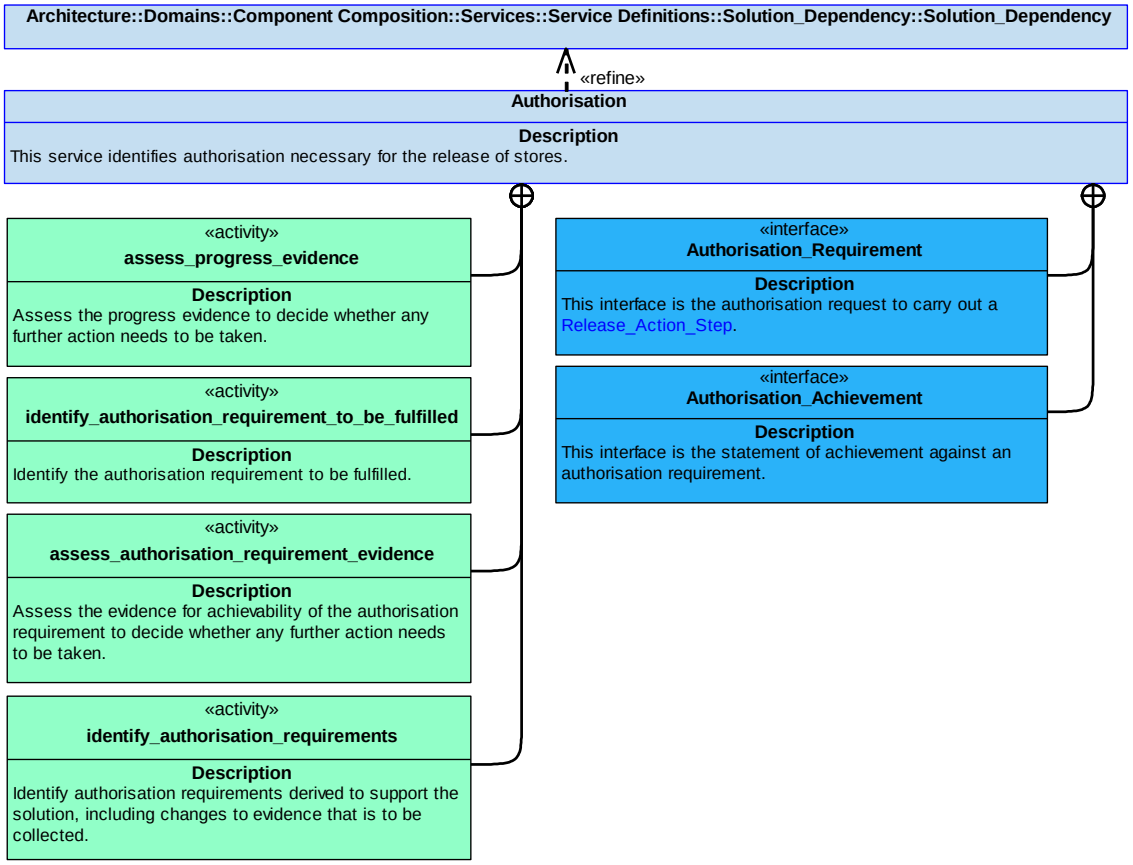

Figure 1058: Authorisation Service Policy

Authorisation

This service identifies authorisation necessary for the release of stores.

Interfaces

Authorisation_Requirement

This interface is the authorisation request 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.

Authorisation_Achievement

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

Activities

assess_progress_evidence

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

identify_authorisation_requirement_to_be_fulfilled

Identify the authorisation requirement to be fulfilled.

identify_authorisation_requirements

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

assess_authorisation_requirement_evidence

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

B.2.59.7.1.4 Operational_Information

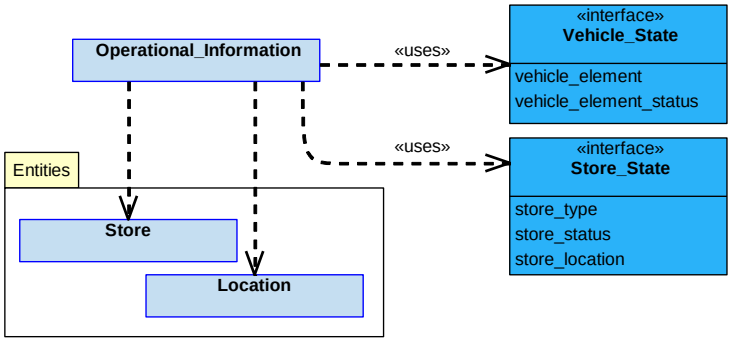

Figure 1059: Operational_Information Service Definition

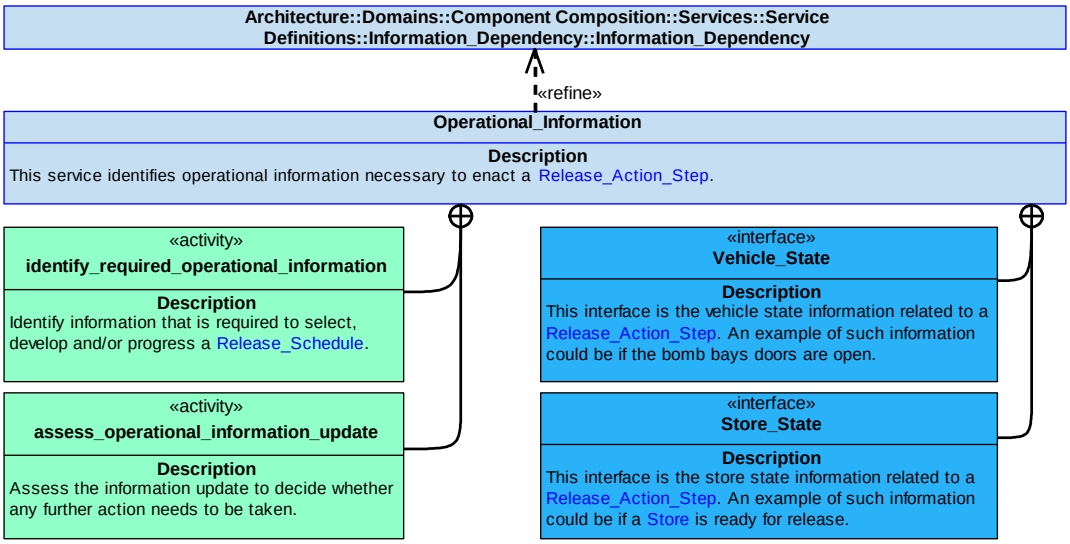

Figure 1060: Operational_Information Service Policy

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.

B.2.59.7.1.5 Constraint

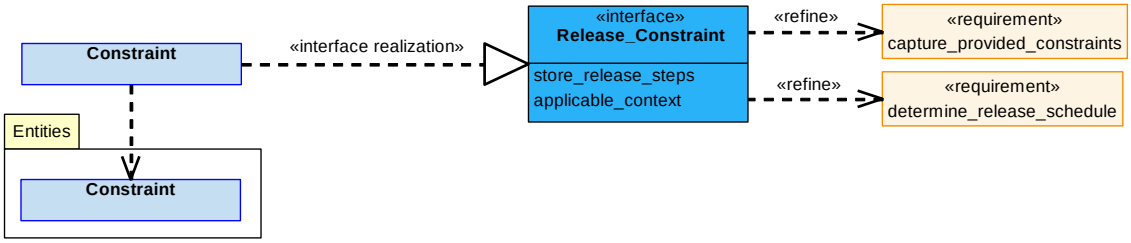

Figure 1061: Constraint Service Definition

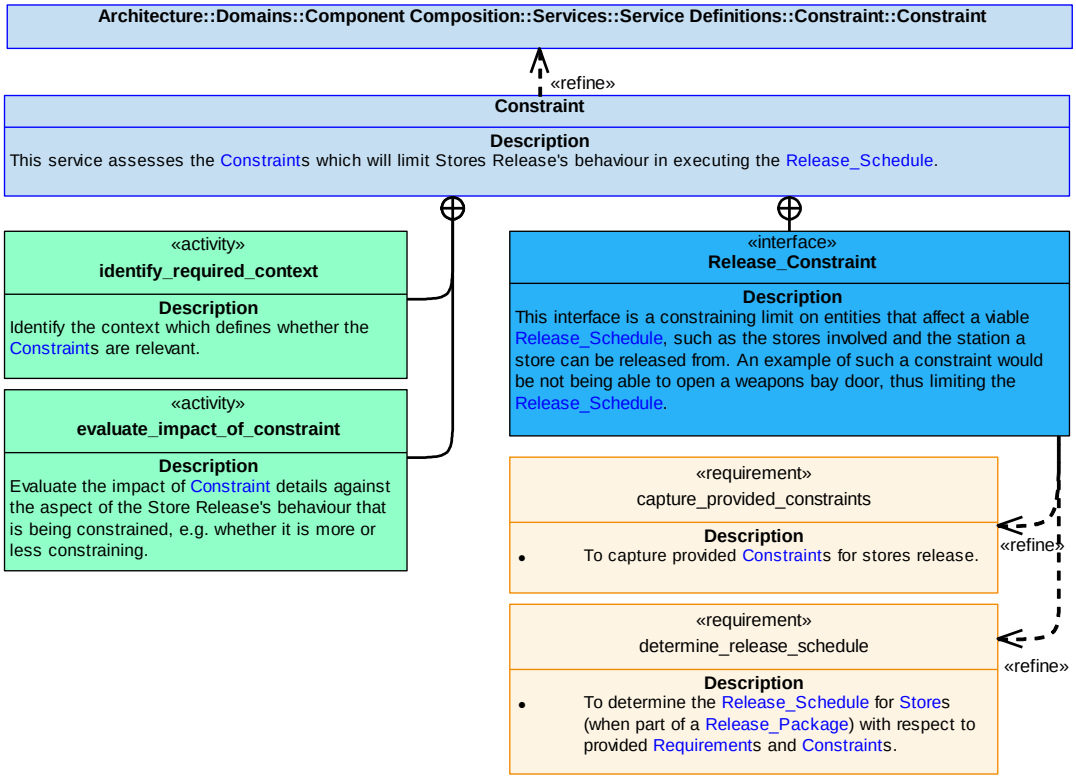

Figure 1062: Constraint Service Policy

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.

B.2.59.7.1.6 Capability

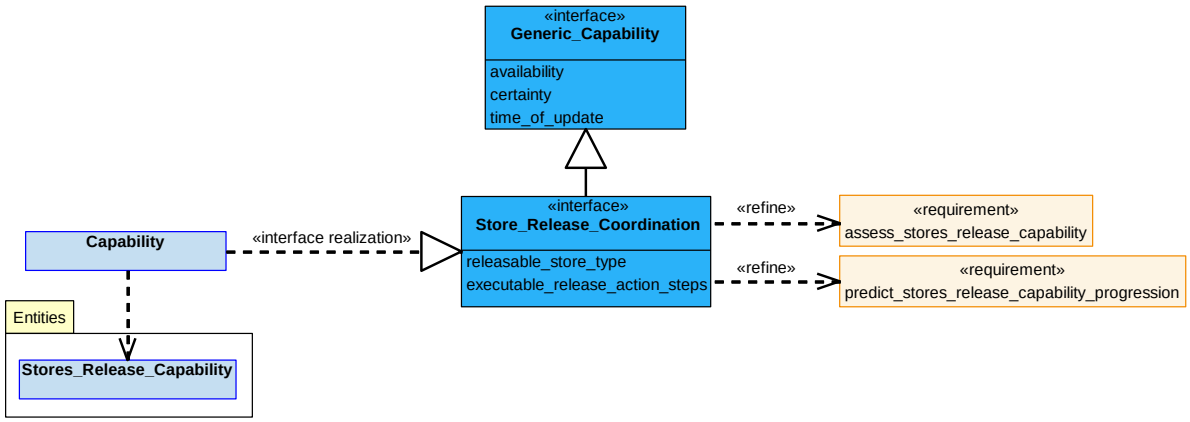

Figure 1063: Capability Service Definition

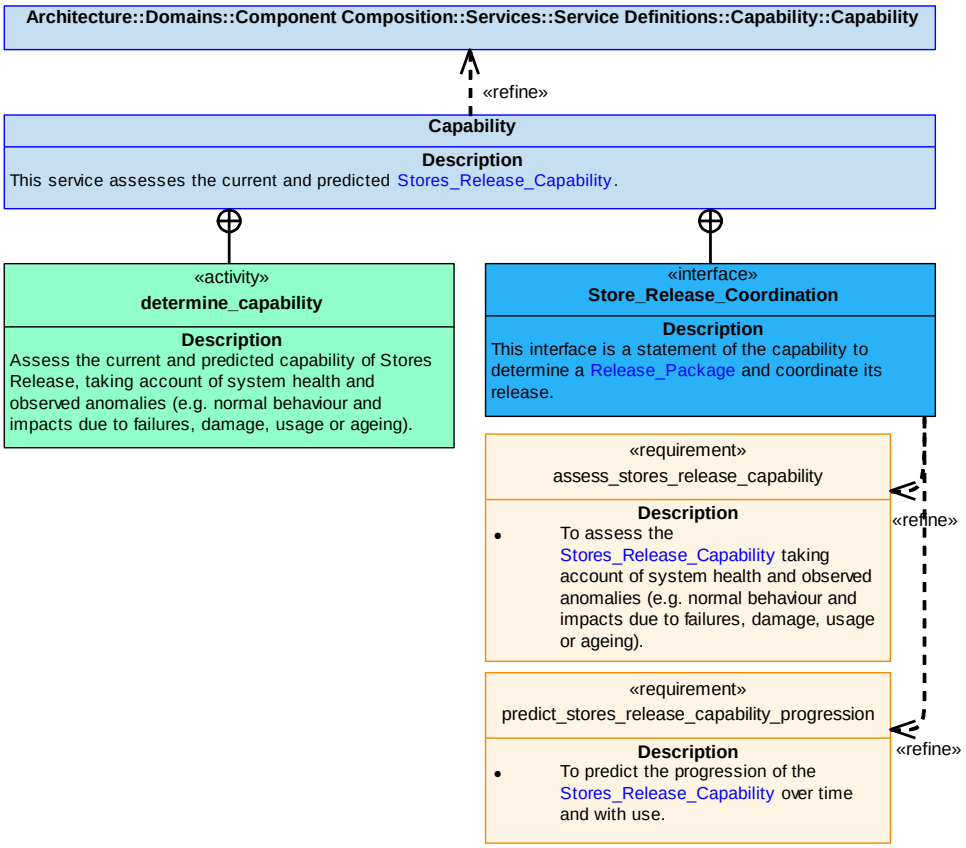

Figure 1064: Capability Service Policy

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_store_type	The types of Stores that can be released.
executable_release_action_steps	The executable steps required to coordinate a store release.

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).

B.2.59.7.1.7 Capability_Evidence

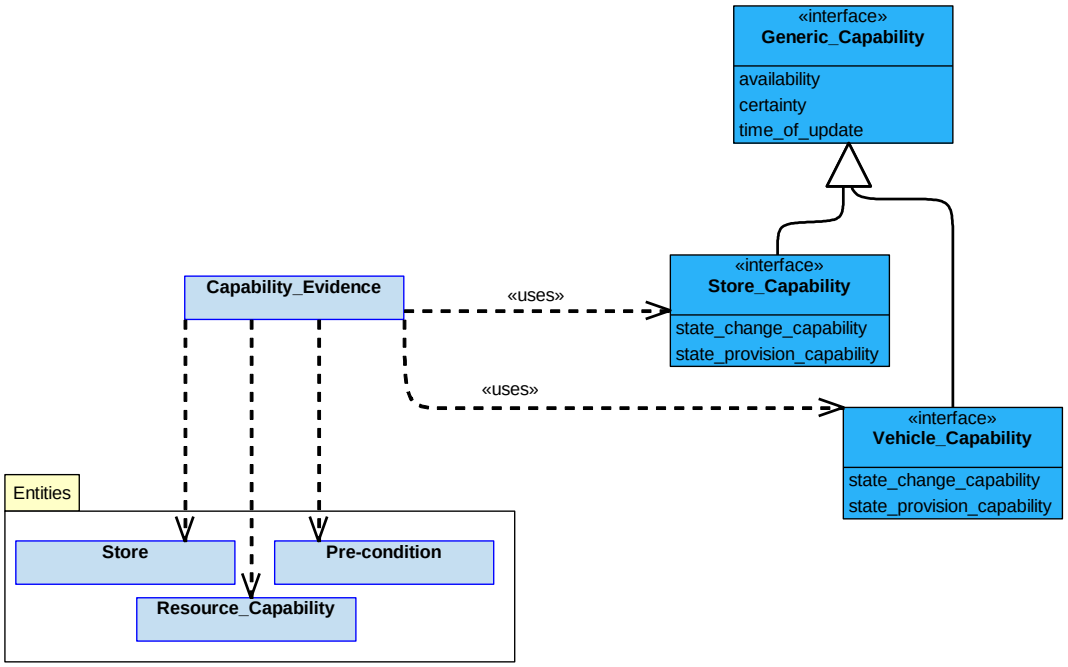

Figure 1065: Capability_Evidence Service Definition

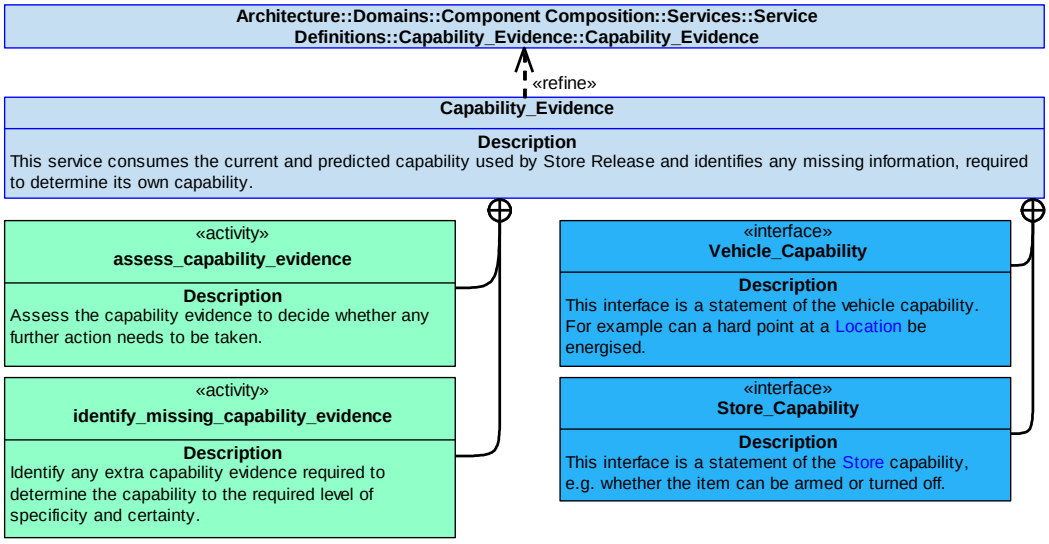

Figure 1066: Capability_Evidence Service Policy

Capability_Evidence

This service consumes the current and predicted capability used by Store 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 can 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.

B.2.59.7.2 Service Dependencies

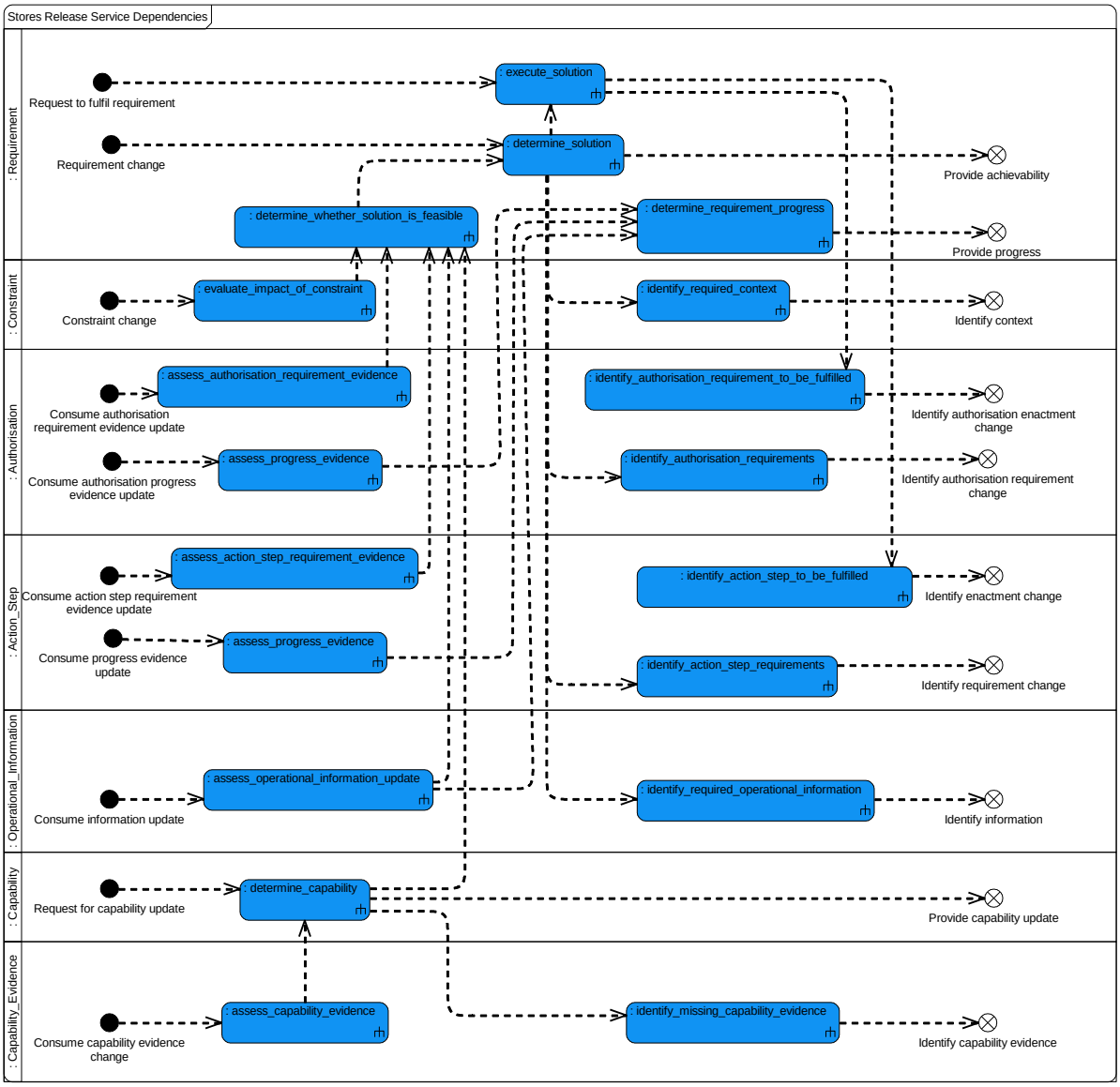

Figure 1067: Stores Release Service Dependencies

B.2.60 Susceptibility

B.2.60.1 Role

The role of Susceptibility is to provide a view of a subject's potential susceptibility to aggressor actions.

B.2.60.2 Overview

Control Architecture

Susceptibility is a service component as defined in the Control Architecture policy.

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.

B.2.60.3 Service Summary

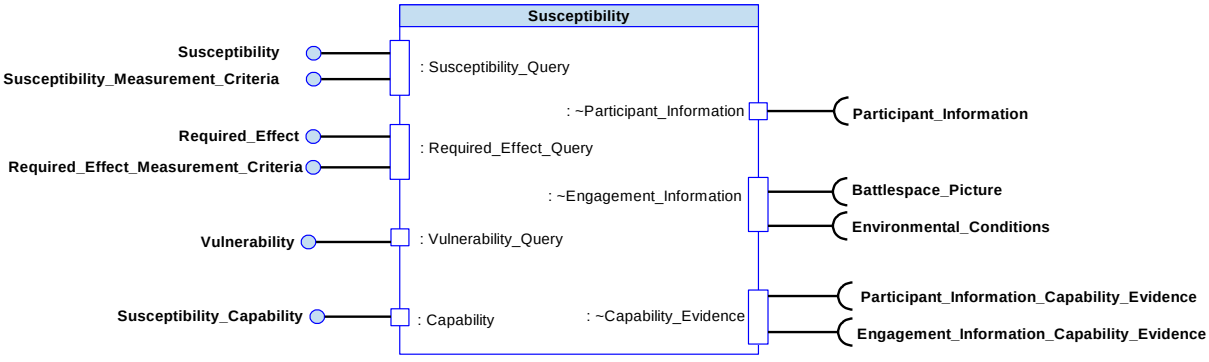

Figure 1068: Susceptibility Service Summary

B.2.60.4 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.

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

- To determine the Susceptibility of a Subject to the Offensive_Capability of an Aggressor.

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).

predict_capability_progression

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

identify_missing_information

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

B.2.60.5 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.

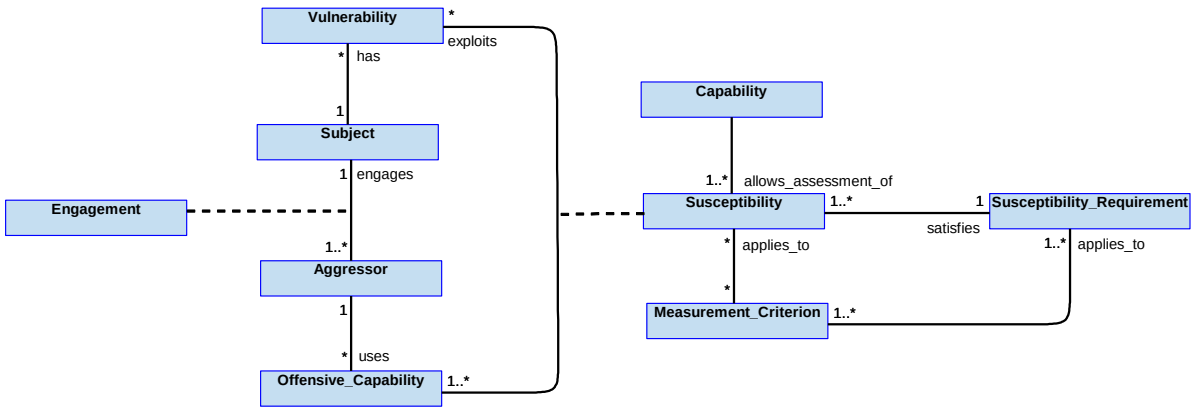

Figure 1069: Susceptibility Semantics

B.2.60.5.1 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.

B.2.60.6 Design Rationale

B.2.60.6.1 Assumptions

- Weather conditions can be considered to be Aggressors to the Subject, as well as more typical threats such as hostile vehicles.

B.2.60.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.60.6.3 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.

B.2.60.6.4 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.

B.2.60.7 Services

B.2.60.7.1 Service Definitions

B.2.60.7.1.1 Susceptibility_Query

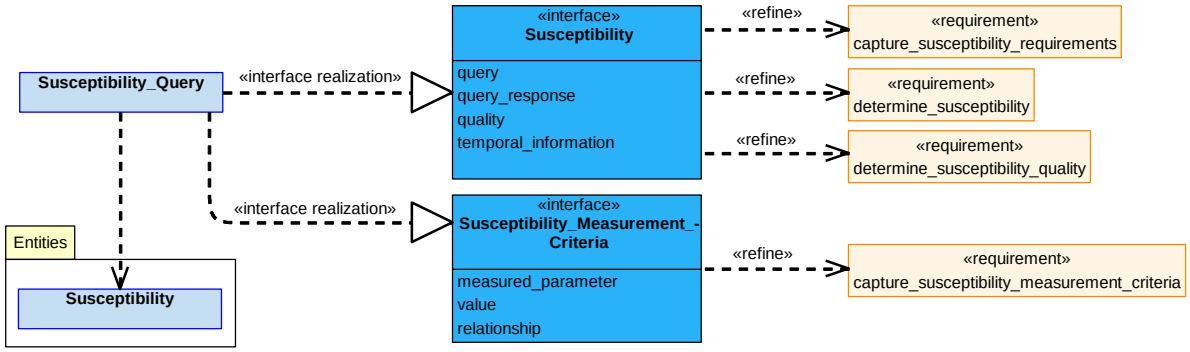

Figure 1070: Susceptibility_Query Service Definition

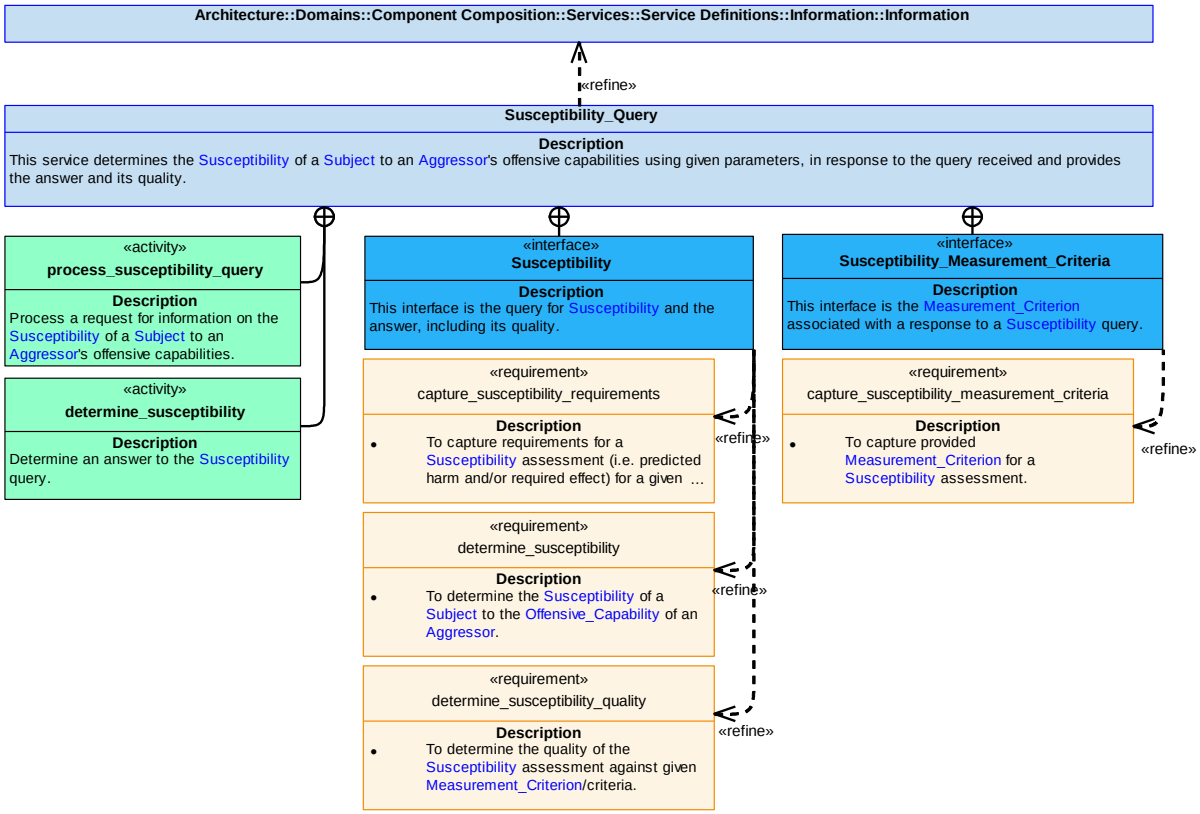

Figure 1071: Susceptibility_Query Service Policy

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. E.g. can the potency of a storm 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.

B.2.60.7.1.2 Required_Effect_Query

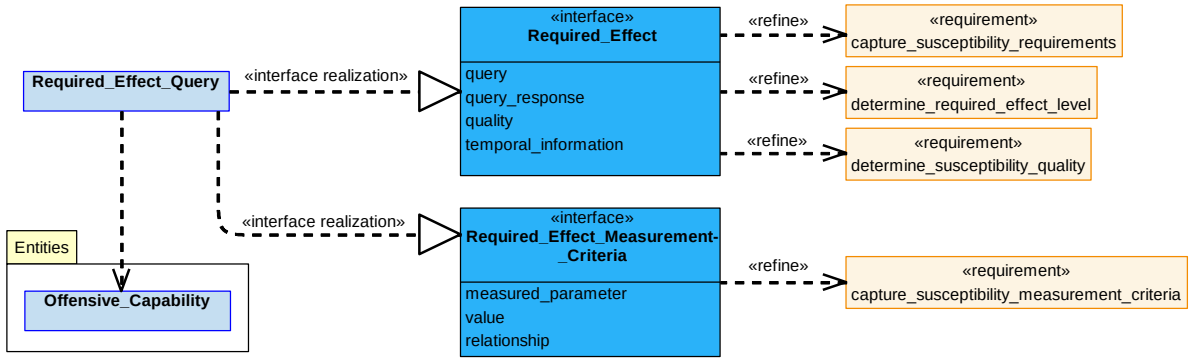

Figure 1072: Required_Effect_Query Service Definition

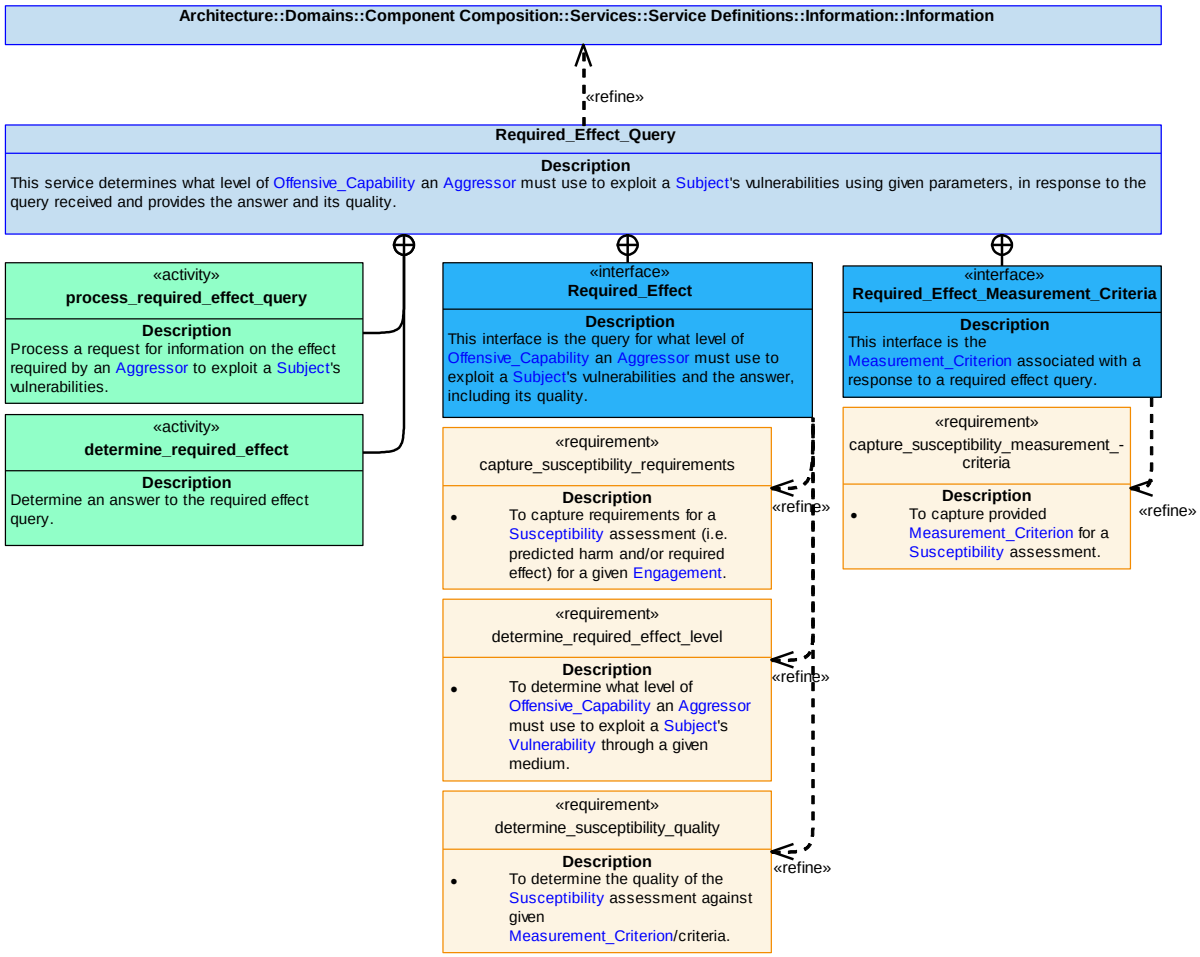

Figure 1073: Required_Effect Service Policy

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. E.g. 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.

B.2.60.7.1.3 Vulnerability_Query

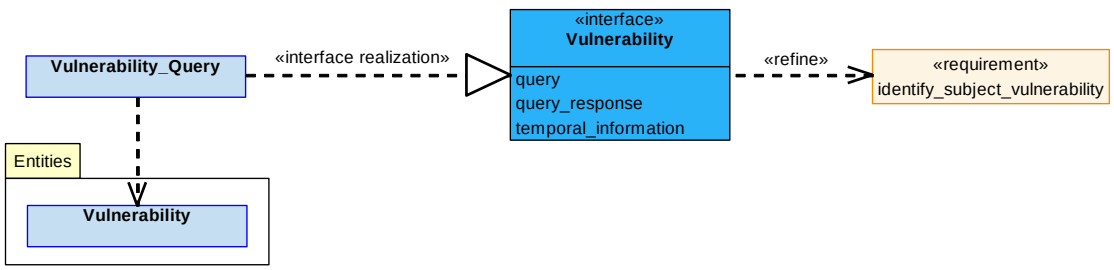

Figure 1074: Vulnerability_Query Service Definition

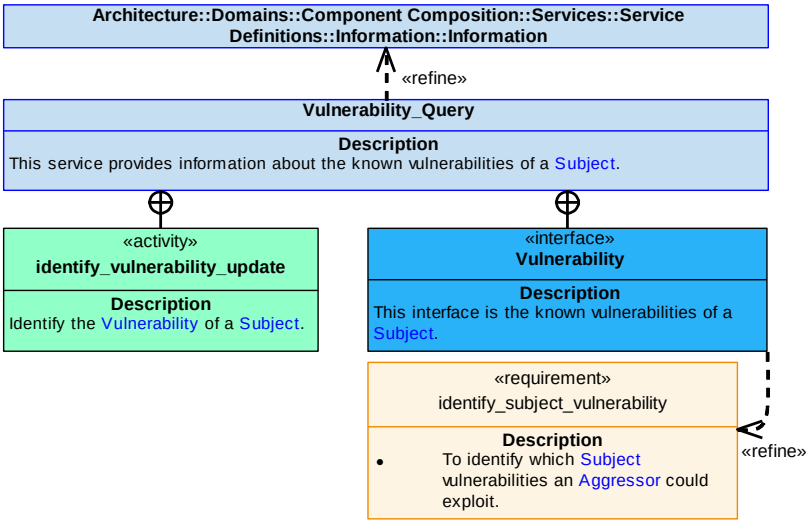

Figure 1075: Vulnerability_Query Service Policy

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.

B.2.60.7.1.4 Participant_Information

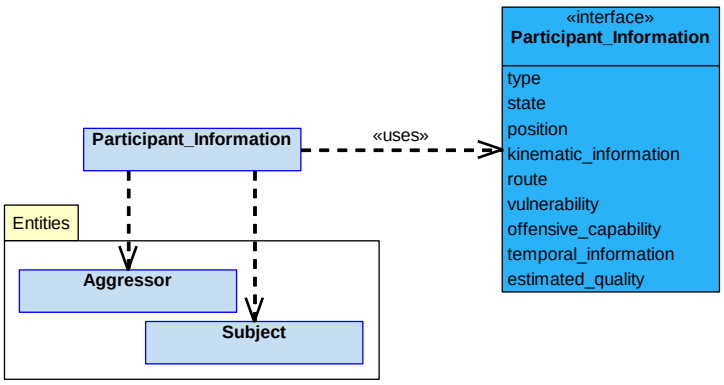

Figure 1076: Participant_Information Service Definition

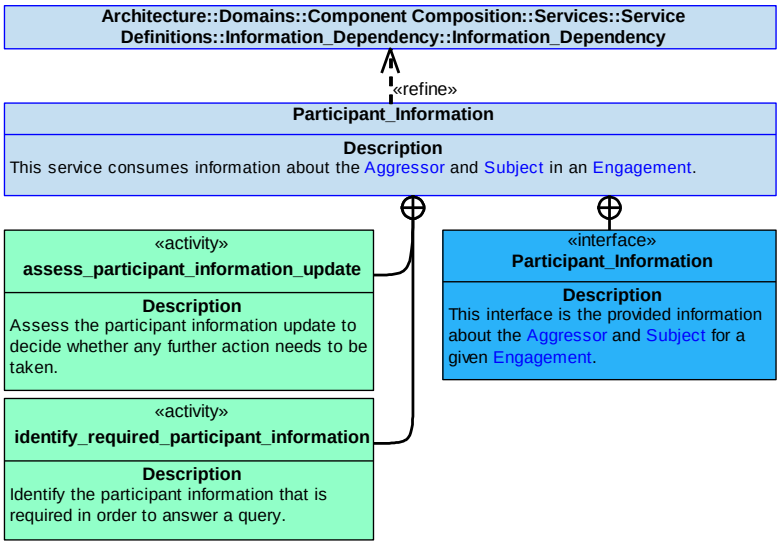

Figure 1077: Participant_Information Service Policy

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.

B.2.60.7.1.5 Engagement_Information

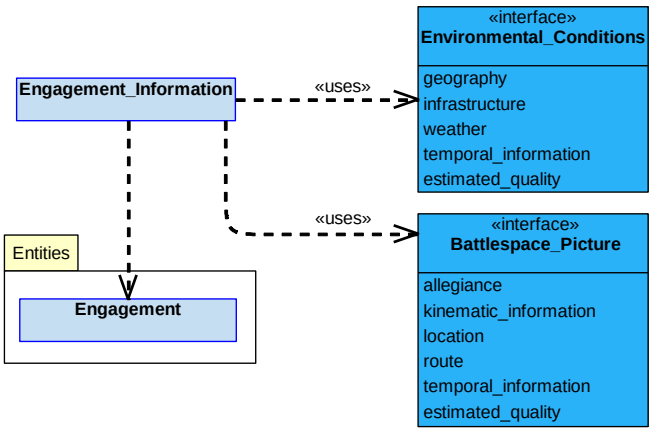

Figure 1078: Engagement_Information Service Definition

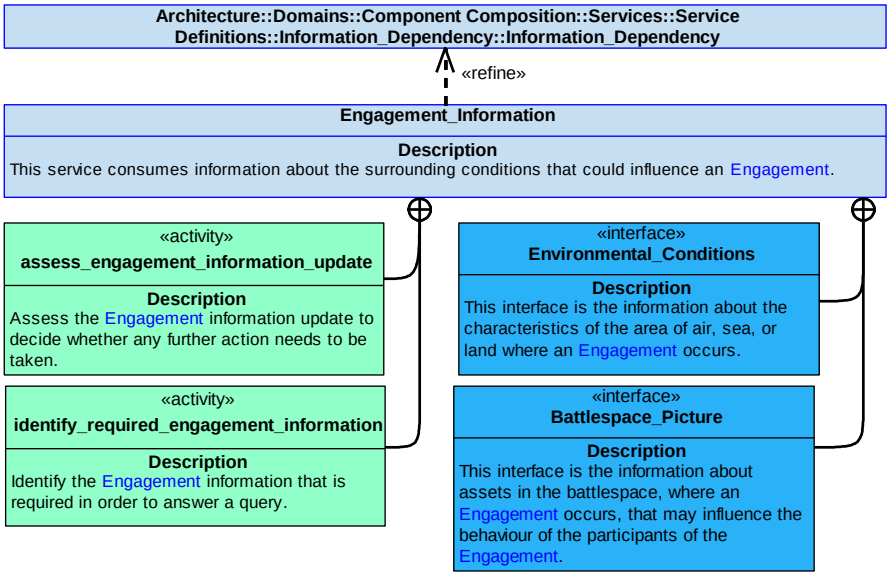

Figure 1079: Engagement_Information Service Policy

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.

B.2.60.7.1.6 Capability

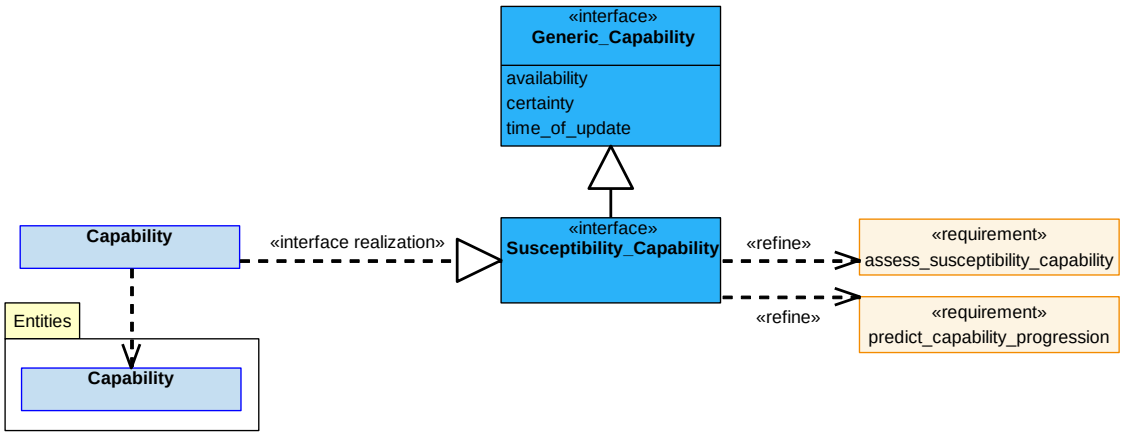

Figure 1080: Capability Service Definition

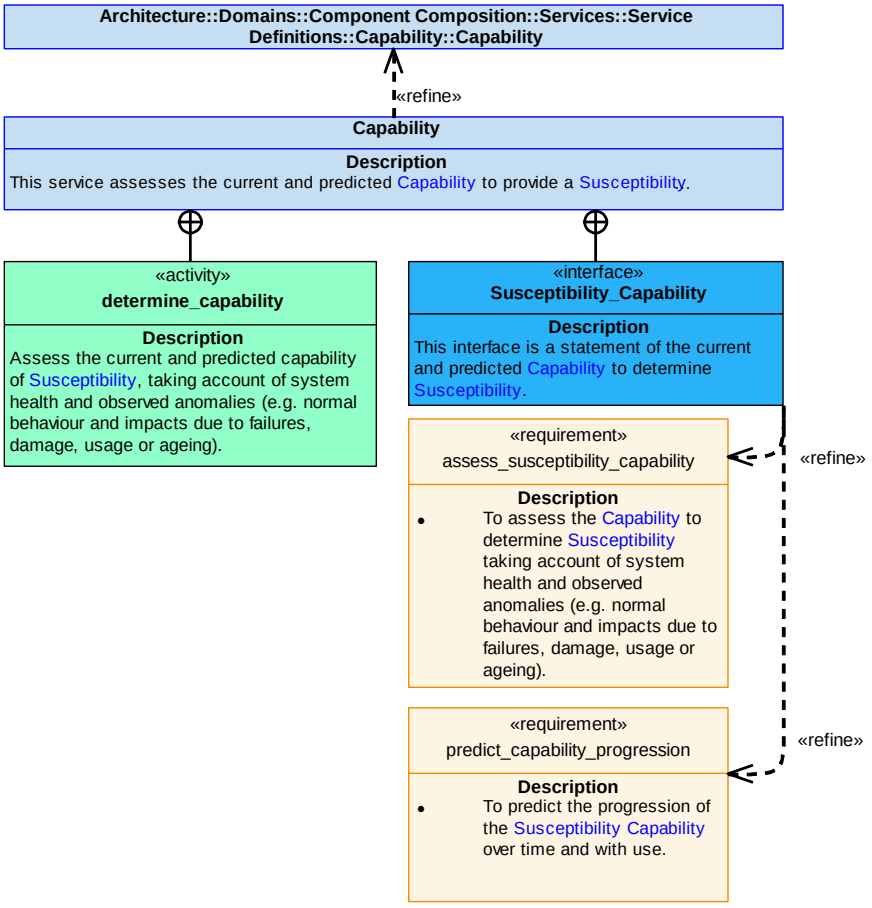

Figure 1081: Capability Service Policy

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).

B.2.60.7.1.7 Capability_Evidence

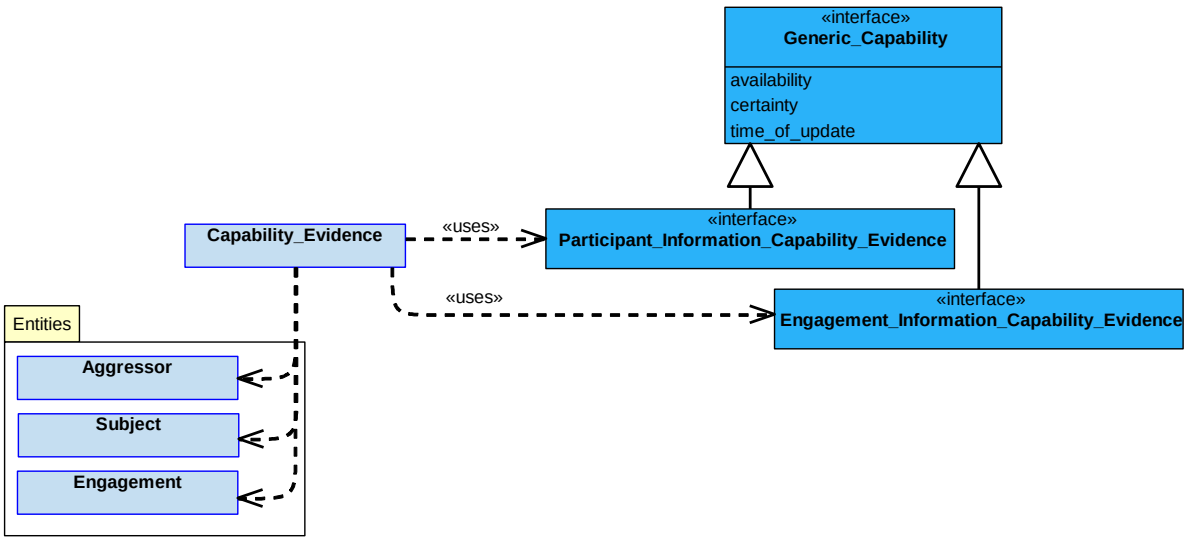

Figure 1082: Capability_Evidence Service Definition

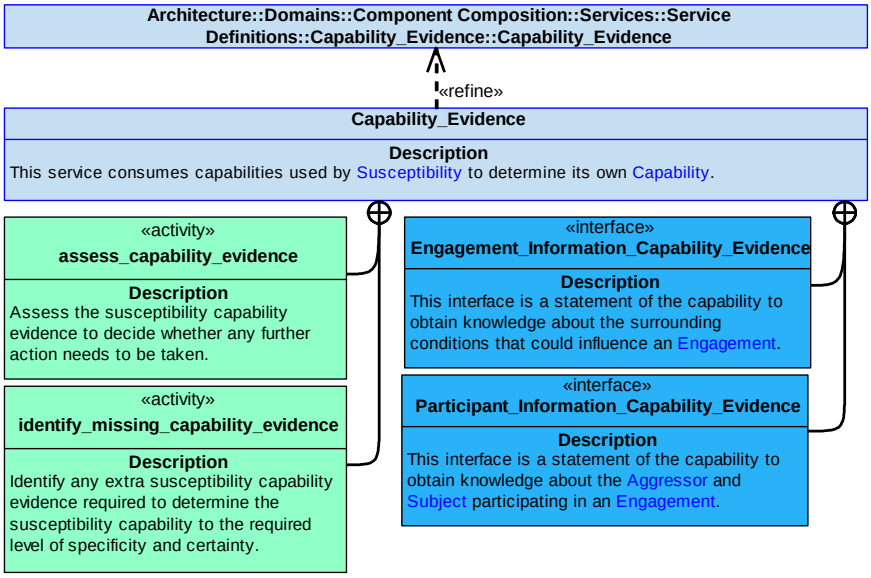

Figure 1083: Capability_Evidence Service Policy

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.

B.2.60.7.2 Service Dependencies

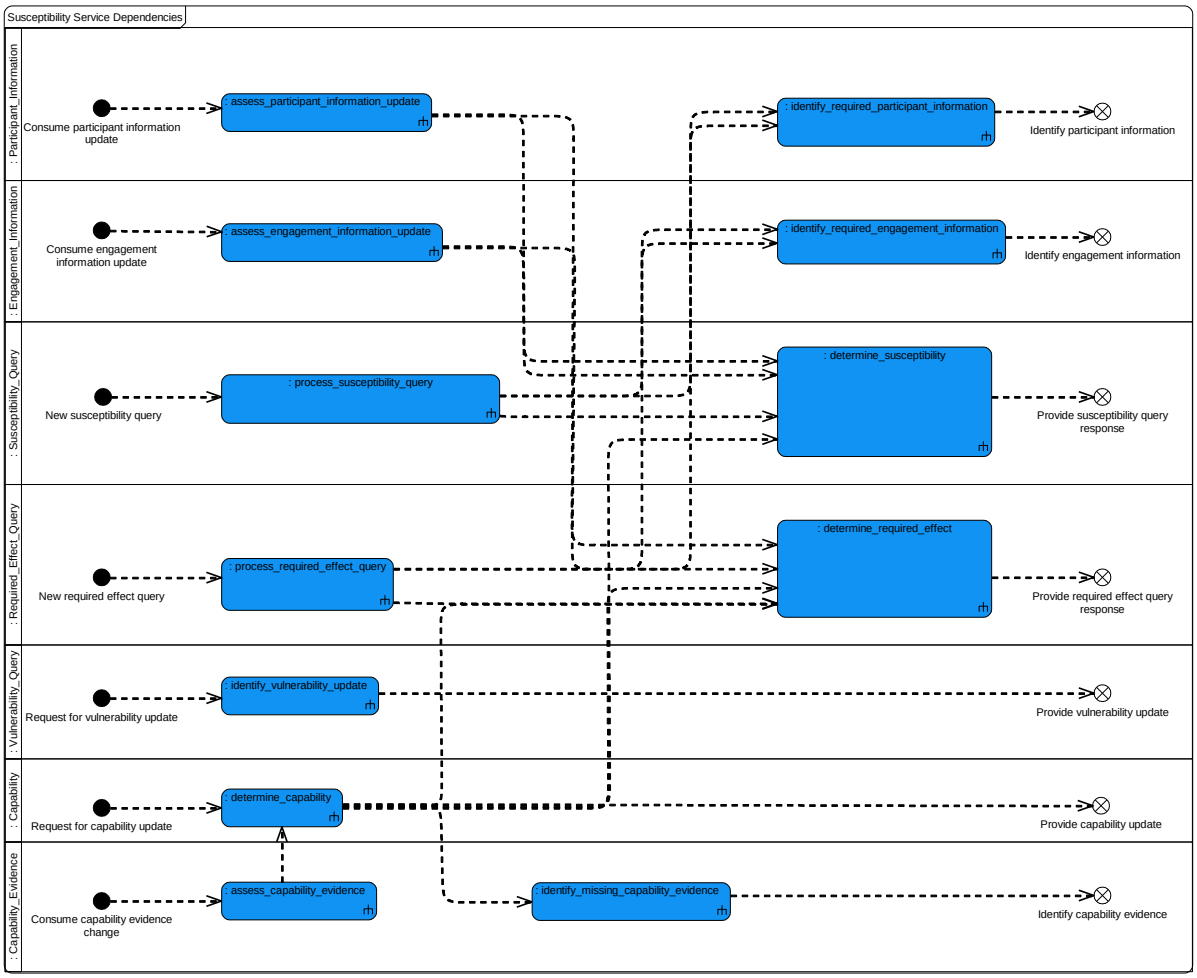

Figure 1084: Susceptibility Service Dependencies

B.2.61 Tactical Objects

B.2.61.1 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.

B.2.61.2 Overview

Control Architecture

Tactical Objects is a service component as defined in the Control Architecture policy.

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.

B.2.61.3 Service Summary

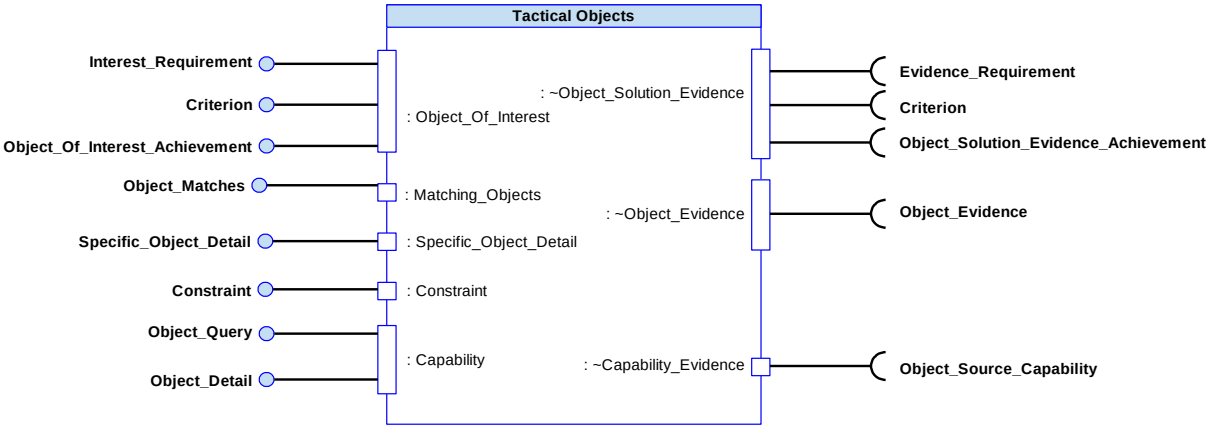

Figure 1085: Tactical Objects Service Summary

B.2.61.4 Responsibilities

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).

capture_measurement_criteria_for_tactical_object

- To capture provided Measurement_Criterion/criteria for Tactical_Objects.

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_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_object_probability_densities

- To capture the probability densities of particular Tactical_Objects within locations.

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).

determine_additional_information

- To determine additional information required to satisfy Object_Interest_Requirements, e.g. improved Tactical_Object confidence required.

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_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).

determine_potential_objects

- To determine the potential for Tactical_Objects to exist at locations.

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.

determine_requirement_solution

- To determine a solution to Object_Interest_Requirements.

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_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.

identify_whether_requirement_is_achievable

- To identify whether an Object_Interest_Requirement is still achievable given current or predicted Capability and Constraints.

identify_progress

- To identify the progress against an Object_Interest_Requirement.

predict_capability_progression

- To predict the progression of the Tactical Objects Capability over time and with use.

B.2.61.5 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.

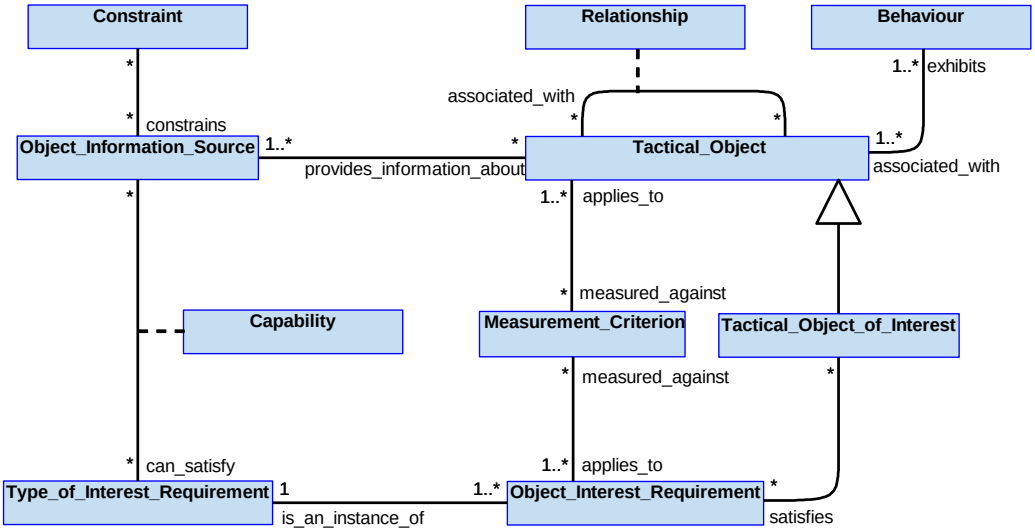

Figure 1086: Tactical Objects Semantics

B.2.61.5.1 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.

B.2.61.6 Design Rationale

B.2.61.6.1 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.

B.2.61.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy is applicable because Tactical_Object is classified as a tactical information component within this policy.

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.

B.2.61.6.3 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. In accordance with UK MoD direction (see Safety Analysis policy) this drives a DAL B indicative IDAL.

B.2.61.6.4 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.

B.2.61.7 Services

B.2.61.7.1 Service Definitions

B.2.61.7.1.1 Object_Of_Interest

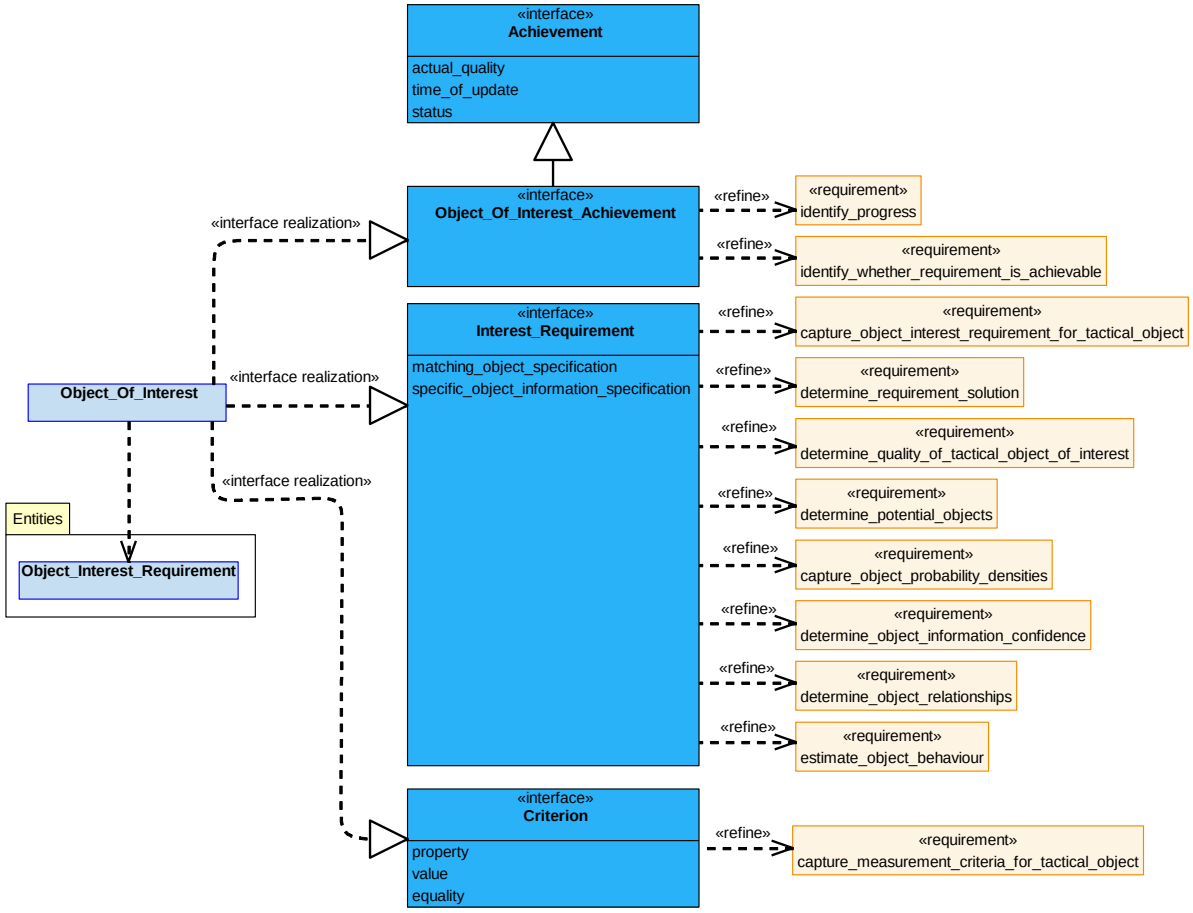

Figure 1087: Object_Of_Interest Service Definition

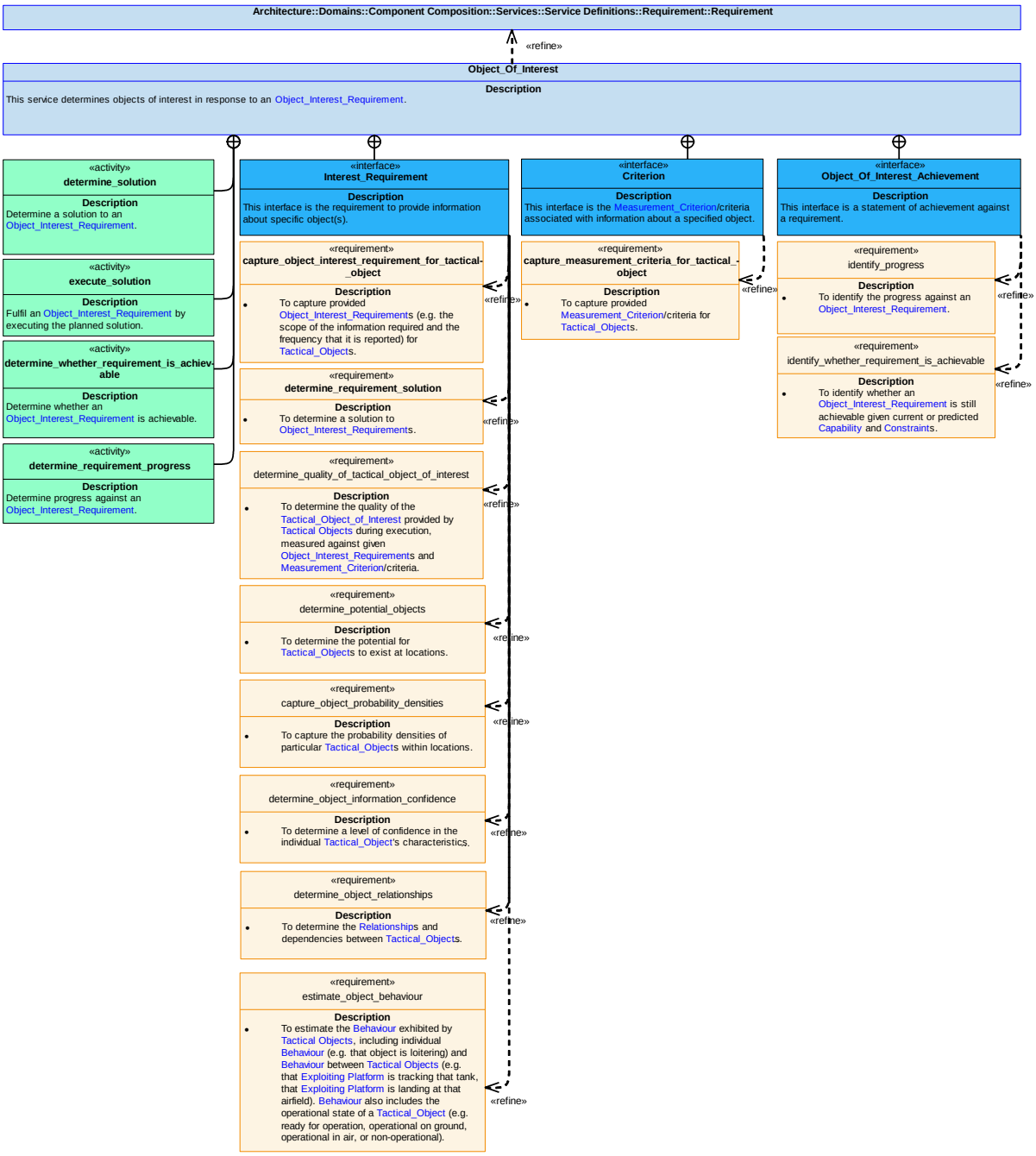

Figure 1088: Object_Of_Interest Service Policy

Object_Of_Interest

This service determines objects of interest in response to an Object_Interest_Requirement.

Interfaces

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.

B.2.61.7.1.2 Object_Solution_Evidence

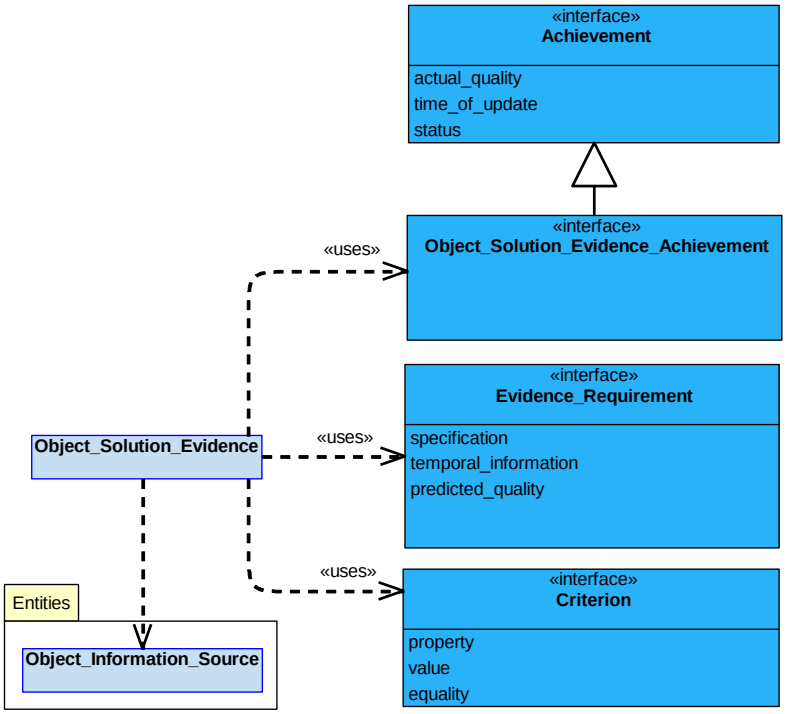

Figure 1089: Object_Solution_Evidence Service Definition

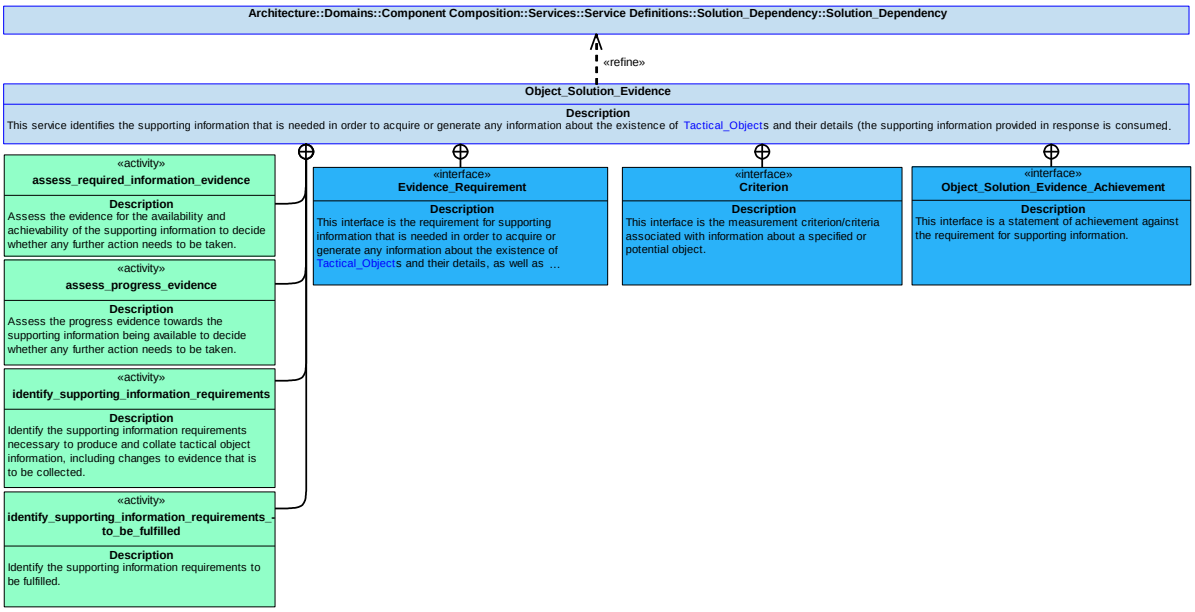

Figure 1090: Object_Solution_Evidence Service Policy

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.

B.2.61.7.1.3 Matching_Objects

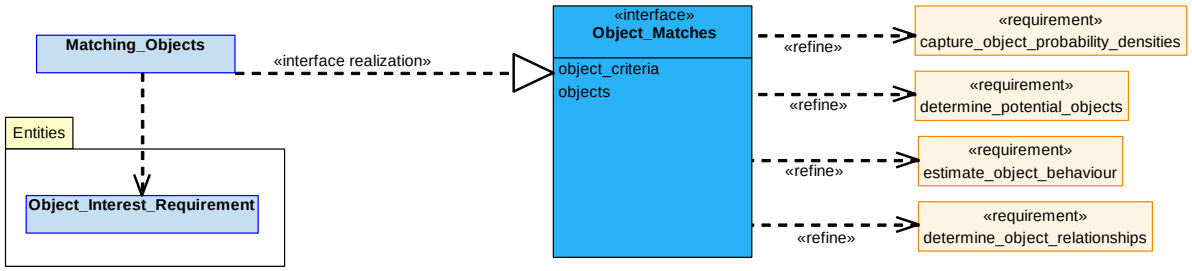

Figure 1091: Matching_Objects Service Definition

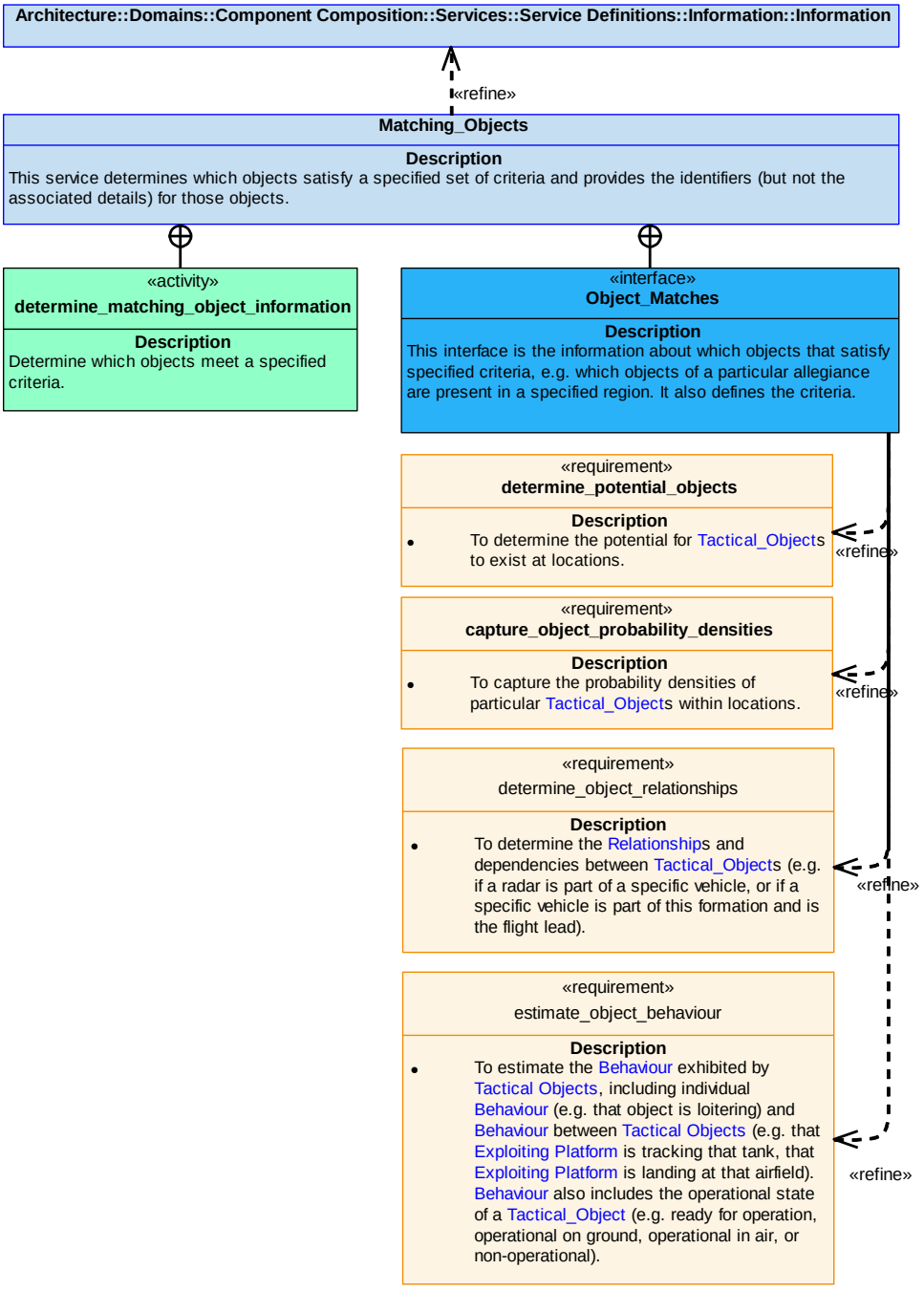

Figure 1092: Matching_Objects Service Policy

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 a specified criteria.

B.2.61.7.1.4 Specific_Object_Detail

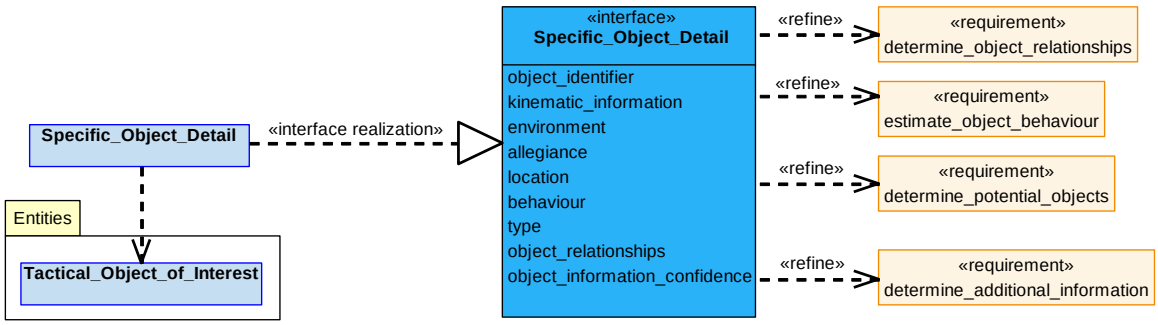

Figure 1093: Specific_Object_Detail Service Definition

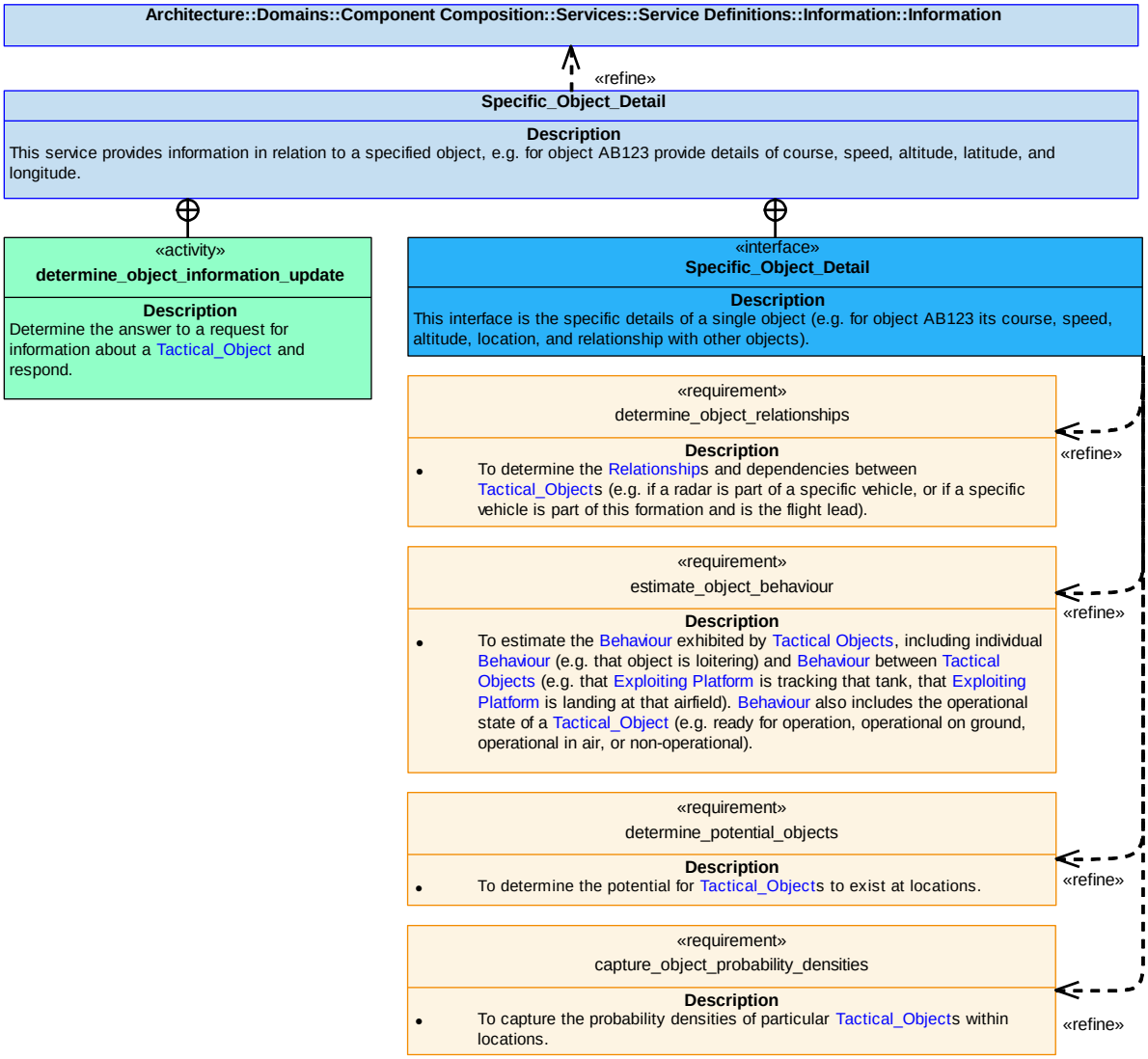

Figure 1094: Specific_Object_Detail Service Policy

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.

B.2.61.7.1.5 Object_Evidence

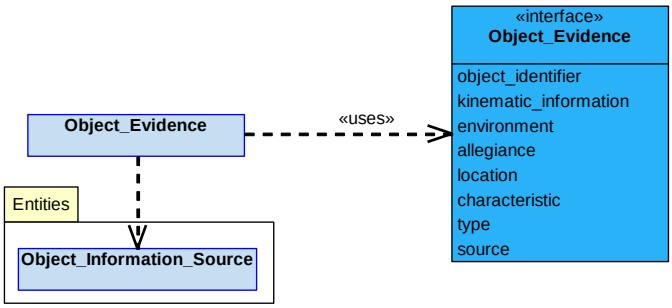

Figure 1095: Object_Evidence Service Definition

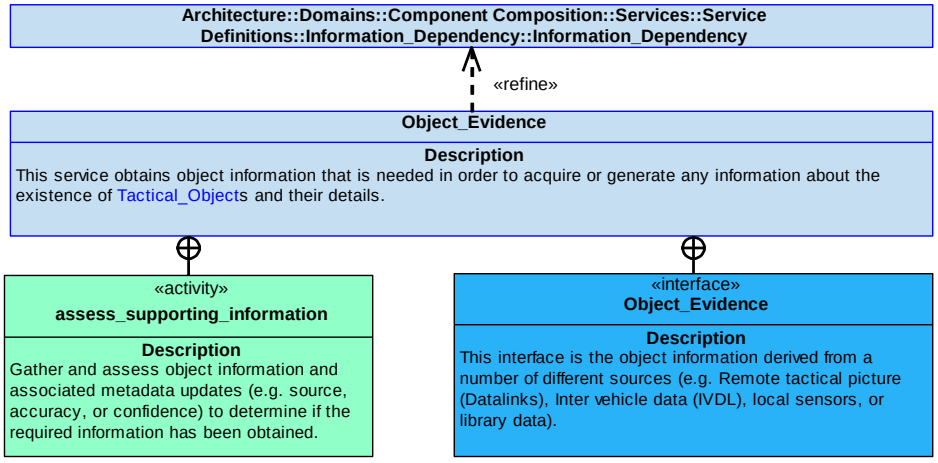

Figure 1096: Object_Evidence Service Policy

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.

B.2.61.7.1.6 Constraint

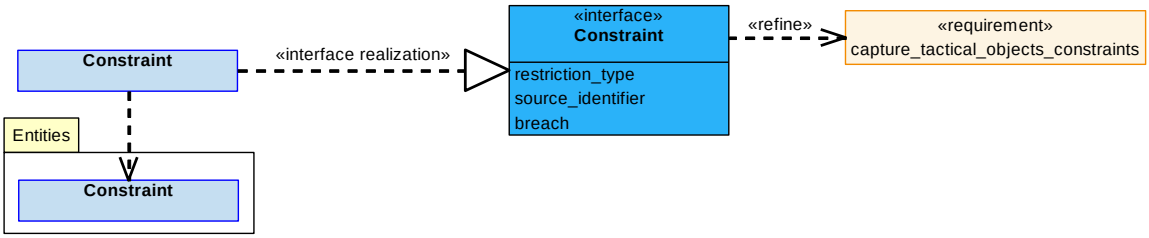

Figure 1097: Constraint Service Definition

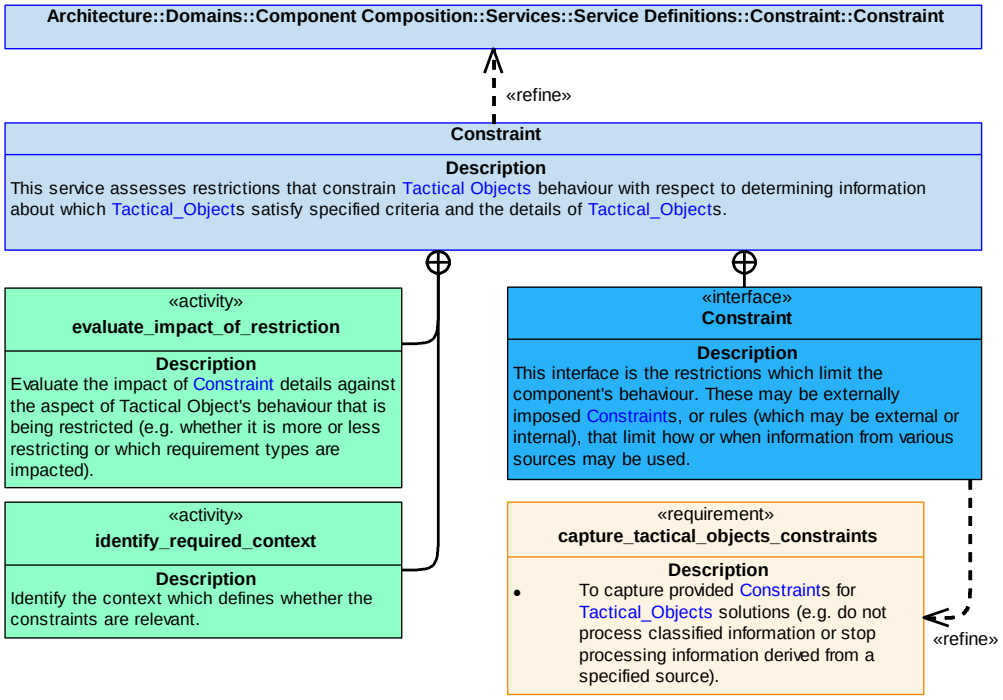

Figure 1098: Constraint Service Policy

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

This interface is the restrictions which limit the component's behaviour. These may be externally imposed Constraints, or rules (which may be external or internal), that limit how or when information from various sources may be used.

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.

B.2.61.7.1.7 Capability

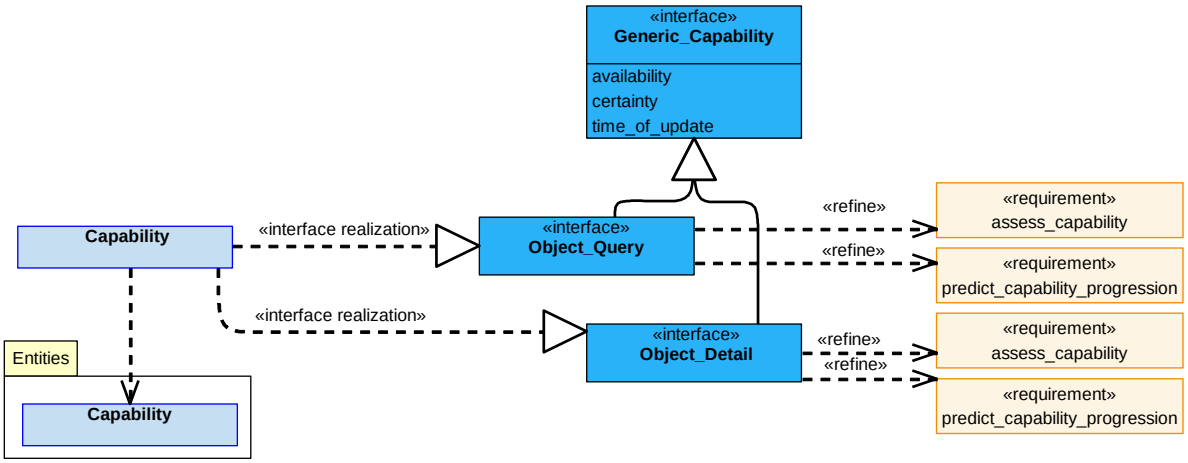

Figure 1099: Capability Service Definition

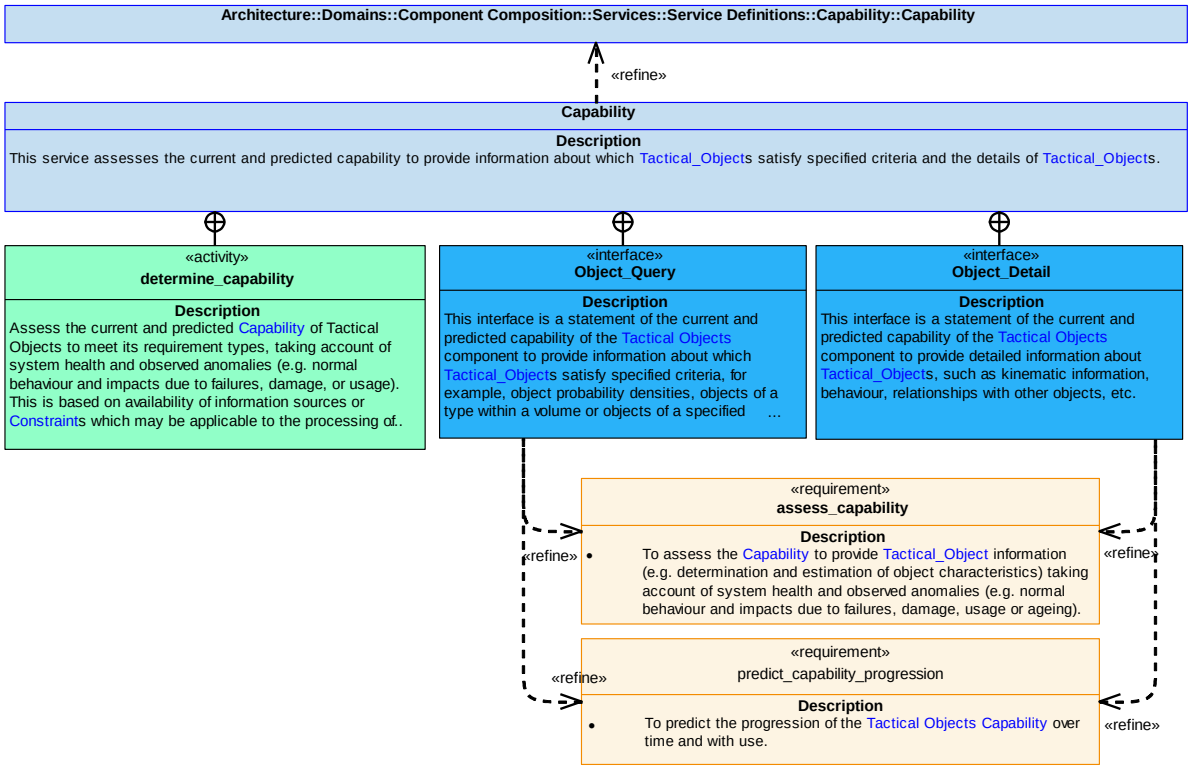

Figure 1100: Capability Service Policy

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.

B.2.61.7.1.8 Capability_Evidence

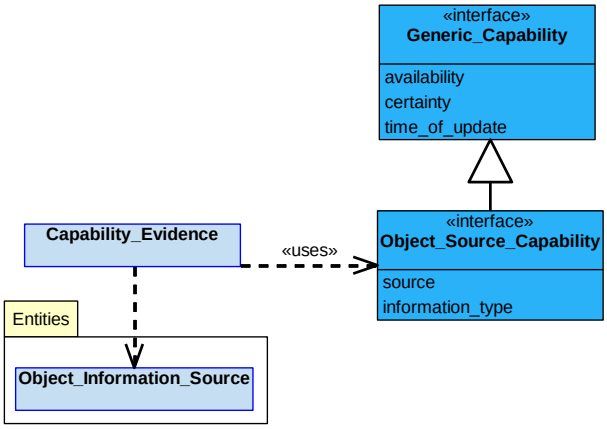

Figure 1101: Capability_Evidence Service Definition

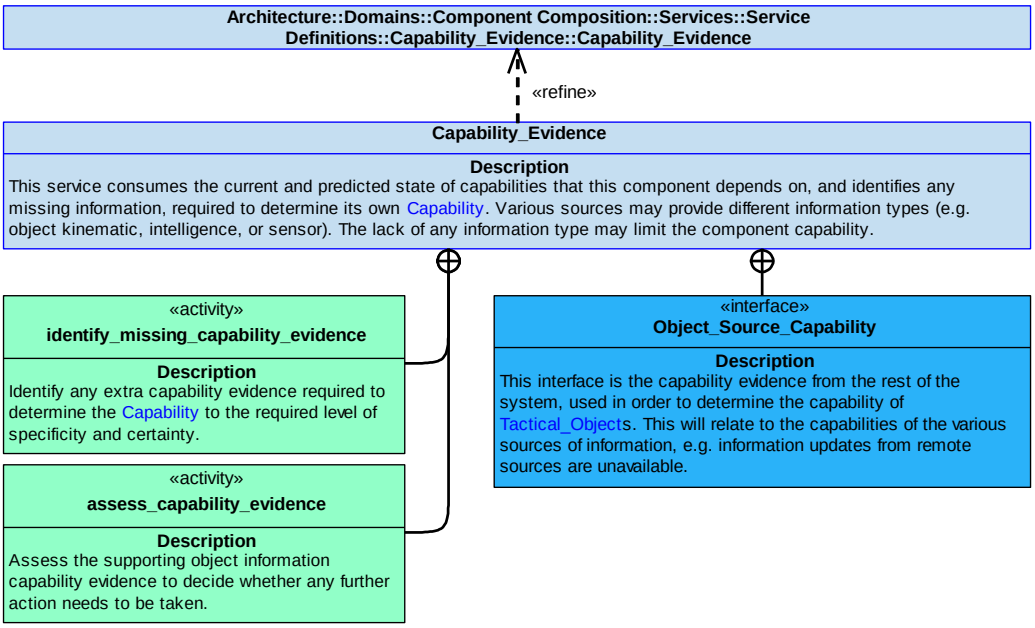

Figure 1102: Capability_Evidence Service Policy

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, 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.

B.2.61.7.2 Service Dependencies

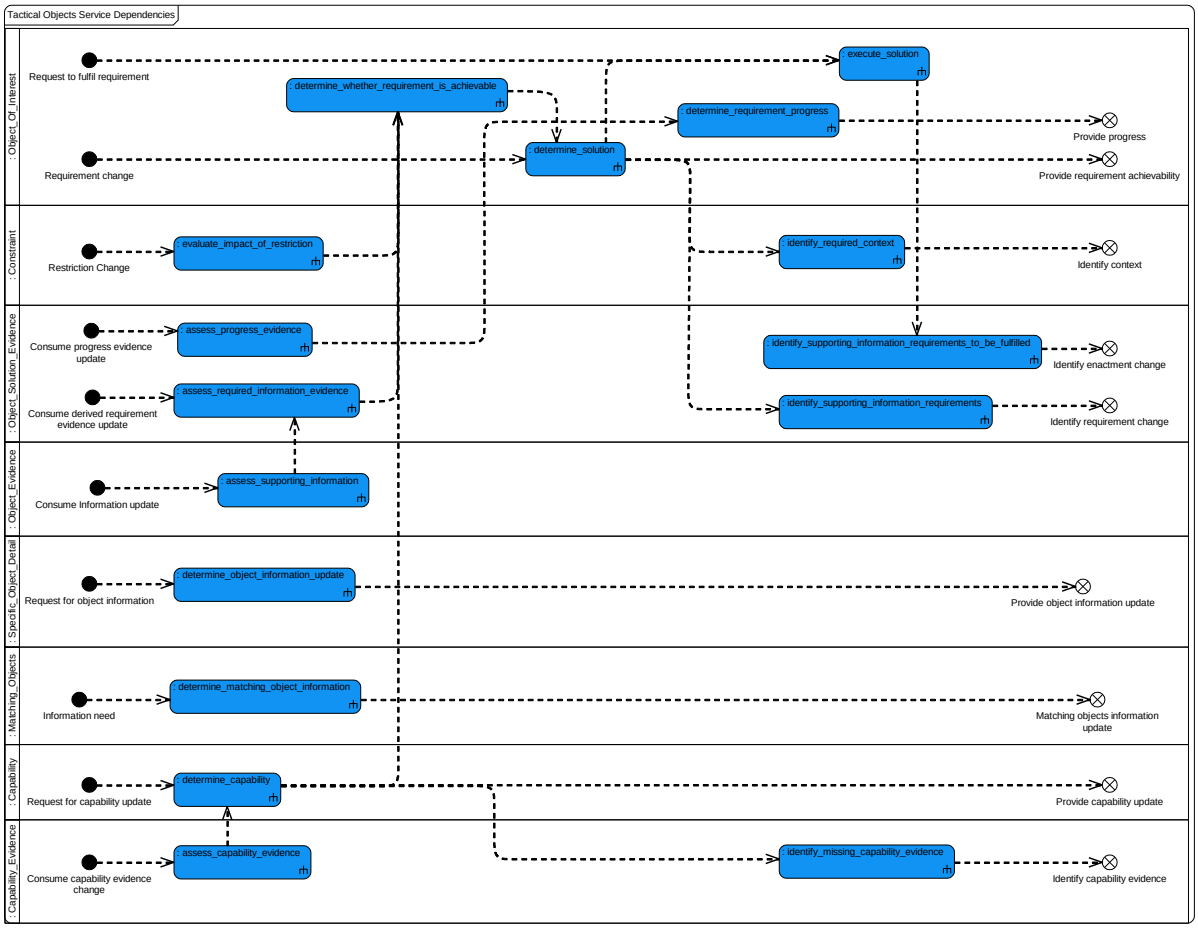

Figure 1103: Tactical Objects Service Dependencies

B.2.62 Target Engagement

B.2.62.1 Role

The role of Target Engagement is to influence, disrupt, damage, destroy, mark, or deliver physical assets to, a target.

B.2.62.2 Overview

Control Architecture

Target Engagement is an action component as defined in the Control Architecture policy.

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 targets 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.

B.2.62.3 Service Summary

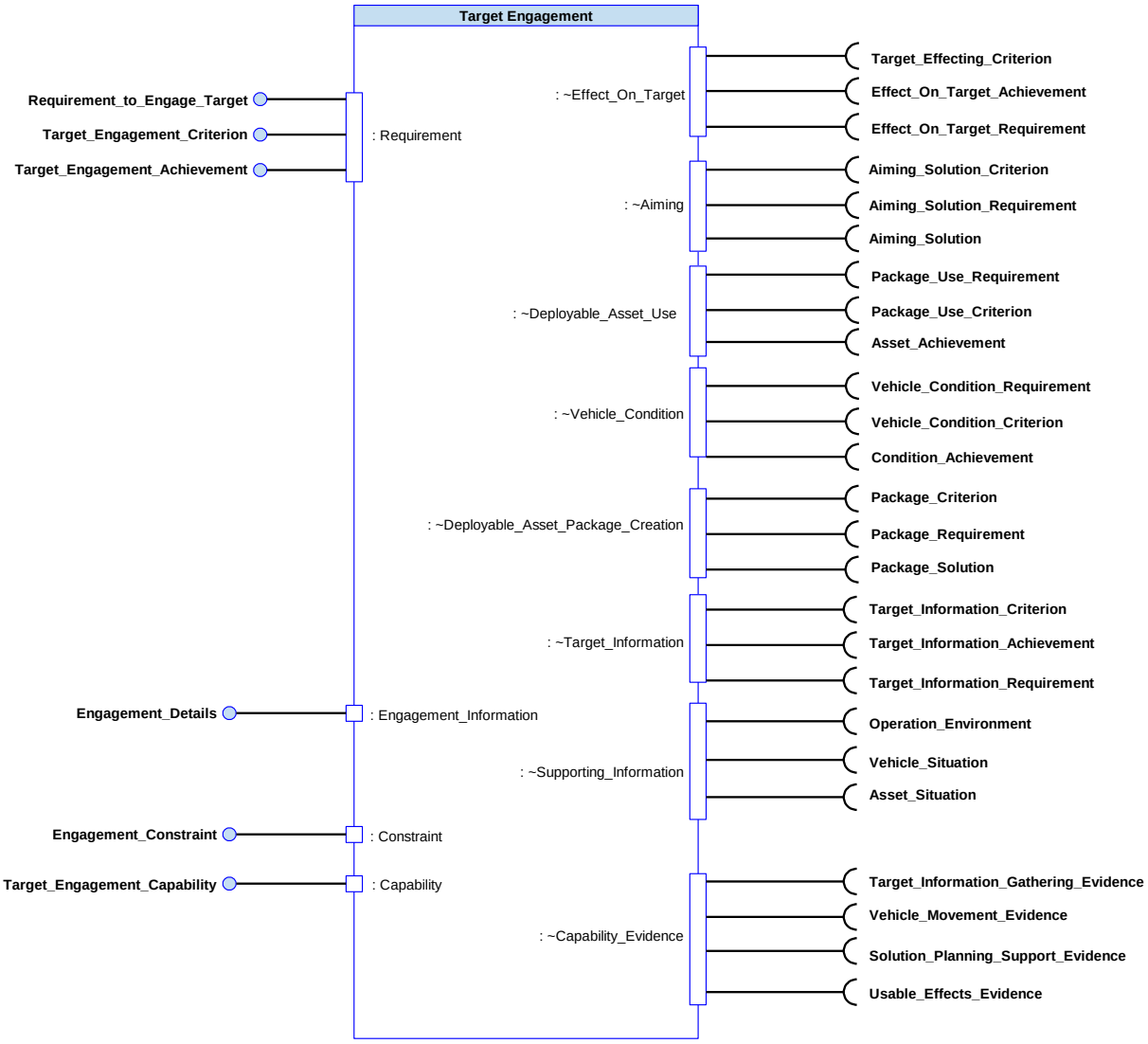

Figure 1104: Target Engagement Service Summary

B.2.62.4 Responsibilities

capture_requirements_for_target_engagement

- To capture provided Requirements (including 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.

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).

predict_progression_of_capability

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

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.

identify_whether_requirement_remains_achievable

- To identify whether a Requirement is still achievable given current Target_Engagement_Resources and Constraints.

identify_missing_information

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

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.

capture_constraints_for_target_engagement

- To capture provided Constraints for target engagement.

B.2.62.5 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.

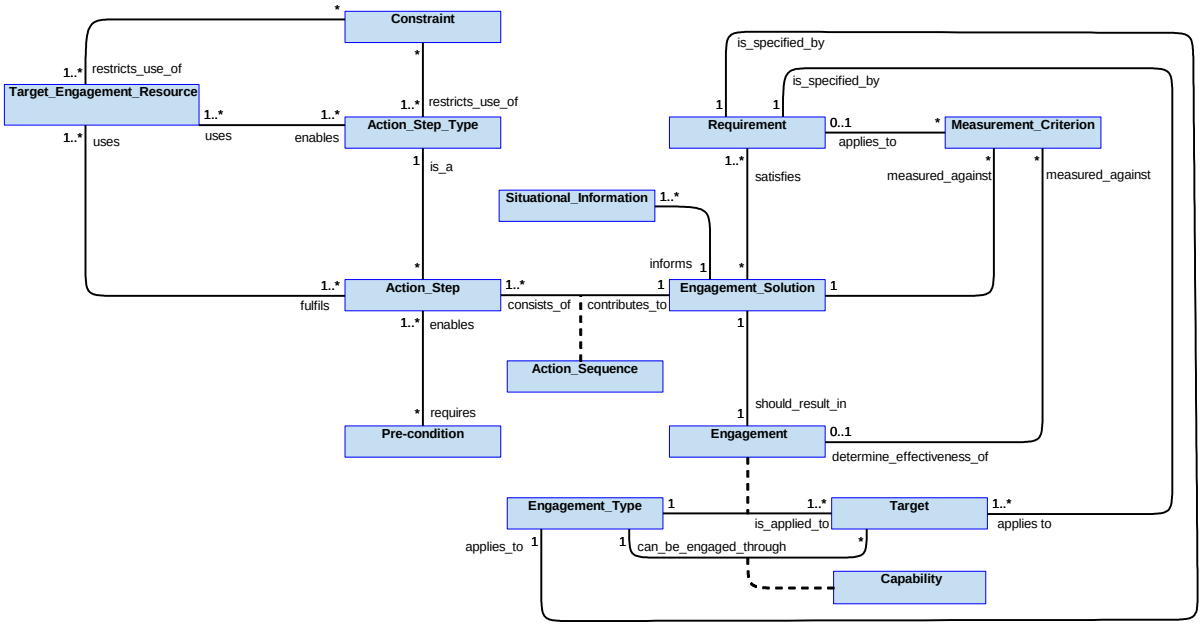

Figure 1105: Target Engagement Semantics

B.2.62.5.1 Entities

Action_Sequence

The order in which action-steps must be performed to achieve an Engagement_Solution.

Action_Step

An activity that contributes to the engagement of a Target (either fully or in part).

Action_Step_Type

A type of activity that can contribute to the engagement of a target, either fully or in part (e.g. an activity contributing to the use of a weapon or effector, or an activity 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 activity 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.

B.2.62.6 Design Rationale

B.2.62.6.1 Assumptions

- The types of weapons, effectors, or deployable assets available will be updated rarely (e.g. would 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.

B.2.62.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Target Engagement:

- Data Driving - This policy 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 policy 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 policies.

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 policy).

B.2.62.6.3 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. In accordance with UK MOD direction (see Safety Analysis policy) this drives a DAL B indicative IDAL.

B.2.62.6.4 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.

B.2.62.7 Services

B.2.62.7.1 Service Definitions

B.2.62.7.1.1 Requirement

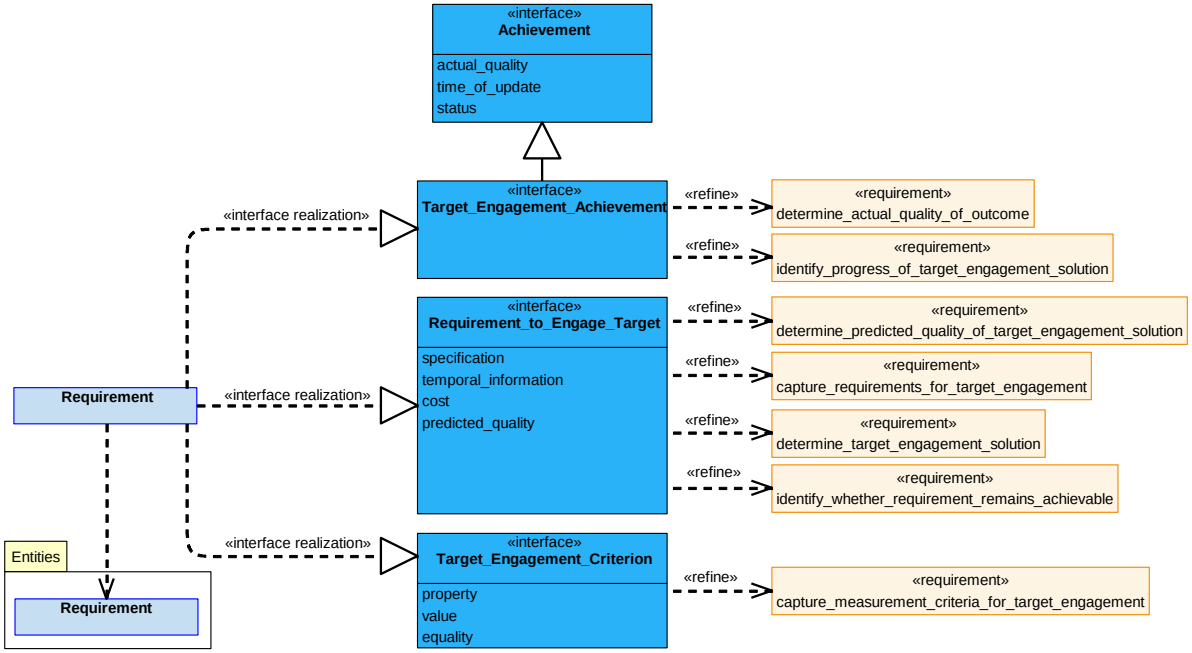

Figure 1106: Requirement Service Definition

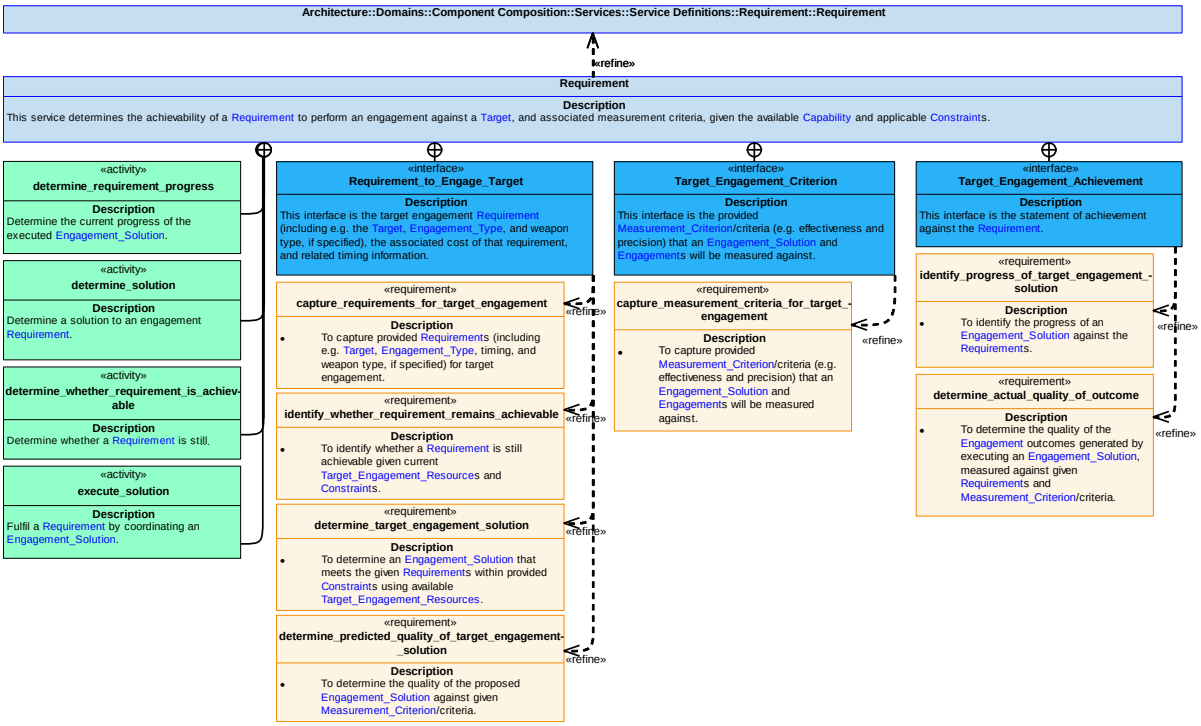

Figure 1107: Requirement Service Policy

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 (including e.g. the Target, Engagement_Type, and weapon type, 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 Target Engagement Requirement is still achievable.

B.2.62.7.1.2 Effect_On_Target

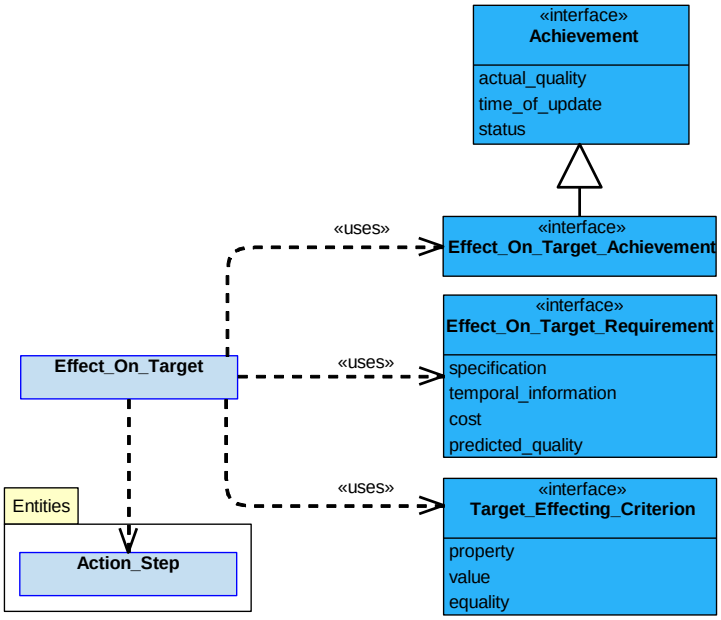

Figure 1108: Effect_On_Target Service Definition

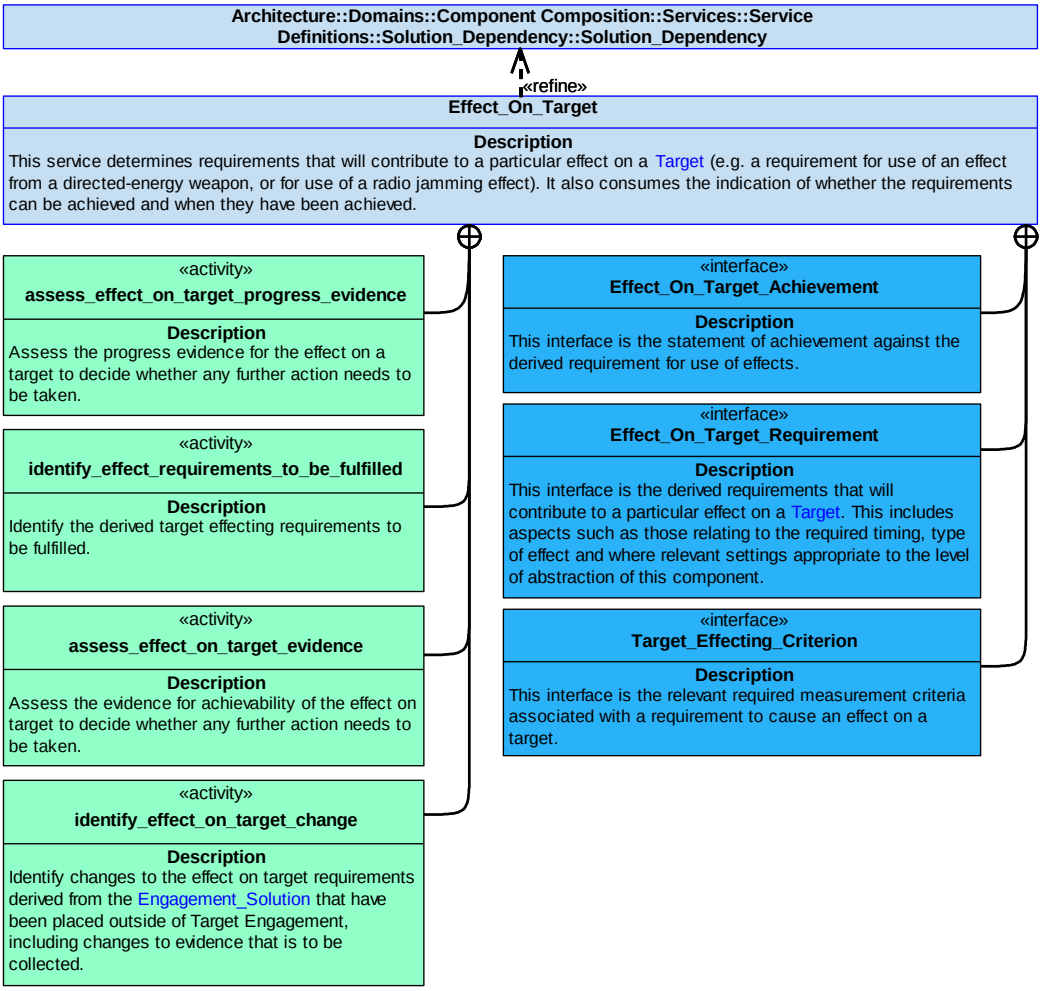

Figure 1109: Effect_On_Target Service Policy

Effect_On_Target

This service determines requirements that will contribute to a particular effect on a Target (e.g. a requirement for use of an effect from a directed-energy weapon, 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

Target_Effecting_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).

Effect_On_Target_Requirement

This interface is the derived requirements that will contribute to a particular effect on a Target. 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.

Effect_On_Target_Achievement

This interface is the statement of achievement against the derived requirement for use of effects.

Activities

assess_effect_on_target_progress_evidence

Assess the progress evidence for the effect on a target to decide whether any further action needs to be taken.

identify_effect_on_target_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 evidence that is to be collected.

identify_effect_requirements_to_be_fulfilled

Identify the derived target effecting requirements to be fulfilled.

assess_effect_on_target_evidence

Assess the evidence for achievability of the effect on target to decide whether any further action needs to be taken.

B.2.62.7.1.3 Aiming

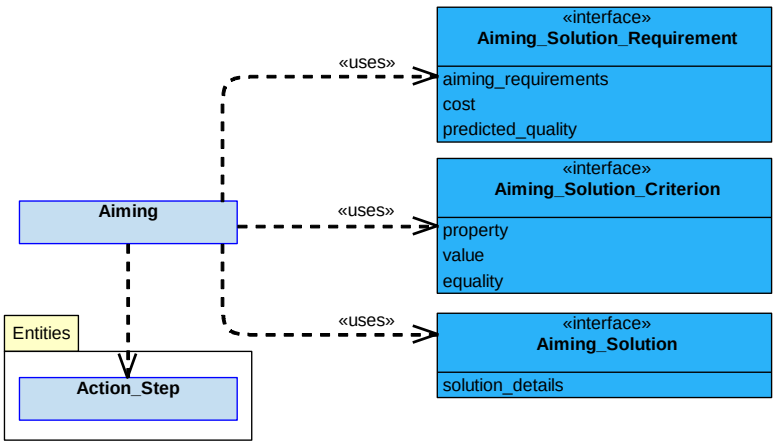

Figure 1110: Aiming Service Definition

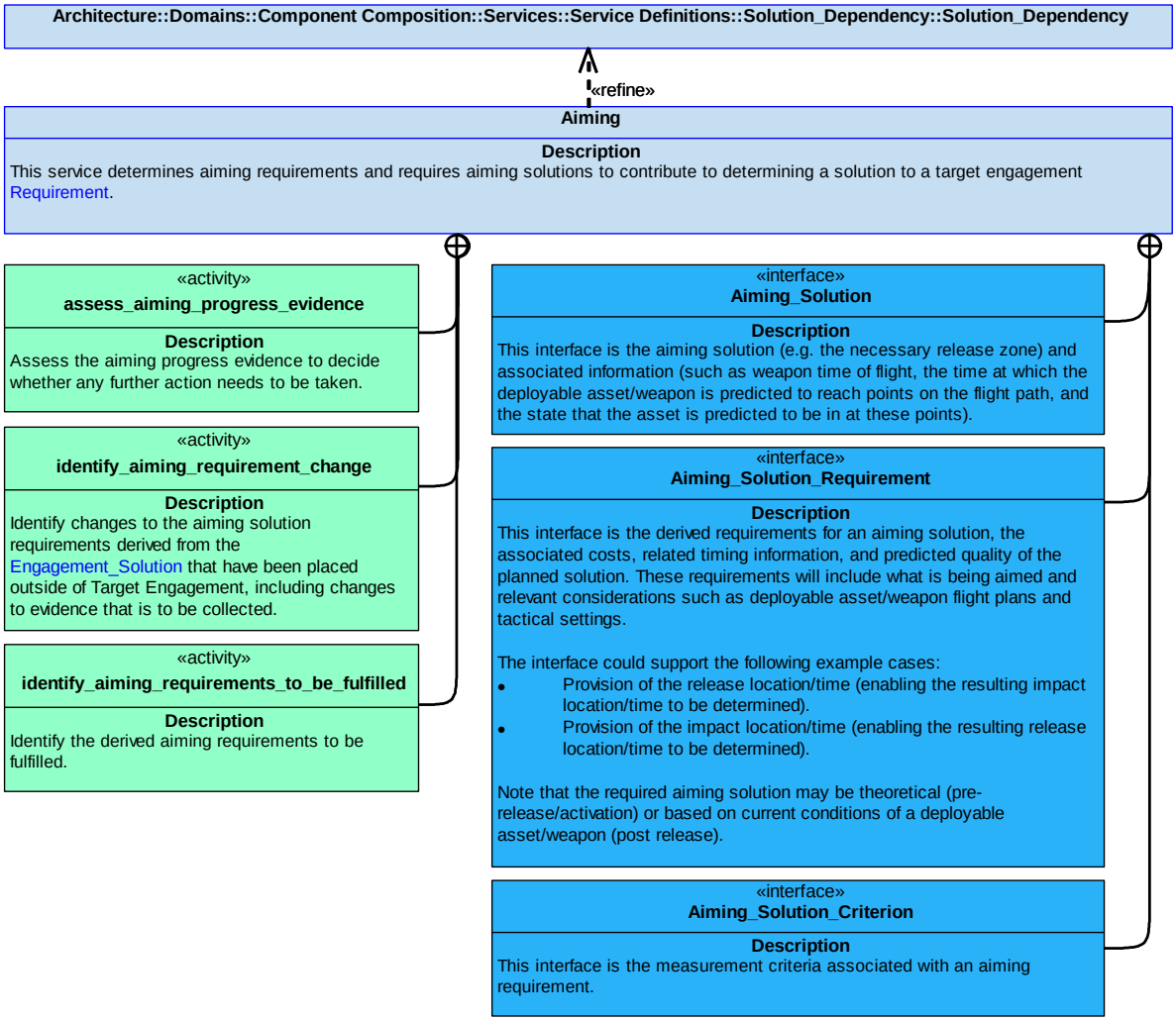

Figure 1111: Aiming Service Policy

Aiming

This service determines aiming requirements and requires aiming solutions to contribute to determining a solution to a target engagement Requirement.

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.

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.

B.2.62.7.1.4 Deployable_Asset_Use

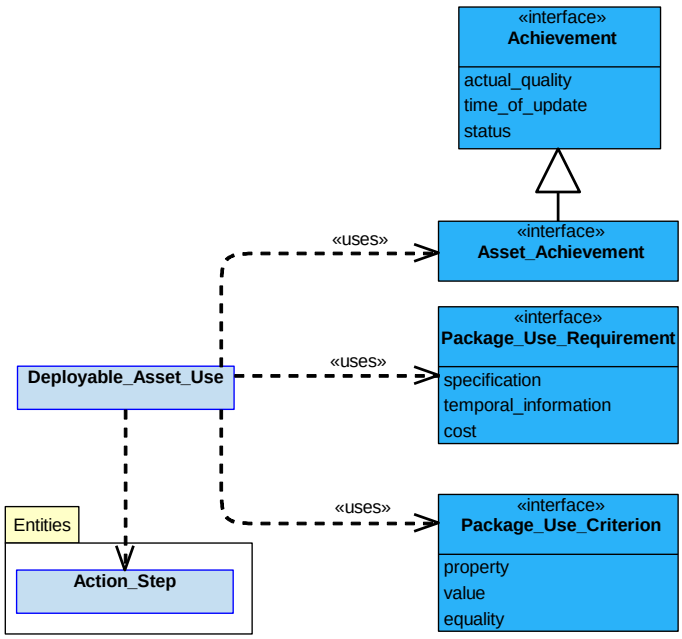

Figure 1112: Deployable_Asset_Use Service Definition

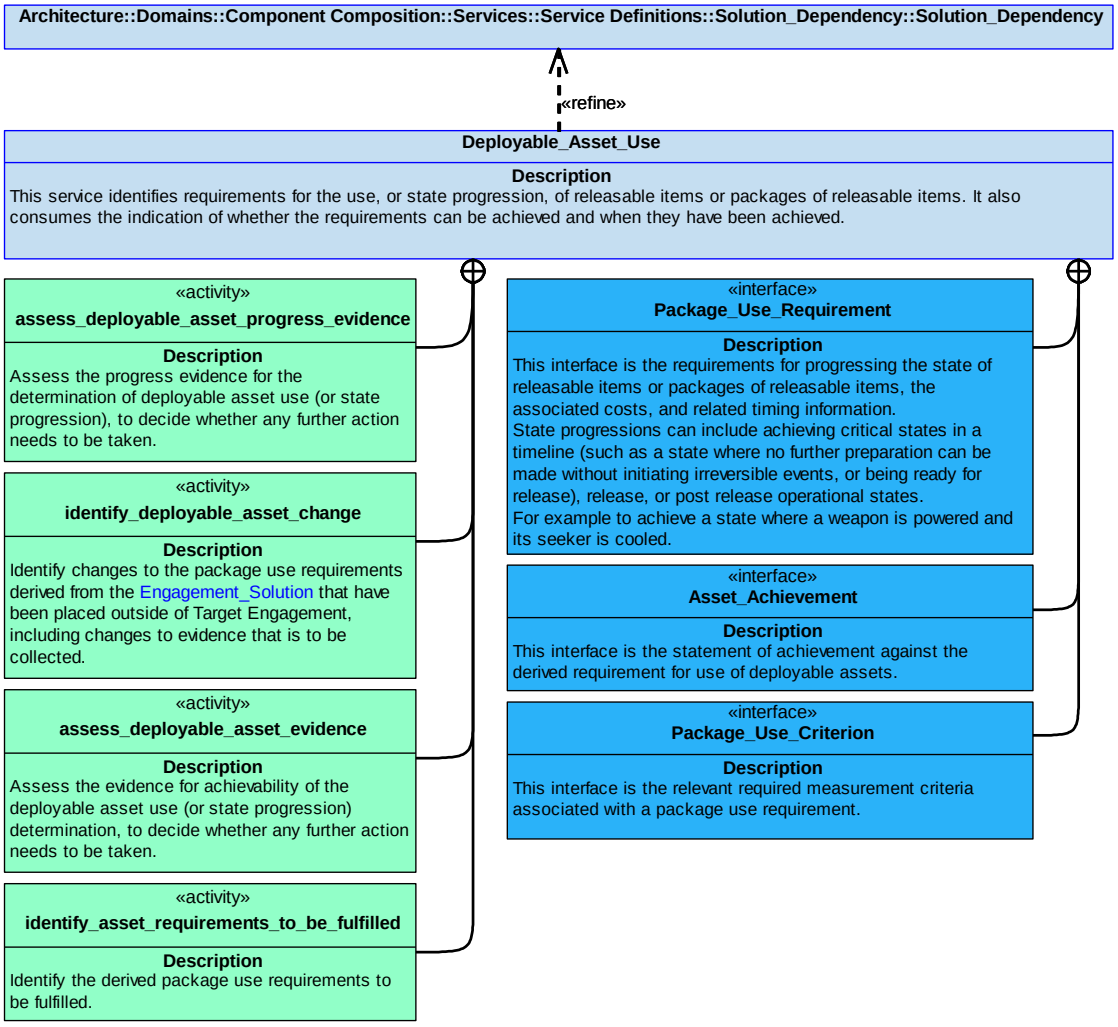

Figure 1113: Deployable_Asset_Use Service Policy

Deployable_Asset_Use

This service identifies requirements for the use, or state progression, of releasable items or packages of releasable items. It also consumes the indication of whether the requirements can be achieved and when they have been achieved.

Interfaces

Package_Use_Criterion

This interface is the relevant required measurement criteria associated with a package 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).

Package_Use_Requirement

This interface is the requirements for progressing the state of releasable items or packages of releasable items, 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.

For example to achieve a state where a 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 deployable assets.

Activities

assess_deployable_asset_progress_evidence

Assess the progress evidence for the determination of deployable asset use (or state progression), to decide whether any further action needs to be taken.

identify_deployable_asset_change

Identify changes to the package 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_requirements_to_be_fulfilled

Identify the derived package use requirements to be fulfilled.

assess_deployable_asset_evidence

Assess the evidence for achievability of the deployable asset use (or state progression) determination, to decide whether any further action needs to be taken.

B.2.62.7.1.5 Vehicle_Condition

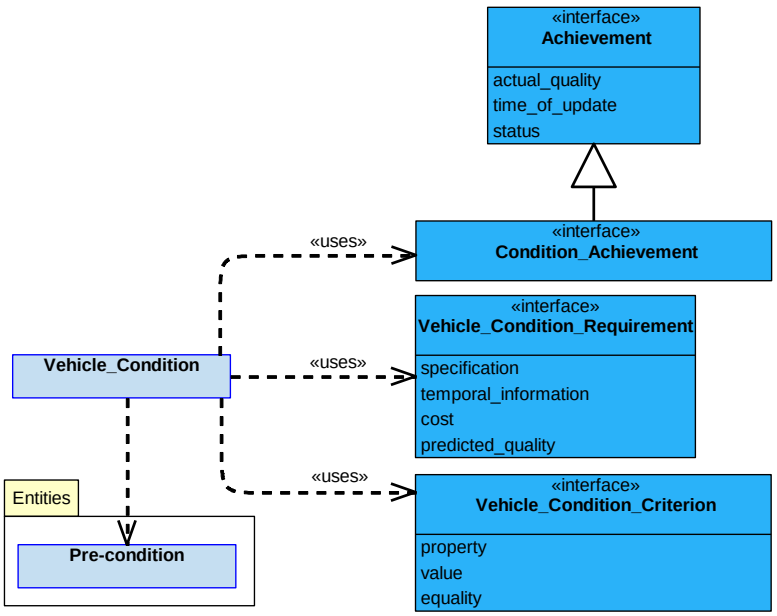

Figure 1114: Vehicle_Condition Service Definition

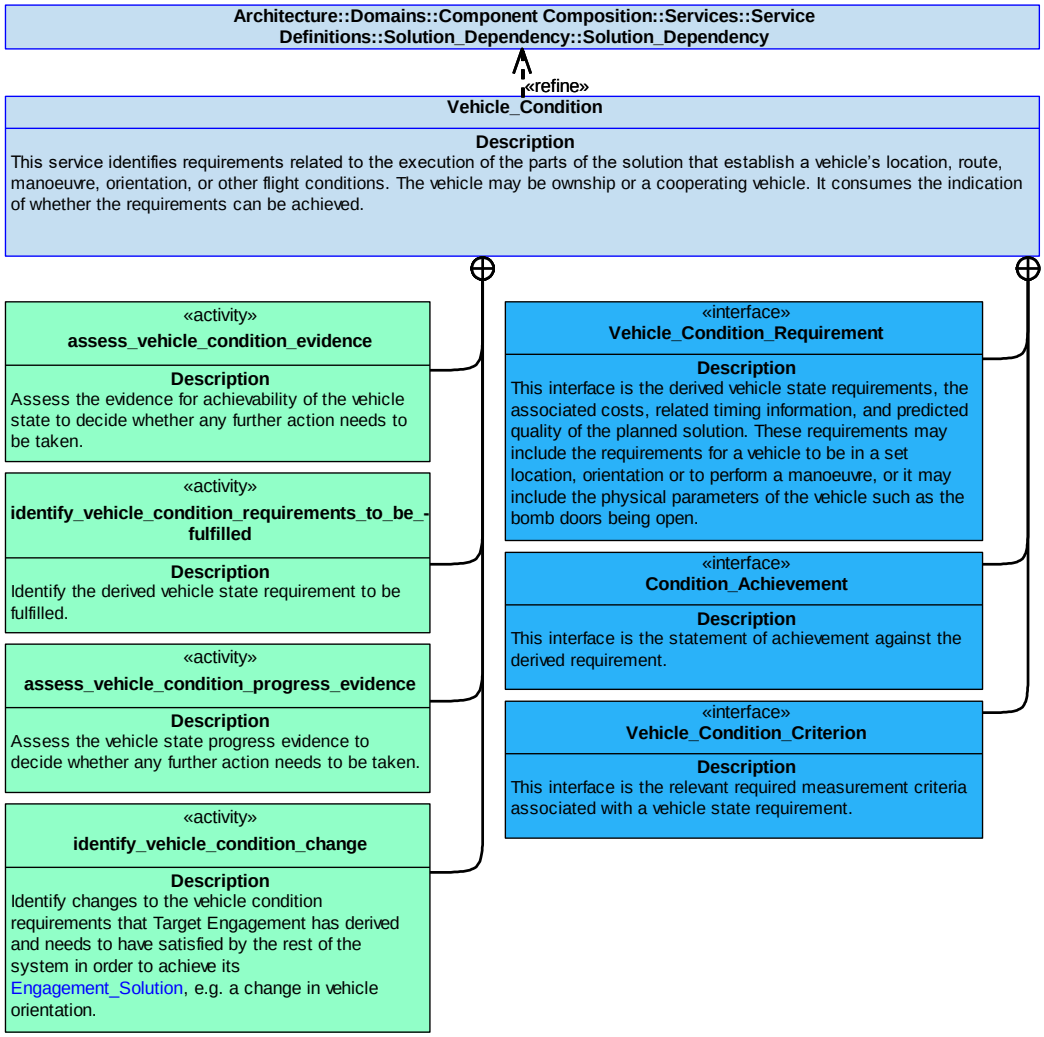

Figure 1115: Vehicle_Condition Service Policy

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.

B.2.62.7.1.6 Deployable_Asset_Package_Creation

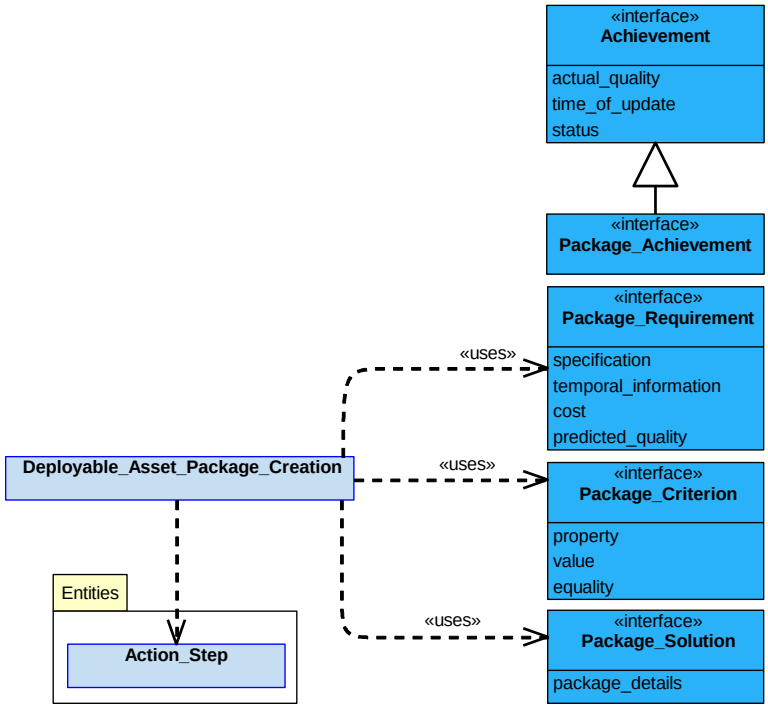

Figure 1116: Deployable_Asset_Package_Creation Service Definition

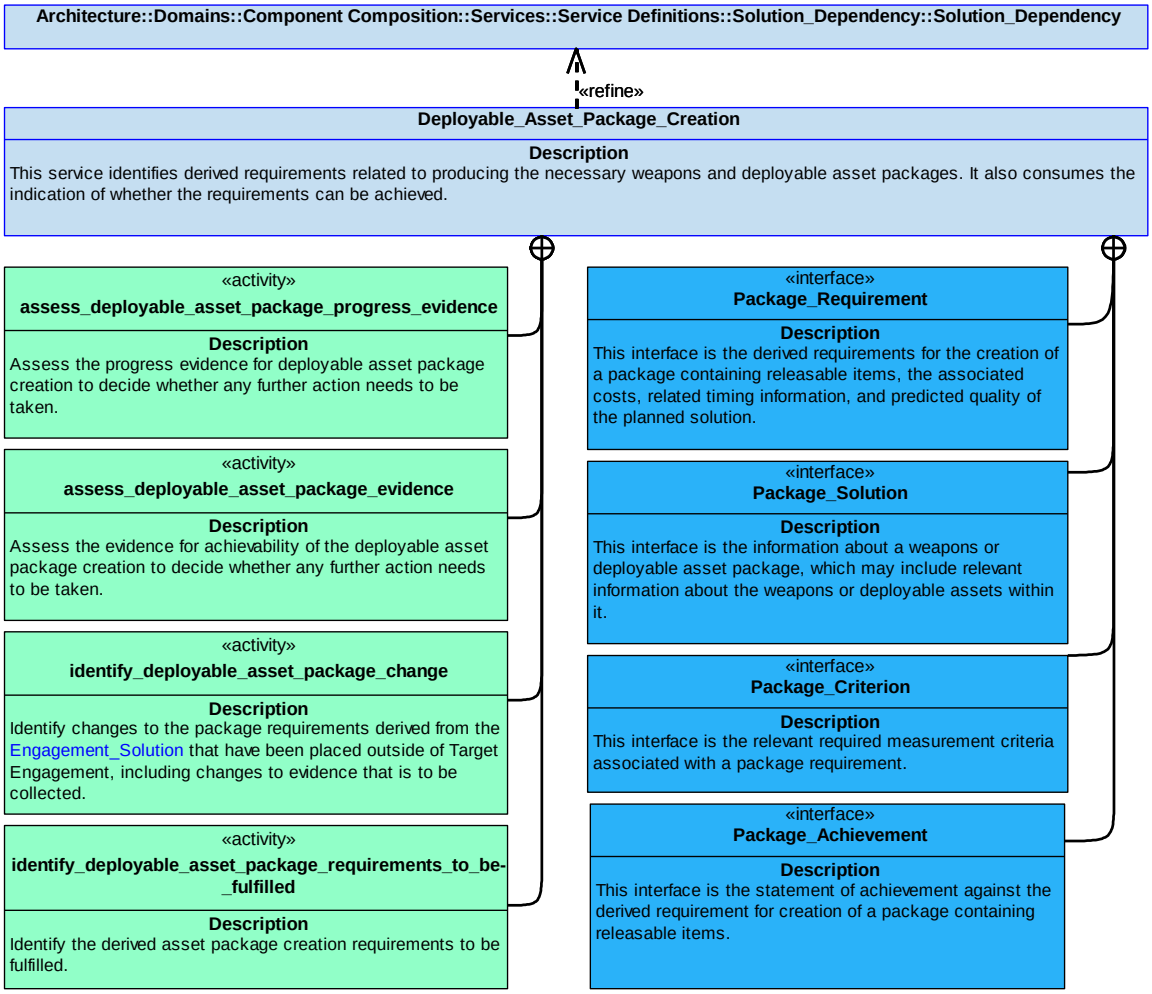

Figure 1117: Deployable_Asset_Package_Creation Service Policy

Deployable_Asset_Package_Creation

This service identifies derived requirements related to producing the necessary weapons and deployable asset packages. It also consumes the indication of whether the requirements can be 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_Requirement

This interface is the derived requirements for the creation of a package containing releasable items, 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 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.

B.2.62.7.1.7 Target_Information

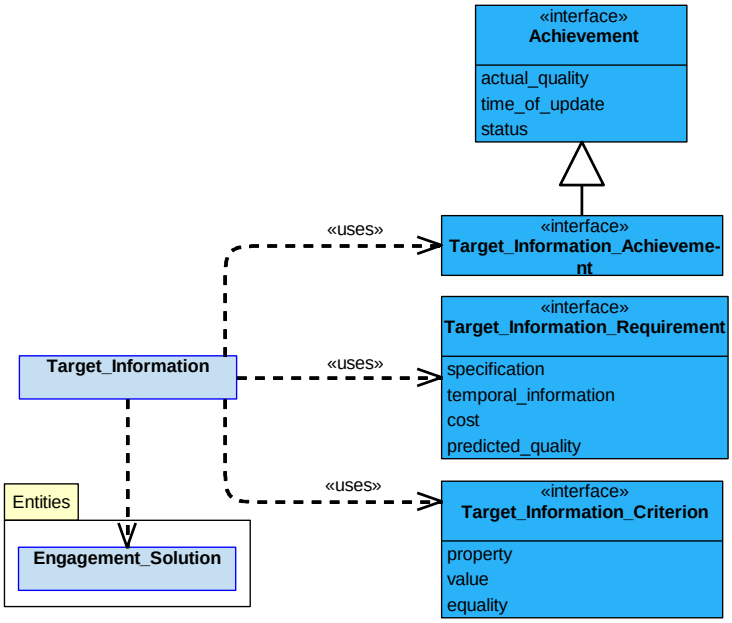

Figure 1118: Target_Information Service Definition

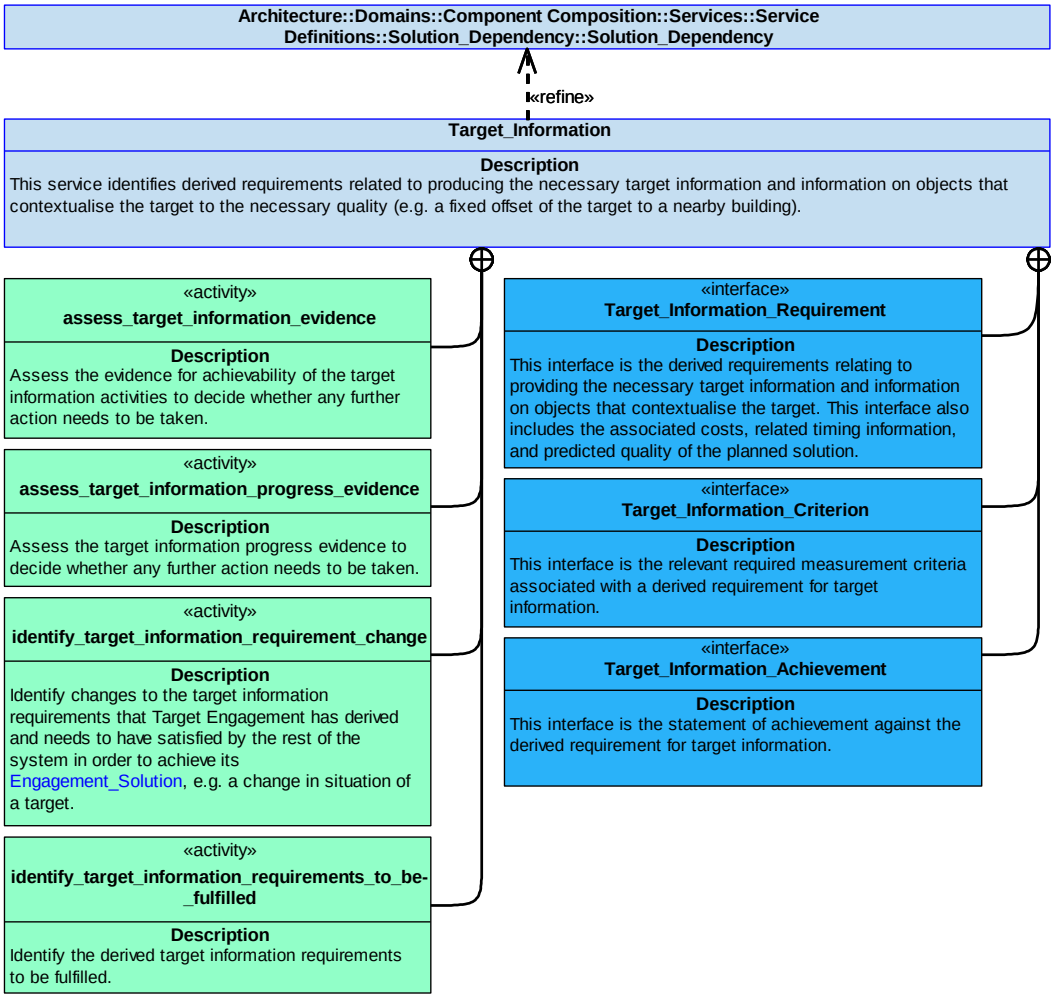

Figure 1119: Target_Information Service Policy

Target_Information

This service identifies derived requirements related to producing the necessary target information and information on objects that contextualise the target to the necessary quality (e.g. a fixed offset of the target to a nearby building).

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 a derived requirement for target information.

Attributes

property	The property to be measured, e.g. minimum level of target accuracy.
value	The measured value of the property, e.g. category 2.
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 target 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.

B.2.62.7.1.8 Engagement_Information

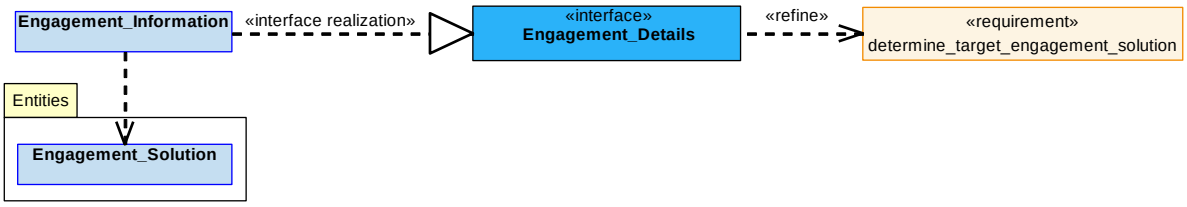

Figure 1120: Engagement_Information Service Definition

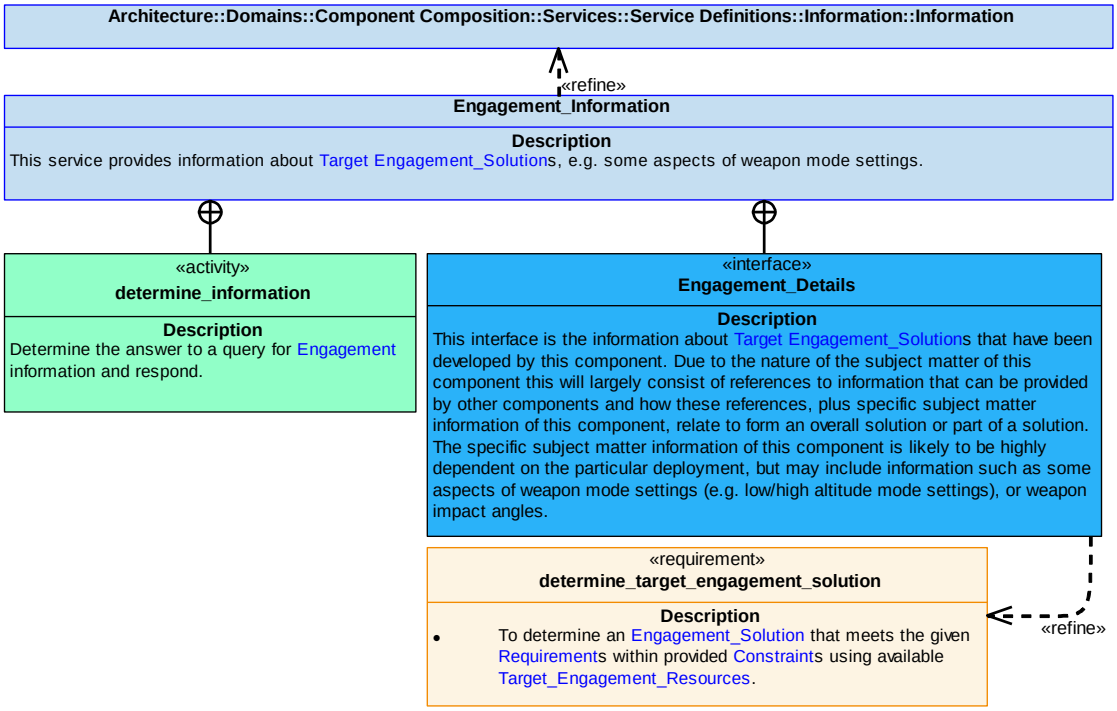

Figure 1121: Engagement_Information Service Policy

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.

B.2.62.7.1.9 Supporting_Information

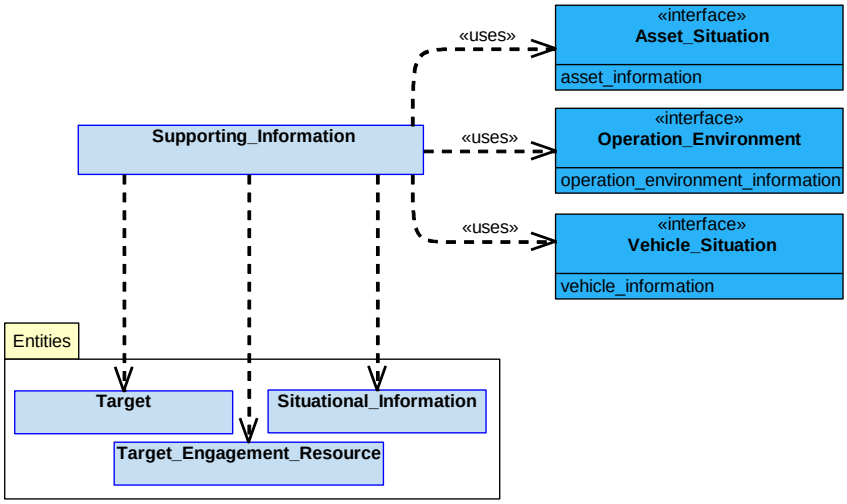

Figure 1122: Supporting_Information Service Definition

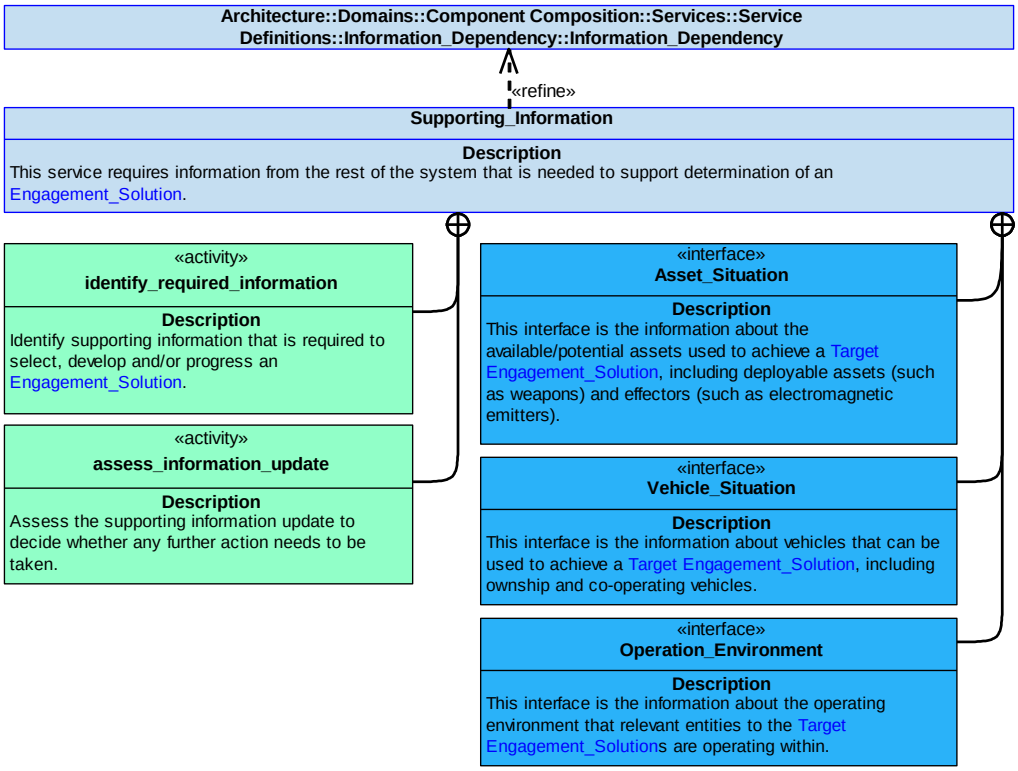

Figure 1123: Supporting_Information Service Policy

Supporting_Information

This service requires 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, 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 can be used to achieve a Target Engagement_Solution, including ownship and co-operating vehicles.

Attribute

vehicle_information	Information about ownship or co-operating vehicles; such as type of vehicle, ID, location, velocity, orientation and state.

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.

B.2.62.7.1.10 Constraint

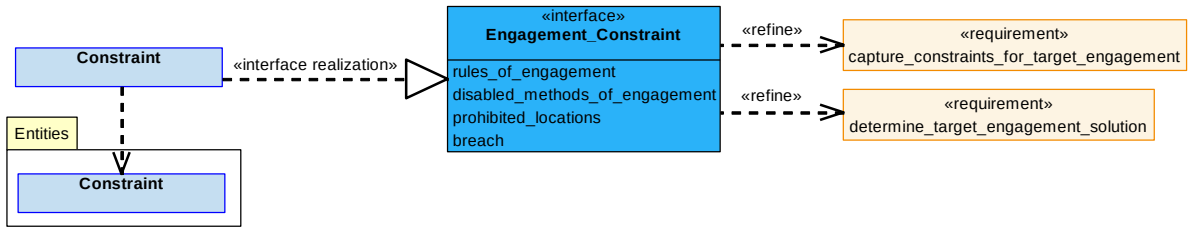

Figure 1124: Constraint Service Definition

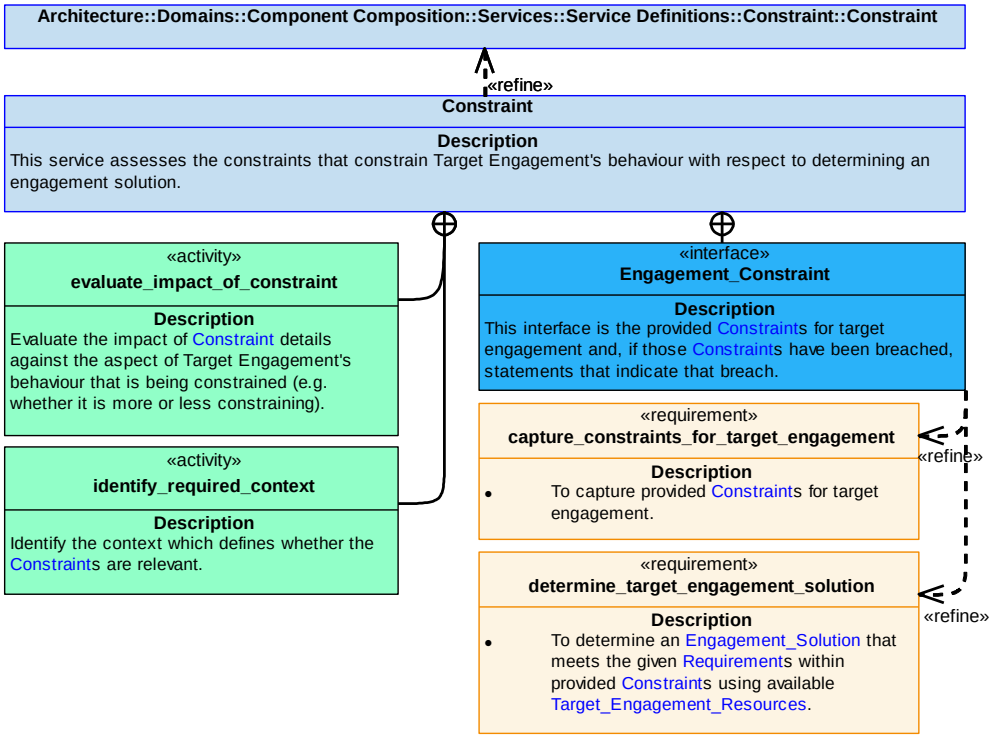

Figure 1125: Constraint Service Policy

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.

B.2.62.7.1.11 Capability

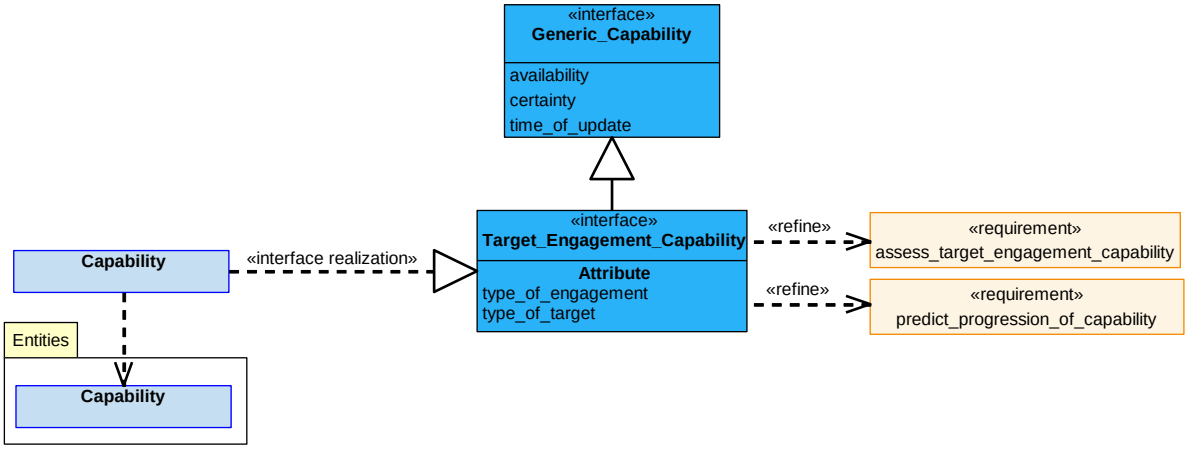

Figure 1126: Capability Service Definition

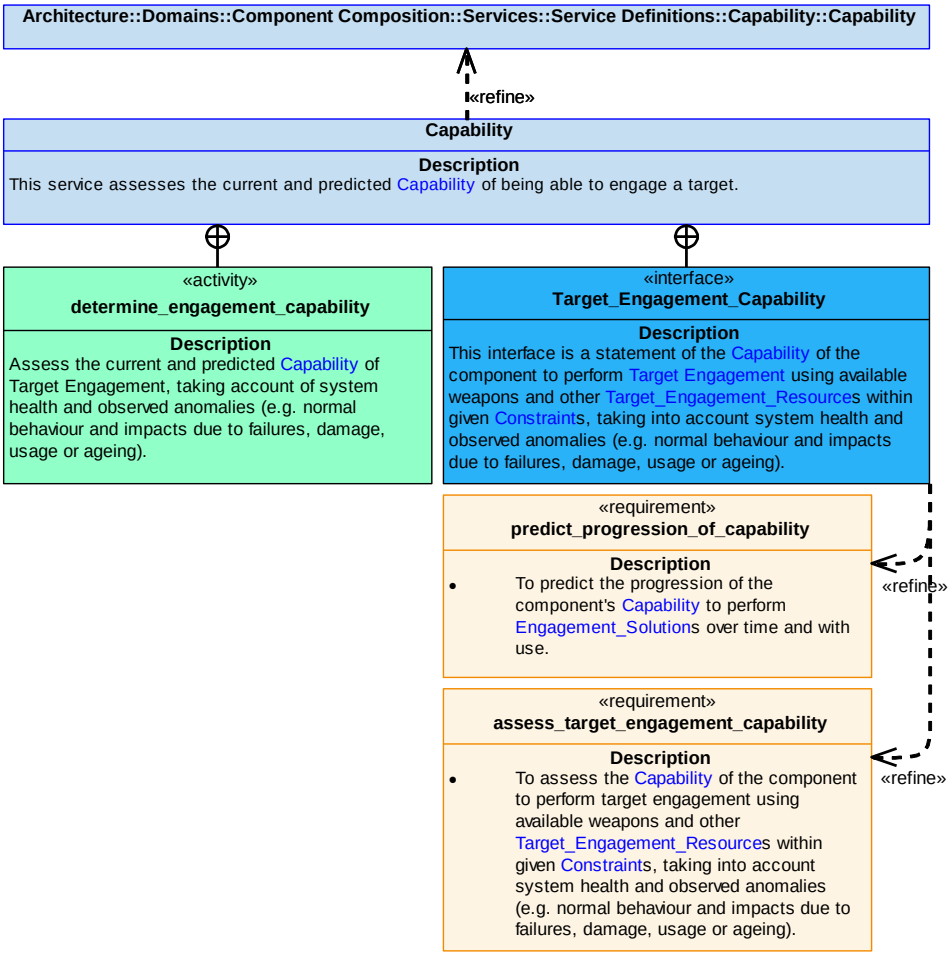

Figure 1127: Capability Service Policy

Capability

This service assesses the current and predicted Capability of being able to engage a target.

Interface

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).

Attributes

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).

B.2.62.7.1.12 Capability_Evidence

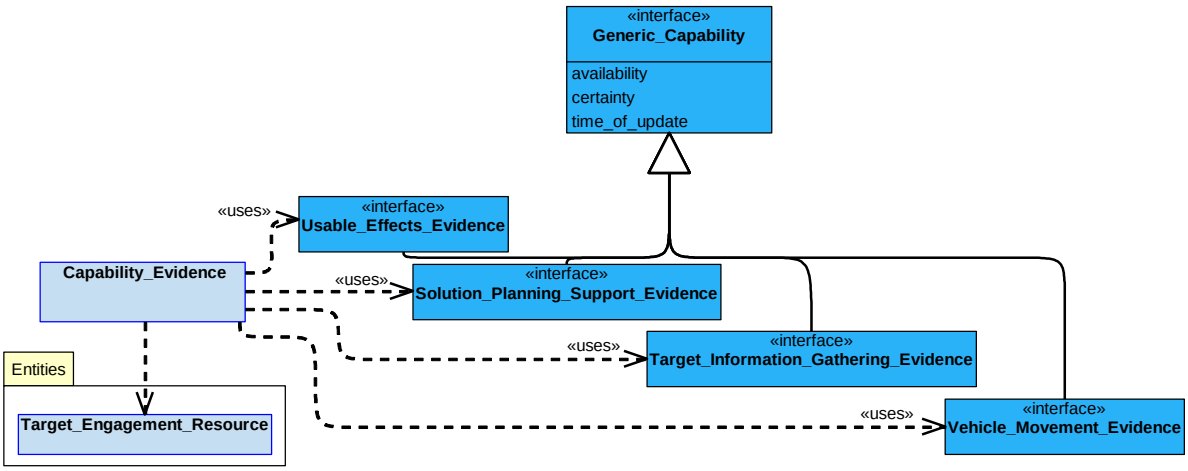

Figure 1128: Capability_Evidence Service Definition

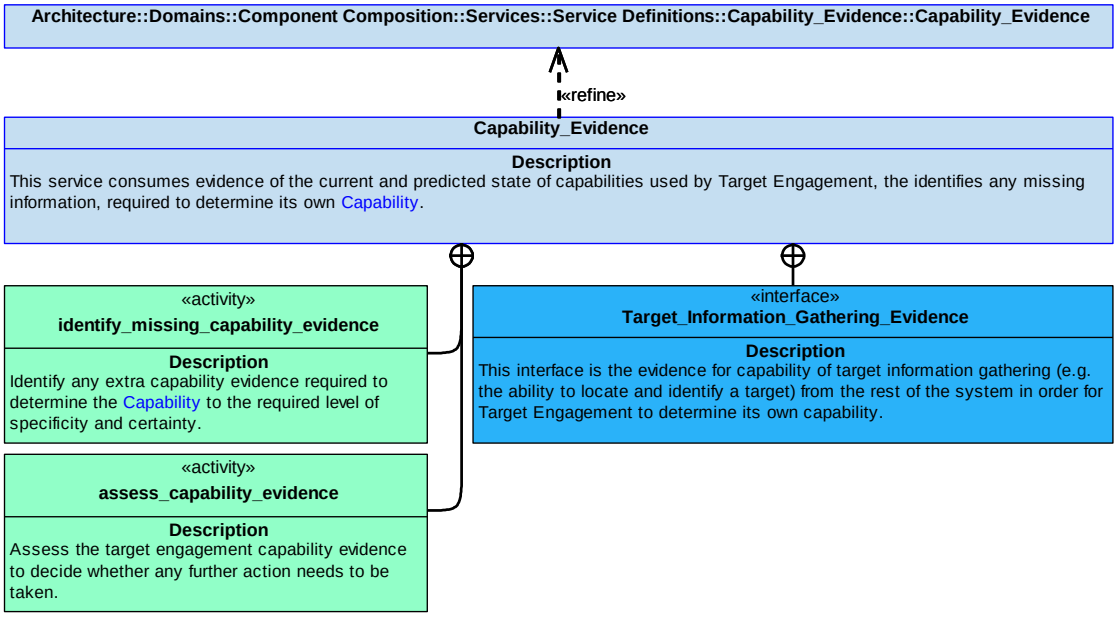

Figure 1129: Capability_Evidence Service Policy

Capability_Evidence

This service consumes evidence of the current and predicted state of capabilities used by Target Engagement, the identifies any missing information, required to determine its own Capability.

Interfaces

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.

Usable_Effects_Evidence

This interface is the evidence for capability of usable effects (e.g. from deployable assets, weapons, effectors, or cooperating vehicles) from the rest of the system in order for Target Engagement to determine its own capability.

Solution_Planning_Support_Evidence

This interface is the consumes evidence for capability of solution planning support (e.g. aiming or target susceptibility) from the rest of the system in order for Target Engagement to determine its own capability.

Vehicle_Movement_Evidence

This interface is the evidence for capability of vehicle movement from the rest of the system 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.

B.2.62.7.2 Service Dependencies

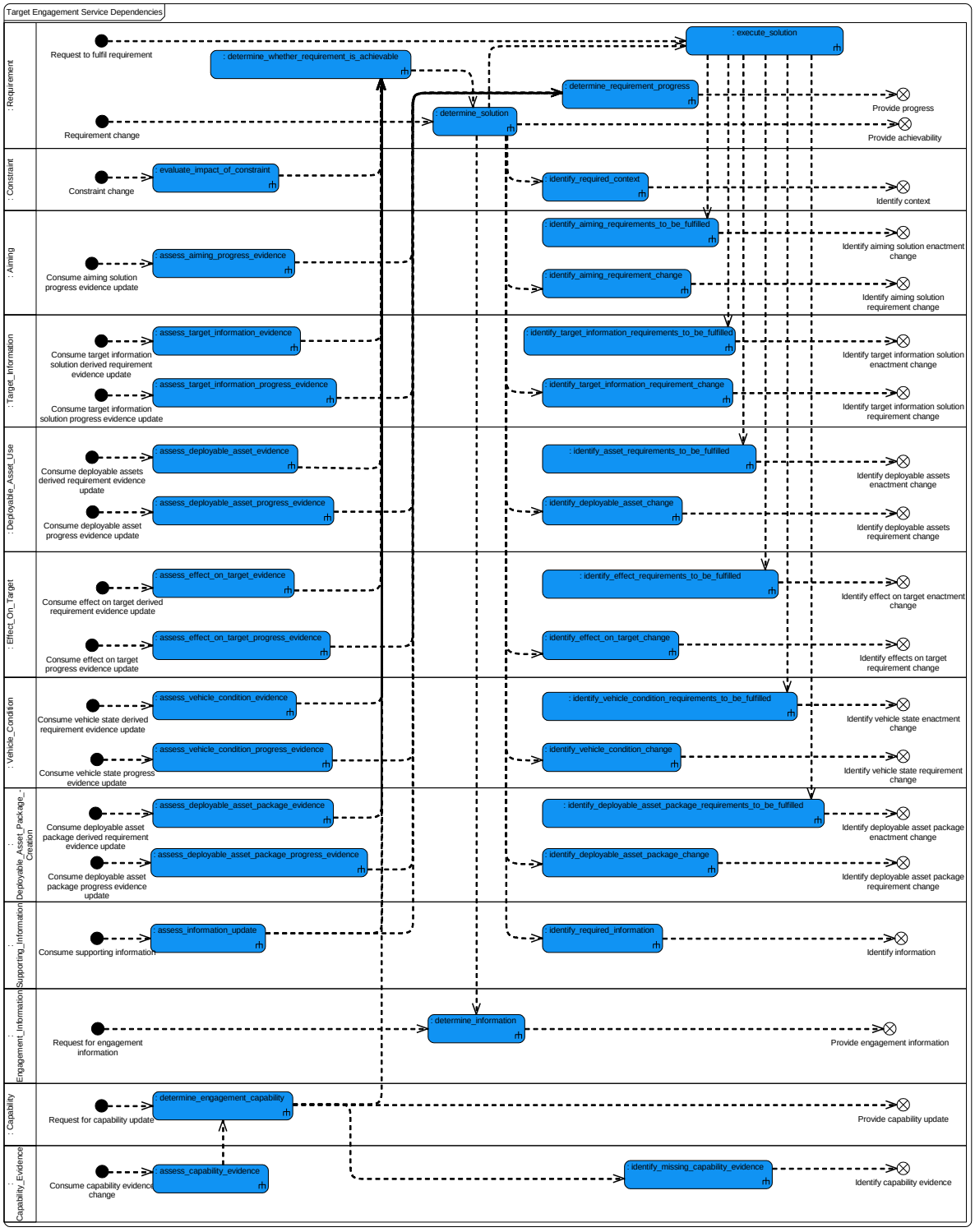

Figure 1130: Target Engagement Service Dependencies

B.2.63 Tasks

B.2.63.1 Role

The role of Tasks is to fulfil task requirements through the coordinated execution of actions.

B.2.63.2 Overview

Control Architecture

Tasks is the only component in the Task Layer, as defined in the Control Architecture policy.

Standard Pattern of Use

When the need for a task is determined, a Task_Requirement will be placed on Tasks. Tasks will determine one or more Implementation_Solutions which are a coordinated sets of Actions, taking into account any Pre-conditions and Constraints. Tasks will instigate execution of an Implementation_Solution and, if this Implementation_Solution then becomes unachievable (for example due to a hardware failure), Tasks will re-plan.

Examples of Use

Tasks will be required where the coordination of Actions by the system may be necessary. For example:

- Where the system is required to generate and fulfil Actions from a provided Task_Requirement to perform a sensing task in a specified location.

B.2.63.3 Service Summary

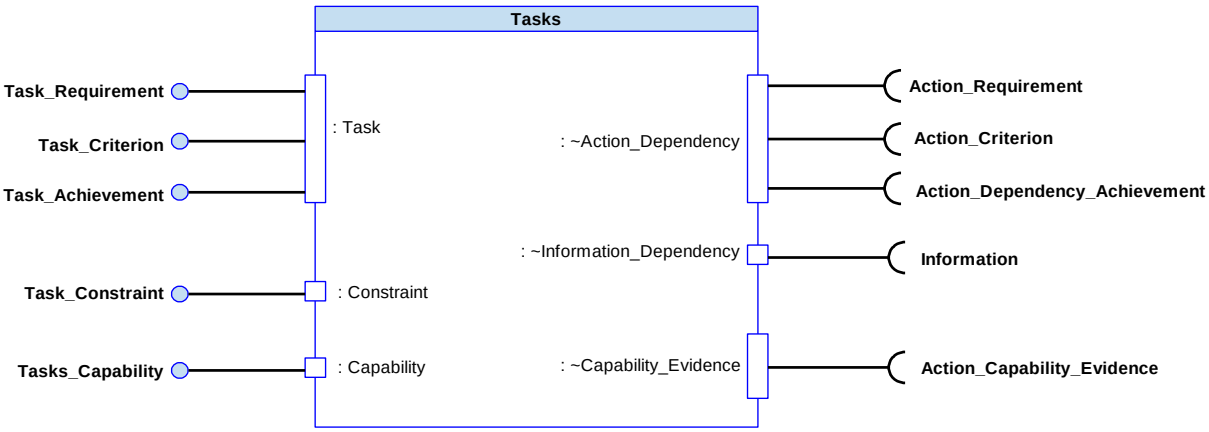

Figure 1131: Tasks Service Summary

B.2.63.4 Responsibilities

capture_task_requirements

- To capture given Task_Requirements (e.g. task type, timing, imperative for success, or risk profile).

capture_task_constraints

- To capture given Constraints (e.g. budgets or operating restrictions).

capture_measurement_criteria

- To capture given Measurement_Criterion for task solutions (e.g. quality, risk, or robustness).

assess_task_capability

- To assess the available Task_Capability based upon available Action_Capability(ies) and observed anomalies.

predict_capability_progression

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

determine_implementation_solution

- To determine an Implementation_Solution to achieve a task by using a Type_of_Solution with available Actions.

determine_implementation_solution_cost

- To determine the Costs of an Implementation_Solution against given Measurement_Criterion/criteria.

identify_pre-conditions

- To identify the Pre-conditions required between Actions within an Implementation_Solution.

satisfy_dependencies_between_actions

- To satisfy the dependencies between Actions by arranging for their Pre-conditions to be satisfied.

coordinate_task_enactment

- To coordinate the enactment of a task solution via the execution of Actions within an Implementation_Solution.

identify_progress_of_task

- To identify the progress of an Implementation_Solution using given Measurement_Criterion against the Task_Requirement.

evaluate_delivered_task_solution_quality

- To evaluate the Quality of the delivered task solution using given Measurement_Criterion.

identify_whether_requirement_remains_achievable

- To identify whether a Task_Requirement is still achievable given current or predicted Solution_Capability and Constraints.

evaluate_predicted_task_solution_quality

- To evaluate the Quality of the predicted task solution using given Measurement_Criterion.

identify_missing_information

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

B.2.63.5 Subject Matter Semantics

The subject matter of Tasks is Actions that can be used to fulfil a Task_Requirement.

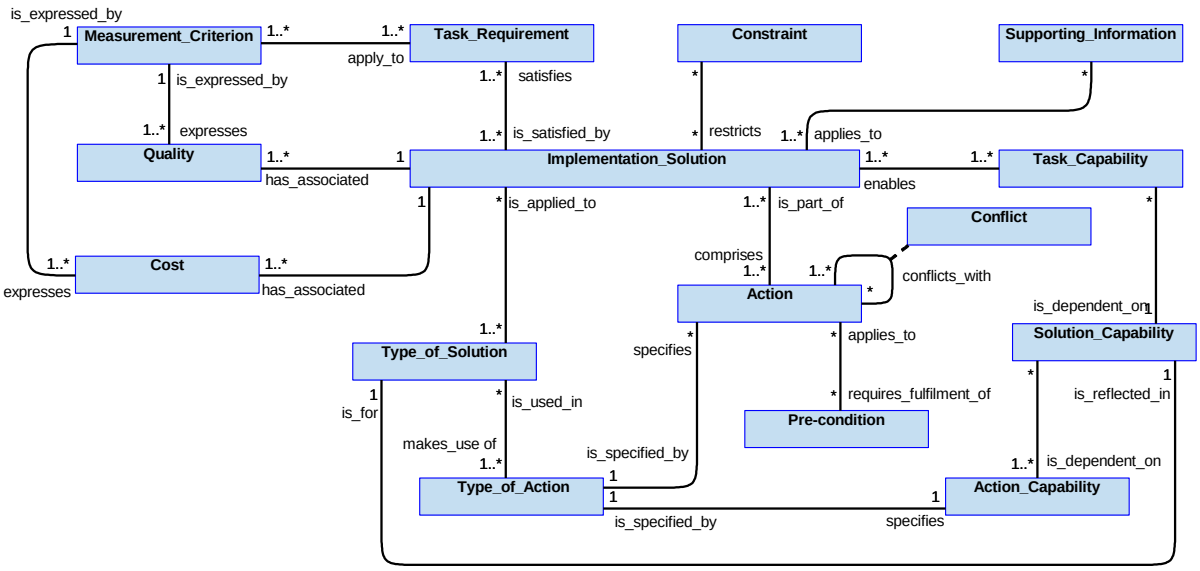

Figure 1132: Tasks Semantics

B.2.63.5.1 Entities

Action

An action is an activity, defined in terms of what needs to be done as part of an Implementation_Solution to meet the Task_Requirement.

Action_Capability

An Action that the system is able to perform with the resources available to it, and the degree of performance with which it can be achieved.

Conflict

A state where the resources required for two or more Actions cannot be satisfied simultaneously.

Constraint

A limitation on behaviour that must not be contravened in the execution of an Implementation_Solution, e.g. prohibited task types.

Cost

A measurement of 'resource' (e.g. time, risk or some other property, specified by Measurement_Criterion) that is used by an Implementation_Solution.

Implementation_Solution

A step-by-step set of Actions to be executed to achieve a given Task_Requirement.

Measurement_Criterion

A method with which to measure and express the Cost and Quality of an Implementation_Solution with respect to the Task_Requirement.

Pre-condition

A constraint on the sequencing of Actions in a given Implementation_Solution, due to inherent timing requirements or because output(s) from one Action are input(s) to another.

Quality

The degree to which the result of an Implementation_Solution satisfies an aspect of Task_Requirement, using the given Measurement_Criterion.

Solution_Capability

The capability to implement Implementation_Solutions of a particular Type_of_Solution.

Task_Capability

The capability to fulfil a particular type of Task_Requirement through the coordinated execution of actions, and the degree of performance with which it can be achieved.

Task_Requirement

A specification that describes a task to be achieved (e.g. task type, timing, risk profile, resource budgets and standard of success using Measurement_Criterion).

Type_of_Action

A specification of a type of Action that the component may use when determining Implementation_Solutions.

Type_of_Solution

A specification of a type of Implementation_Solution that the component is configured to be able to use. This could be provided by a Tactics Extension that will register the ability to implement a specialised tactic to determine a solution.

Supporting_Information

Information that supports planning or strategic decisions.

B.2.63.6 Design Rationale

B.2.63.6.1 Assumptions

- Types of tasks may vary occasionally, but are unlikely to vary during a mission.
- Types of Constraint may vary occasionally, but are unlikely to vary during a mission.
- New types of Measurement_Criterion may be defined from time to time, but not within a mission.

B.2.63.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Tasks:

- Component Extensions - See below.
- Constraint Management - The tactics used by Tasks are an example of solution-based constraints.
- Data Driving - Types of Task_Requirement, types of Constraint, types of Pre-condition and types of Measurement_Criterion should be data-driven.
- Dependency Management - Describes the process for carrying out Actions that are generated by the component.
- Multi-Vehicle Coordination - A Tasks component may break down tasks into a series of multi-vehicle Actions allocated across flight members.
- Resource Management - Where resources can be shared or used to perform multiple actions, Tasks may need to resolve conflicts to achieve Task_Requirements in line with mission priorities.

Extensions

- The breakdown of a particular type of task into Actions can be achieved by the application of specialised tactics capabilities. 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 PRA 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. (The Component Extensions policy provides further details.)

Exploitation Considerations

- 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.

B.2.63.6.3 Safety Considerations

The indicative IDAL is DAL A.

The rationale behind this is:

- The driving safety consideration for Tasks is avoiding unsuitable weather. The Survival extension Safety Considerations concluded the indicative IDAL should be DAL A. Therefore, the overall Tasks component would also need to be DAL A.
- Where instances of this component contribute to hazards that are less severe, then the Exploiting Platform may require a less onerous DAL.

B.2.63.6.4 Security Considerations

The indicative security classification is SNEO.

This component receives tasking orders, and contains the tactics, techniques and procedures (as extensions) required to plan tasks and coordinates the Actions 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.

B.2.63.7 Services

B.2.63.7.1 Service Definitions

B.2.63.7.1.1 Task

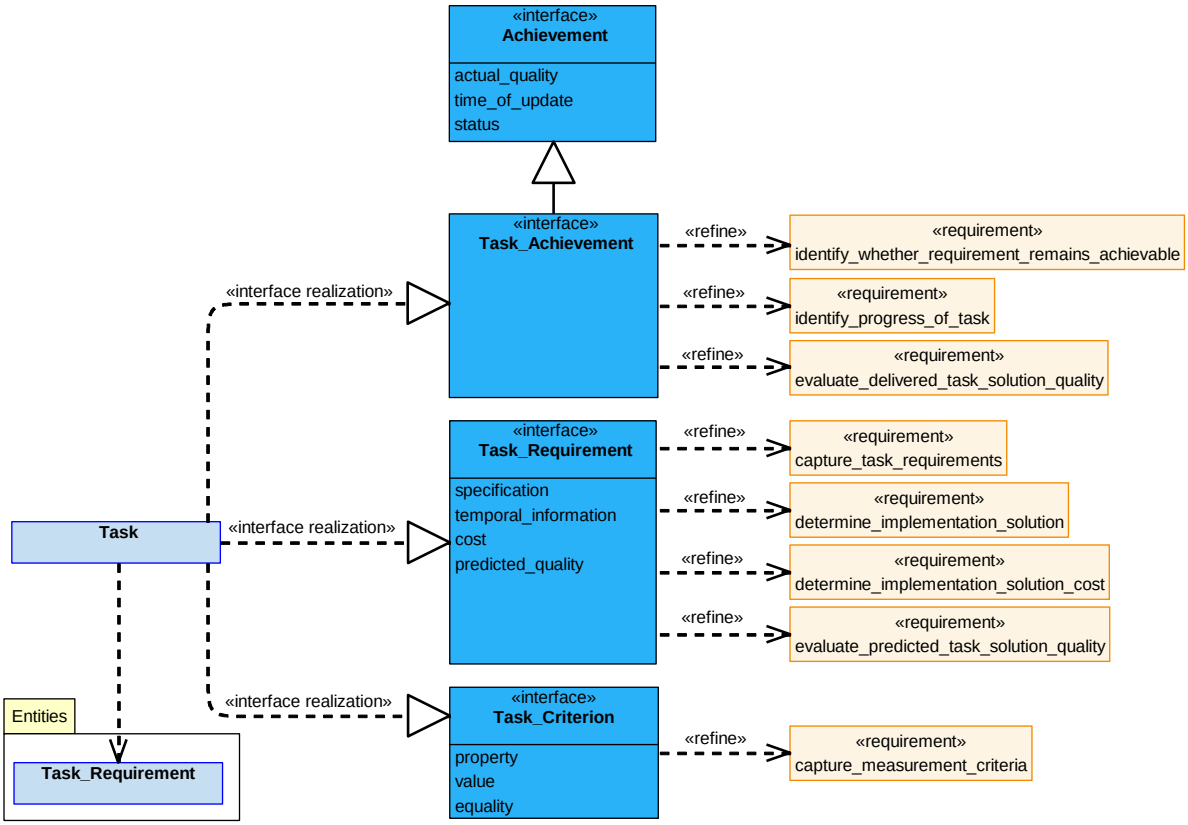

Figure 1133: Task Service Definition

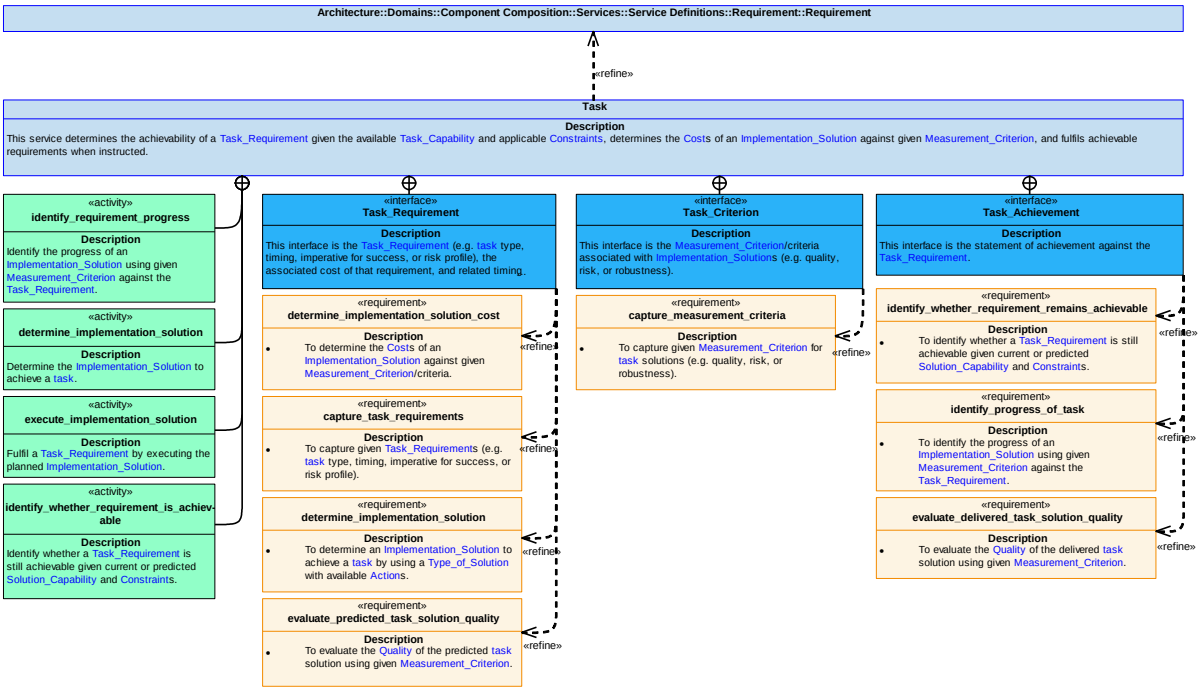

Figure 1134: Task Service Policy

Task

This service determines the achievability of a Task_Requirement given the available Task_Capability and applicable Constraints, determines the Costs of an Implementation_Solution against given Measurement_Criterion, and fulfils achievable requirements when instructed.

Interfaces

Task_Criterion

This interface is the Measurement_Criterion/criteria associated with Implementation_Solutions (e.g. quality, risk, or robustness).

Attributes

property	The property to be measured, e.g. time to fulfil task requirement by.
value	The measured value of the property, e.g. 14:00.
equality	The relationship between the value and any limit on the measurement, e.g. less than, or equal to.

Task_Requirement

This interface is the Task_Requirement (e.g. task type, timing, imperative for success, or risk profile), the associated cost of that requirement, and related timing information.

Attributes

specification	The definition of a Task_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.
cost	The cost of executing the solution, for example: resources used, time taken.
predicted_quality	How well the proposed Implementation_Solution is predicted to satisfy the Task_Requirement.

Task_Achievement

This interface is the statement of achievement against the Task_Requirement.

Activities

identify_requirement_progress

Identify the progress of an Implementation_Solution using given Measurement_Criterion against the Task_Requirement.

determine_implementation_solution

Determine the Implementation_Solution to achieve a task.

execute_implementation_solution

Fulfil a Task_Requirement by executing the planned Implementation_Solution.

identify_whether_requirement_is_achievable

Identify whether a Task_Requirement is still achievable given current or predicted Solution_Capability and Constraints.

B.2.63.7.1.2 Action_Dependency

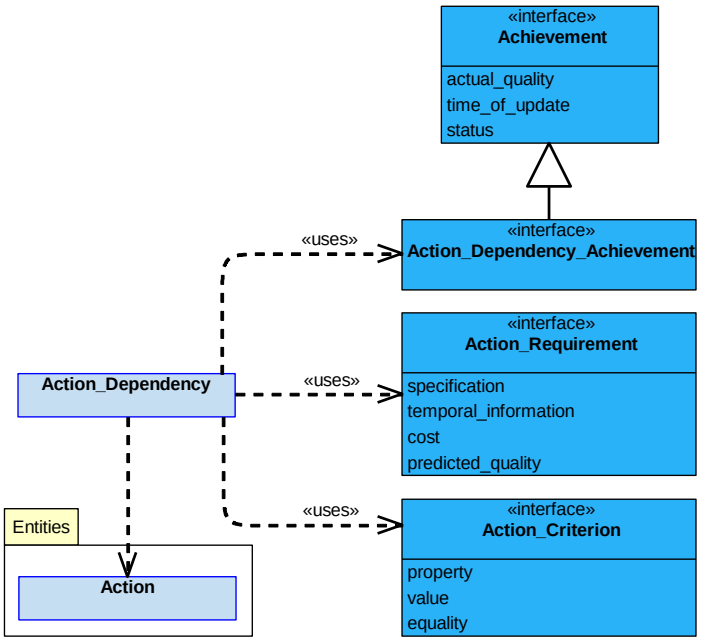

Figure 1135: Action_Dependency Service Definition

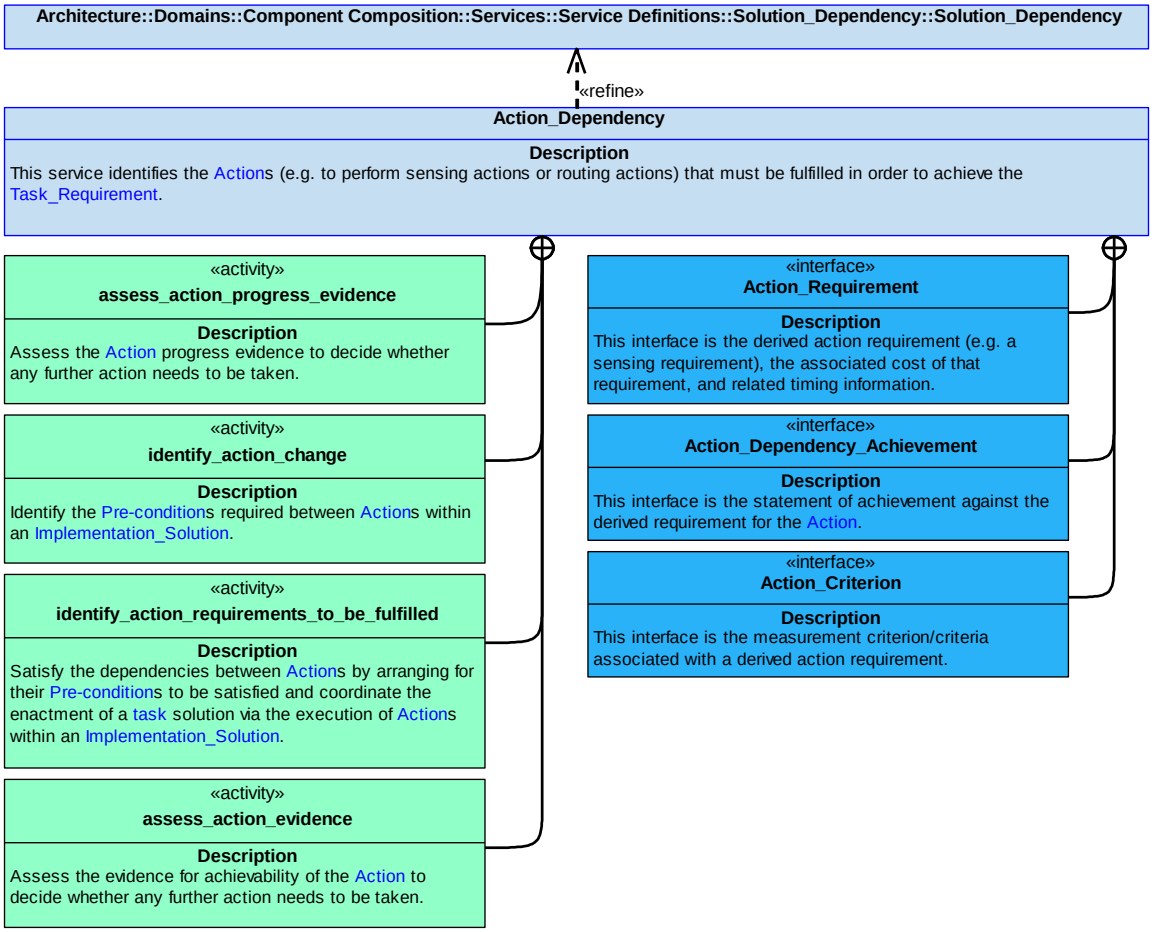

Figure 1136: Action_Dependency Service Policy

Action_Dependency

This service identifies the Actions (e.g. to perform sensing actions or routing actions) that must be fulfilled in order to achieve the Task_Requirement.

Interfaces

Action_Requirement

This interface is the derived action requirement (e.g. a sensing requirement), the associated cost of that requirement, and related timing information.

Attributes

specification	The definition of the action 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.
predicted_quality	How well the proposed action solution is predicted to satisfy the requirement.

Action_Criterion

This interface is the measurement criterion/criteria associated with a derived action 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.

Action_Dependency_Achievement

This interface is the statement of achievement against the derived requirement for the Action.

Activities

assess_action_evidence

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

assess_action_progress_evidence

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

identify_action_change

Identify the Pre-conditions required between Actions within an Implementation_Solution.

identify_action_requirements_to_be_fulfilled

Satisfy the dependencies between Actions by arranging for their Pre-conditions to be satisfied and coordinate the enactment of a task solution via the execution of Actions within an Implementation_Solution.

B.2.63.7.1.3 Information_Dependency

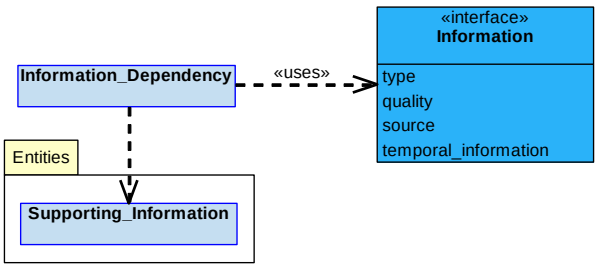

Figure 1137: Information_Dependency Service Definition

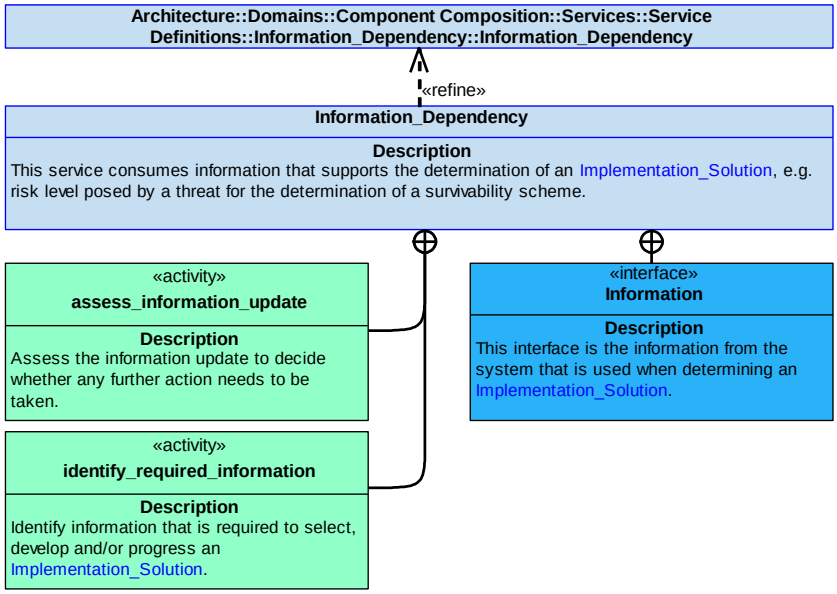

Figure 1138: Information_Dependency Service Policy

Information_Dependency

This service consumes information that supports the determination of an Implementation_Solution, e.g. risk level posed by a threat for the determination of a survivability scheme.

Interface

Information

This interface is the information from the system that is used when determining an Implementation_Solution.

Attributes

type	The type of information
quality	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 an Implementation_Solution.

B.2.63.7.1.4 Constraint

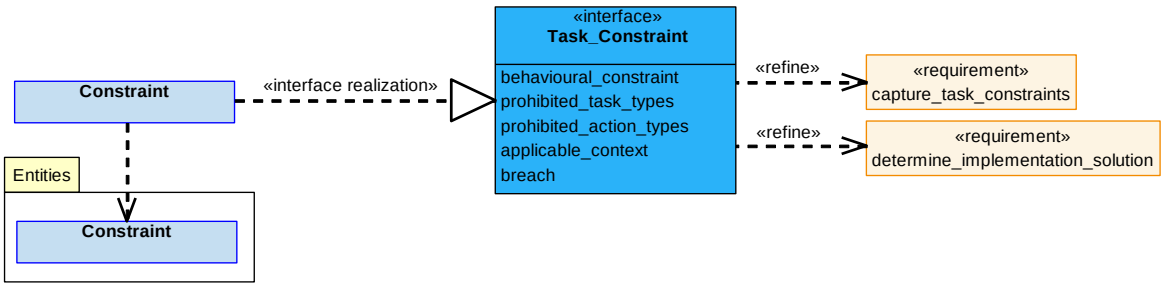

Figure 1139: Constraint Service Definition

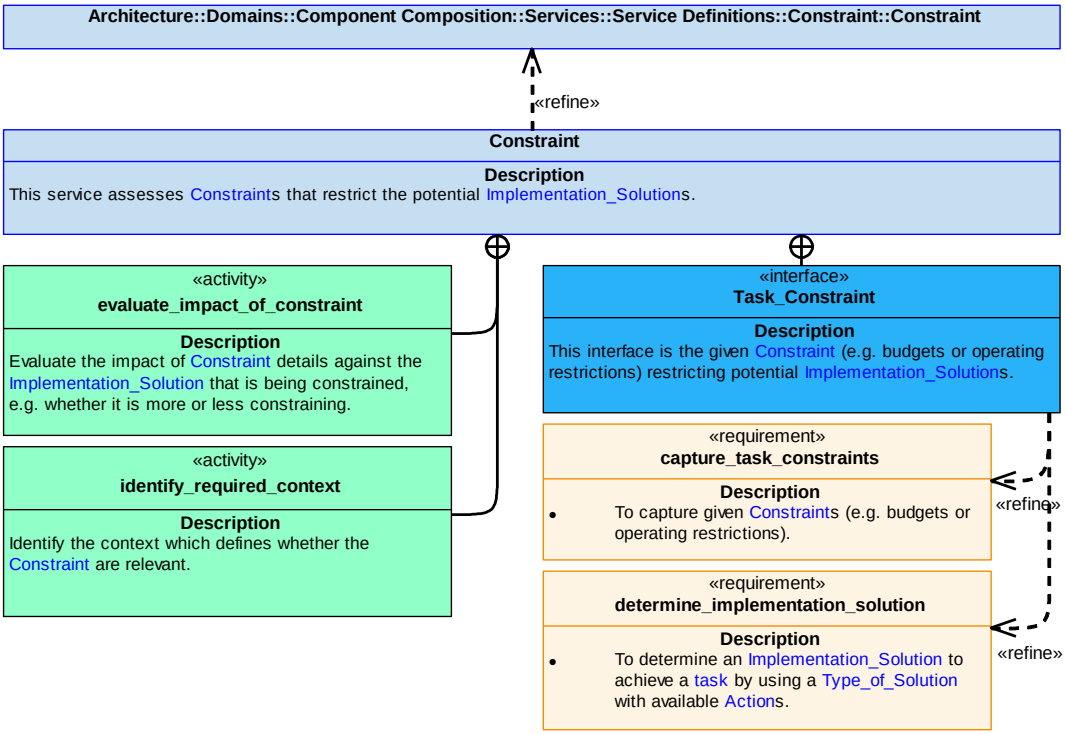

Figure 1140: Constraint Service Policy

Constraint

This service assesses Constraints that restrict the potential Implementation_Solutions.

Interface

Task_Constraint

This interface is the given Constraint (e.g. budgets or operating restrictions) restricting potential Implementation_Solutions.

Attributes

behavioural_constraint	A Constraint that limits the behaviour of Tasks, such as operational rules and limits or budgets.
prohibited_task_types	A category of task that is prohibited from being performed (e.g. the RoE currently in force may prohibit attacks).
prohibited_action_types	A category of Action that is prohibited from being performed (e.g. Actions that should only be performed during ground operations).
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 Implementation_Solution that is being constrained, e.g. whether it is more or less constraining.

identify_required_context

Identify the context which defines whether the Constraint are relevant.

B.2.63.7.1.5 Capability

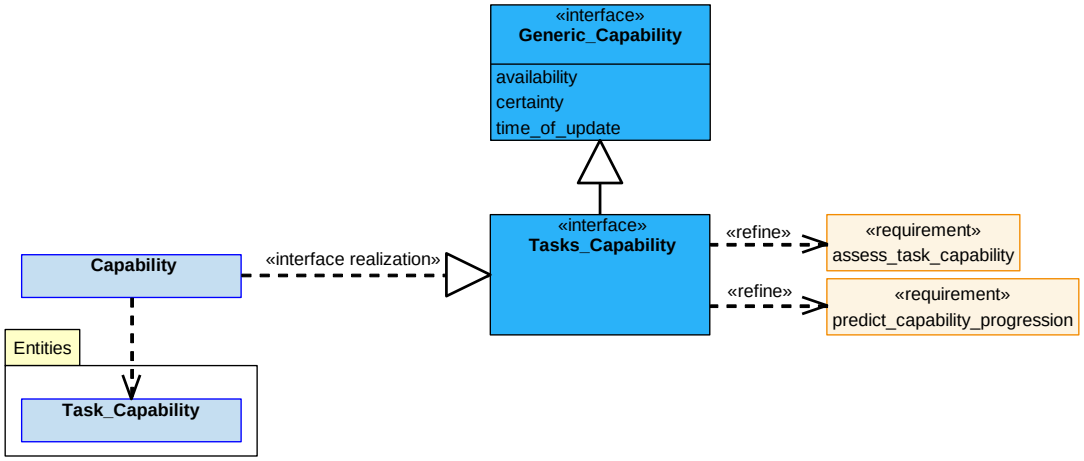

Figure 1141: Capability Service Definition

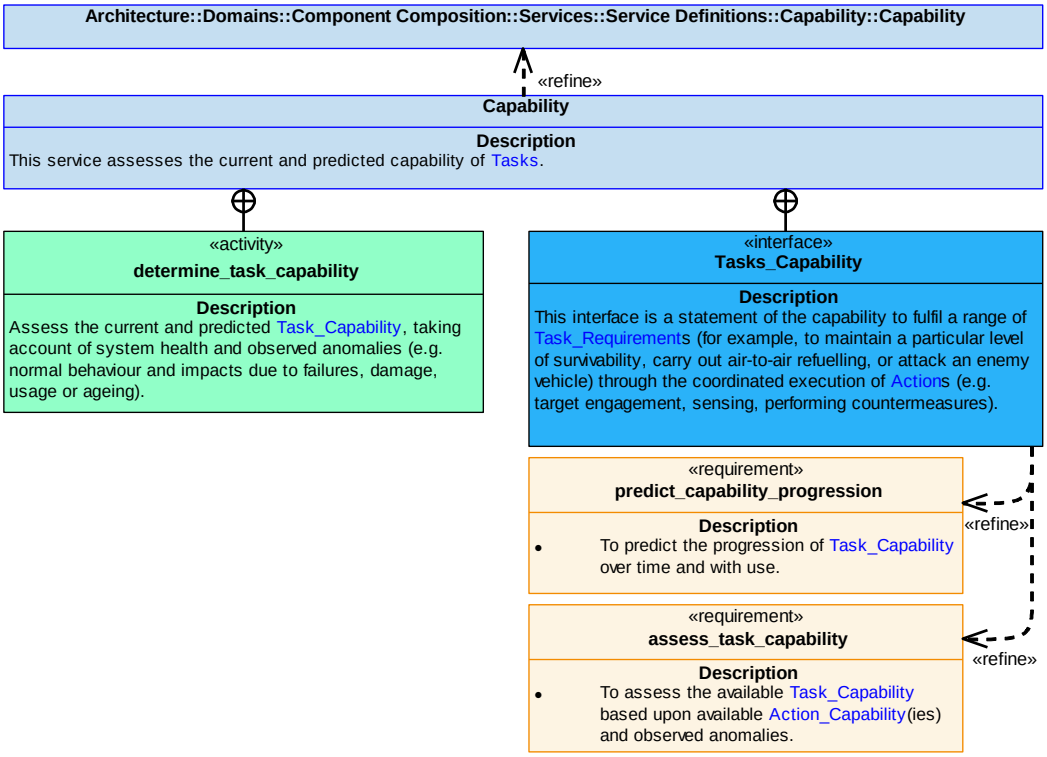

Figure 1142: Capability Service Policy

Capability

This service assesses the current and predicted capability of Tasks.

Interface

Tasks_Capability

This interface is a statement of the capability to fulfil a range of Task_Requirements (for example, to maintain a particular level of survivability, carry out air-to-air refuelling, or attack an enemy vehicle) through the coordinated execution of Actions (e.g. target engagement, sensing, performing countermeasures).

Activity

determine_task_capability

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

B.2.63.7.1.6 Capability_Evidence

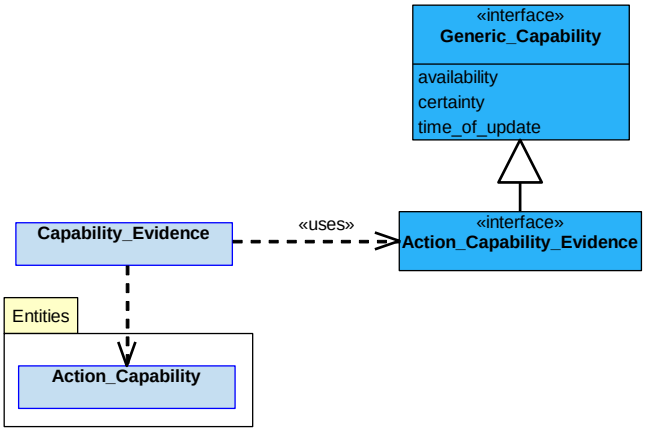

Figure 1143: Capability_Evidence Service Definition

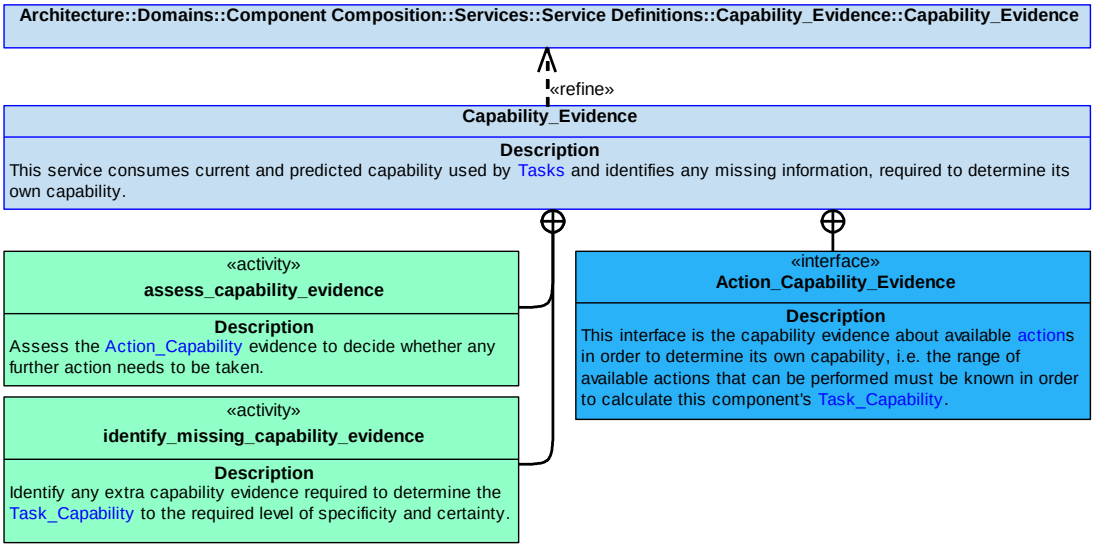

Figure 1144: Capability_Evidence Service Policy

Capability_Evidence

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

Interface

Action_Capability_Evidence

This interface is the capability evidence about available actions in order to determine its own capability, i.e. the range of available actions that can be performed must be known in order to calculate this component's Task_Capability.

Activities

assess_capability_evidence

Assess the Action_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 Task_Capability to the required level of specificity and certainty.

B.2.63.7.2 Service Dependencies

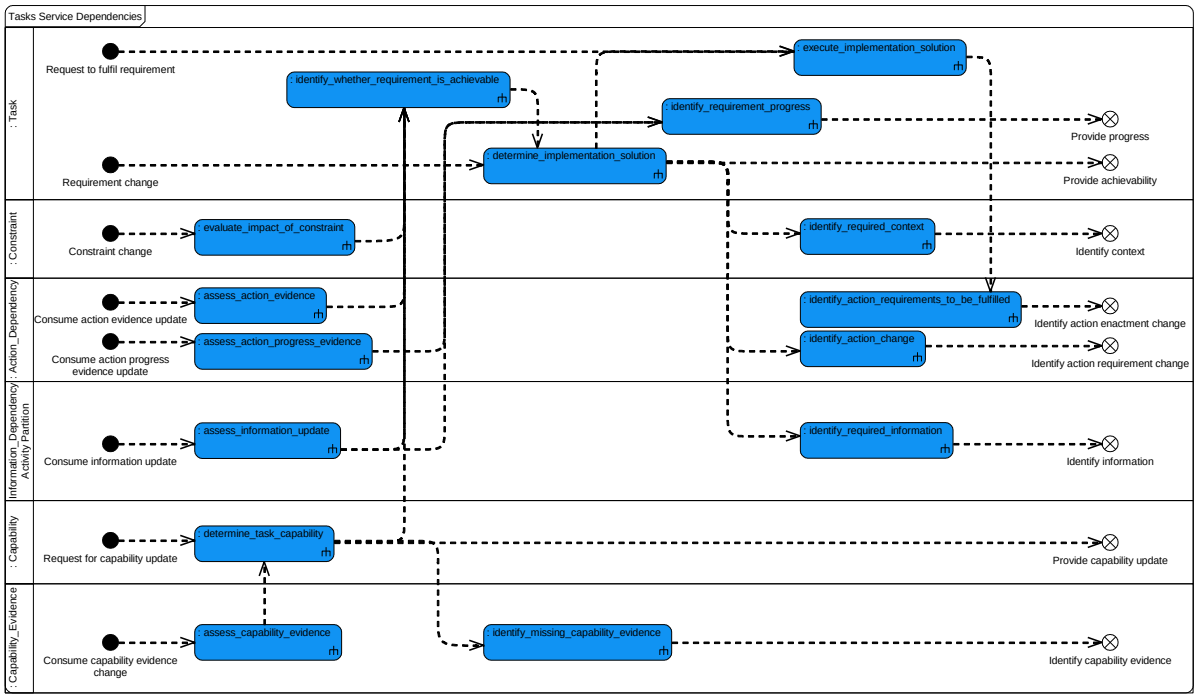

Figure 1145: Tasks Service Dependencies

B.2.64 Tasks Extension: Tactics: Aerial Refuelling

B.2.64.1 Role

The role of the Aerial Refuelling Tactics Extension is to apply tactics to determine the actions required to achieve aerial refuelling tasks.

B.2.64.2 Responsibilities

assess_aerial_refuelling_capability

- To assess the available aerial refuelling task capability based upon available Action capabilities and observed anomalies.

determine_aerial_refuelling_implementation_solution_costs

- To determine the Costs of aerial refuelling against given Measurement_Criterion (e.g. power, signature or rule transgression).

predict_aerial_refuelling_capability_progression

- To predict the progression of aerial refuelling capability over time.

determine_aerial_refuelling_pre-conditions

- To determine the Pre-conditions (e.g. air vehicle position, authorisation or tanker) required to support the delivery of a refuelling Implementation_Solution.

determine_aerial_refuelling_implementation_solution

- To determine an Implementation_Solution for aerial refuelling, which includes joining the tanker, connection, transfer of fuel and separation.

B.2.64.3 Design Rationale

B.2.64.3.1 Assumptions

- See Tasks component.
- This Tactics Extension covers the receiver role in aerial refuelling. A different Tactics Extension would be required for the tanker role.

B.2.64.3.2 Design Considerations

- The Aerial Refuelling Tactics Extension is an extension for the Tasks component in line with the Component Extensions policy.
- The Aerial Refuelling Tactics Extension is required for tasks that enable the air vehicle to receive fuel from a tanker.
- Performing aerial refuelling involves actions that might be proposed in response to the situation, such as maintaining formation with the tanker whilst fuelling.

B.2.64.3.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component could result in an air-to-air refuelling formation being formed when not required or formatting on an inappropriate aircraft (e.g. civil airliner). Whilst it is expected that other components (e.g. Formations and Collision Prediction) would ensure a mid-air collision (or other catastrophic hazard) did not occur this situation would breach the expected separation distance between aircraft. This is considered a "large reduction in safety margins" (critical severity) and so the indicative IDAL is DAL B.

Note: the inability to refuel is considered to be of less significance, as safety analysis assumes that an air vehicle would have sufficient fuel to land (return to base or diversion), even if the refuelling was not successful. This may not always be the case as "operational imperative" may allow a higher level of risk to be accepted (by the aircraft operating authority), but this is not a consideration for the design.

B.2.64.3.4 Security Considerations

This extension component will receive information and contains air-to-air refuelling TTP appropriate for the mission being performed, therefore the extension can be assumed to have an indicative security classification of SNEO.

See the Tasks component for security functions.

B.2.65 Tasks Extension: Tactics: Attack

B.2.65.1 Role

The role of the Attack Tactics Extension is to apply tactics to determine the actions required to achieve attack tasks.

B.2.65.2 Responsibilities

assess_attack_capability

- To assess the available attack task capability based upon available Action capabilities and observed anomalies.

predict_attack_capability_progression

- To predict the progression of attack capability over time.

determine_attack_implementation_solution

- To determine an Implementation_Solution for an attack using available attack Actions.

determine_attack_implementation_solution_costs

- To determine the Costs of an attack Implementation_Solution against given Measurement_Criterion (e.g. power, signature, rule transgression).

determine_attack_pre-conditions

- To determine the Pre-conditions (e.g. air vehicle position, target sensing/designation, authorisation) required to support the delivery of an attack Implementation_Solution.

B.2.65.3 Design Rationale

B.2.65.3.1 Assumptions

- See Tasks component.

B.2.65.3.2 Design Considerations

- The Attack Tactics Extension is an extension for the Tasks component, see Component Extensions policy.
- The Attack Tactics Extension is required for Tasks where destructive or suppressive effects are needed against a hostile entity or entities. These tasks will typically combine Target Engagement with other actions.
- Prosecuting an attack will require multiple Actions such as sensing and stores release.

B.2.65.3.3 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. In accordance with UK MOD direction (see Safety Analysis policy) this drives a DAL B indicative IDAL.

B.2.65.3.4 Security Considerations

This extension component will receive information and contains attack TTP appropriate for the Mission being performed, therefore the extension can be assumed to have an indicative security classification of SNEO.

See the Tasks component for security functions.

B.2.66 Tasks Extension: Tactics: Communication

B.2.66.1 Role

The role of the Communication Tactics Extension is to apply tactics to determine the actions required to achieve communications tasks.

B.2.66.2 Responsibilities

determine_communication_implementation_solution

- To determine a communication Implementation_Solution to fulfil an actual or potential communication need using communication and other Actions (e.g. movement, link, service, network and crypto).

determine_communication_implementation_solution_costs

- To determine the Costs of communication against given Measurement_Criterion (e.g. power, signature, location or information value).

determine_communication_pre-conditions

- To determine the Pre-conditions (e.g. position and compatibility of communicating parties or crypto availability) required to support the delivery of a communications Implementation_Solution.

predict_communication_capability_progression

- To predict the progression of communications Task_Capability over time.

assess_communication_capability

- To assess the available communications Task_Capability based upon available Action capabilities (e.g. protocol, network, crypto or links) and observed anomalies (e.g. throughput or constrictions).

B.2.66.3 Design Rationale

B.2.66.3.1 Assumptions

- See Tasks component.

B.2.66.3.2 Design Considerations

- The Communication Tactics Extension is an extension for the Tasks component, see Component Extensions policy.
- The Communication Tactics Extension is required to determine how to provide communications coverage for required areas and usage patterns.
- Non-communications type actions might be needed to fulfil the communication task. For example, an inter-vehicle data link may not be possible to maintain if a mountain intersects the line of sight between air vehicles, so one or both air vehicles will need re-routing to ensure continuous communications.

B.2.66.3.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component may result in the inability to communicate 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 in 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.

B.2.66.3.4 Security Considerations

This extension component will receive information and contains communications TTP appropriate for the mission being performed and methods of communications available, therefore the extension can be assumed to have an indicative security classification of SNEO.

See the Tasks component for security functions.

B.2.67 Tasks Extension: Tactics: Contingency

B.2.67.1 Role

The role of the Contingency Tactics Extension is to determine how to react to abnormal or emergency situations.

B.2.67.2 Responsibilities

determine_mitigation_implementation_solution

- To determine an Implementation_Solution to mitigate an actual or potential situation using mitigation Actions.

determine_mitigation_implementation_solution_costs

- To determine the Costs of a mitigation Implementation_Solution against given Measurement_Criterion (e.g. fuel use, safety margin or level of authorisation required).

determine_mitigation_pre-conditions

- To determine the Pre-conditions (e.g. capabilities, resource levels, jettison points or authorisation requirements) required to support the delivery of a mitigation Implementation_Solution.

predict_mitigation_capability_progression

- To predict the progression of mitigation capability over time.

assess_mitigation_capability

- To assess the available mitigation Task_Capability based upon available Action capabilities and observed anomalies.

B.2.67.3 Design Rationale

B.2.67.3.1 Assumptions

- See Tasks component.

B.2.67.3.2 Design Considerations

- The Contingency Tactics Extension is an extension for the Tasks component, see Component Extensions policy.
- The Contingency Tactics Extension is required for circumstances where the outcome of the mission is compromised due to abnormal or emergency situations. For example, an engine failure means that, although a search task can still be executed, it is not possible to subsequently complete a standard recovery.
- Multiple mitigation Actions might be proposed in response to the situation, such as performing a jettison and an emergency landing.

B.2.67.3.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- The handling of contingencies needs to be a high availability function. If a failure occurs and the correct contingency is not performed this could lead to an uncontrolled crash resulting in loss of air vehicle and fatalities. There is only a need to perform a contingency if an initial failure has occurred, hence an indicative IDAL of DAL B is considered appropriate rather than DAL A.
- Whilst Tasks will be involved in initiating Actions by the air vehicle with safety consequences (e.g. stores jettison), interlocks feeding into the Action and Resource Layers of the Control Architecture will prevent them occurring when not required.

B.2.67.3.4 Security Considerations

This extension component will receive information and contains contingency TTP appropriate for the mission being performed, therefore the extension can be assumed to have an indicative security classification of SNEO.

See the Tasks component for security functions.

B.2.68 Tasks Extension: Tactics: Search

B.2.68.1 Role

The role of the Search Tactics Extension is to apply tactics to determine the actions required to achieve search tasks.

B.2.68.2 Responsibilities

assess_search_capability

- To assess the available search Task_Capability based upon available Action capabilities and their observed anomalies.

determine_search_implementation_solution

- To determine an Implementation_Solution for a search task using available search Actions (including routing, sensing, and tactical data processing).

determine_search_implementation_solution_costs

- To determine the Costs of a search against given Measurement_Criterion (e.g. power, signature or rule transgression).

determine_search_pre-conditions

- To determine the Pre-conditions (e.g. air vehicle position and orientation, power availability, authorisation, processing or sensing) required to support the delivery of a search Implementation_Solution.

predict_search_capability_progression

- To predict the progression of search Task_Capability over time.

B.2.68.3 Design Rationale

B.2.68.3.1 Assumptions

- See Tasks component.

B.2.68.3.2 Design Considerations

- The Search Tactics Extension is an extension for the Tasks component, see Component Extensions policy.
- The Search Tactics Extension is required for search type tasks that combine sensing activities with other Actions to detect, perform geolocation, recognise, identify, or otherwise increase knowledge of objects or features in the tactical environment. For example: locate all the tanks in a region.
- Performing a search will require multiple Actions such as sensing and routing.

B.2.68.3.3 Safety Considerations

The indicative IDAL is DAL C.

The rationale behind this is:

- Failure of this component would result in the inability to search, which does not impact safety. It is expected that if this component attempted to initiate search activities which compromised safety (e.g. intended route impacted terrain) that other, high integrity, components (e.g. Path Demands or Interlocks) would prevent any unsafe Actions from being performed. Therefore, an IDAL no more onerous than DAL C, is considered appropriate for this component.

B.2.68.3.4 Security Considerations

This extension component will receive information and contains search TTP appropriate for the Mission being performed, therefore the extension can be assumed to have an indicative security classification of SNEO.

See the Tasks component for security functions.

B.2.69 Tasks Extension: Tactics: Survival

B.2.69.1 Role

The role of the Survival Tactics Extension is to apply tactics to determine the actions required to achieve survival tasks.

B.2.69.2 Responsibilities

assess_survival_capability

- To assess the available survival Task_Capability based upon available Action capabilities and observed anomalies.

determine_survival_implementation_solution

- To determine an Implementation_Solution to achieve a survival task using available Actions.

determine_survival_implementation_solution_costs

- To determine the Costs of an Implementation_Solution against given Measurement_Criterion (e.g. impact on mission goal, fuel expended, safety margin, resources utilised, wider threat risk assessment).

determine_survival_pre-conditions

- To determine the Pre-conditions (e.g. routing dependencies like air vehicle location and position requirements, manoeuvre dependencies like attitude and orientation of air vehicle, or identification dependencies required like threat identification and confidence level) required to support the delivery of a survival Implementation_Solution.

predict_survival_capability_progression

- To predict the progression and viability of survival Task_Capability over time and with use against the measurement criteria and task requirements.

B.2.69.3 Design Rationale

B.2.69.3.1 Assumptions

- See Tasks component.

B.2.69.3.2 Design Considerations

- The Survival Tactics Extension is an extension for the Tasks component in line with the Component Extensions policy.
- The Survival Tactics Extension is required for tasks that increase the survivability of the air vehicle by mitigating threats using a combination of Actions that are likely to include countermeasures, routing and others.
- Multiple survival Actions might be proposed in response to the situation, such as reducing signature or deploying countermeasures.

B.2.69.3.3 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.

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.

B.2.69.3.4 Security Considerations

This extension component will receive information and contains survival TTP appropriate for the mission being performed, therefore the extension can be assumed to have an indicative security classification of SNEO.

See the Tasks component for security functions.

B.2.70 Tasks Extension: Tactics: Transit

B.2.70.1 Role

The role of the Transit Tactics Extension is to apply tactics to determine the actions required to achieve transit tasks.

B.2.70.2 Responsibilities

determine_transit_implementation_solution

- To determine an Implementation_Solution for a transit task using available transit Actions to reposition the air vehicle.

determine_transit_implementation_solution_costs

- To determine the Costs of transit against given Measurement_Criterion (e.g. fuel use, safety margin or level of authorisation required).

determine_transit_pre-conditions

- To determine the Pre-conditions (e.g. capabilities, resource levels or authorisation requirements) required to support the delivery of a transit Implementation_Solution.

assess_transit_capability

- To assess the available transit Task_Capability based upon available Action capabilities and observed anomalies.

predict_transit_capability_progression

- To predict the progression of transit Task_Capability over time.

B.2.70.3 Design Rationale

B.2.70.3.1 Assumptions

- See Tasks component.

B.2.70.3.2 Design Considerations

- The Transit Tactics Extension is an extension for the Tasks component in line with the Component Extensions policy.
- The Transit Tactics Extension is required for tasks that reposition the air vehicle so that it can perform another task.
- Multiple Actions might be proposed in response to the situation.

B.2.70.3.3 Safety Considerations

The indicative IDAL is DAL B.

The rationale behind this is:

- Failure of this component could result in an inappropriate transit activity being carried out. As it is expected that other components (e.g. Routes) would ensure the transit route is achievable (in terms of fuel, etc.), this is considered a "large reduction in safety margins" (critical severity) and so the indicative IDAL is DAL B.

B.2.70.3.4 Security Considerations

This extension component will receive information and contains transit TTP appropriate for the mission being performed, therefore the extension can be assumed to have an indicative security classification of SNEO.

See the Tasks component for security functions.

B.2.71 Test

B.2.71.1 Role

The role of Test is to coordinate Tests by managing the resources available to it.

B.2.71.2 Overview

Control Architecture

Test is an action component as defined in the Control Architecture policy.

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.

B.2.71.3 Service Summary

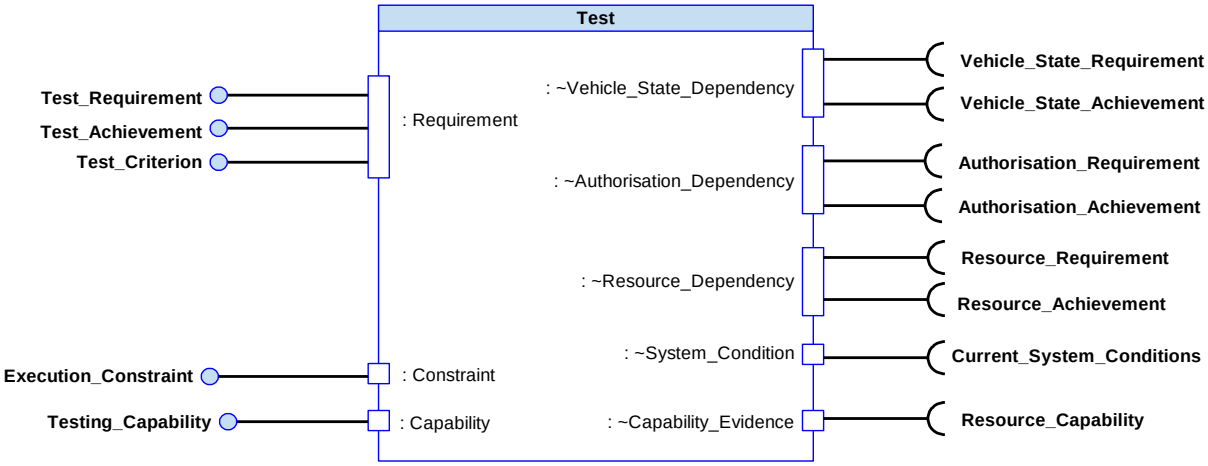

Figure 1146: Test Service Summary

B.2.71.4 Responsibilities

capture_test_requirements

- To capture Test_Requirements (e.g. system characteristics to test or a test methodology).

capture_test_constraints

- To capture test Constraints (e.g. operational limits or safety limits).

determine_test_outcome

- To determine the Test_Outcome of an enacted Test_Strategy.

determine_test_which_meets_requirements

- To determine a Test_Strategy that meets the given Test_Requirement and Constraints using available resources.

determine_cost_of_test_solution

- To determine the cost of a Test_Strategy (e.g. affected capabilities or cost of using 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, 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).

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.

predict_capability_progression

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

identify_whether_requirement_remains_achievable

- To identify whether a Test_Requirement is still achievable given current or predicted Capability and Constraints.

capture_measurement_criteria

- To capture given Measurement_Criterion for the Test_Strategy.

identify_missing_information

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

B.2.71.5 Subject Matter Semantics

The subject matter of Test is tests that can be run to establish the condition of part of a system.

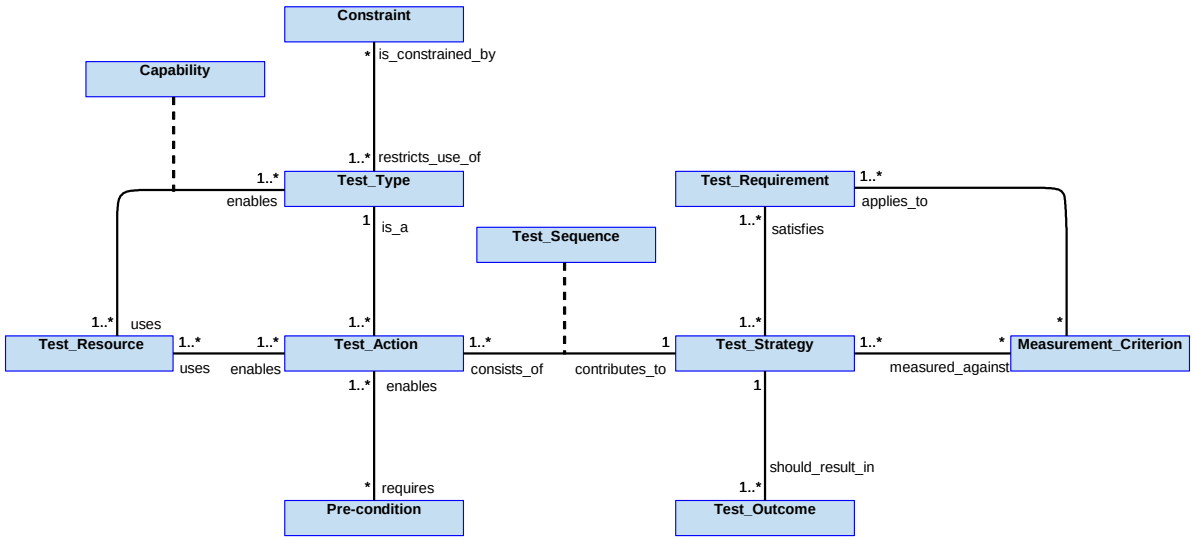

Figure 1147: Test Semantics

B.2.71.5.1 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.

B.2.71.6 Design Rationale

B.2.71.6.1 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). Startup tests would not ordinarily be managed by the Test component.

B.2.71.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Test:

- Test - This policy 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 policy.

Extensions

- Extensions to this component could be used to accommodate different types of test and specifics of the system-under-test's behaviour.

B.2.71.6.3 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.

B.2.71.6.4 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.

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.

B.2.71.7 Services

B.2.71.7.1 Service Definitions

B.2.71.7.1.1 Requirement

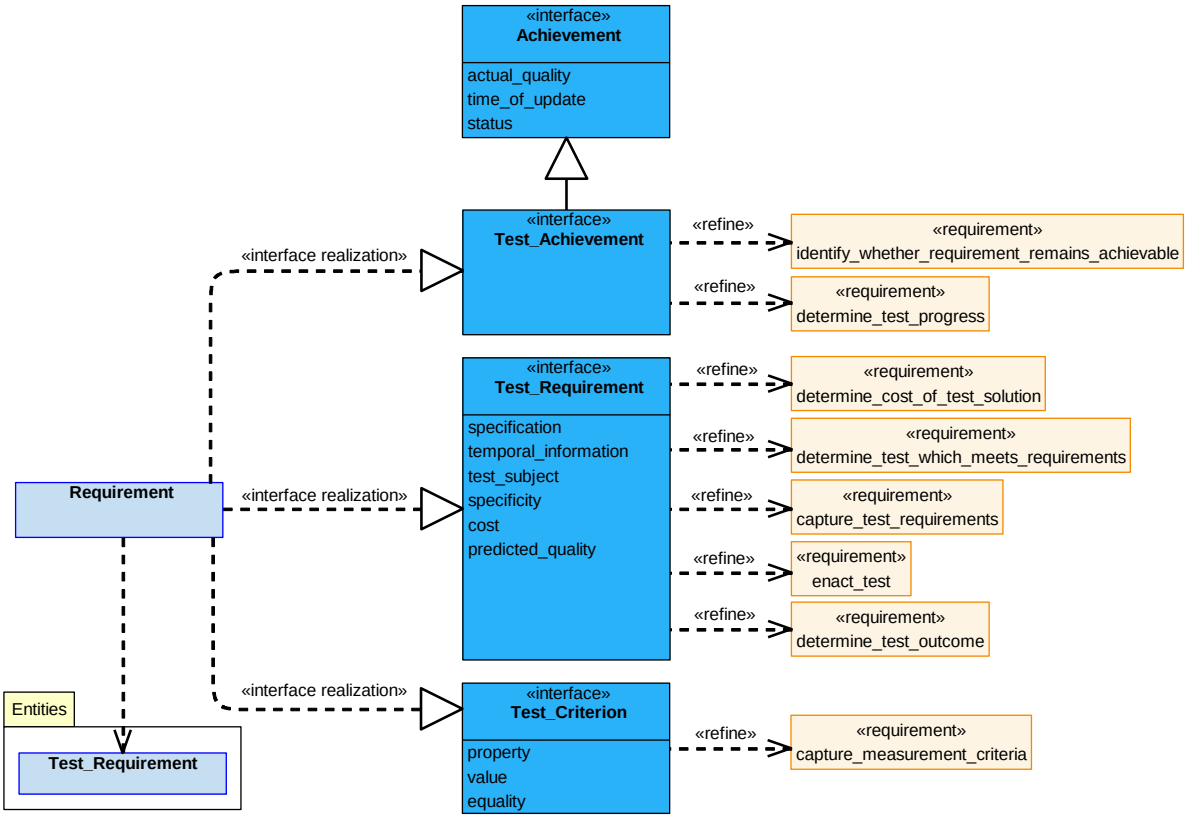

Figure 1148: Requirement Service Definition

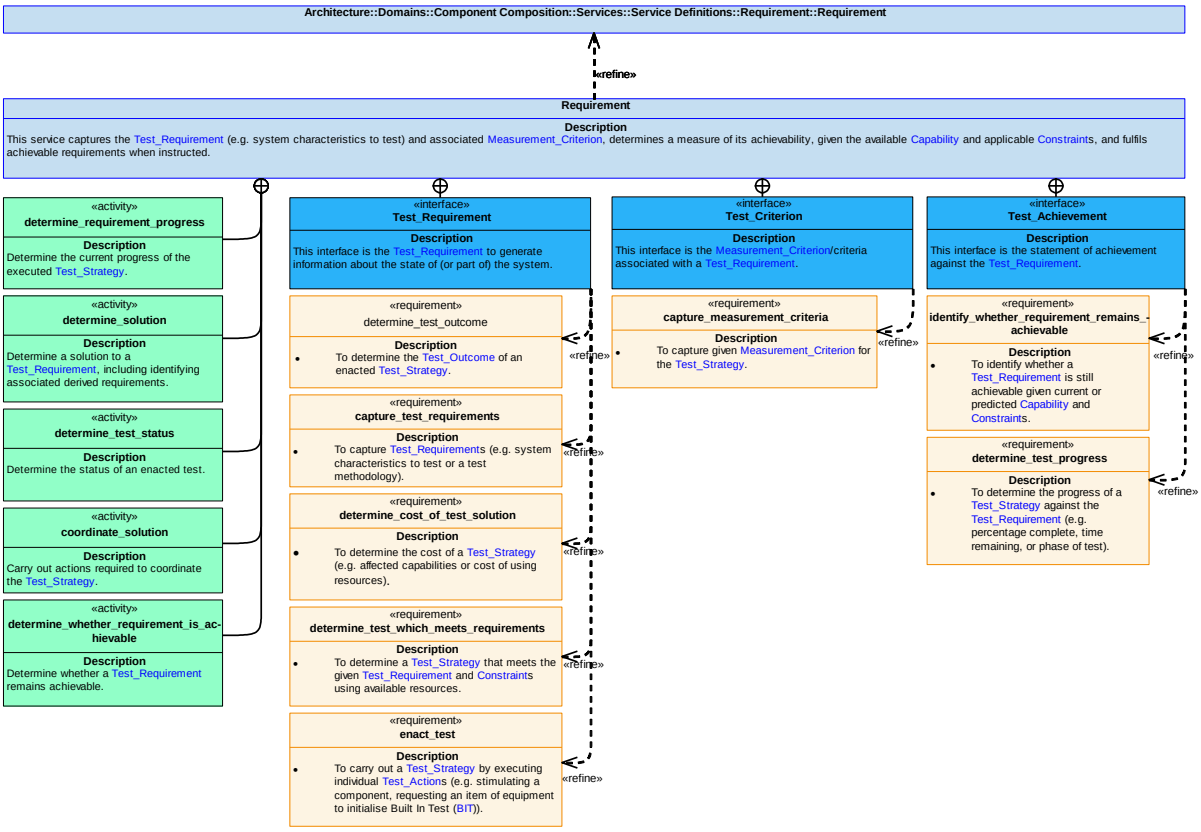

Figure 1149: Requirement Service Policy

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.

B.2.71.7.1.2 Vehicle_State_Dependency

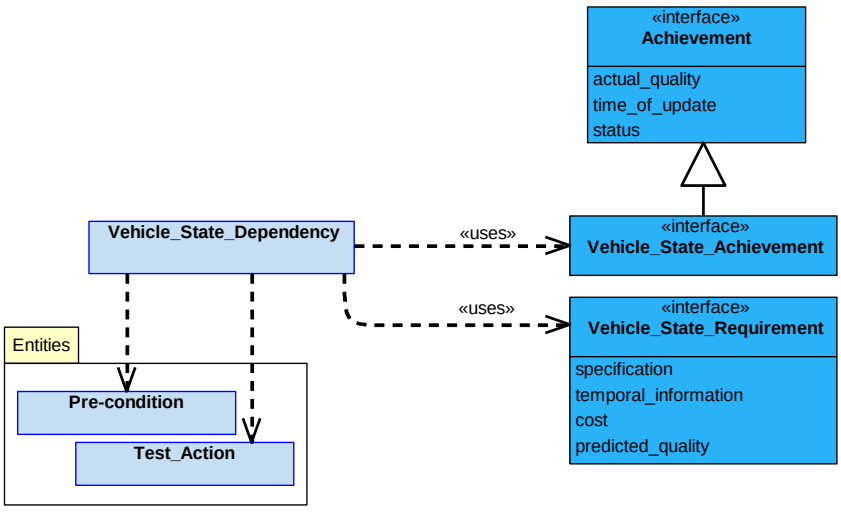

Figure 1150: Vehicle_State_Dependency Service Definition

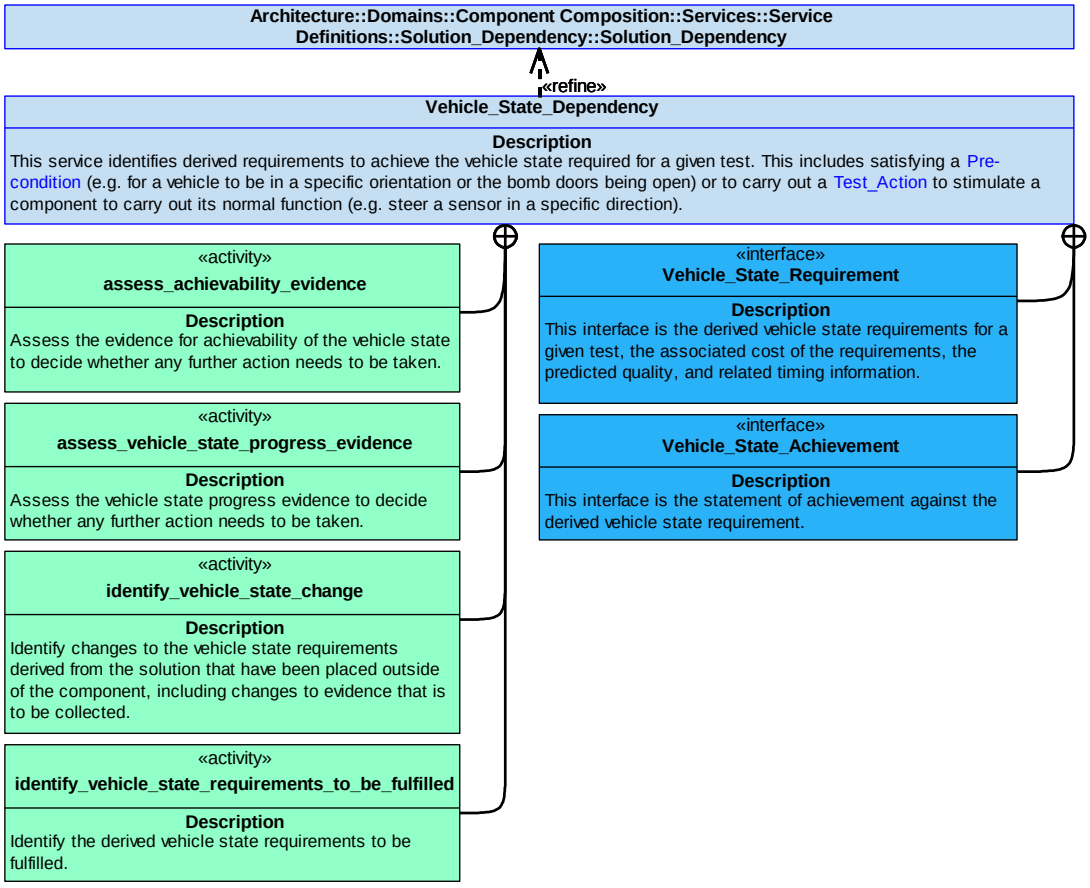

Figure 1151: Vehicle_State_Dependency Service Policy

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.

B.2.71.7.1.3 Authorisation_Dependency

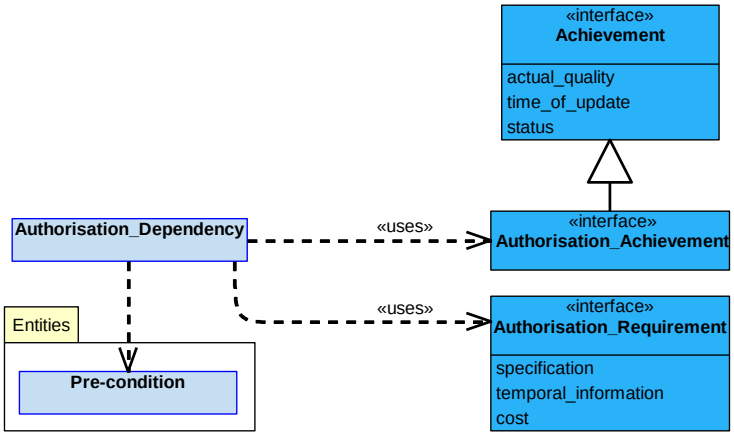

Figure 1152: Authorisation_Dependency Service Definition

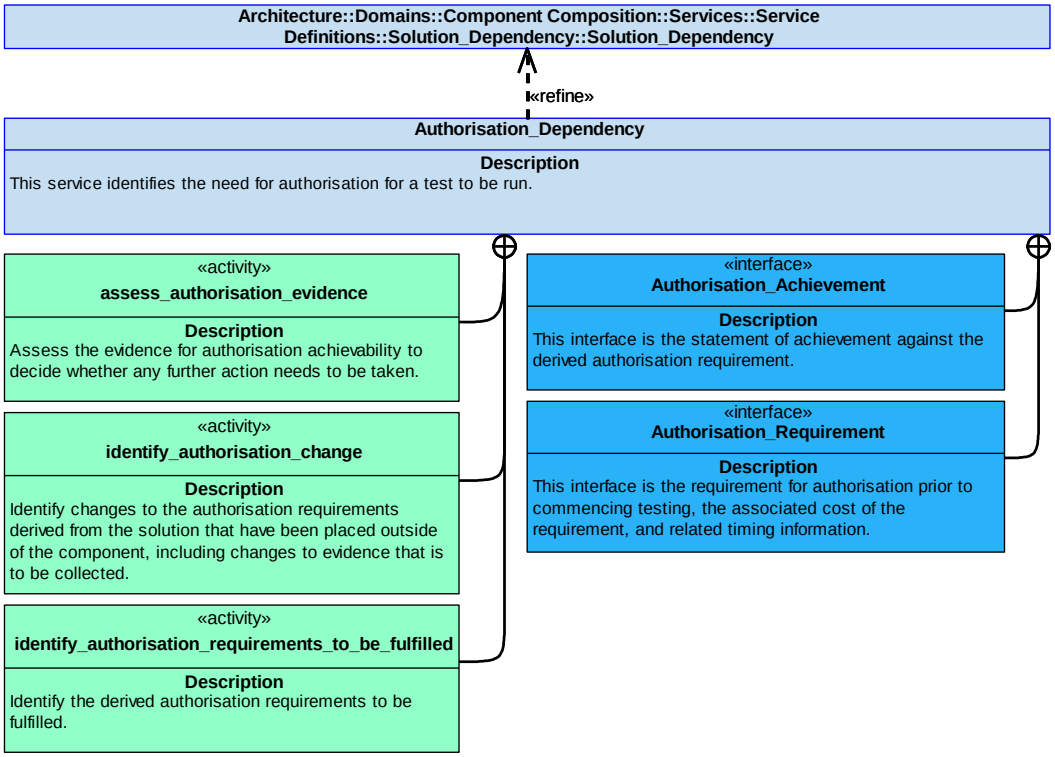

Figure 1153: Authorisation_Dependency Service Policy

Authorisation_Dependency

This service identifies the need for authorisation for a test to be run.

Interfaces

Authorisation_Requirement

This interface is the requirement for authorisation prior to commencing testing, the associated cost of the requirement, and related timing information.

Attributes

specification	The authorisation required to enact the test.
temporal_information	Information covering timing, such as start and end times of the authorisation.
cost	Information about the cost, usually in terms of resources, required to execute the solution.

Authorisation_Achievement

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

Activities

assess_authorisation_evidence

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

identify_authorisation_change

Identify changes to the authorisation requirements derived from the solution that have been placed outside of the component, including changes to evidence that is to be collected.

identify_authorisation_requirements_to_be_fulfilled

Identify the derived authorisation requirements to be fulfilled.

B.2.71.7.1.4 Resource_Dependency

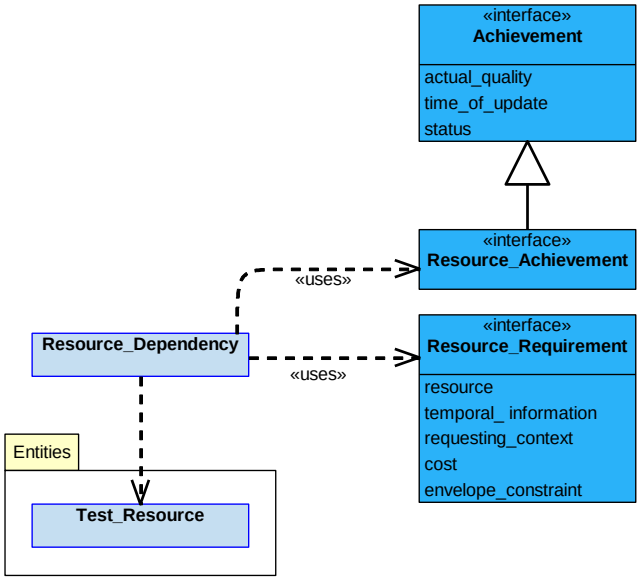

Figure 1154: Resource_Dependency Service Definition

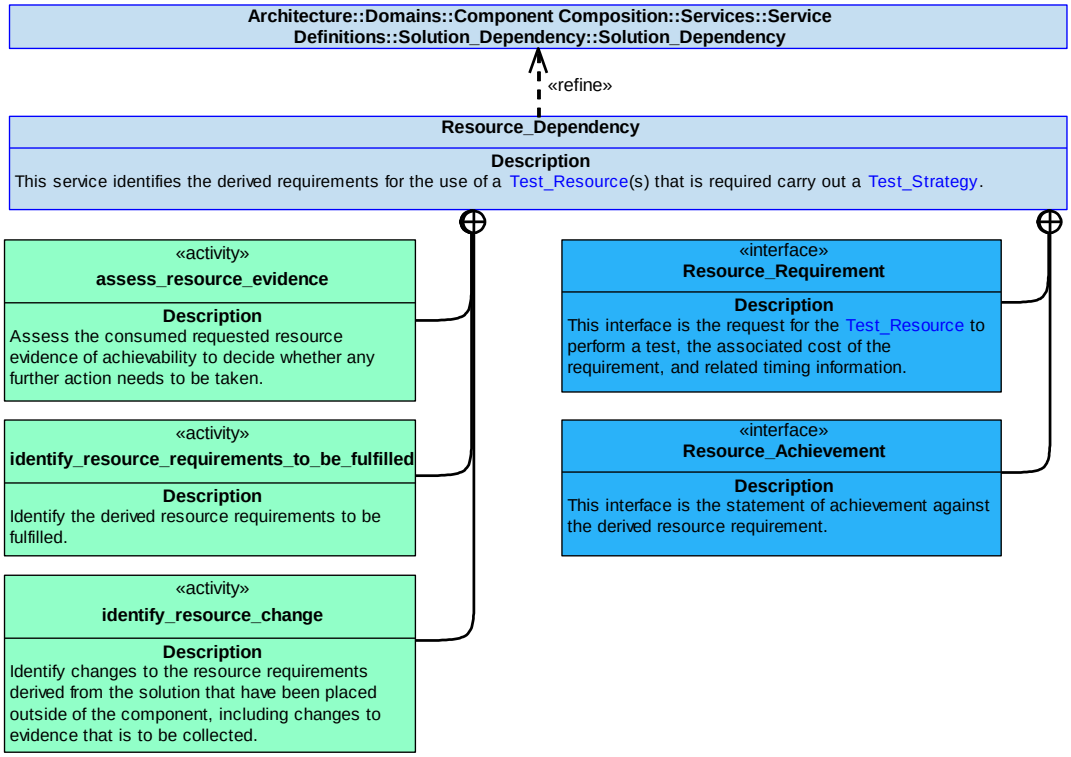

Figure 1155: Resource_Dependency Service Policy

Resource_Dependency

This service identifies the derived requirements for the use of a Test_Resource(s) that is required 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.

B.2.71.7.1.5 System_Condition

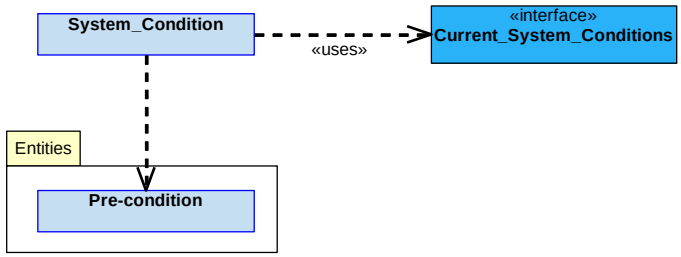

Figure 1156: System_Condition Service Definition

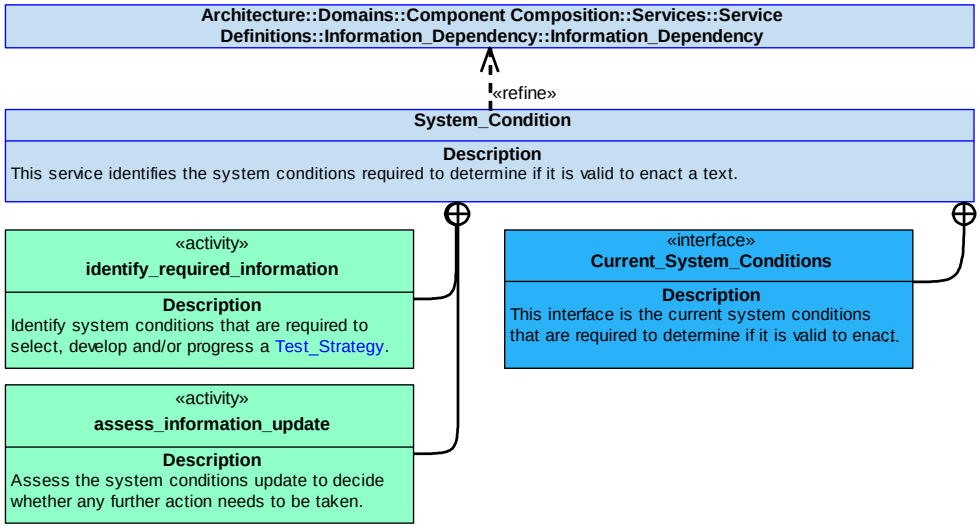

Figure 1157: System_Condition Service Policy

System_Condition

This service identifies the system conditions required to determine if it is valid to enact a text.

Interface

Current_System_Conditions

This interface is the current system conditions that are required to determine if it is valid to enact a test, e.g. if the weight is on wheels, if the aircraft is below the constraint g limit, or if the Test_Resource is online or offline.

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.

B.2.71.7.1.6 Constraint

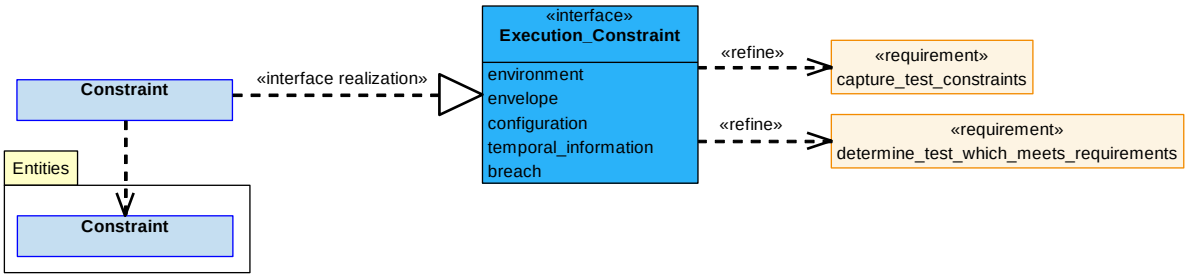

Figure 1158: Constraint Service Definition

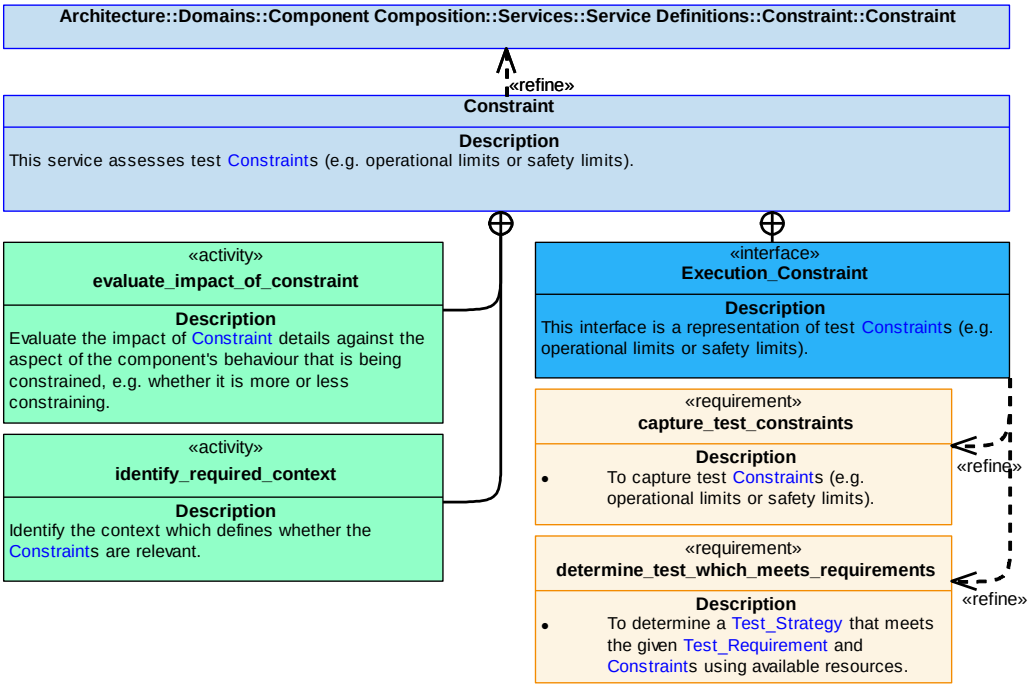

Figure 1159: Constraint Service Policy

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, 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.

B.2.71.7.1.7 Capability

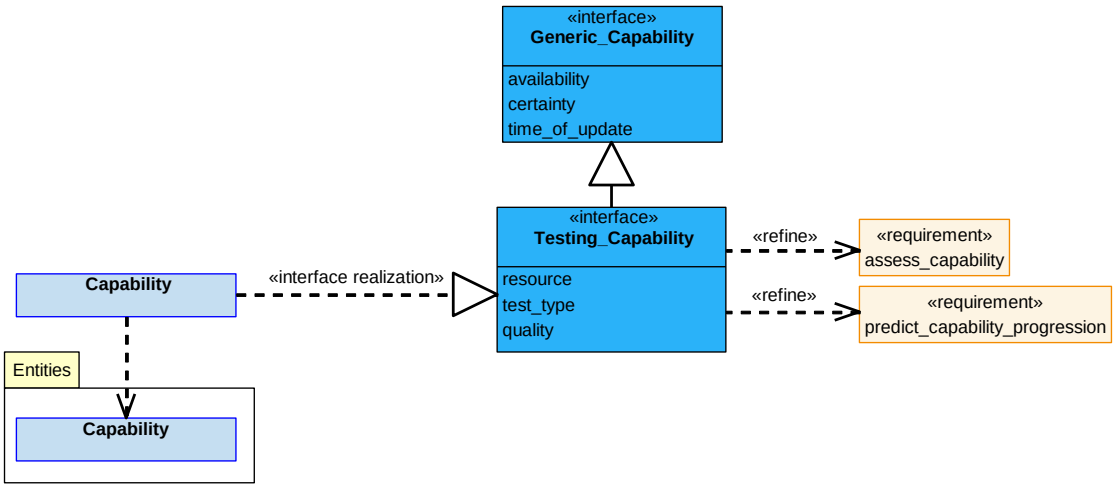

Figure 1160: Capability Service Definition

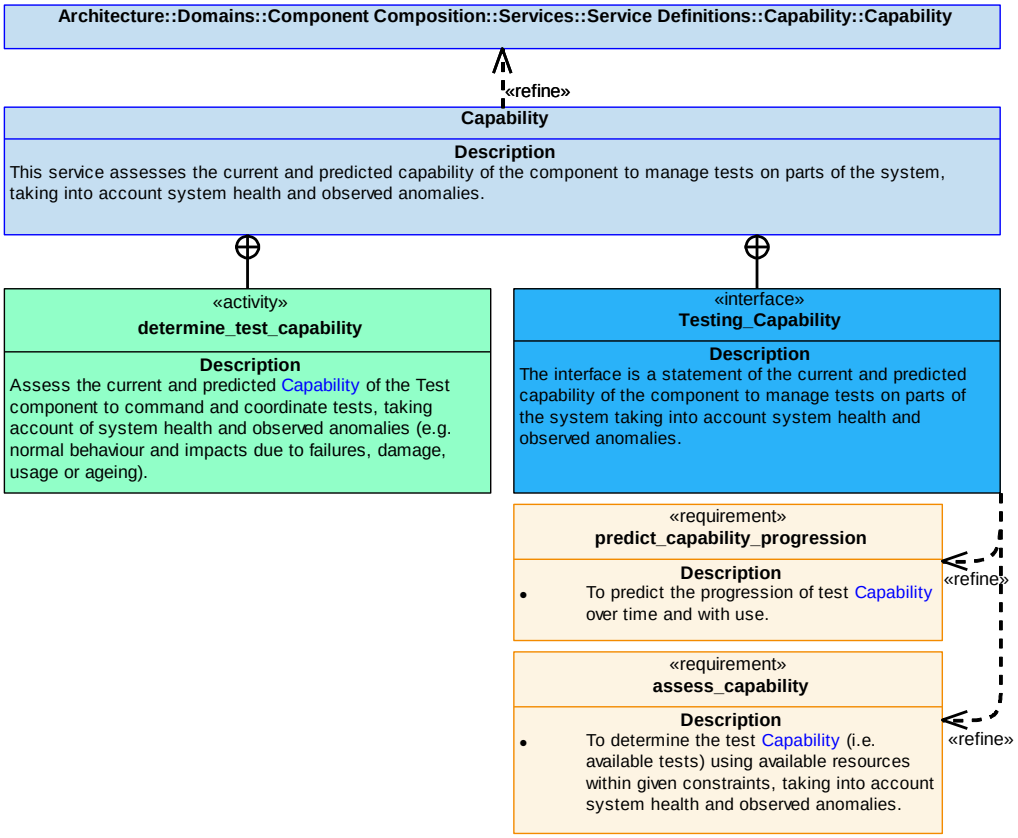

Figure 1161: Capability Service Policy

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).

B.2.71.7.1.8 Capability_Evidence

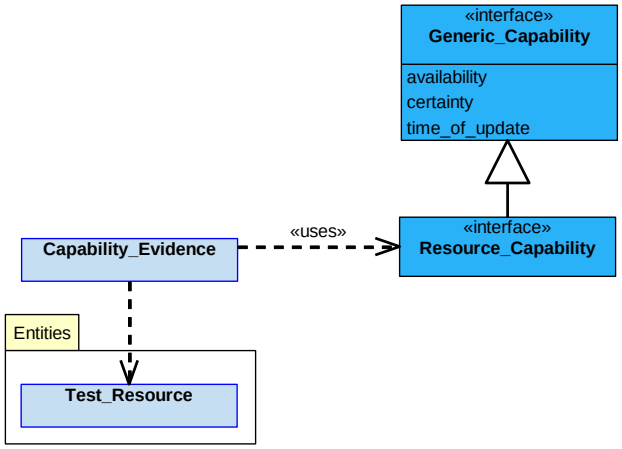

Figure 1162: Capability_Evidence Service Definition

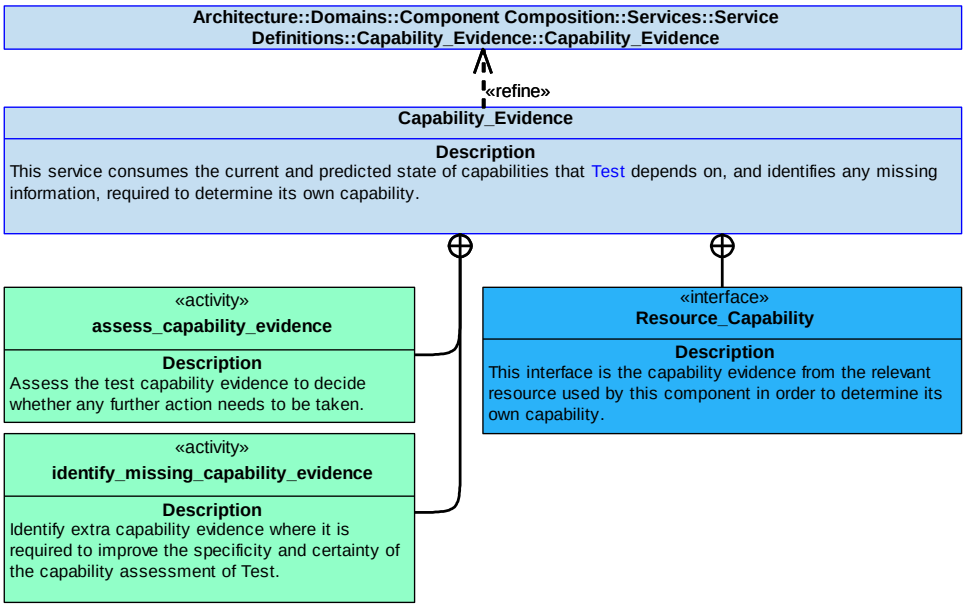

Figure 1163: Capability_Evidence Service Policy

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.

Interface

Resource_Capability

This interface is the capability evidence from the relevant resource used by this component in order to determine its own capability.

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.

B.2.71.7.2 Service Dependencies

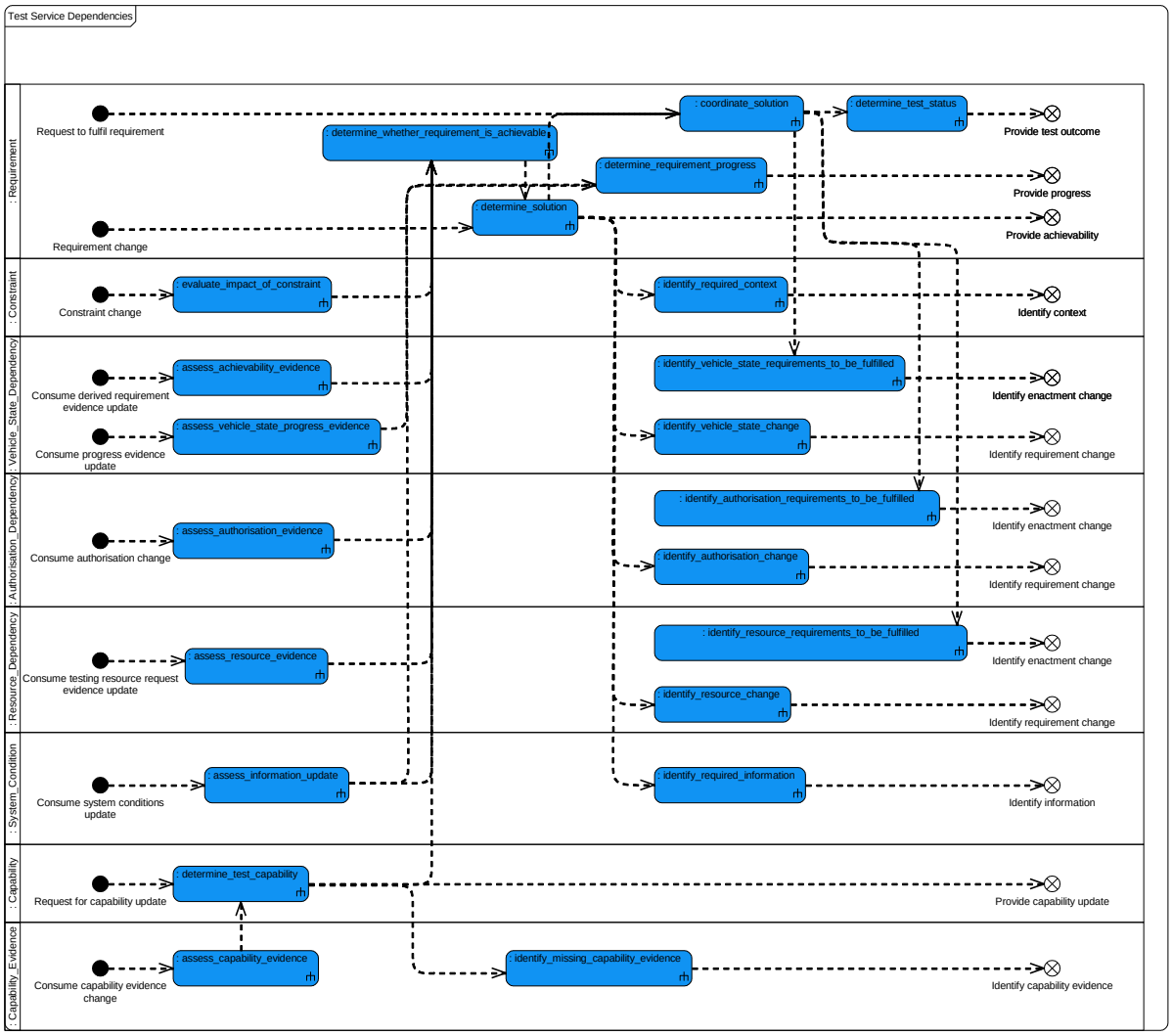

Figure 1164: Test Service Dependencies

B.2.72 Threats

B.2.72.1 Role

The role of Threats is to determine the level of risk posed by a tactical object to another tactical object.

B.2.72.2 Overview

Control Architecture

Threats is a service component as defined in the Control Architecture policy.

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.

B.2.72.3 Service Summary

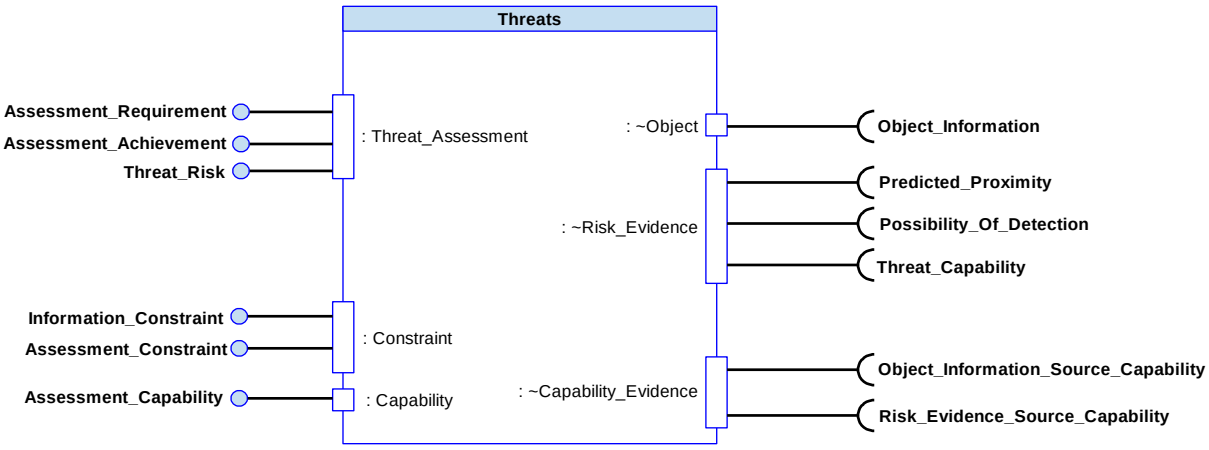

Figure 1165: Threats Service Summary

B.2.72.4 Responsibilities

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).

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).

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.

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).

identify_whether_requirement_remains_achievable

- To identify whether a Threat_Assessment is still achievable given current Capability and dependencies.

predict_capability_progression

- To predict the progression of Threat_Assessment generation Capability over time and with use.

B.2.72.5 Subject Matter Semantics

The subject matter of Threats is entities in the external environment that has 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.

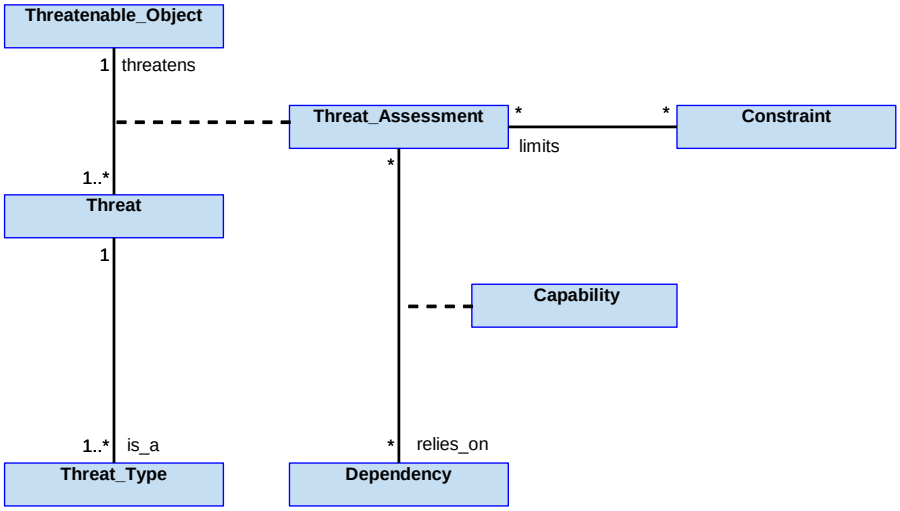

Figure 1166: Threats Semantics

B.2.72.5.1 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).

B.2.72.6 Design Rationale

B.2.72.6.1 Assumptions

- The component will have access to information about tactical objects, their location and their capabilities against the Threatenable_Object.

B.2.72.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy).

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. if the object is able to observe ownship or if 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).

B.2.72.6.3 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.

B.2.72.6.4 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.

B.2.72.7 Services

B.2.72.7.1 Service Definitions

B.2.72.7.1.1 Threat_Assessment

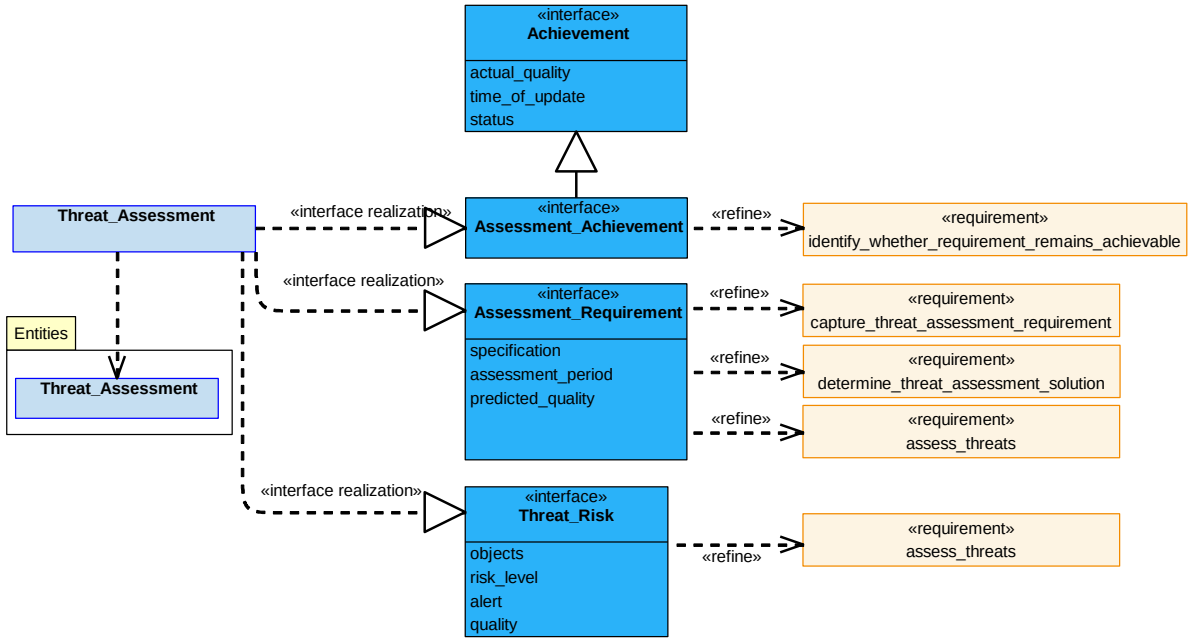

Figure 1167: Threat_Assessment Service Definition

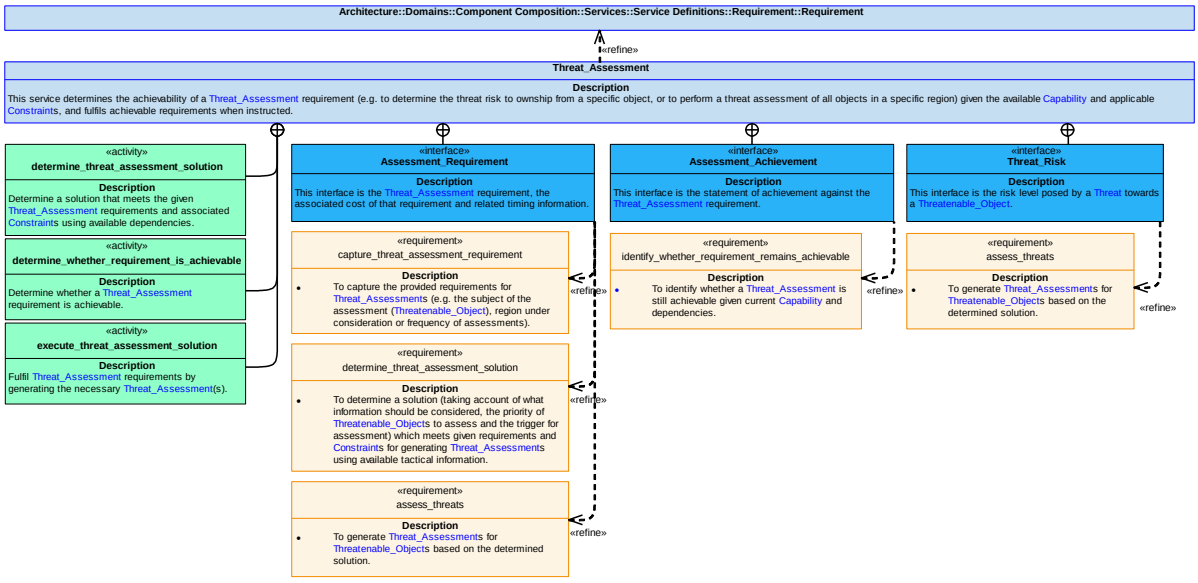

Figure 1168: Threat_Assessment Service Policy

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).

B.2.72.7.1.2 Object

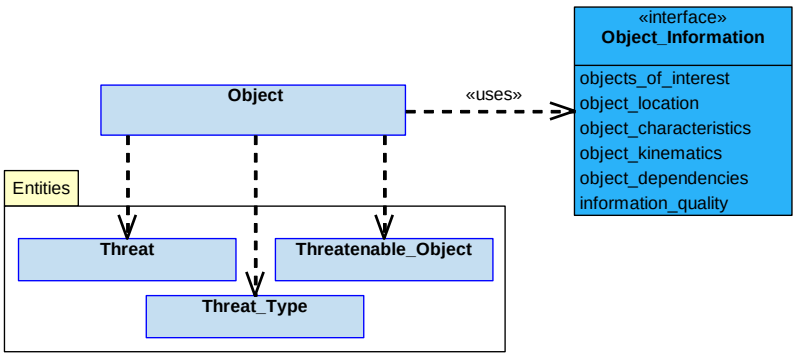

Figure 1169: Object Service Definition

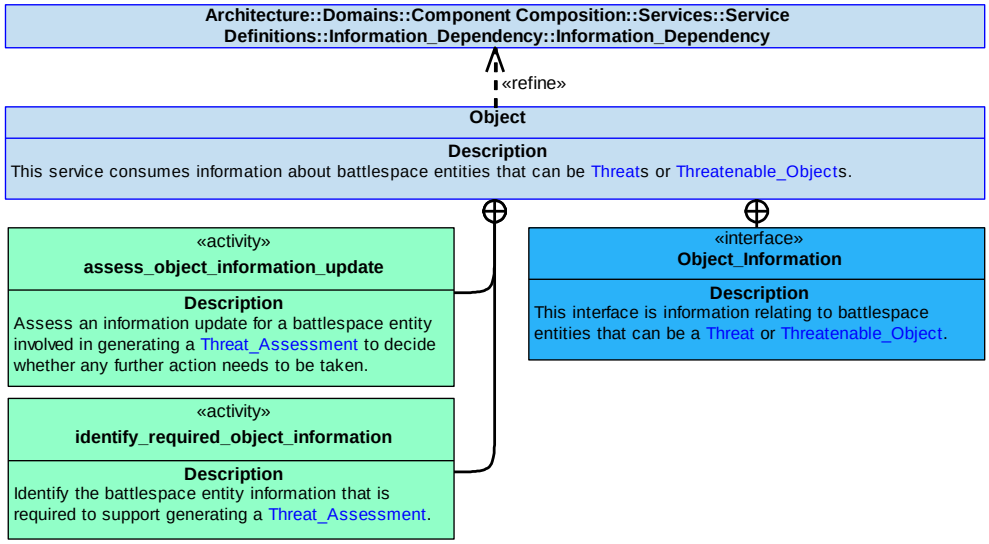

Figure 1170: Object Service Policy

Object

This service consumes information about battlespace entities that can be Threats or Threatenable_Objects.

Interface

Object_Information

This interface is information relating to battlespace entities that can be a Threat or Threatenable_Object.

Attributes

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.

B.2.72.7.1.3 Risk_Evidence

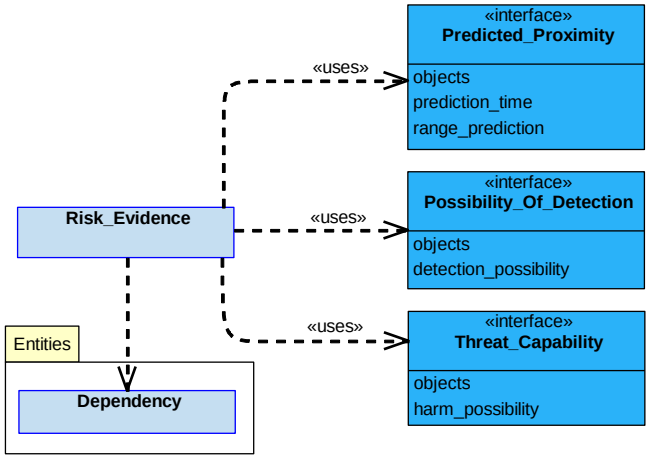

Figure 1171: Risk_Evidence Service Definition

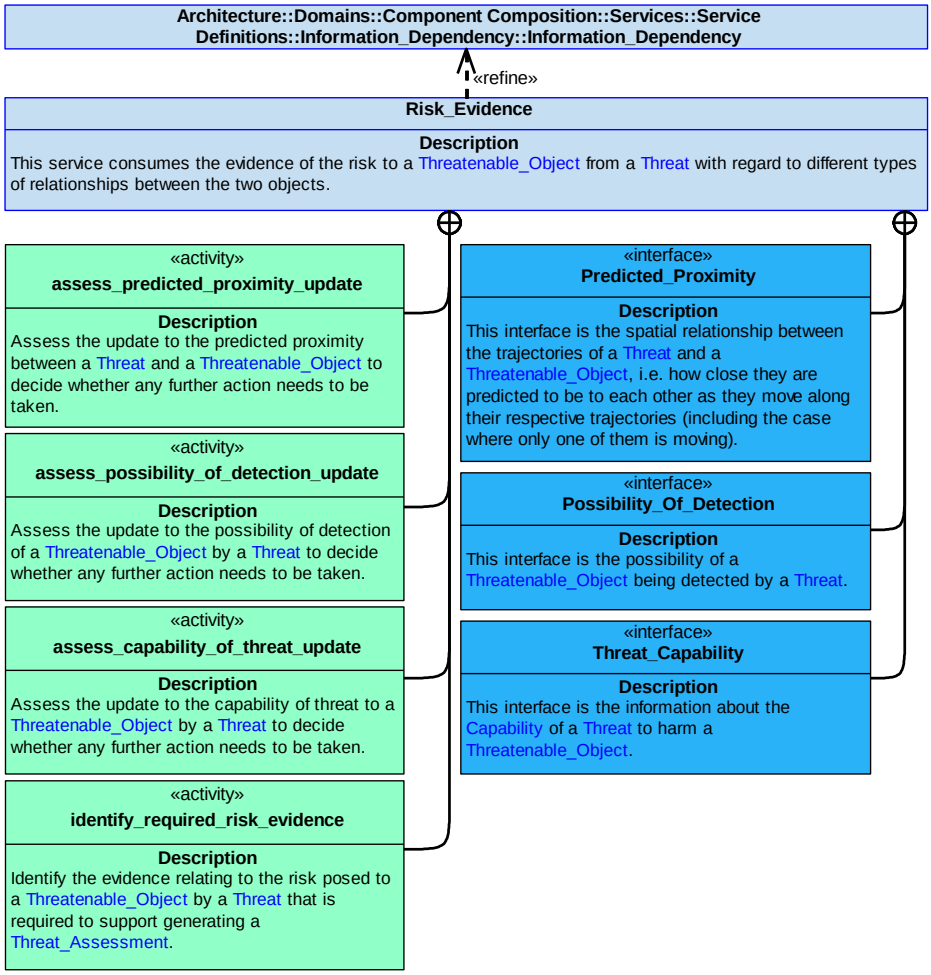

Figure 1172: Risk_Evidence Service Policy

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

Predicted_Proximity

This interface is the spatial relationship between the trajectories of a Threat and a Threatenable_Object, i.e. how close they are predicted to be to each other as they move along their respective trajectories (including the case where only one of them is moving).

Attributes

objects	The two objects that are the subject of a proximity prediction assessment.
prediction_time	The time period over which the proximity prediction assessment is valid.
range_prediction	The range between two objects based on their predicted 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_predicted_proximity_update

Assess the update to the 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.

B.2.72.7.1.4 Constraint

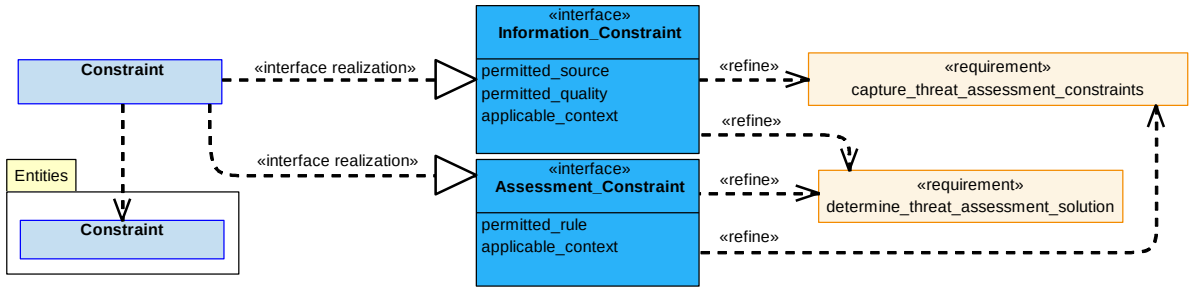

Figure 1173: Constraint Service Definition

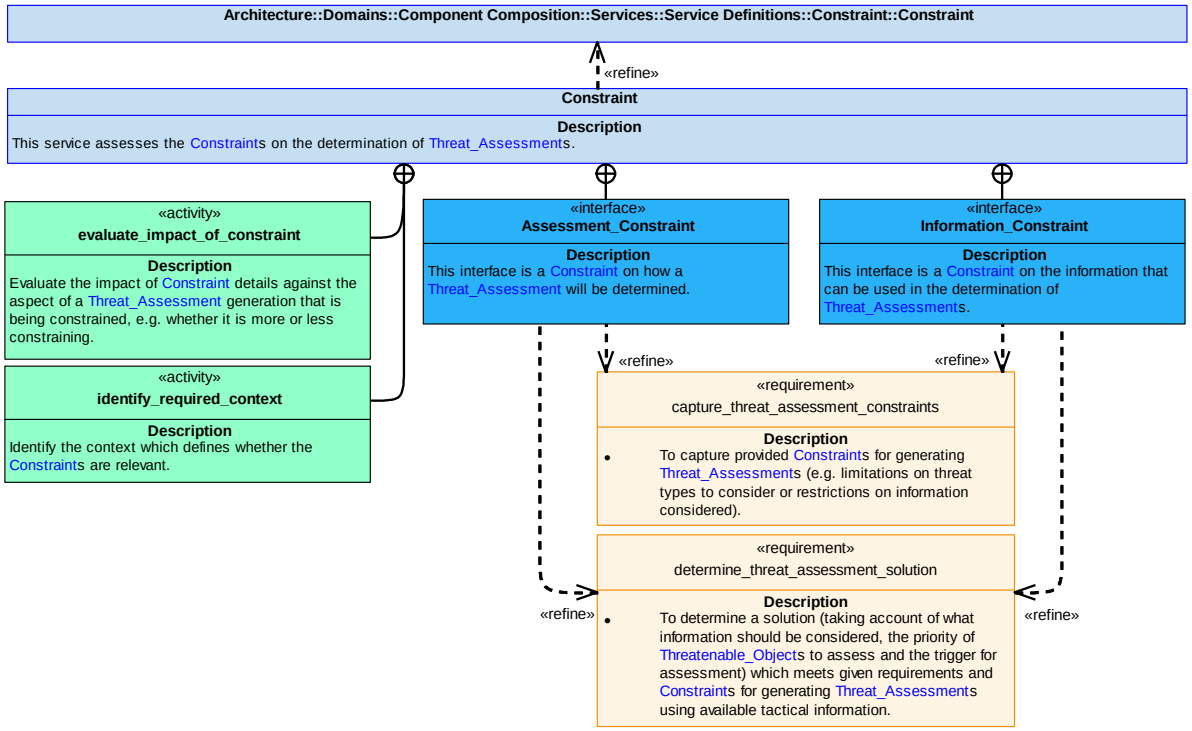

Figure 1174: Constraint Service Policy

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.

B.2.72.7.1.5 Capability

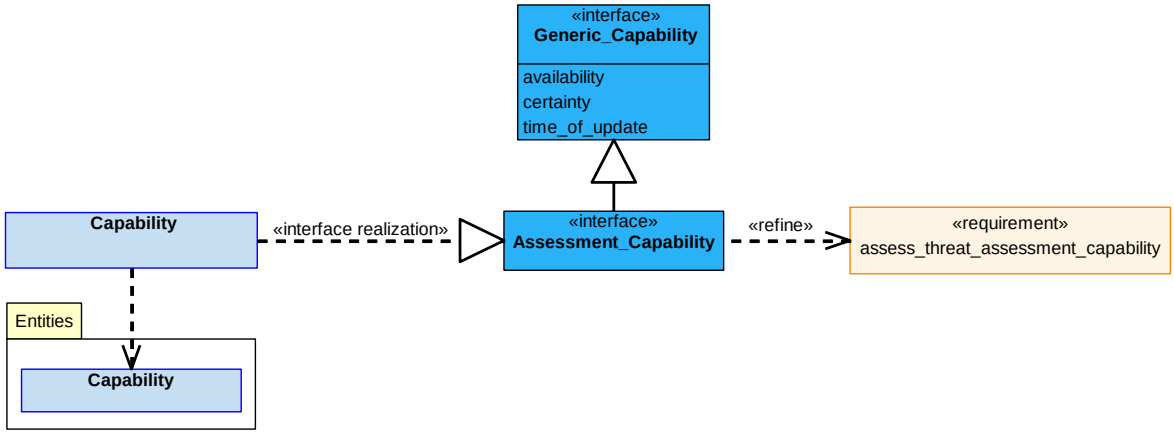

Figure 1175: Capability Service Definition

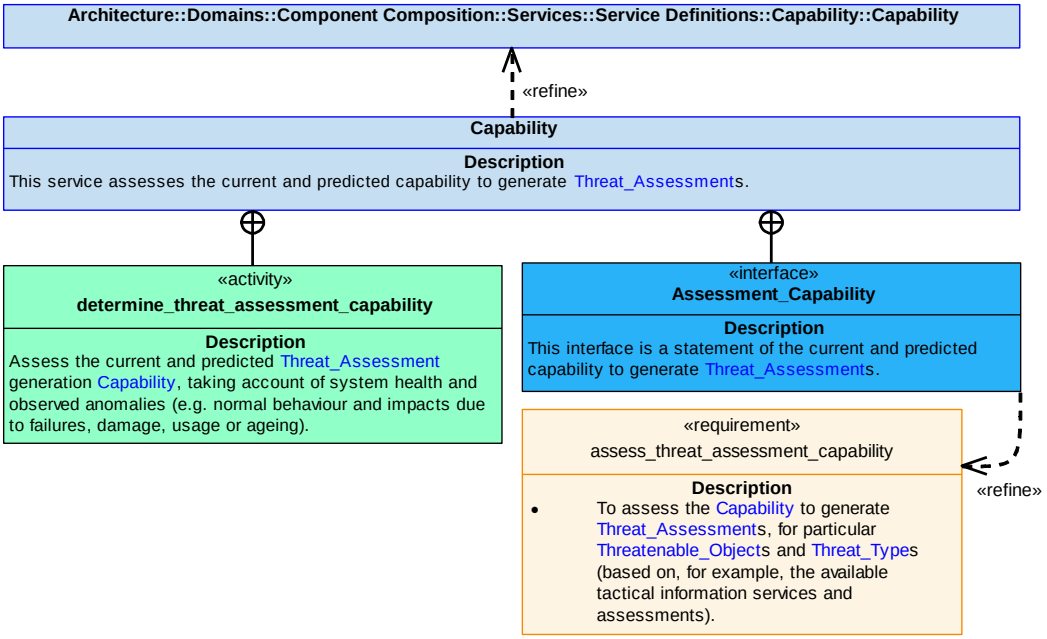

Figure 1176: Capability Service Policy

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).

B.2.72.7.1.6 Capability_Evidence

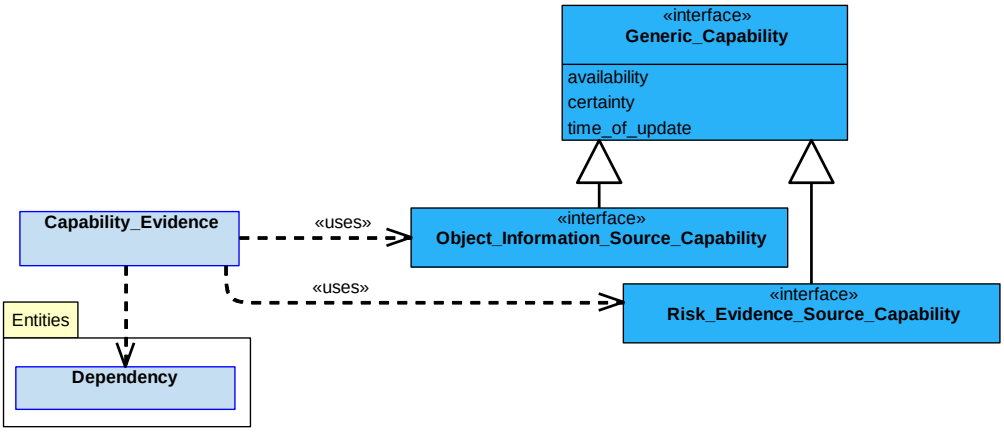

Figure 1177: Capability_Evidence Service Definition

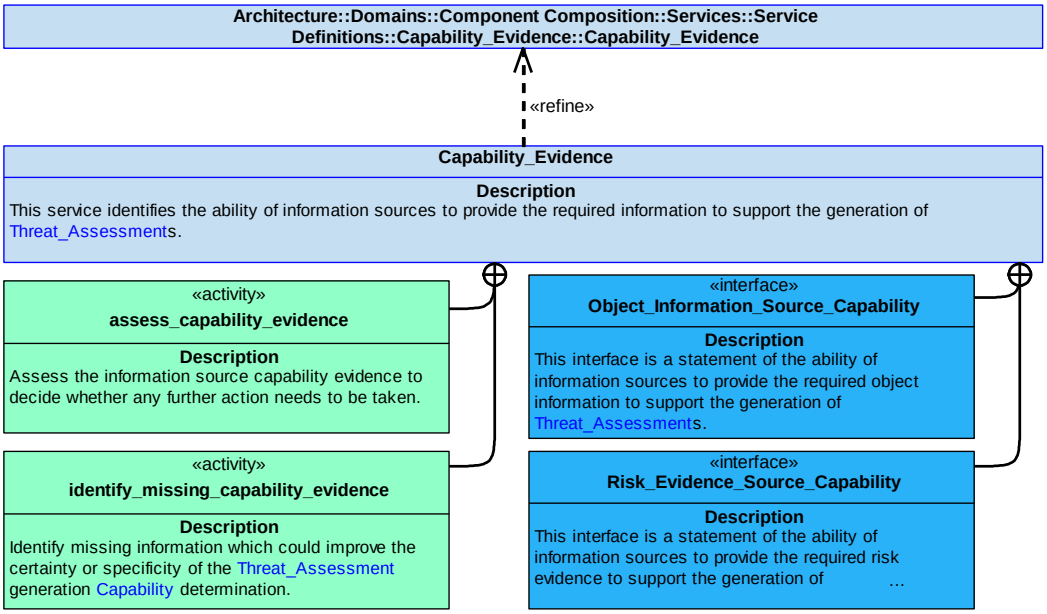

Figure 1178: Capability_Evidence Service Policy

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.

B.2.72.7.2 Service Dependencies

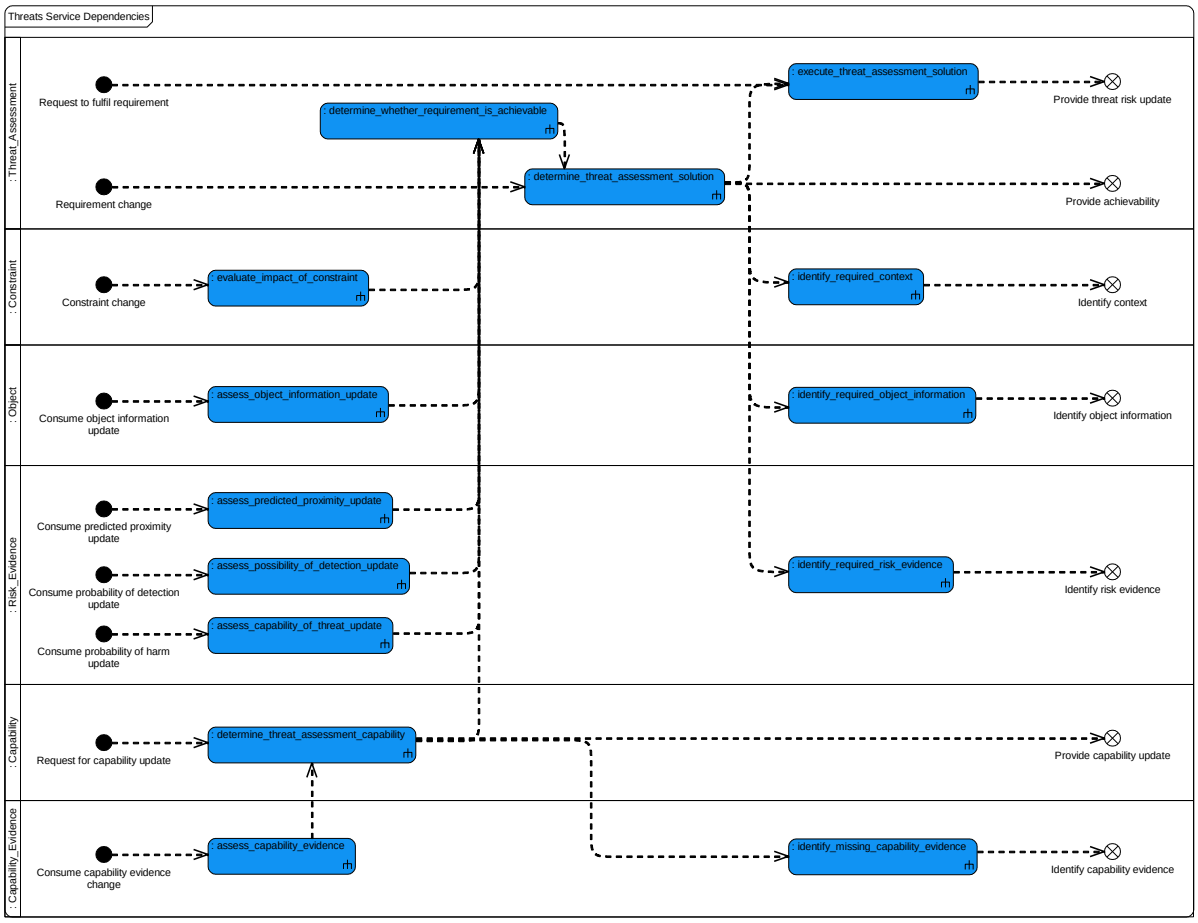

Figure 1179: Threats Service Dependencies

B.2.73 Trajectory Prediction

B.2.73.1 Role

The role of Trajectory Prediction is to predict a spatial trajectory or trajectories for real world objects.

B.2.73.2 Overview

Control Architecture

Trajectory Prediction is a service component as defined in the Control Architecture policy.

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.

B.2.73.3 Service Summary

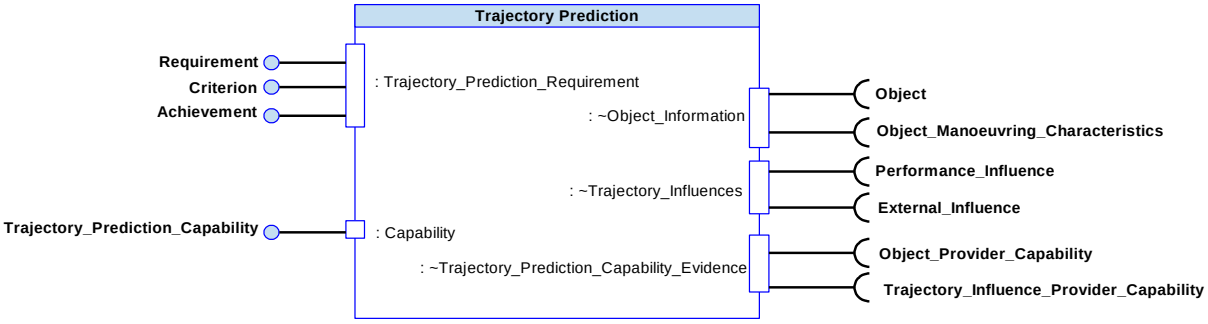

Figure 1180: Trajectory Prediction Service Summary

B.2.73.4 Responsibilities

capture_requirements_for_trajectory_prediction

- To capture the Requirements to be satisfied by the Trajectory_Prediction_Solution.

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).

capture_measurement_criteria_for_trajectory_prediction

- To capture Measurement_Criterion/ criteria for Trajectory_Prediction_Solutions.

capture_object_information

- To capture information about an Object's spatial state.

determine_trajectory_prediction_solution_for_object

- To determine the Trajectory_Prediction_Solution for an Object.

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.

capture_manoeuvring_characteristics

- To capture the Manoeuvring_Characteristics of an Object or type of object.

identify_progress

- To identify the progress against a Requirement.

identify_whether_requirement_remains_achievable

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

B.2.73.5 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.

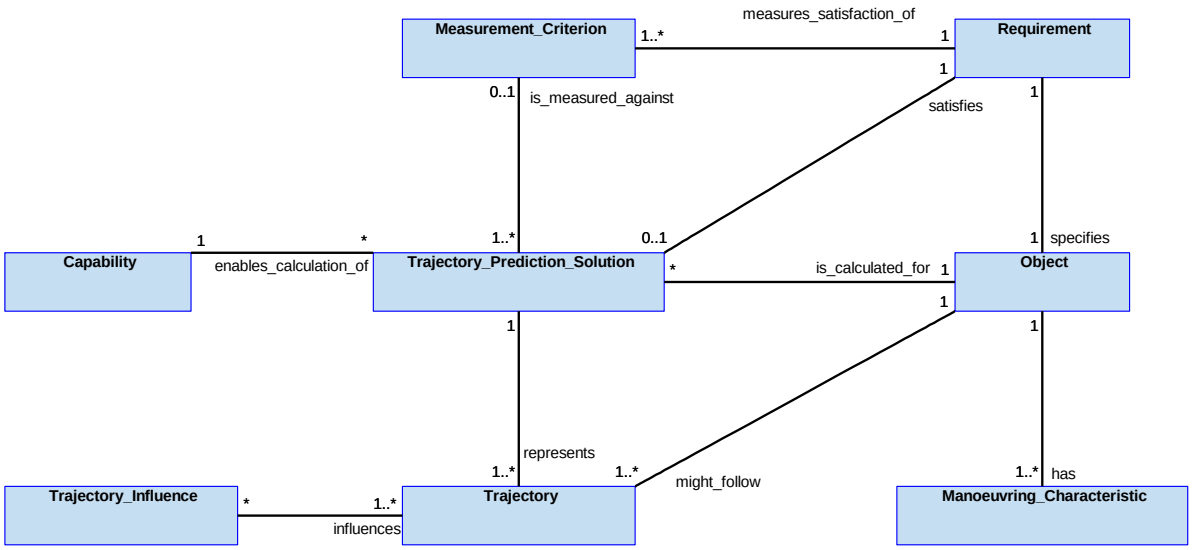

Figure 1181: Trajectory Prediction Semantics

B.2.73.5.1 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).

B.2.73.6 Design Rationale

B.2.73.6.1 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.

B.2.73.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy).

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.

B.2.73.6.3 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. In accordance with UK MOD direction (see Safety Analysis policy) 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.

B.2.73.6.4 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.

B.2.73.7 Services

B.2.73.7.1 Service Definitions

B.2.73.7.1.1 Trajectory_Prediction_Requirement

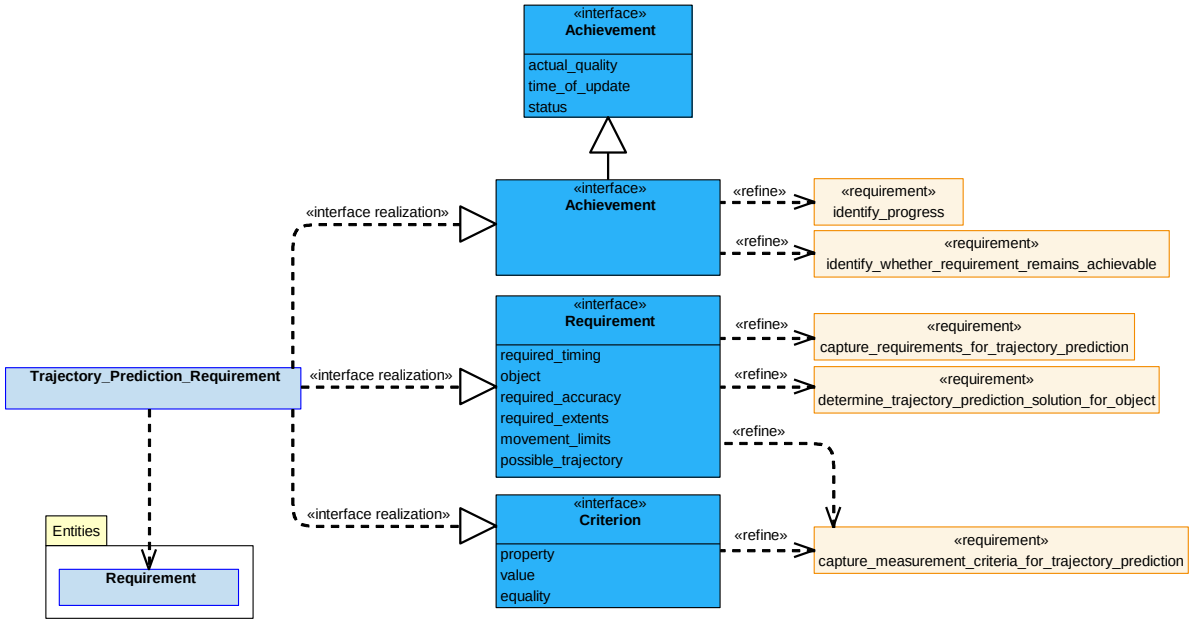

Figure 1182: Trajectory_Prediction_Requirement Service Definition

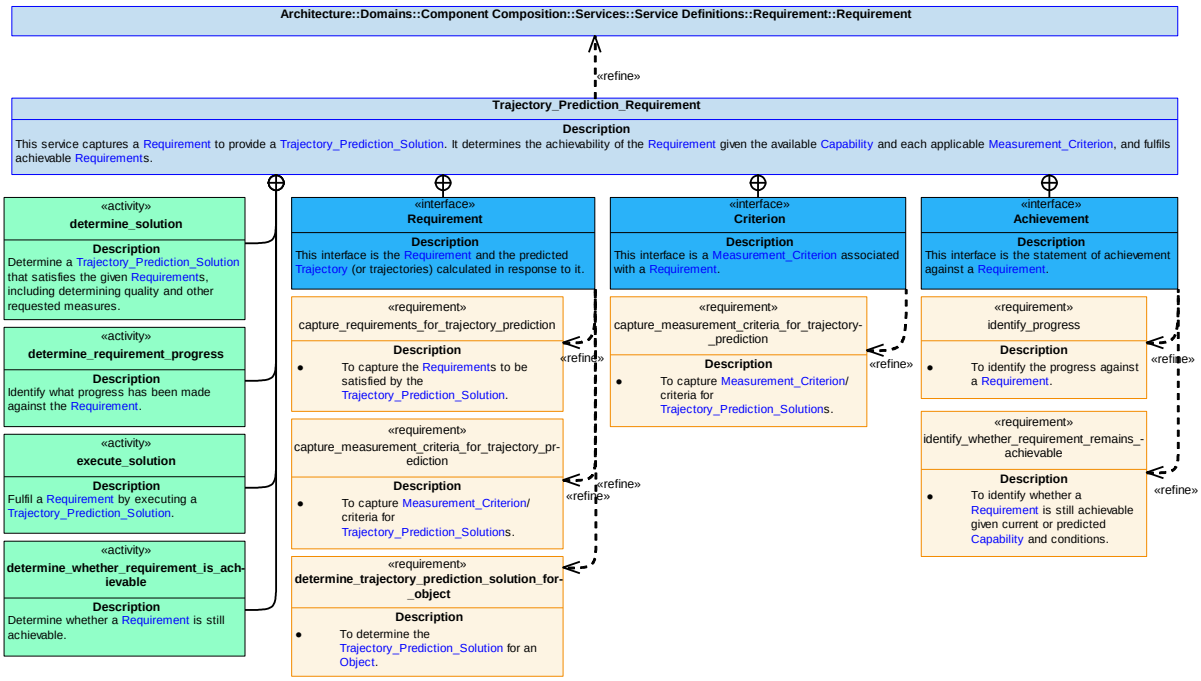

Figure 1183: Trajectory_Prediction_Requirement Service Policy

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.

B.2.73.7.1.2 Object_Information

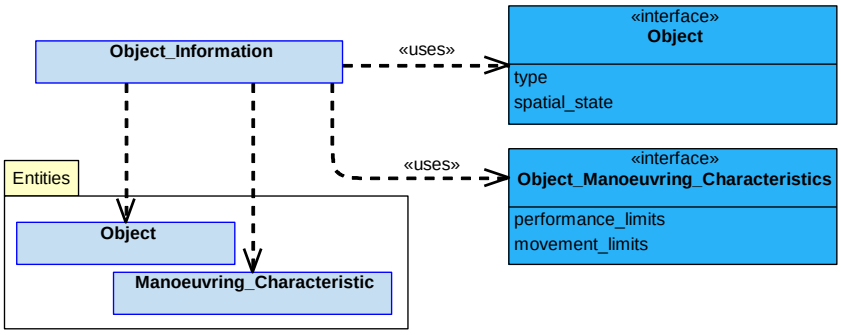

Figure 1184: Object_Information Service Definition

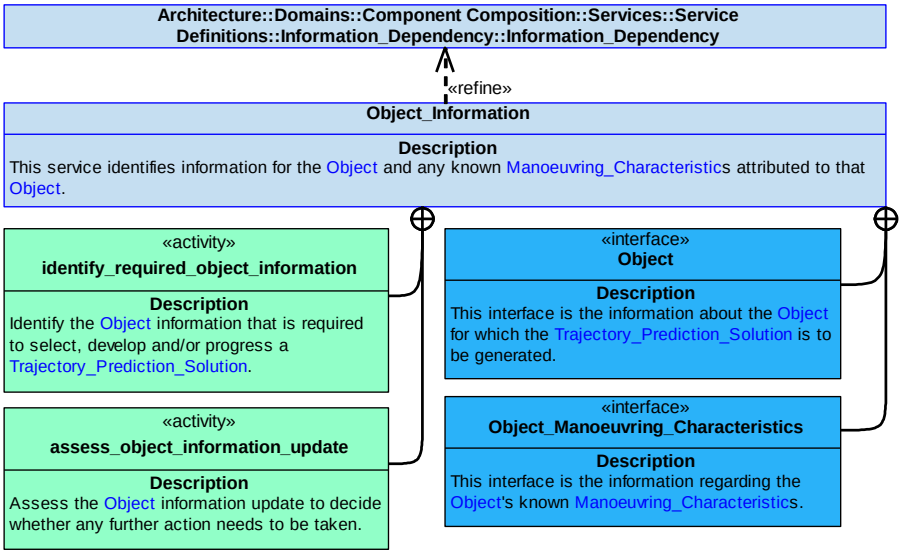

Figure 1185: Object_Information Service Policy

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.

B.2.73.7.1.3 Trajectory_Influences

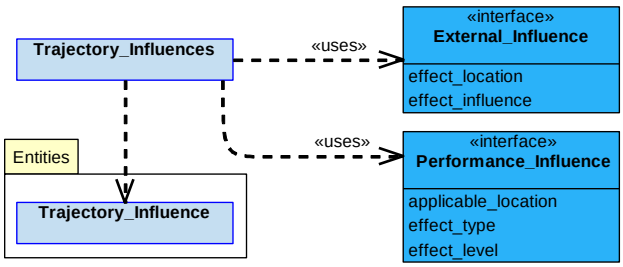

Figure 1186: Trajectory_Influences Service Definition

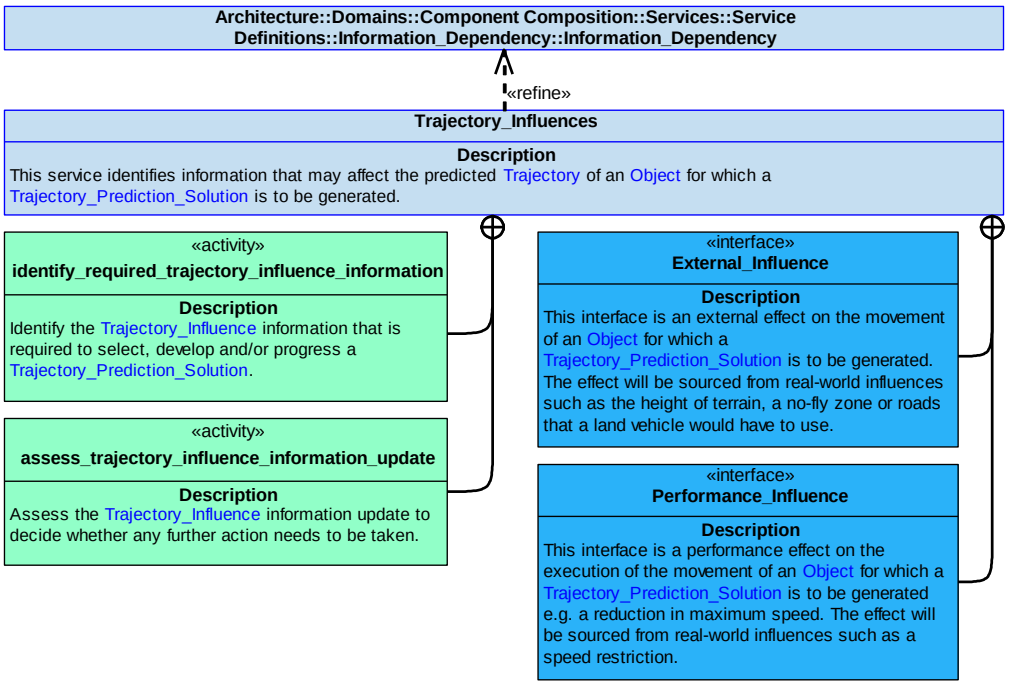

Figure 1187: Trajectory_Influences Service Policy

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.

B.2.73.7.1.4 Capability

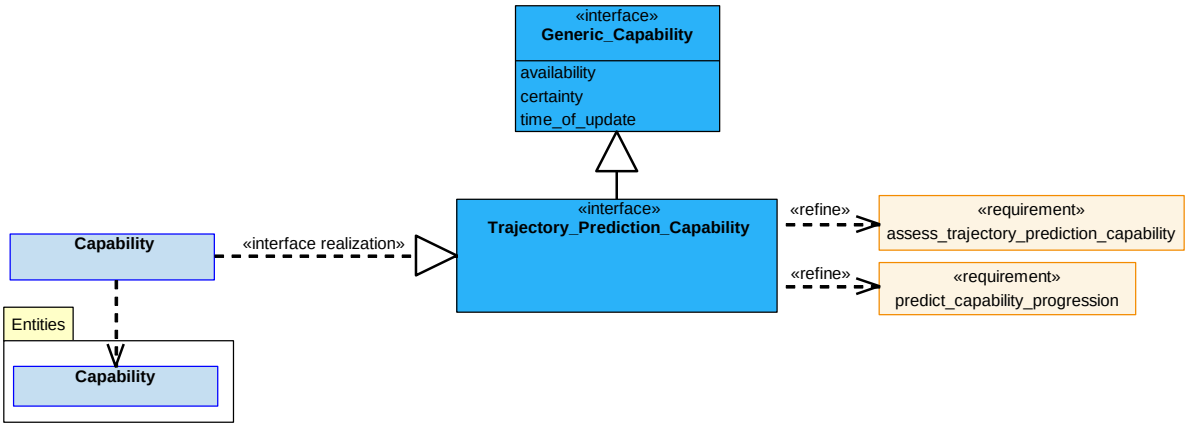

Figure 1188: Capability Service Definition

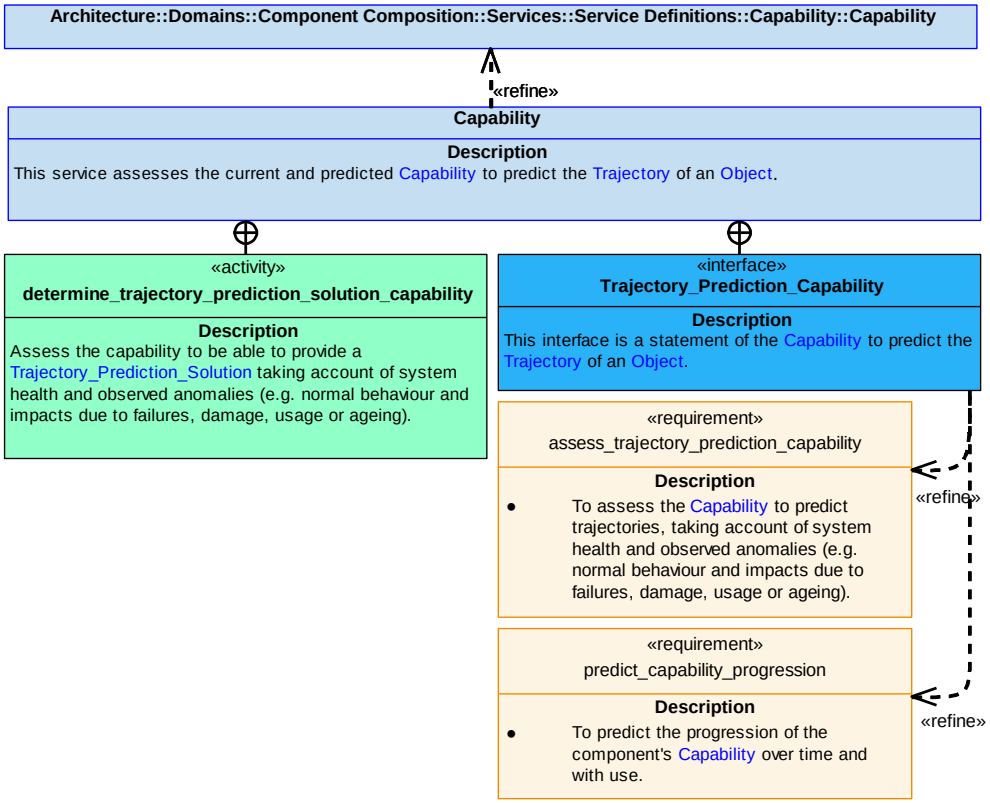

Figure 1189: Capability Service Policy

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).

B.2.73.7.1.5 Trajectory_Prediction_Capability_Evidence

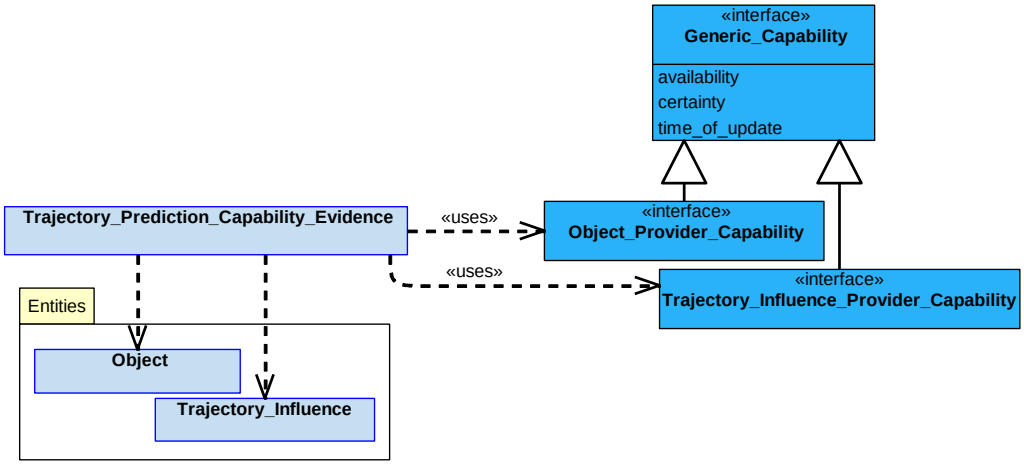

Figure 1190: Trajectory_Prediction_Capability_Evidence Service Definition

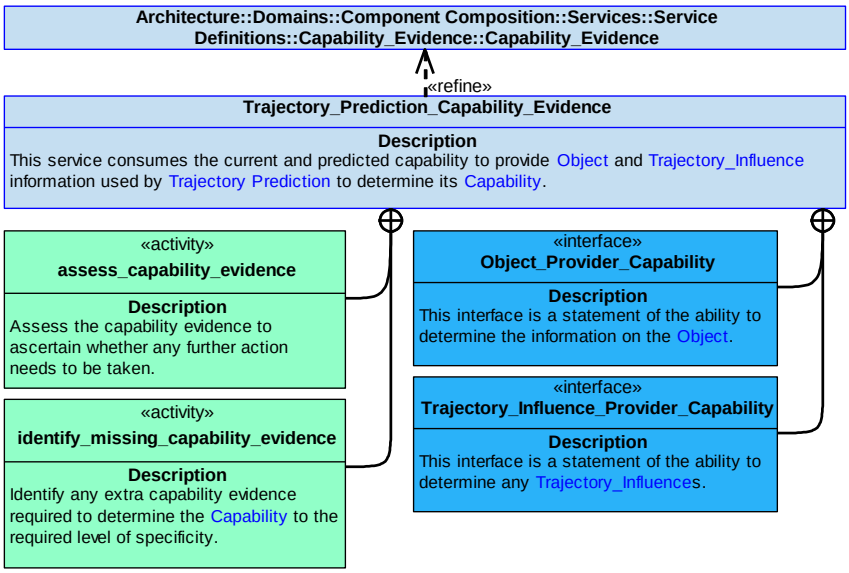

Figure 1191: Trajectory_Prediction_Capability_Evidence Service Policy

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.

Interfaces

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.

Activities

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.

B.2.73.7.2 Service Dependencies

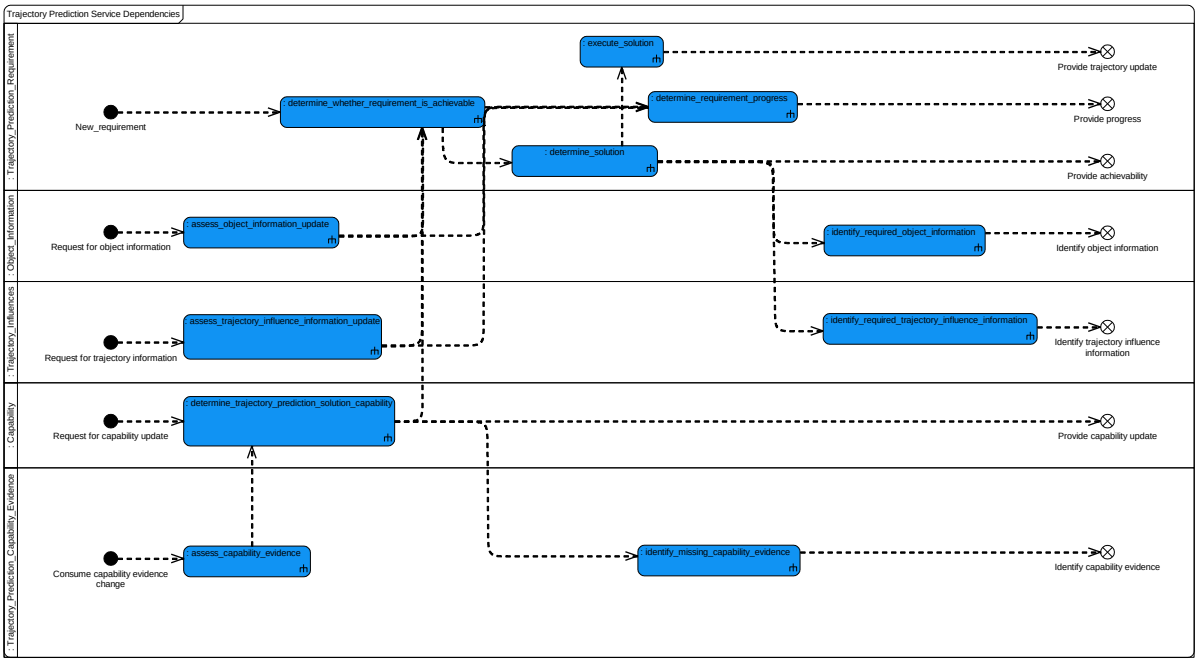

Figure 1192: Trajectory Prediction Service Dependencies

B.2.74 Undercarriage

B.2.74.1 Role

The role of Undercarriage is to control the deployment or retraction of the undercarriage.

B.2.74.2 Overview

Control Architecture

Undercarriage is an action component as defined in the Control Architecture policy.

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.

B.2.74.3 Service Summary

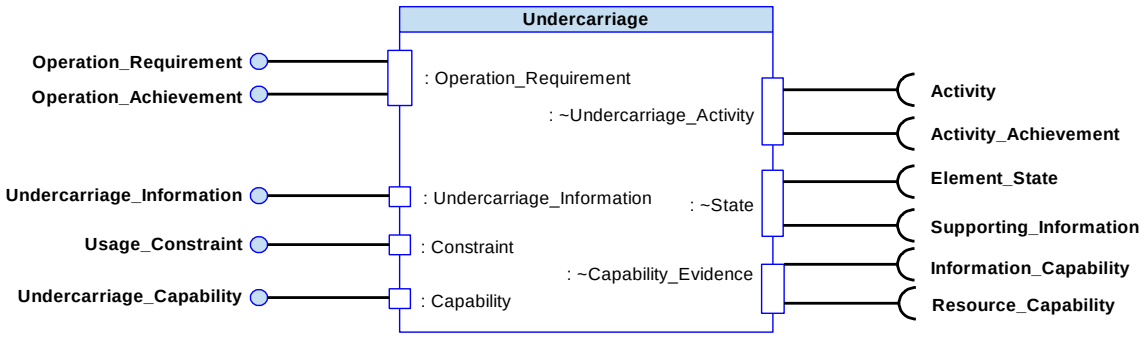

Figure 1193: Undercarriage Service Summary

B.2.74.4 Responsibilities

capture_undercarriage_requirement

- To capture provided Requirements for deployment or retraction of undercarriage.

determine_undercarriage_state

- To determine the Undercarriage_State (e.g. up, travelling, down or weight on or off wheels).

coordinate_undercarriage_movement

- To coordinate the deployment or retraction of the undercarriage to fulfil an undercarriage Requirement.

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).

predict_capability_progression

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

capture_undercarriage_constraints

- To capture undercarriage Constraints.

identify_pre_condition

- To identify Pre-conditions required to support the Undercarriage_Solution or an Undercarriage_Step.

identify_progress_of_undercarriage_solution

- To identify the progress of an Undercarriage_Solution against the Requirements.

identify_missing_information

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

determine_undercarriage_solution

- To determine an Undercarriage_Solution that meets the given Requirements and Constraints.

identify_undercarriage_solution_in_progress_remains_feasible

- To identify whether an Undercarriage_Solution in progress remains feasible given current Capability.

B.2.74.5 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.

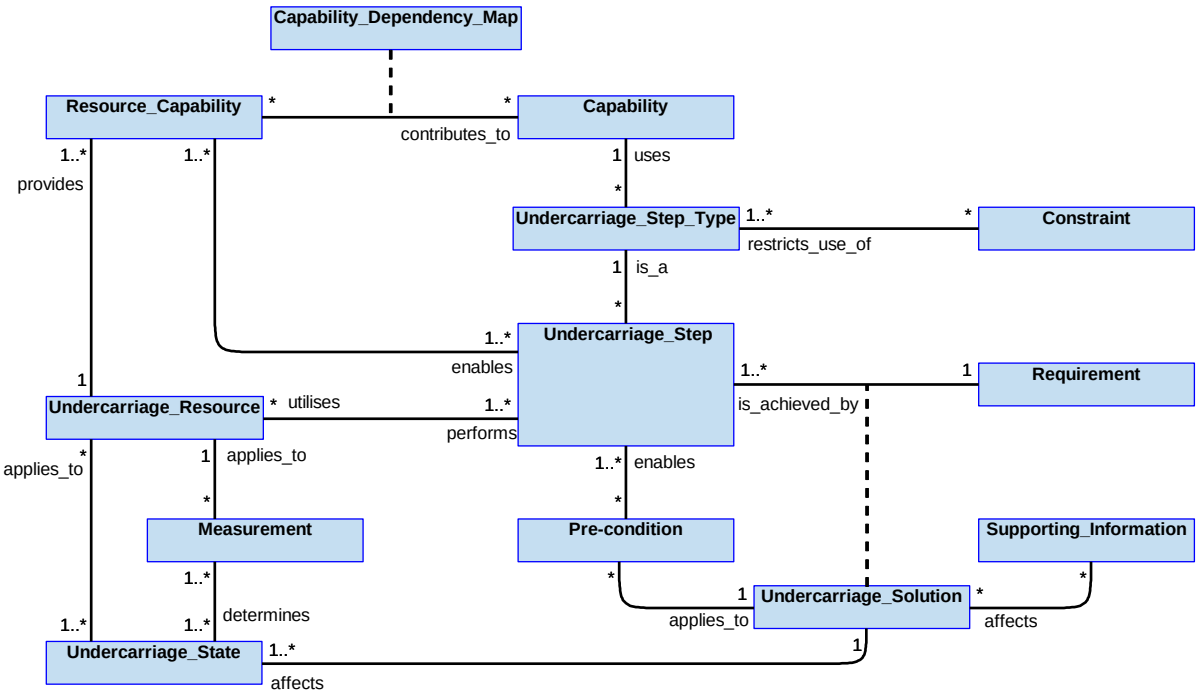

Figure 1194: Undercarriage Semantics

B.2.74.5.1 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 activity 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 activity the component will coordinate that, when performed, contributes to the extension or retraction of the undercarriage.

Undercarriage_Step_Type

The kind of activity this component knows how to coordinate, e.g. opening doors, extending and locking the landing gear, despinning the wheel.

B.2.74.6 Design Rationale

B.2.74.6.1 Assumptions

- Under normal conditions, undercarriage positions may be dependent on factors such as weight on or off wheels, etc.

B.2.74.6.2 Design Considerations

Directly Applicable Policies

These policies 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 deployment 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.

B.2.74.6.3 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.

B.2.74.6.4 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.

B.2.74.7 Services

B.2.74.7.1 Service Definitions

B.2.74.7.1.1 Operation_Requirement

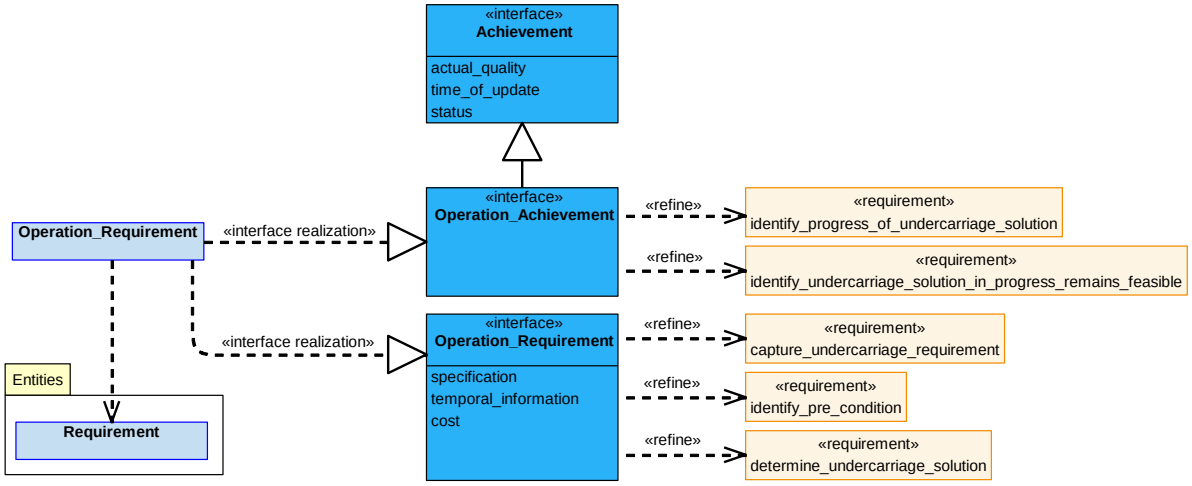

Figure 1195: Operation_Requirement Service Definition

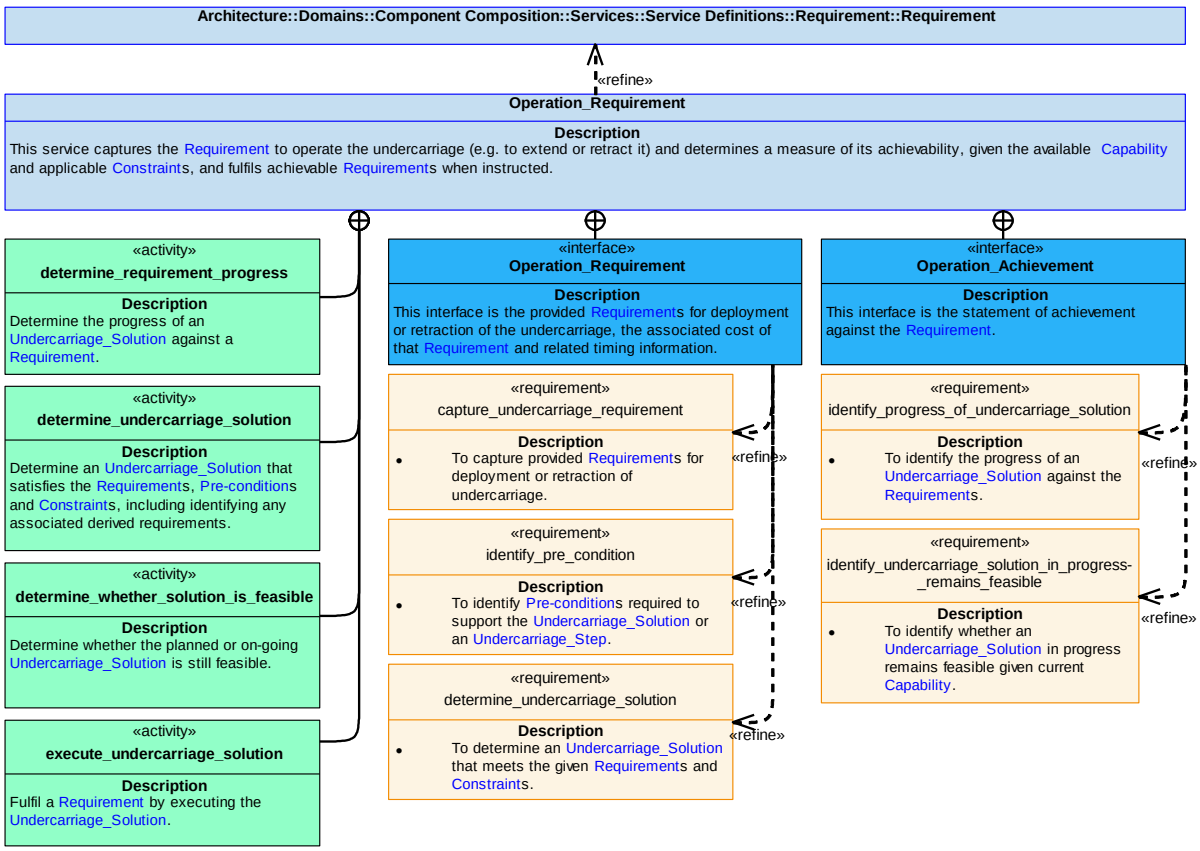

Figure 1196: Operation_Requirement Service Policy

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.

B.2.74.7.1.2 Undercarriage_Activity

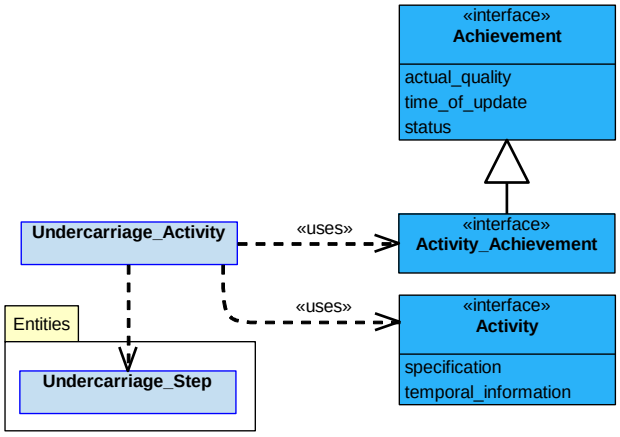

Figure 1197: Undercarriage_Activity Service Definition

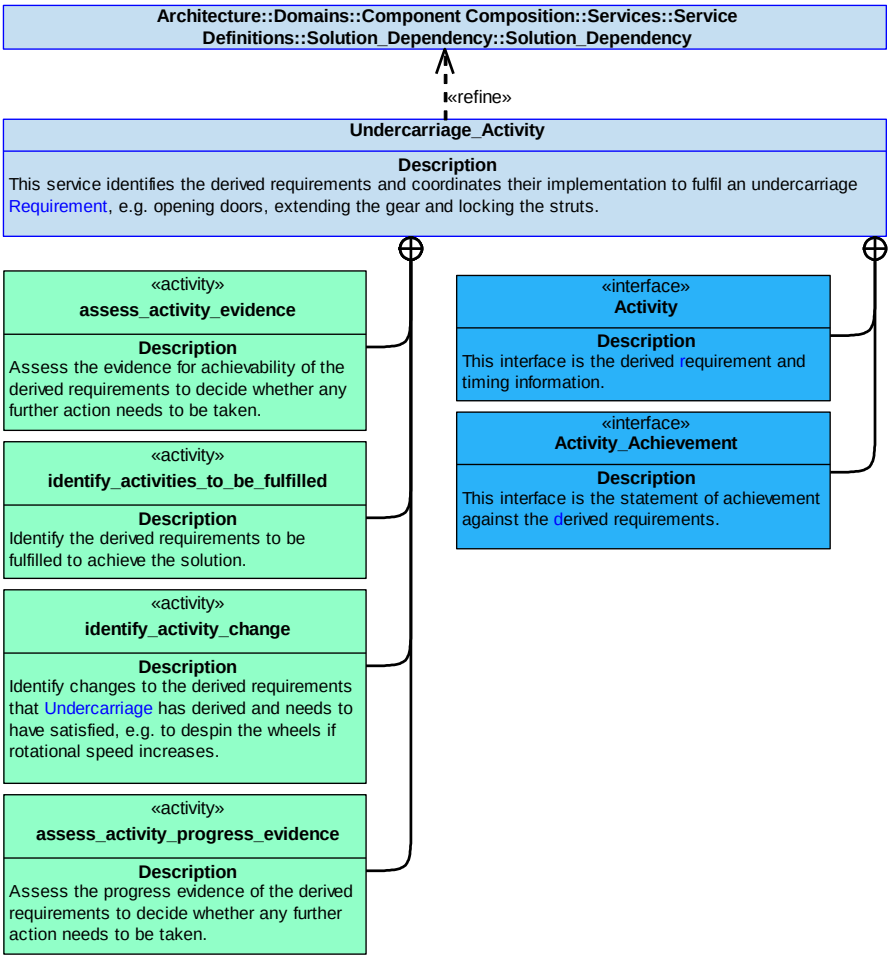

Figure 1198: Undercarriage_Activity Service Policy

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.

B.2.74.7.1.3 Undercarriage_Information

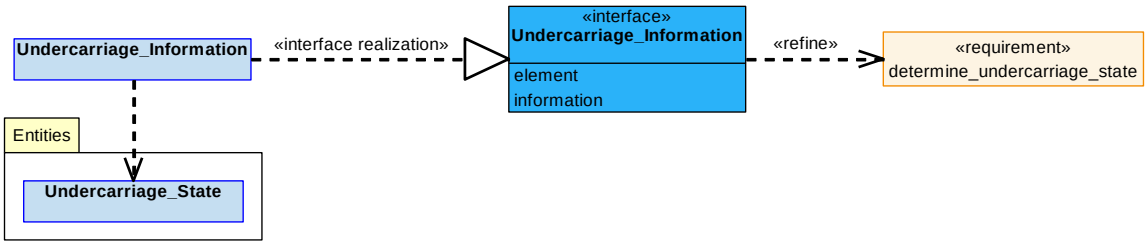

Figure 1199: Undercarriage_Information Service Definition

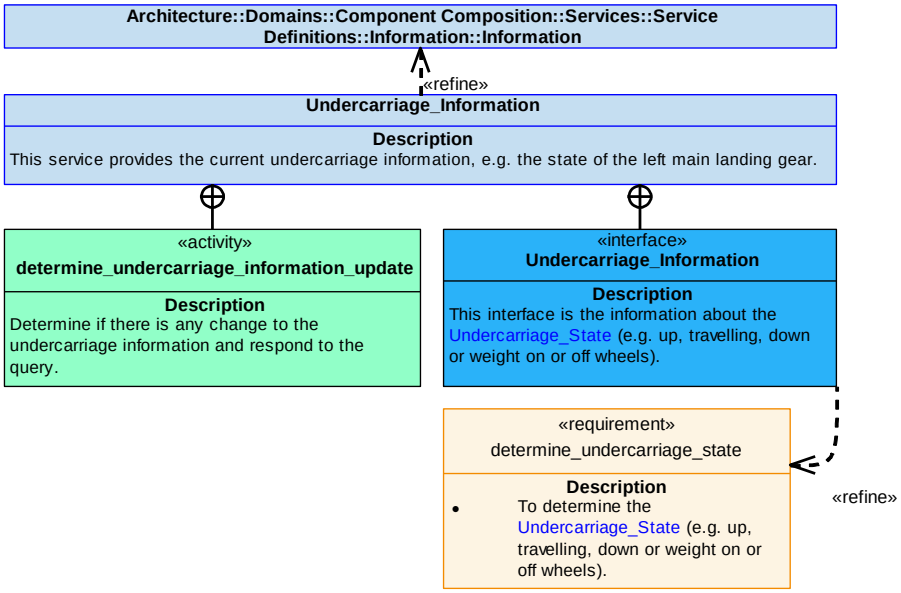

Figure 1200: Undercarriage_Information Service Policy

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.

B.2.74.7.1.4 State

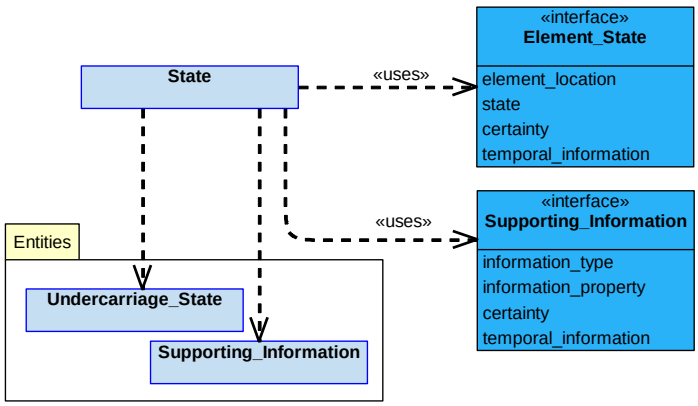

Figure 1201: State Service Definition

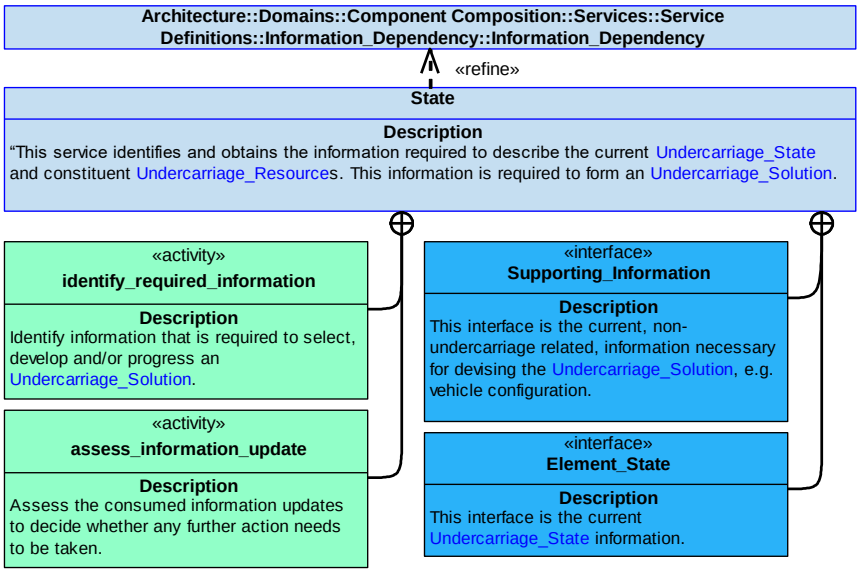

Figure 1202: State Service Policy

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.

B.2.74.7.1.5 Constraint

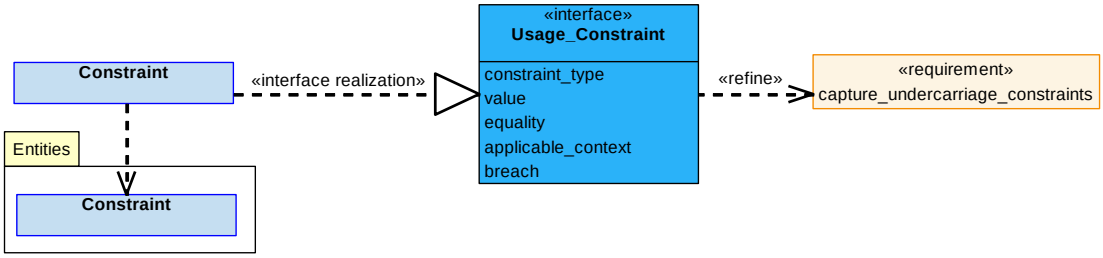

Figure 1203: Constraint Service Definition

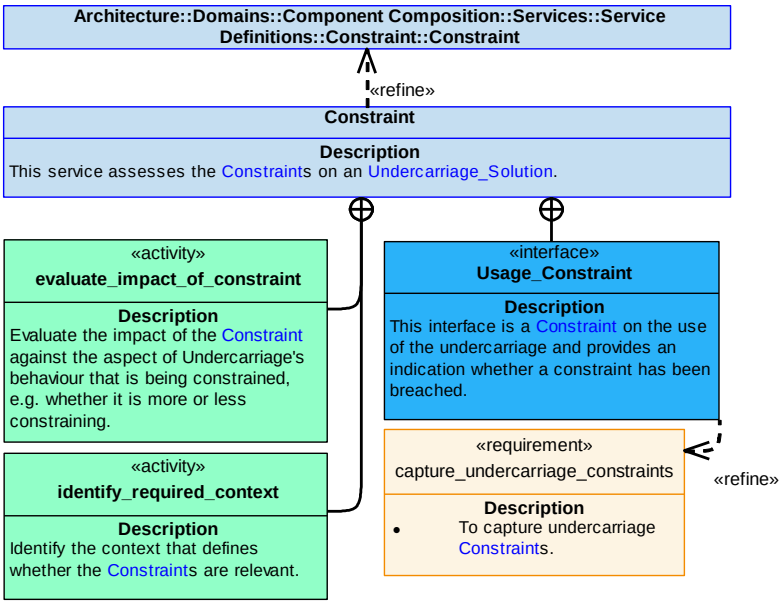

Figure 1204: Constraint Service Policy

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.

B.2.74.7.1.6 Capability

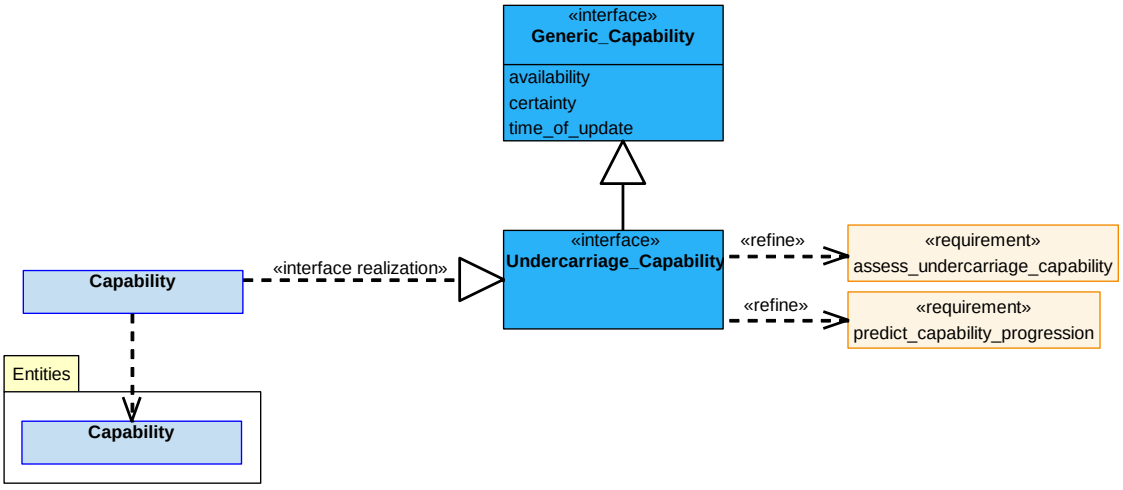

Figure 1205: Capability Service Definition

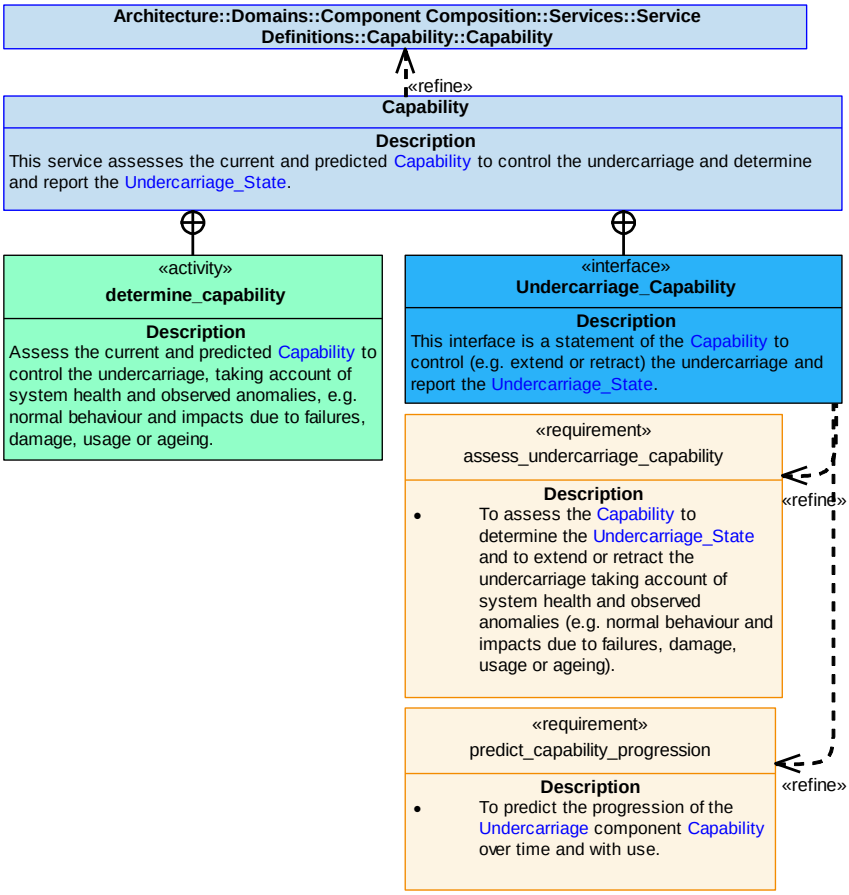

Figure 1206: Capability Policy Diagram

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.

B.2.74.7.1.7 Capability_Evidence

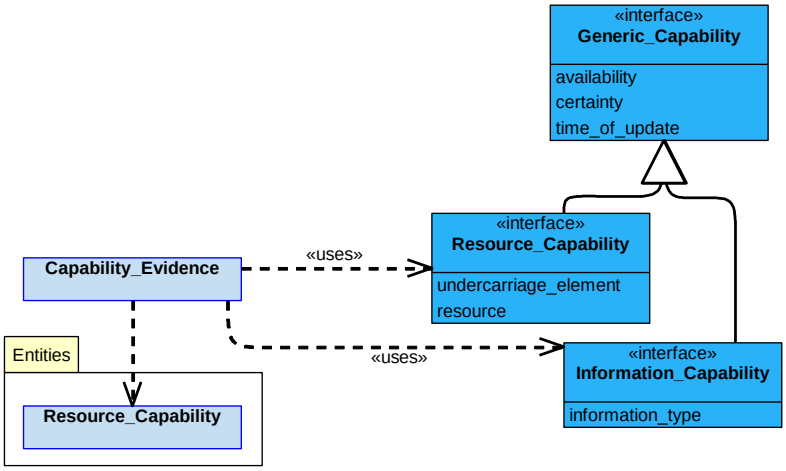

Figure 1207: Capability_Evidence Service Definition

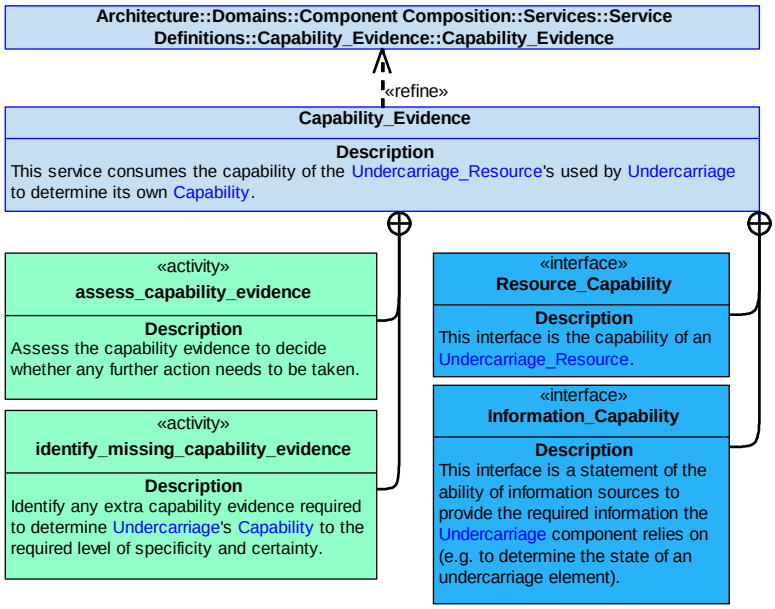

Figure 1208: Capability_Evidence Policy Diagram

Capability_Evidence

This service consumes the capability of the Undercarriage_Resource's used by Undercarriage to determine its own Capability.

Interfaces

Resource_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).

Information_Capability

This interface is a statement of the ability of information sources to provide the required information the Undercarriage component relies on (e.g. to determine the state of an undercarriage element).

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.

B.2.74.7.2 Service Dependencies

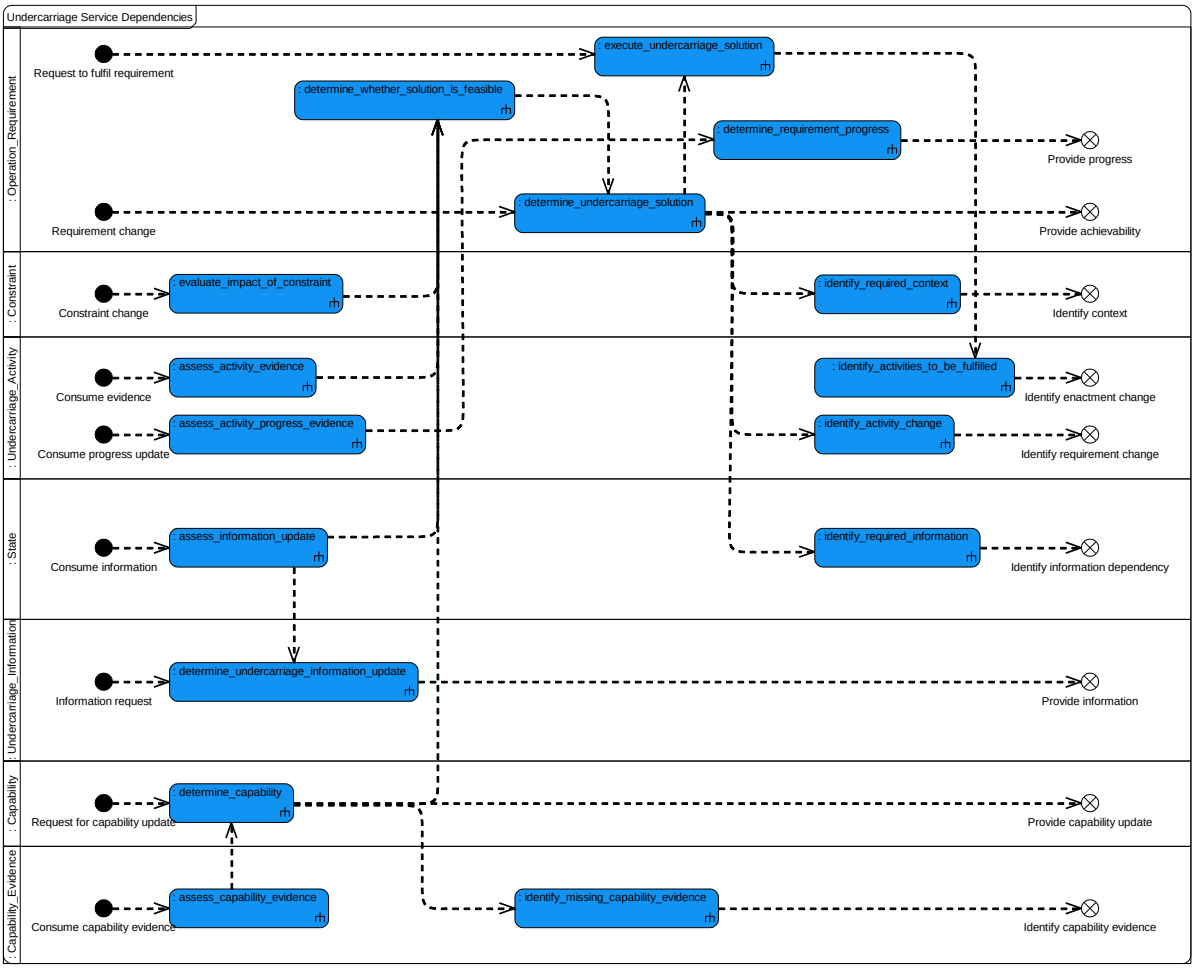

Figure 1209: Undercarriage Service Dependencies

B.2.75 User Accounts

B.2.75.1 Role

The role of User Accounts is to co-ordinate access for users.

B.2.75.2 Overview

Control Architecture

User Accounts is a service component as defined in the Control Architecture policy.

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.

B.2.75.3 Service Summary

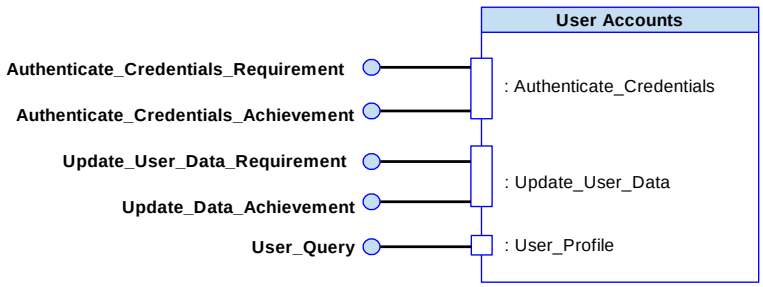

Figure 1210: User Accounts Service Summary

B.2.75.4 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).

B.2.75.5 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.

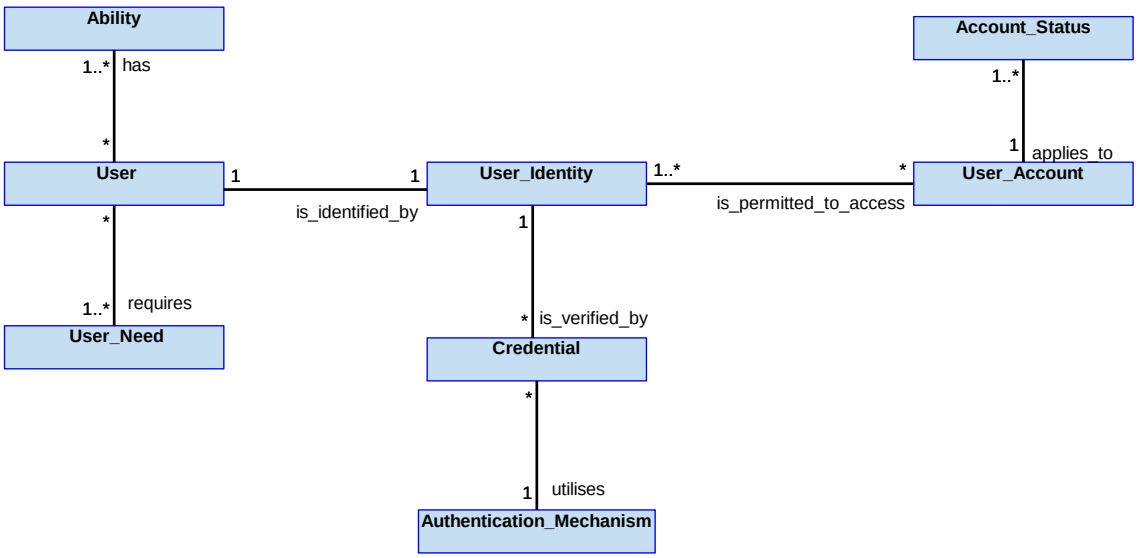

Figure 1211: User Accounts Semantics

B.2.75.5.1 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.

B.2.75.6 Design Rationale

B.2.75.6.1 Assumptions

None.

B.2.75.6.2 Design Considerations

Directly Applicable Policies

These policies 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 policy.
- Recording and Logging - This component will initiate logging of access attempts (successful or failed) in accordance with this policy 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 Assessment policy.

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.

B.2.75.6.3 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.

B.2.75.6.4 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. [22] 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.

B.2.75.7 Services

B.2.75.7.1 Service Definitions

B.2.75.7.1.1 Authenticate_Credentials

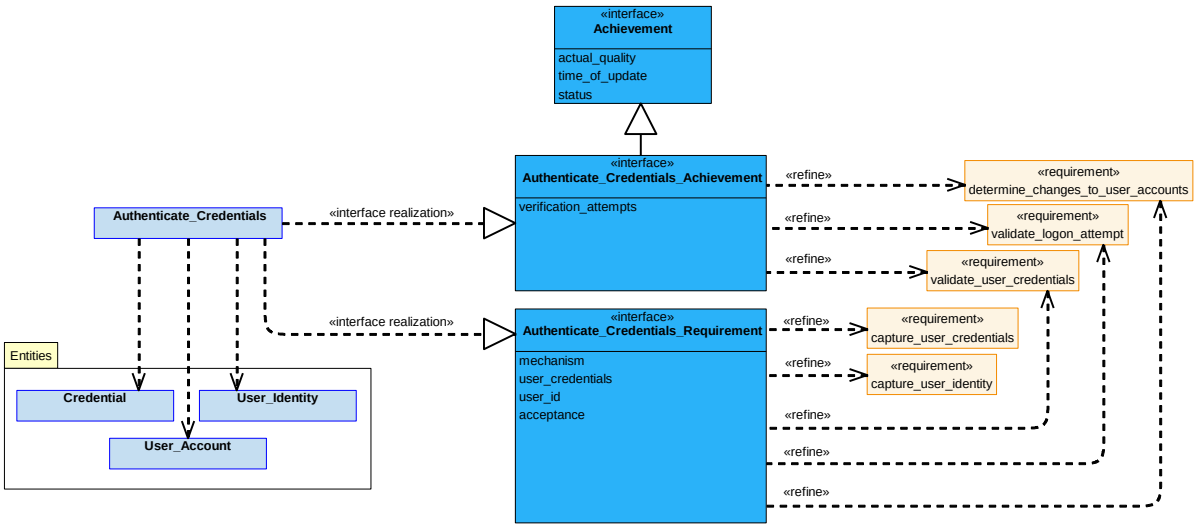

Figure 1212: Authenticate_Credentials Service Definition

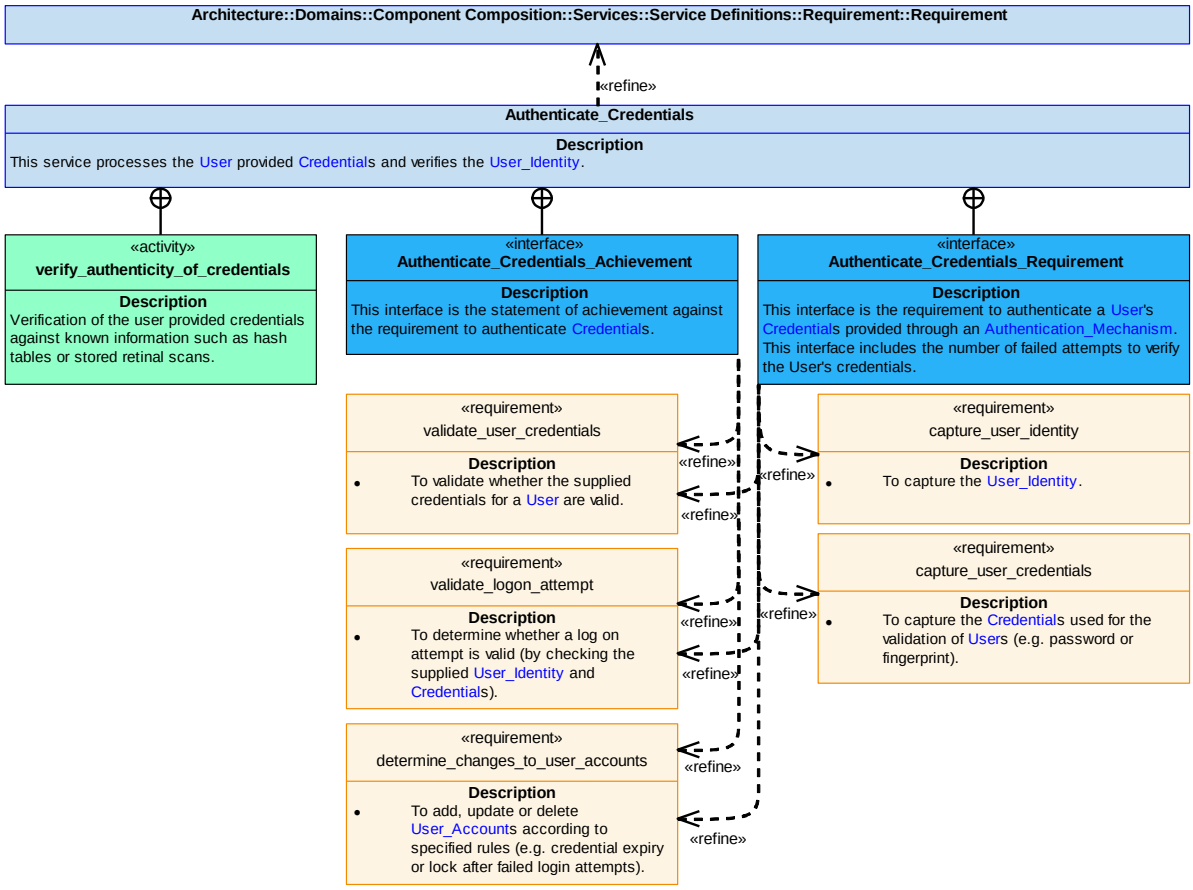

Figure 1213: Authenticate_Credentials Service Policy

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.

B.2.75.7.1.2 Update_User_Data

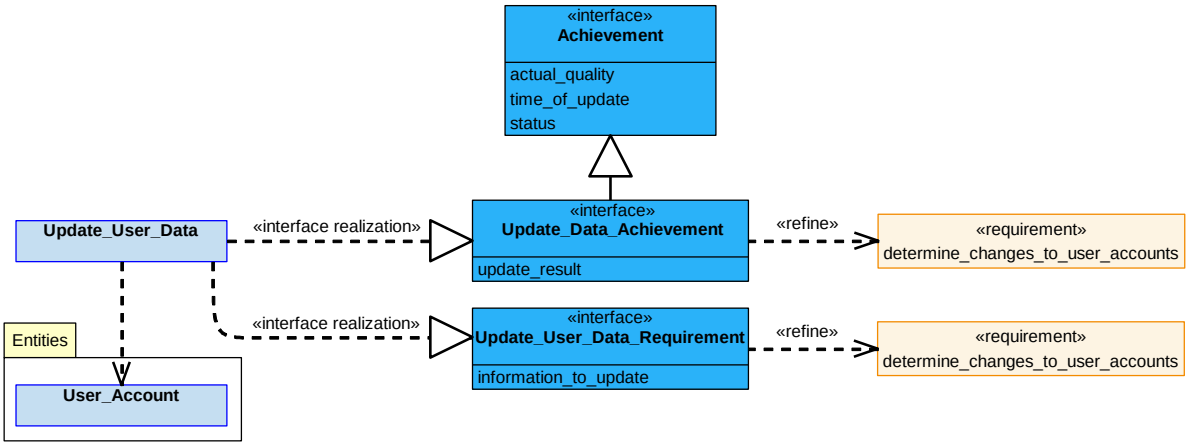

Figure 1214: Update_User_Data Service Definition

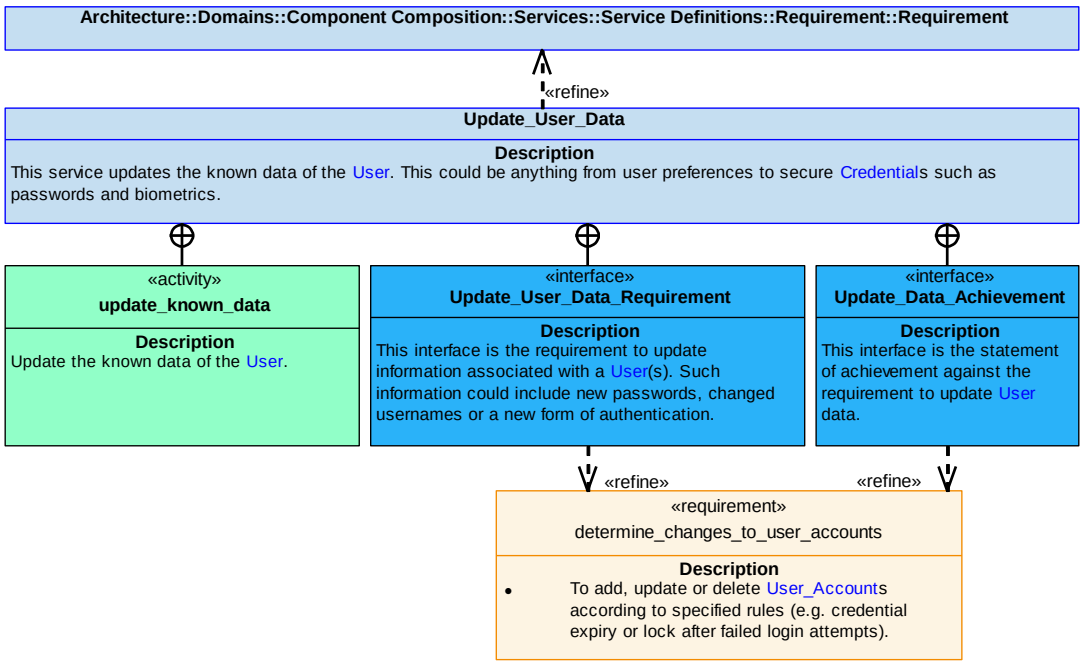

Figure 1215: Update_User_Data Service Policy

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.

B.2.75.7.1.3 User_Profile

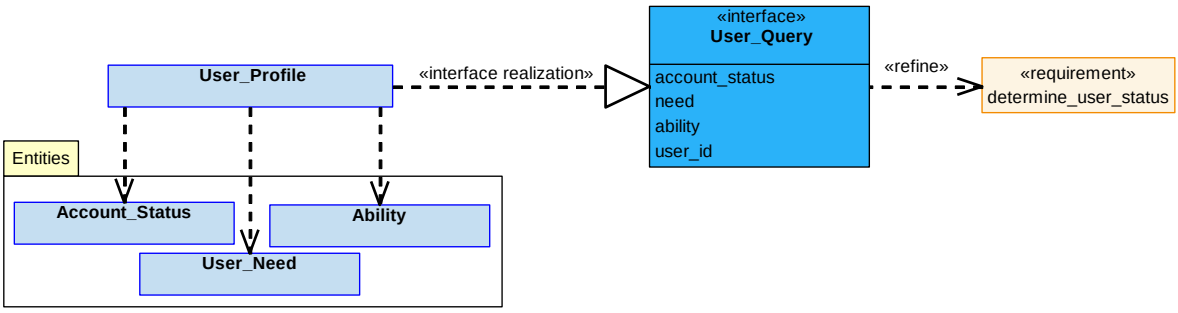

Figure 1216: User_Profile Service Definition

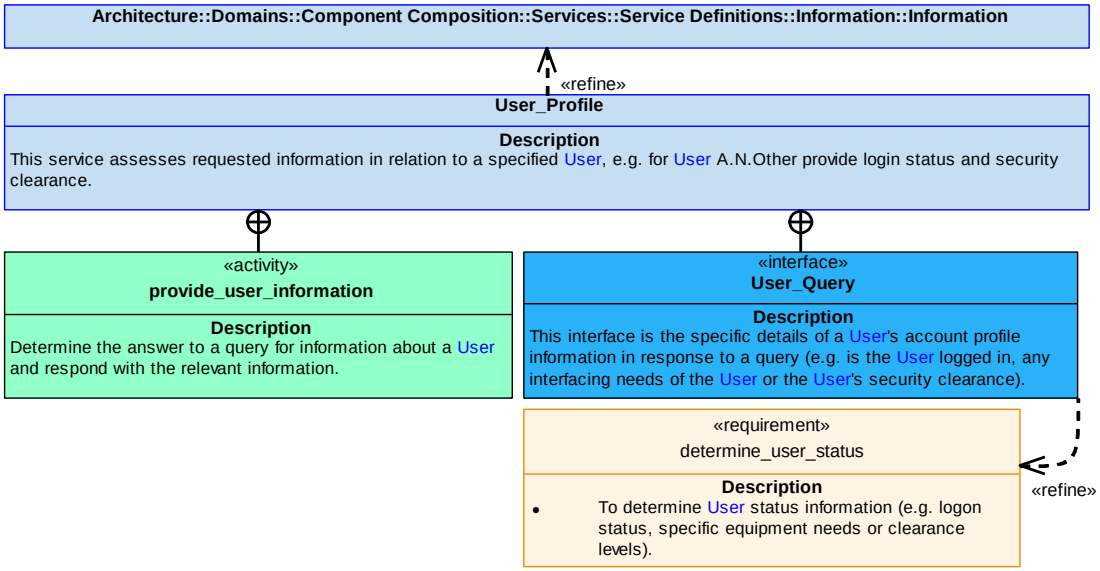

Figure 1217: User_Profile Service Policy

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.

B.2.75.7.2 Service Dependencies

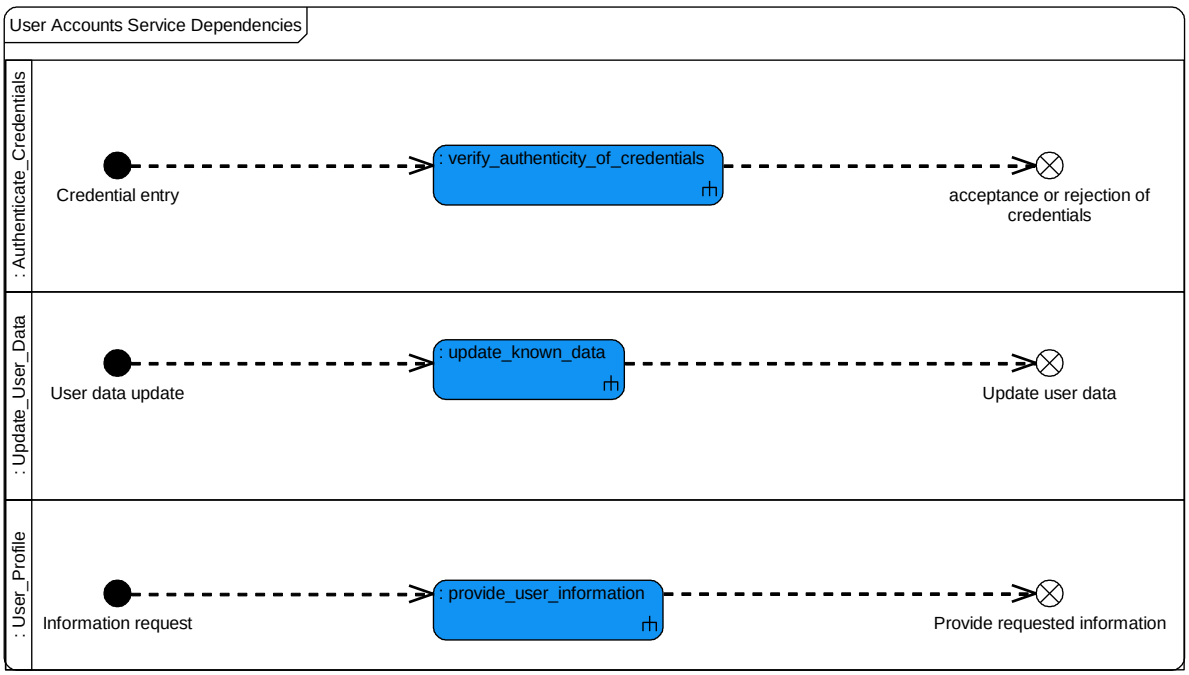

Figure 1218: User Accounts Service Dependencies

B.2.76 User Roles

B.2.76.1 Role

The role of User Roles is to control the allocation of roles to Authorised Operators.

B.2.76.2 Overview

Control Architecture

User Roles is a service component as defined in the Control Architecture policy.

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.

B.2.76.3 Service Summary

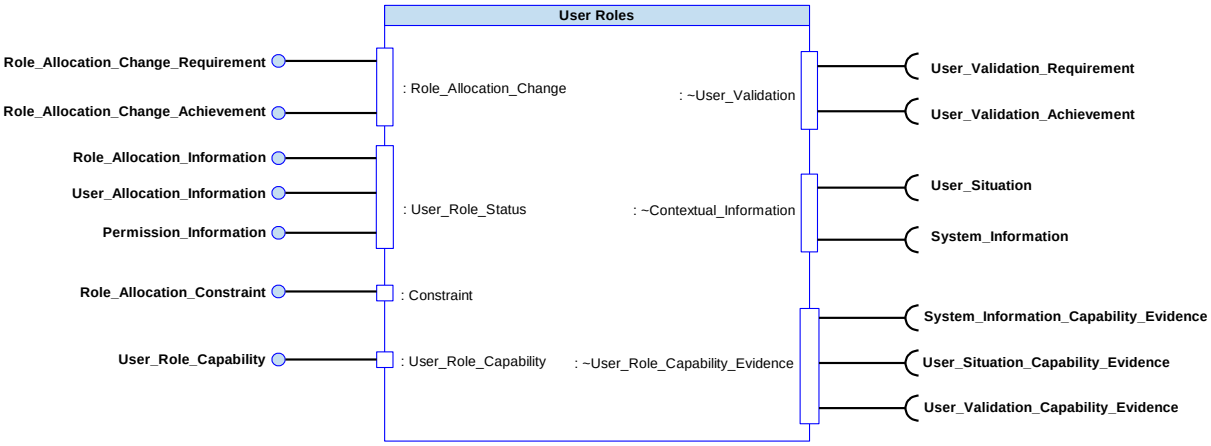

Figure 1219: User Roles Service Summary

B.2.76.4 Responsibilities

allocate_roles

- To allocate, re-allocate and de-allocate Roles to Users.

perform_role_handover

- To perform the handover of Roles between Users.

capture_constraints

- To capture the Constraints on the mapping of a User to a 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.

capture_requirements_for_role_allocation_change

- To capture provided requirements for User to Role allocation, deallocation and handover requests.

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).

identify_missing_information

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

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).

predict_capability_progression

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

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.

B.2.76.5 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.

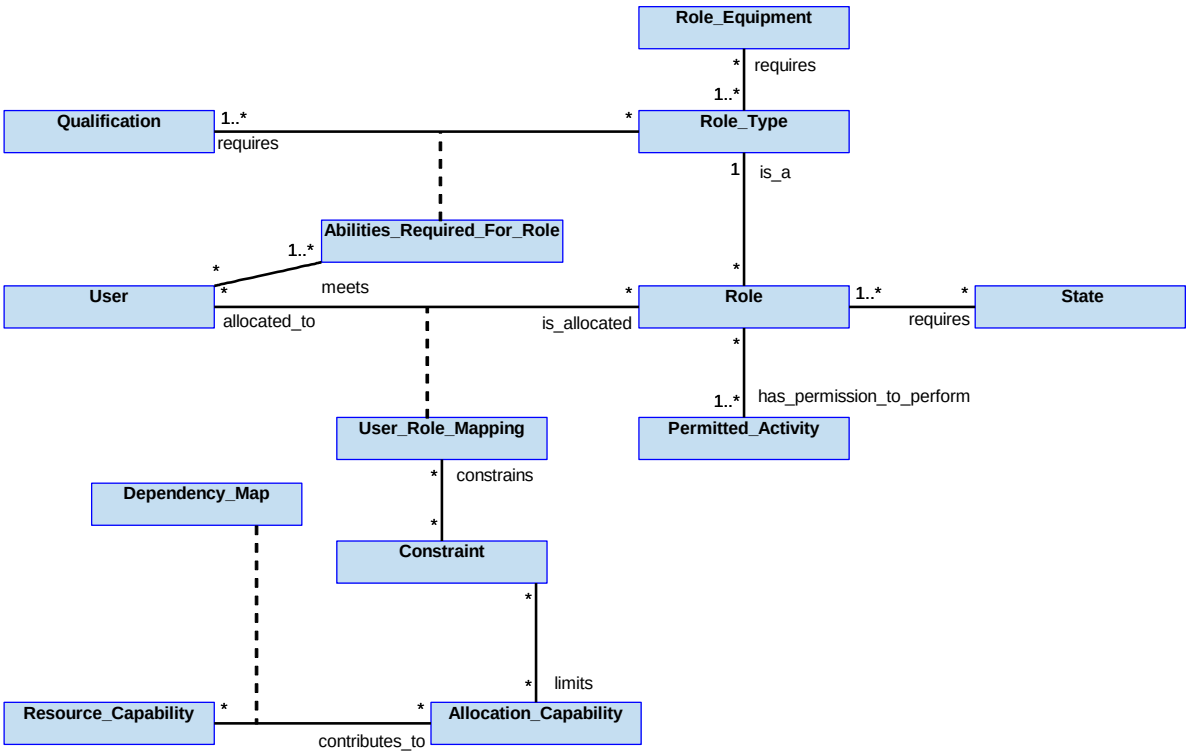

Figure 1220: User Roles Semantics

B.2.76.5.1 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.

B.2.76.6 Design Rationale

B.2.76.6.1 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'.

B.2.76.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

B.2.76.6.3 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 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.

B.2.76.6.4 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 activity 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.

B.2.76.7 Services

B.2.76.7.1 Service Definitions

B.2.76.7.1.1 Role_Allocation_Change

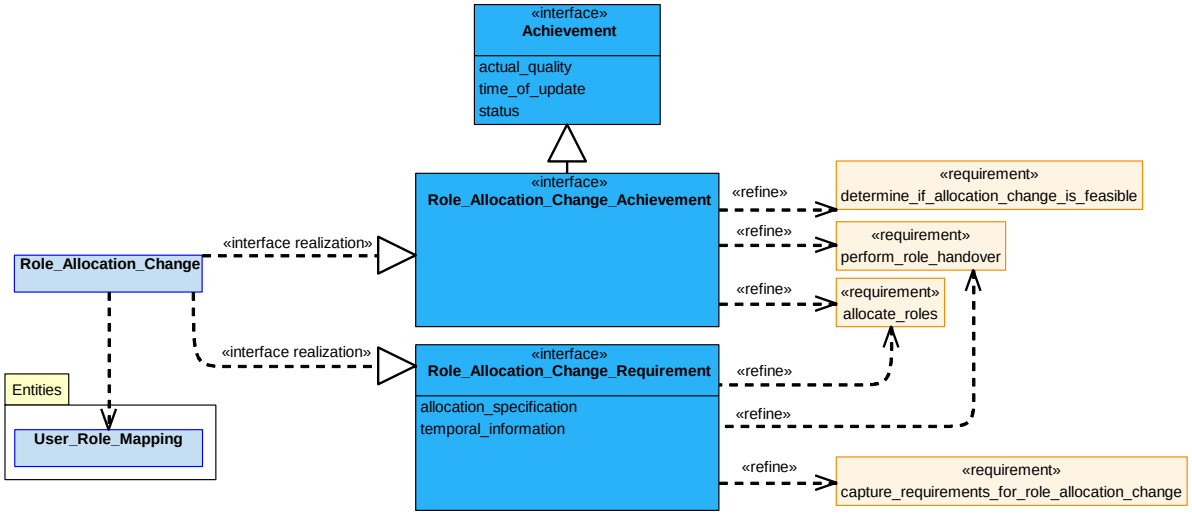

Figure 1221: Role_Allocation_Change Service Definition

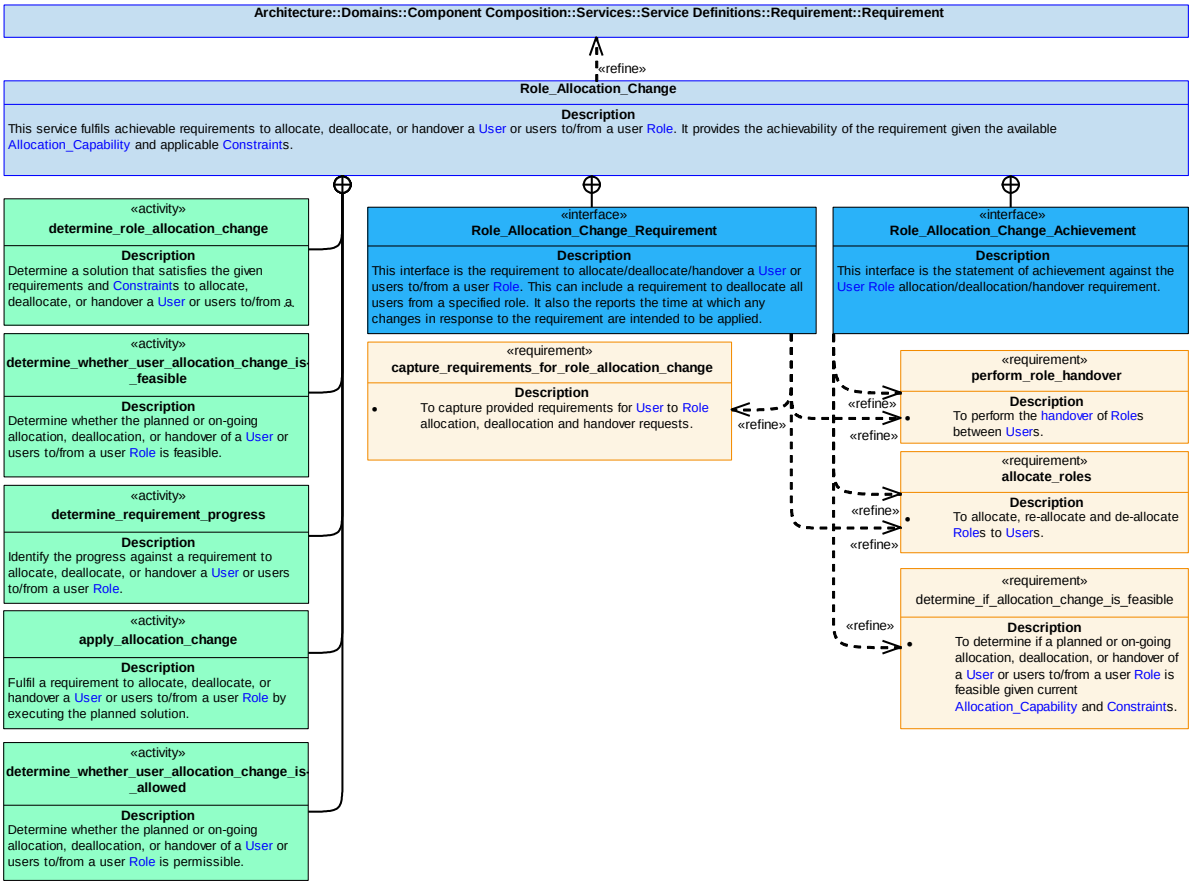

Figure 1222: Role_Allocation_Change Service Policy

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.

B.2.76.7.1.2 User_Validation

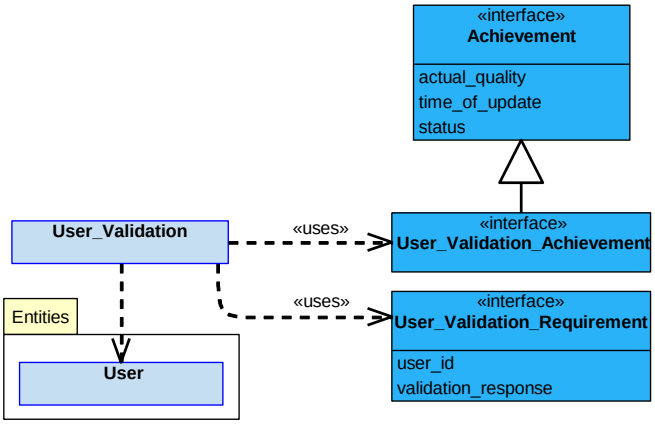

Figure 1223: User_Validation Service Definition

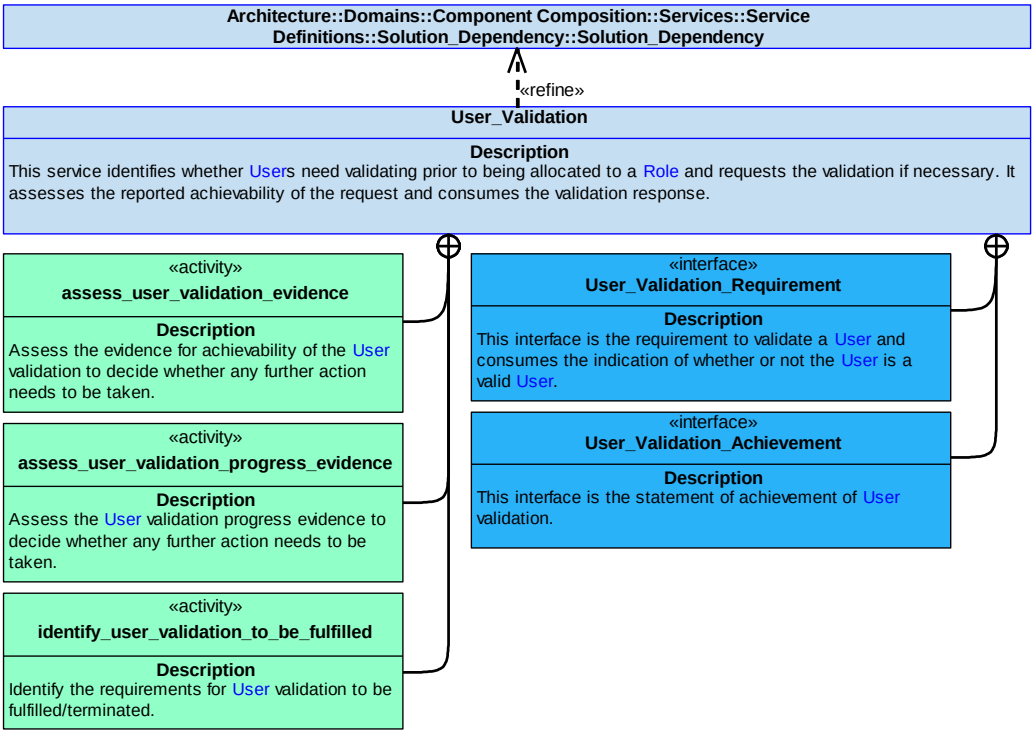

Figure 1224: User_Validation Service Policy

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.

B.2.76.7.1.3 User_Role_Status

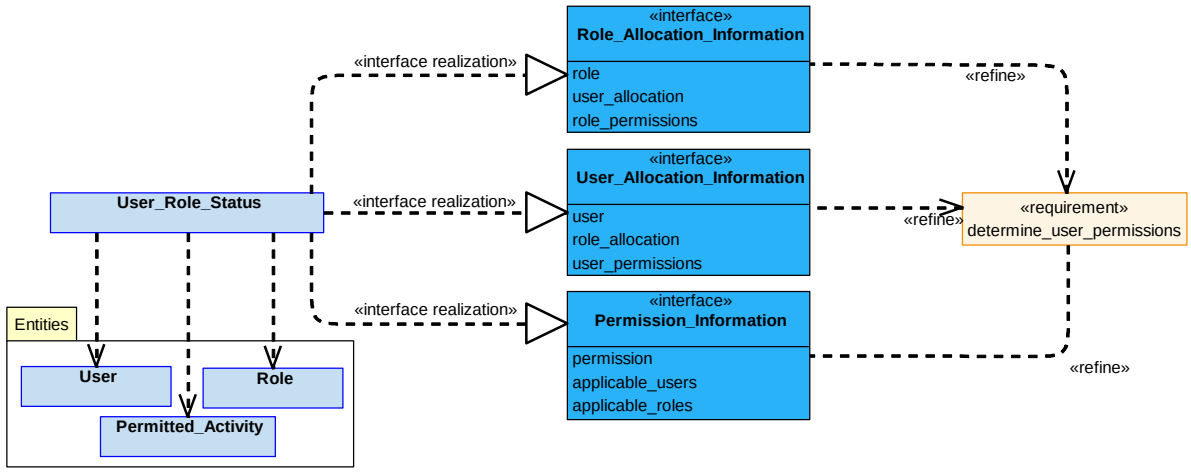

Figure 1225: User_Role_Status Service Definition

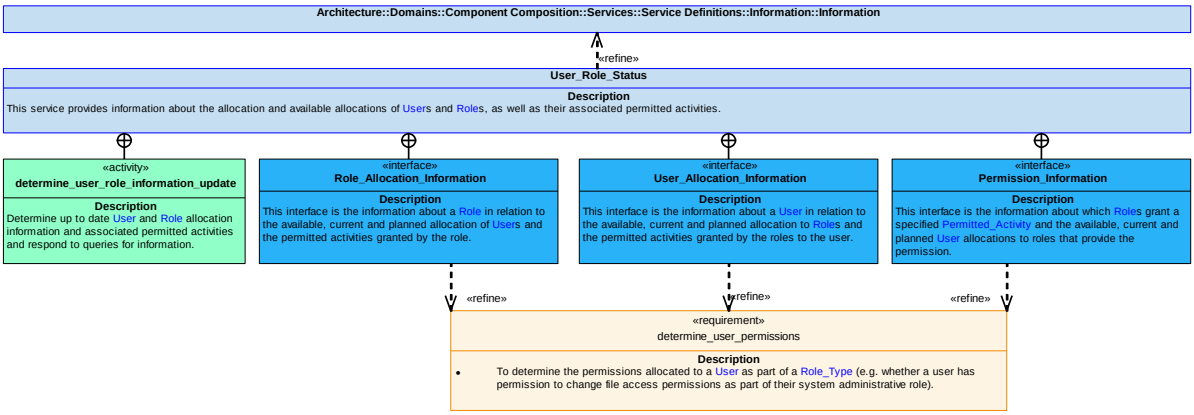

Figure 1226: User_Role_Status Service Policy

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.

B.2.76.7.1.4 Contextual_Information

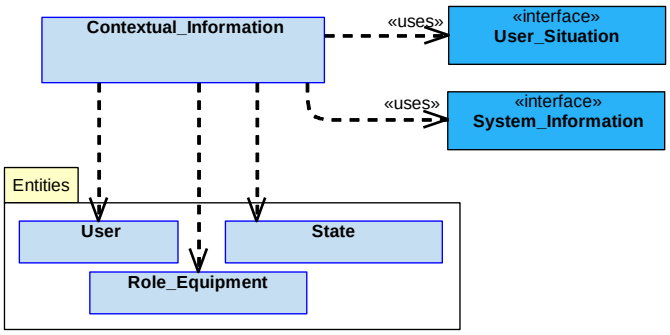

Figure 1227: Contextual_Information Service Definition

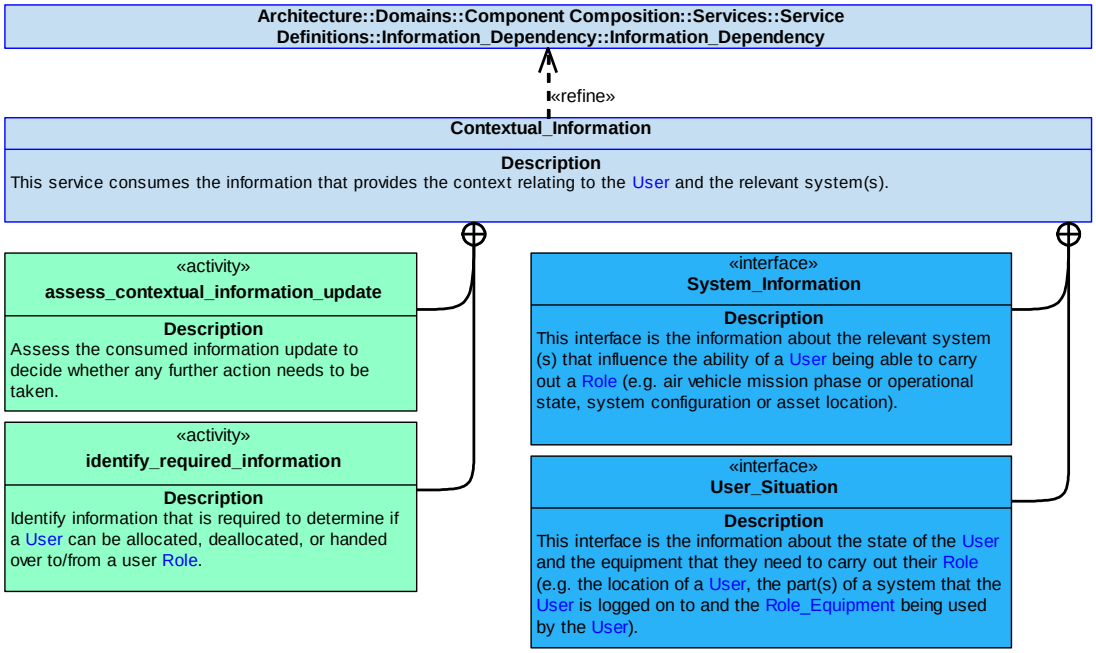

Figure 1228: Contextual_Information Service Policy

Contextual_Information

This service consumes the information that provides the context relating to the User and the relevant system(s).

Interfaces

System_Information

This interface is the information about the relevant system(s) that influence the ability of a User being able to carry out a Role (e.g. air vehicle mission phase or operational state, system configuration or asset location).

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_contextual_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 determine if a User can be allocated, deallocated, or handed over to/from a user Role.

B.2.76.7.1.5 Constraint

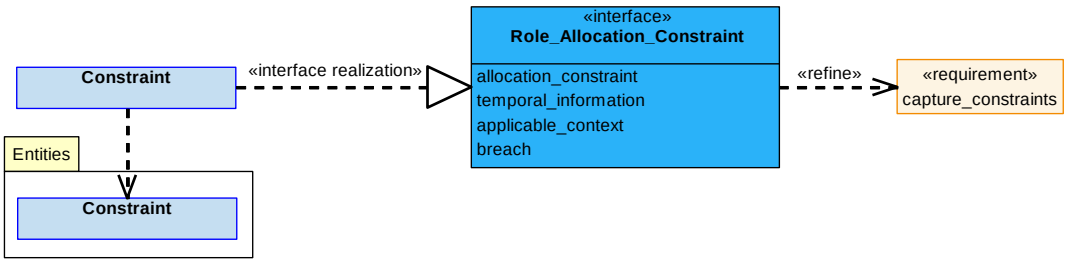

Figure 1229: Constraint Service Definition

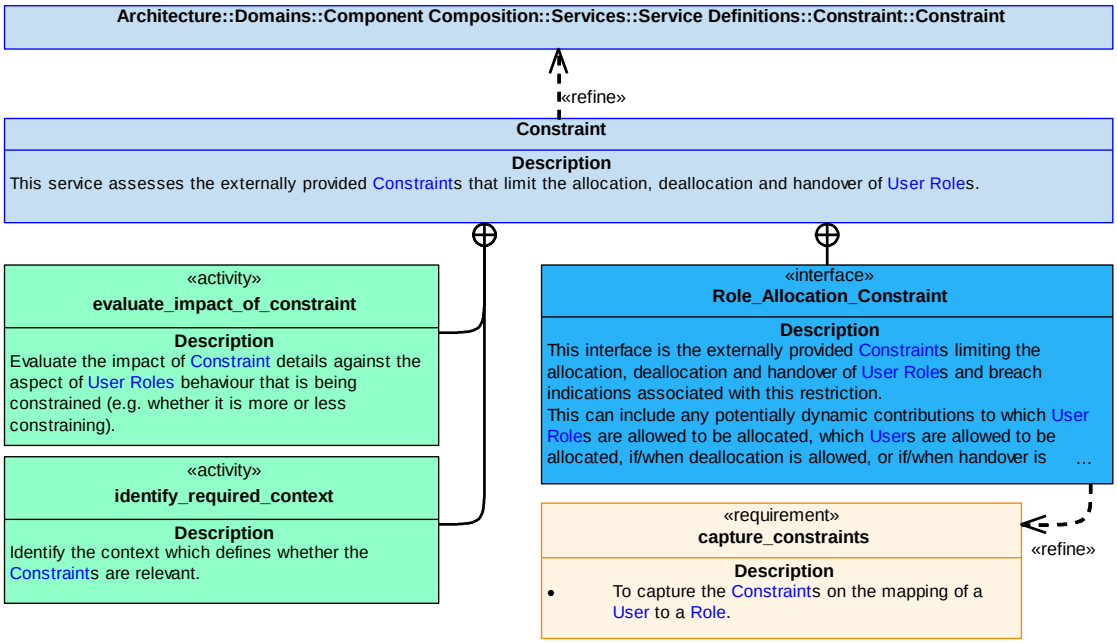

Figure 1230: Constraint Service Policy

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.
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 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.

B.2.76.7.1.6 User_Role_Capability

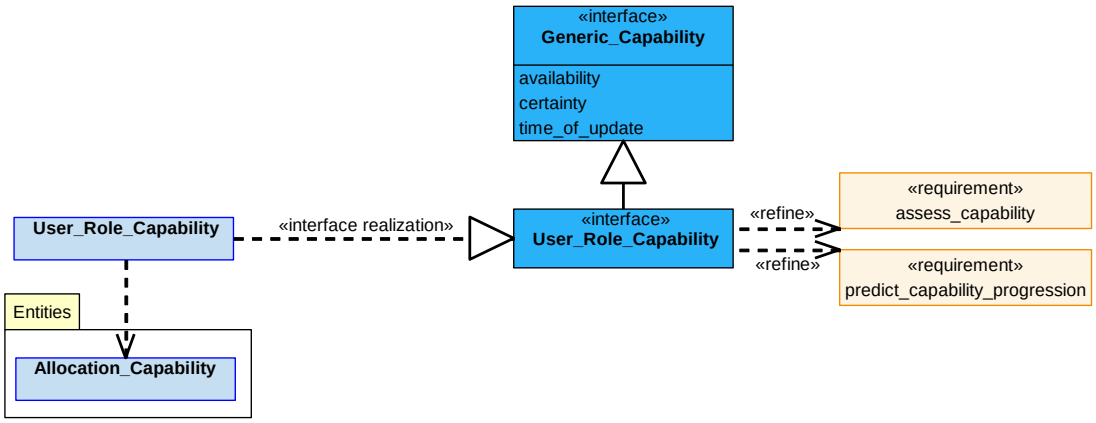

Figure 1231: User_Role_Capability Service Definition

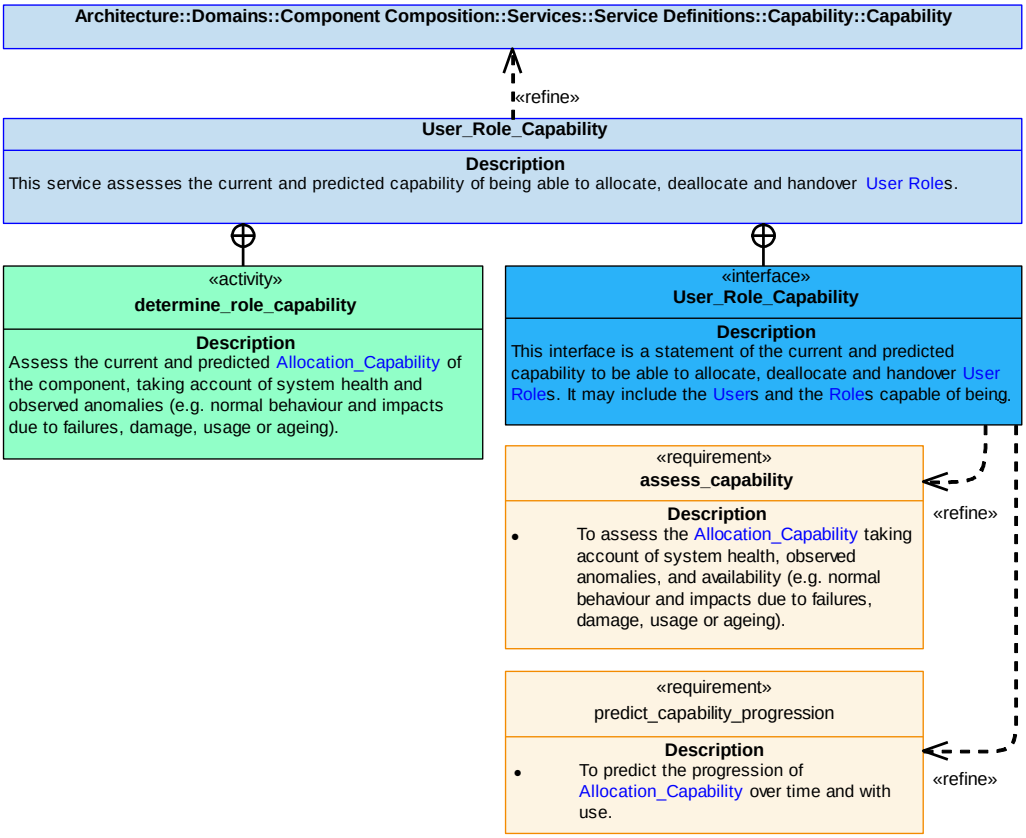

Figure 1232: User_Role_Capability Service Policy

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).

B.2.76.7.1.7 User_Role_Capability_Evidence

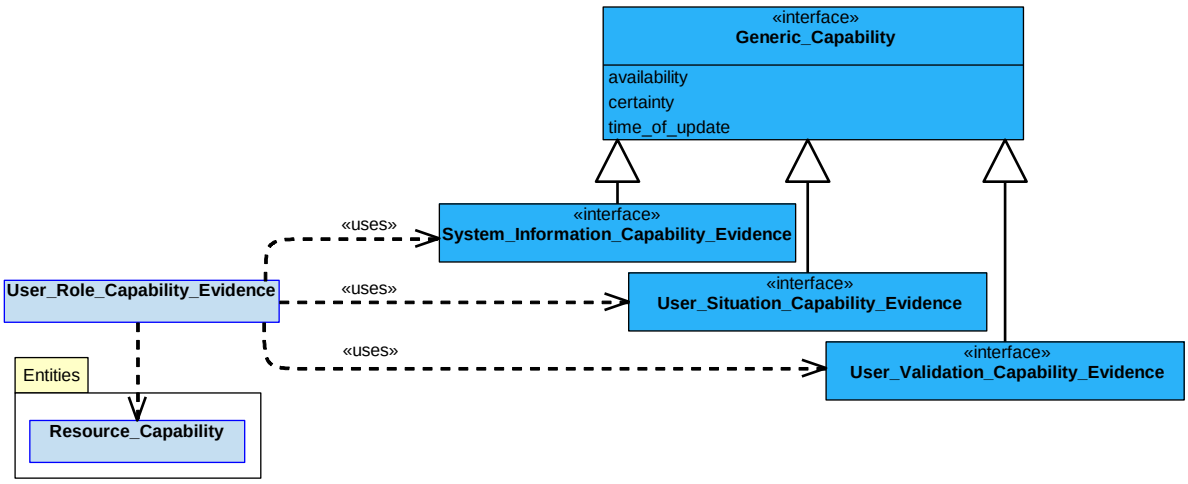

Figure 1233: User_Role_Capability_Evidence Service Definition

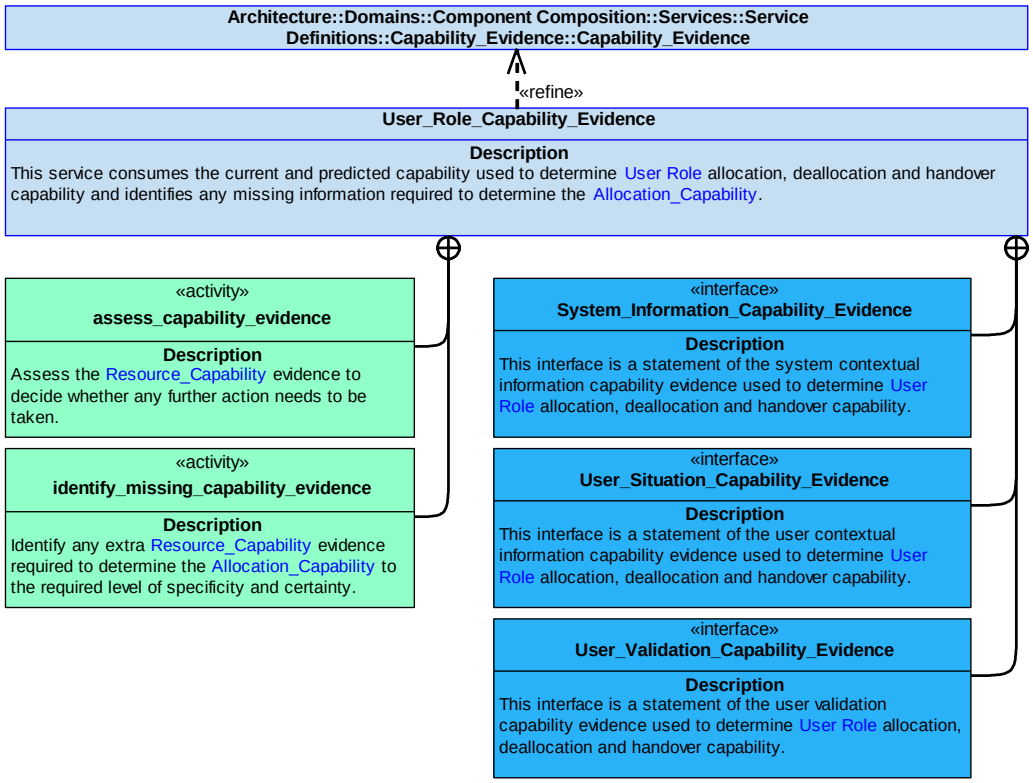

Figure 1234: User_Role_Capability_Evidence Service Policy

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.

B.2.76.7.2 Service Dependencies

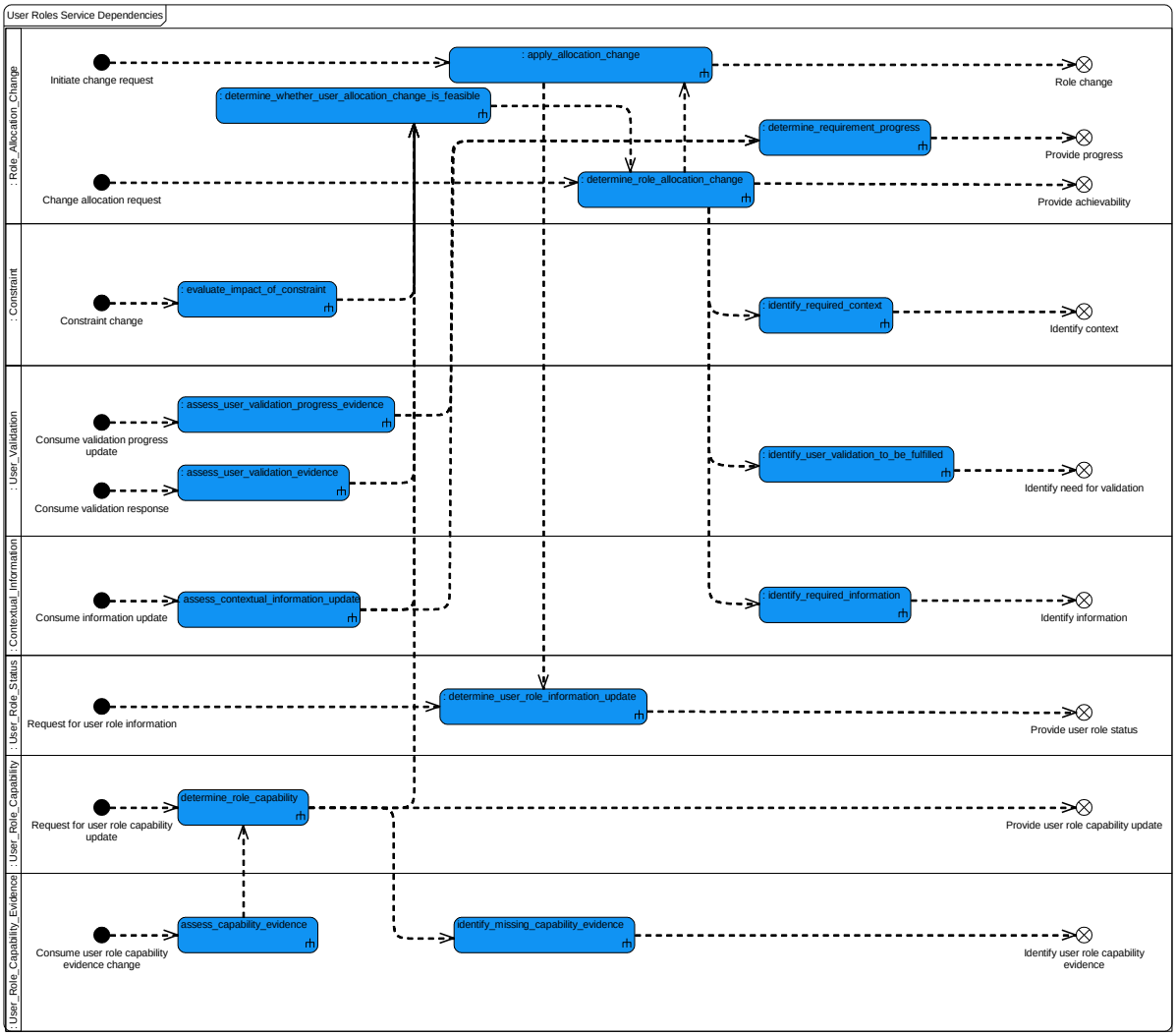

Figure 1235: User Roles Service Dependencies

B.2.77 Vehicle External Environment

B.2.77.1 Role

The role of Vehicle External Environment is to determine information about the immediate environment surrounding a vehicle.

B.2.77.2 Overview

Control Architecture

Vehicle External Environment is a service component as defined in the Control Architecture policy.

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.

B.2.77.3 Service Summary

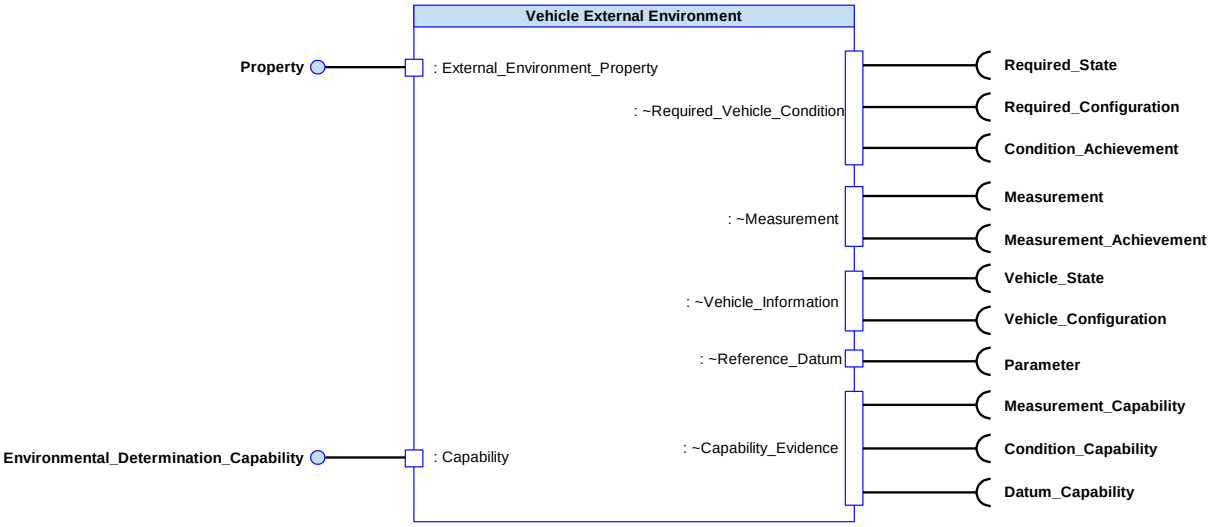

Figure 1236: Vehicle External Environment Service Summary

B.2.77.4 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).

capture_reference_datum

- To capture appropriate Reference_Datum/data from which environmental relationships should be calculated.

determine_required_correction_factors

- To apply Correction_Factors to received sensor data.

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.

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.

B.2.77.5 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.

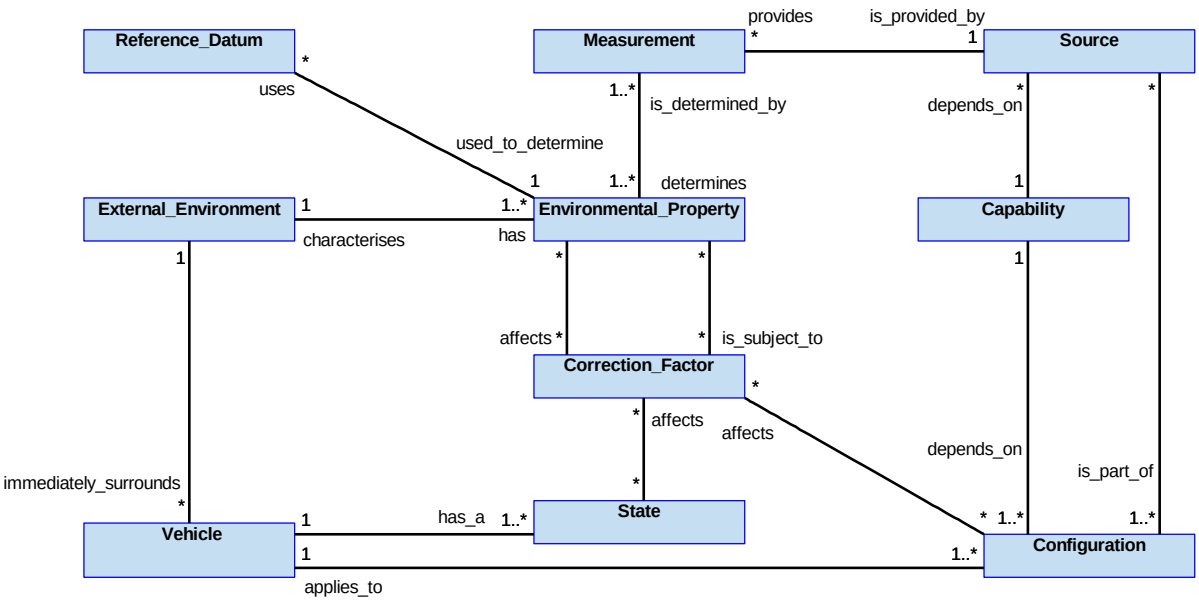

Figure 1237: Vehicle External Environment Semantics

B.2.77.5.1 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.

B.2.77.6 Design Rationale

B.2.77.6.1 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.

B.2.77.6.2 Design Considerations

Directly Applicable Policies

These policies 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 deployment 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 deployment 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 policy.

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.

B.2.77.6.3 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.

B.2.77.6.4 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.

B.2.77.7 Services

B.2.77.7.1 Service Definitions

B.2.77.7.1.1 External_Environment_Property

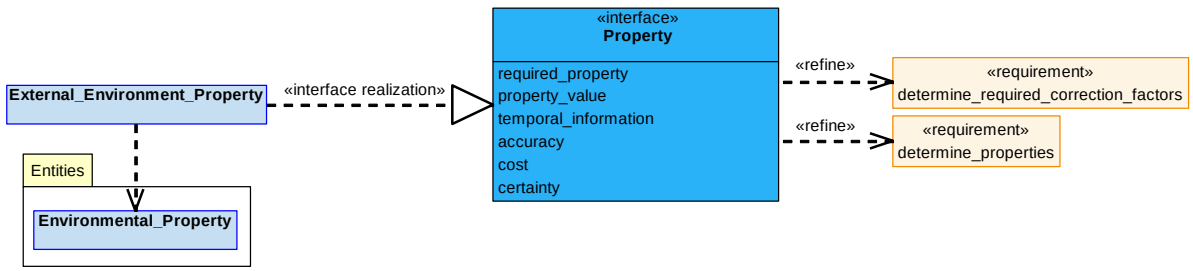

Figure 1238: External_Environment_Property Service Definition

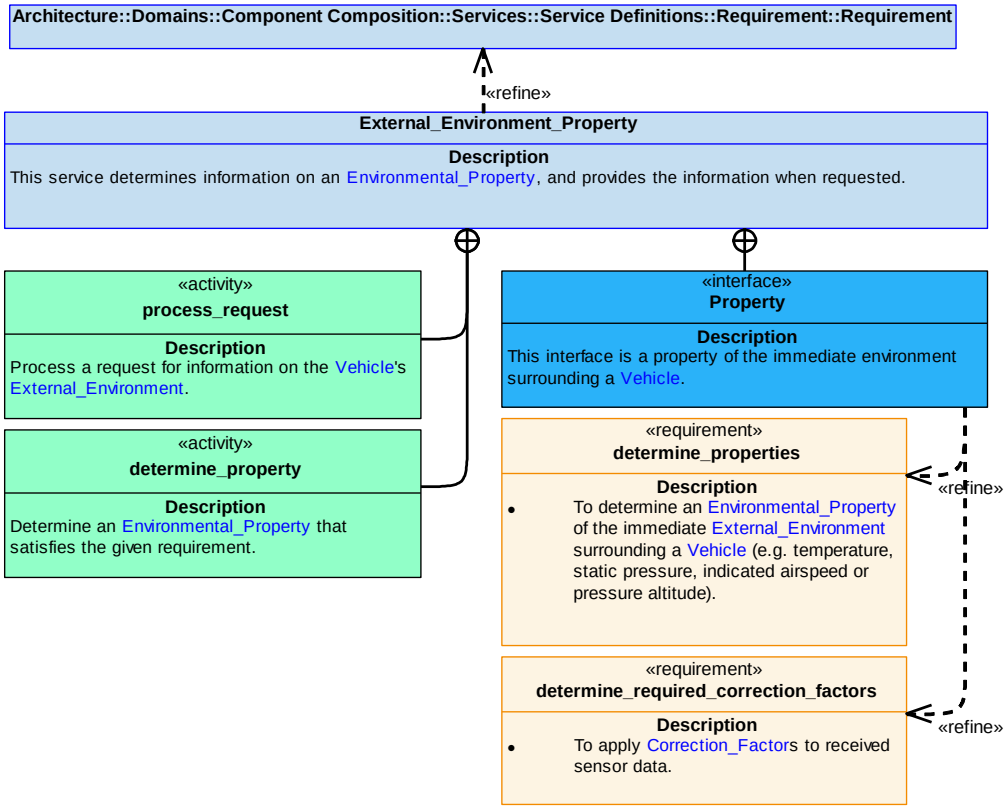

Figure 1239: External_Environment_Property Service Policy

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.

B.2.77.7.1.2 Required_Vehicle_Condition

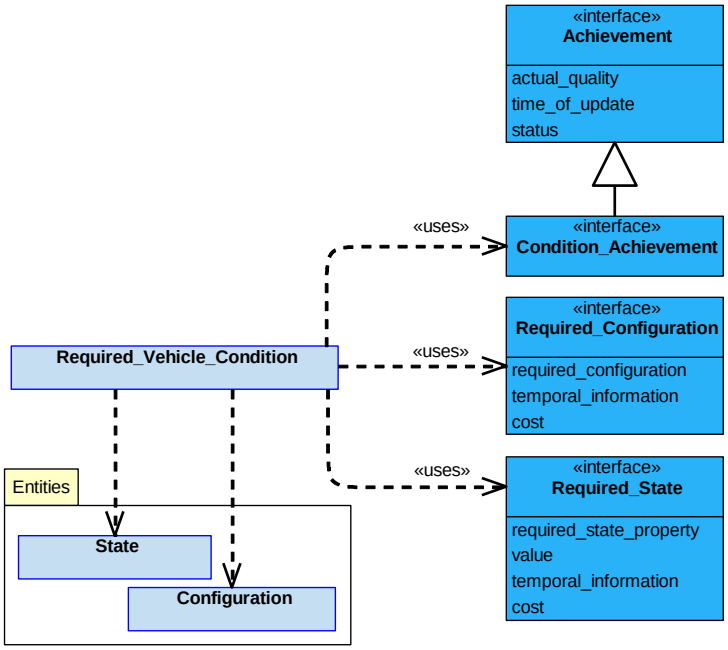

Figure 1240: Required_Vehicle_Condition Service Definition

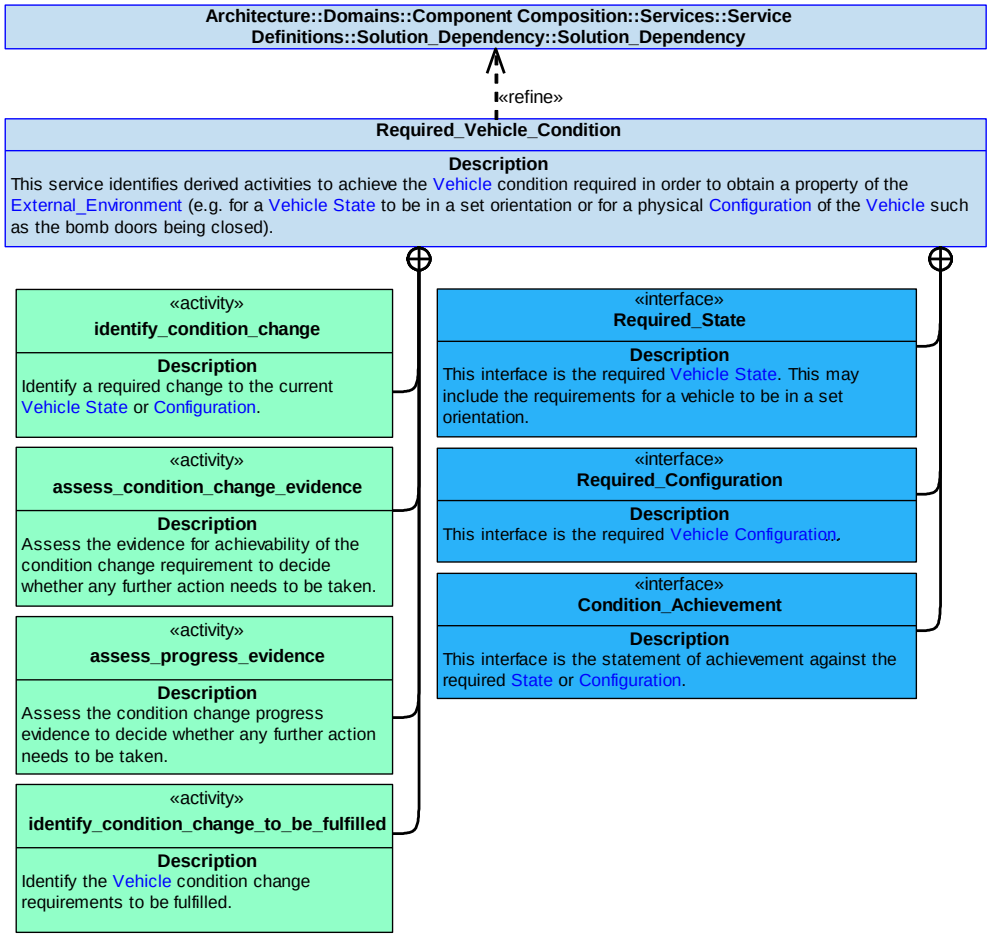

Figure 1241: Required_Vehicle_Condition Service Policy

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.

B.2.77.7.1.3 Measurement

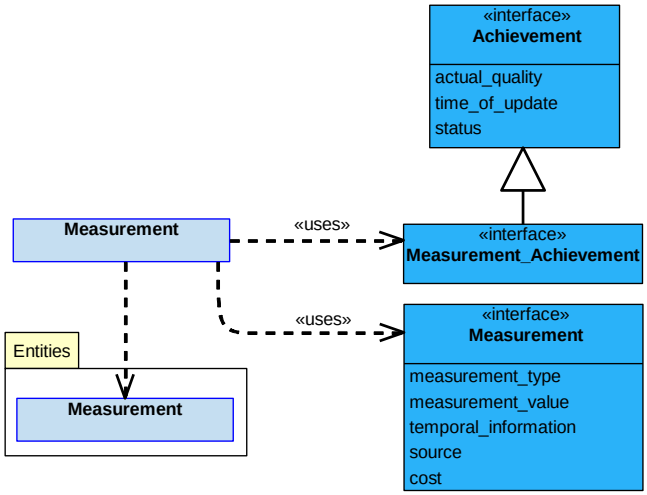

Figure 1242: Measurement Service Definition

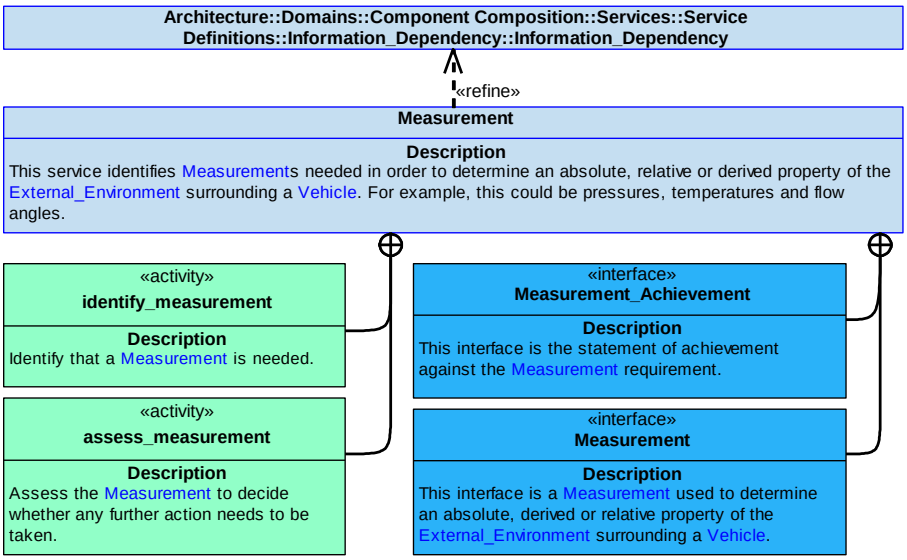

Figure 1243: Measurement Service Policy

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.

B.2.77.7.1.4 Vehicle_Information

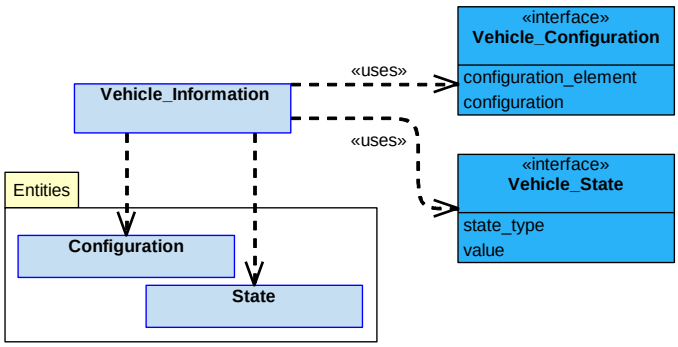

Figure 1244: Vehicle_Information Service Definition

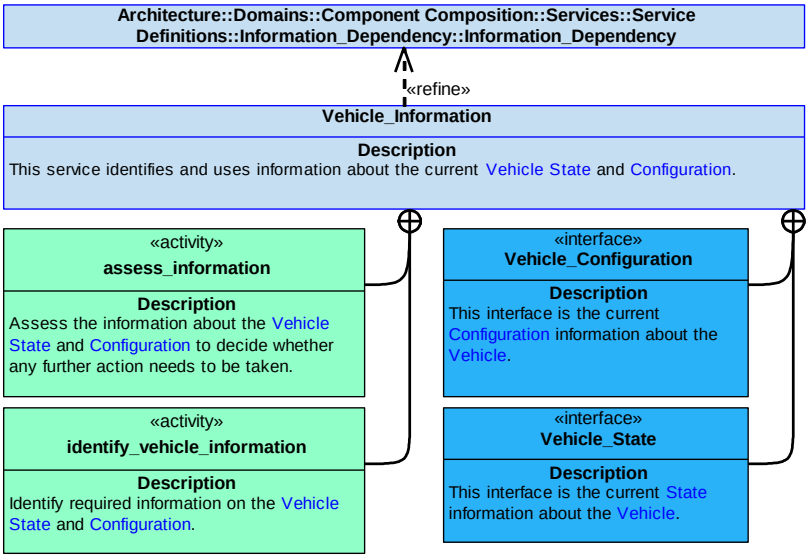

Figure 1245: Vehicle_Information Service Policy

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.

B.2.77.7.1.5 Reference_Datum

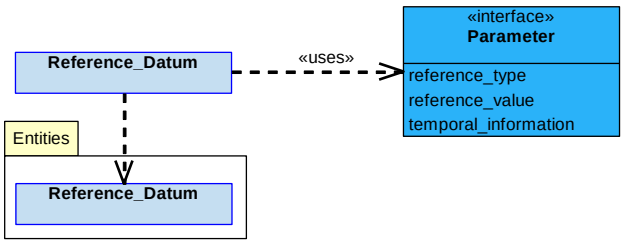

Figure 1246: Reference_Datum Service Definition

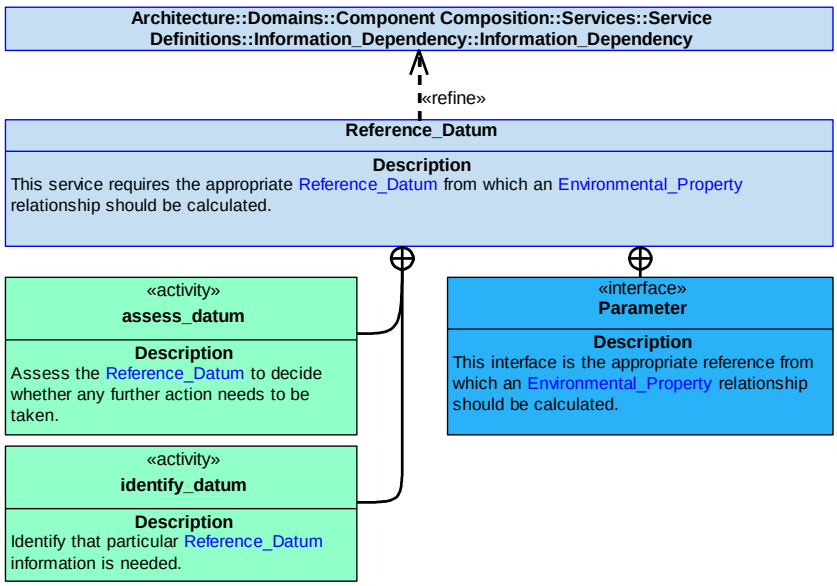

Figure 1247: Reference_Datum Service Policy

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.

B.2.77.7.1.6 Capability

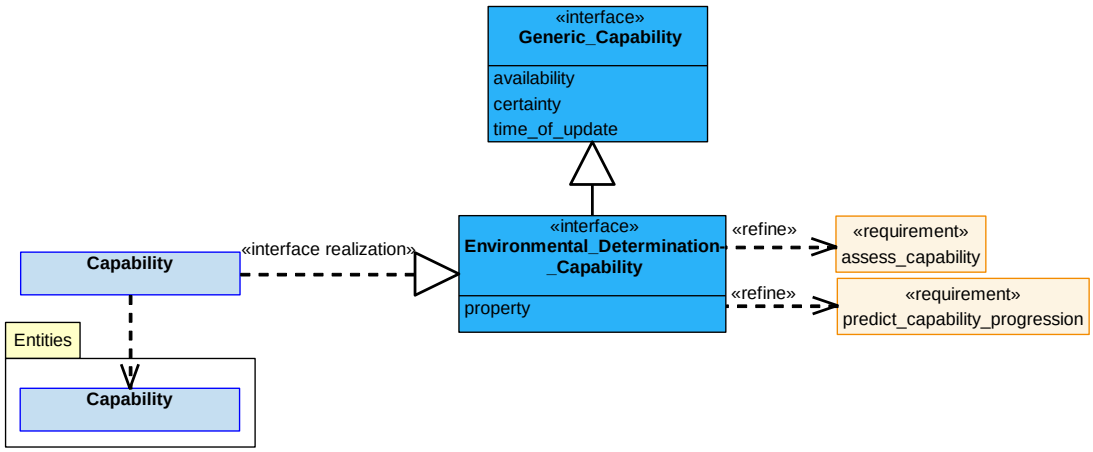

Figure 1248: Capability Service Definition

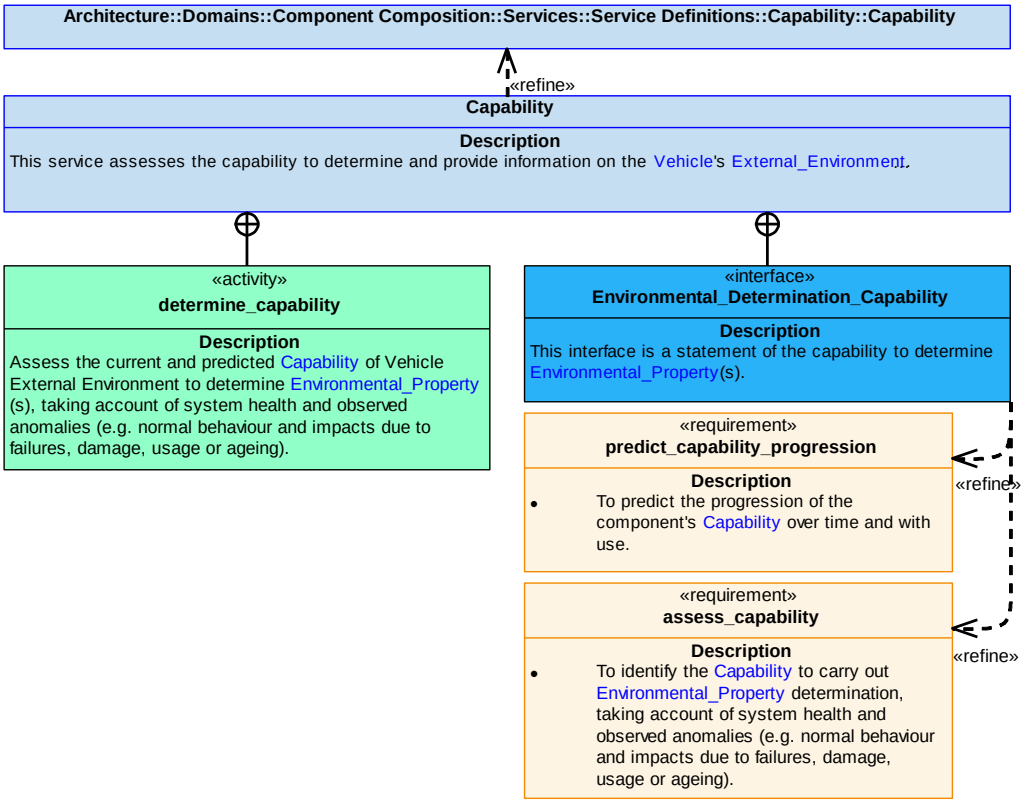

Figure 1249: Capability Service Policy

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 Environmental_Property(s).

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 Environmental_Property(s), taking account of system health and observed anomalies (e.g. normal behaviour and impacts due to failures, damage, usage or ageing).

B.2.77.7.1.7 Capability_Evidence

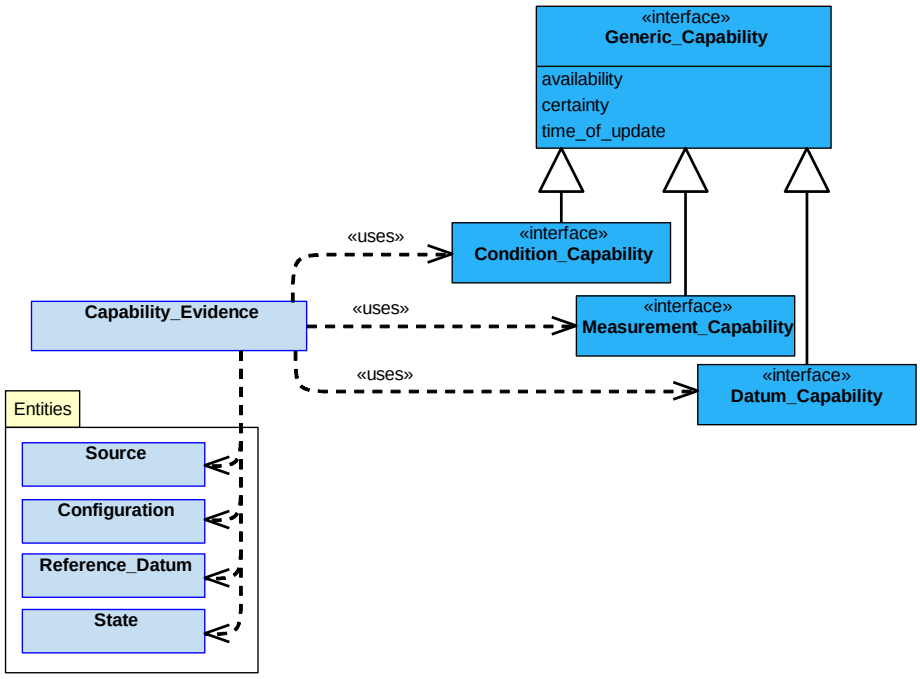

Figure 1250: Capability_Evidence Service Definition

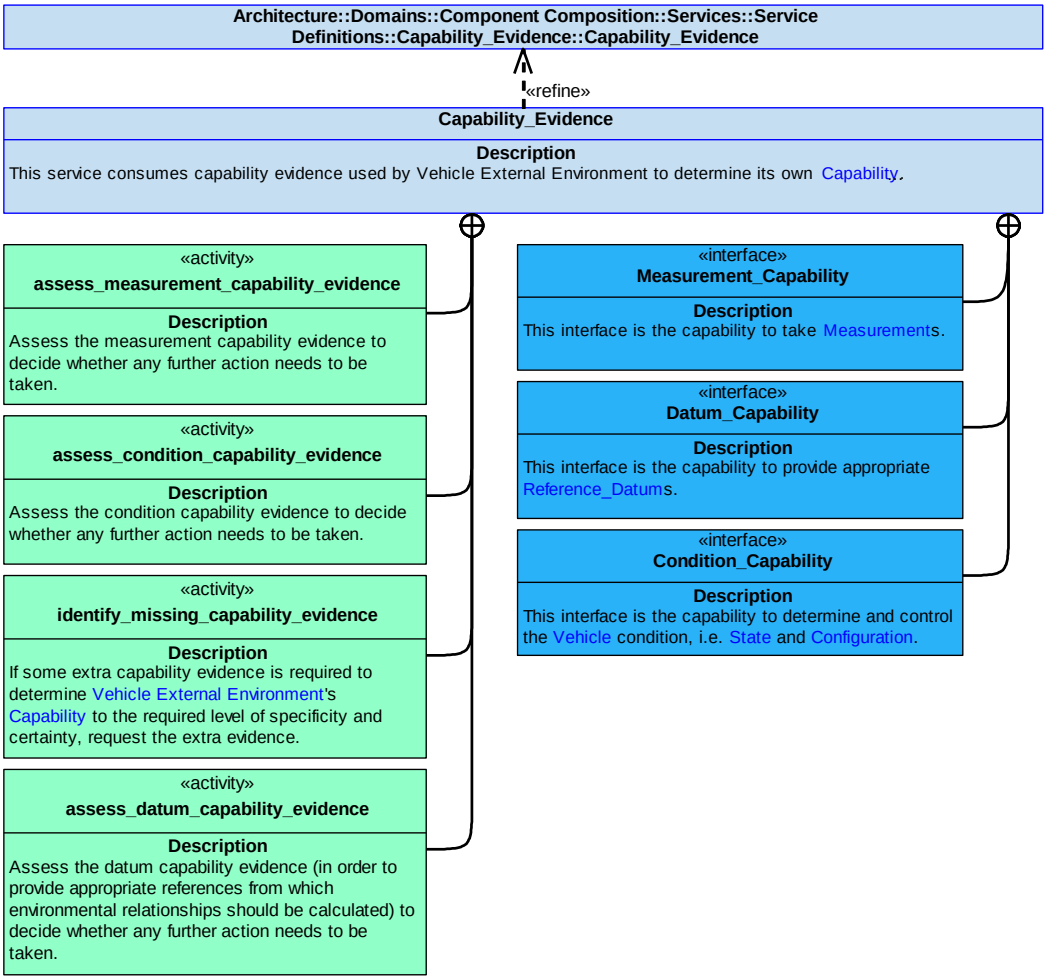

Figure 1251: Capability_Evidence Service Policy

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.

Activities

assess_measurement_capability_evidence

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

assess_condition_capability_evidence

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

assess_datum_capability_evidence

Assess the datum capability evidence (in order to provide appropriate references from which environmental relationships should be calculated) 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.

B.2.77.7.2 Service Dependencies

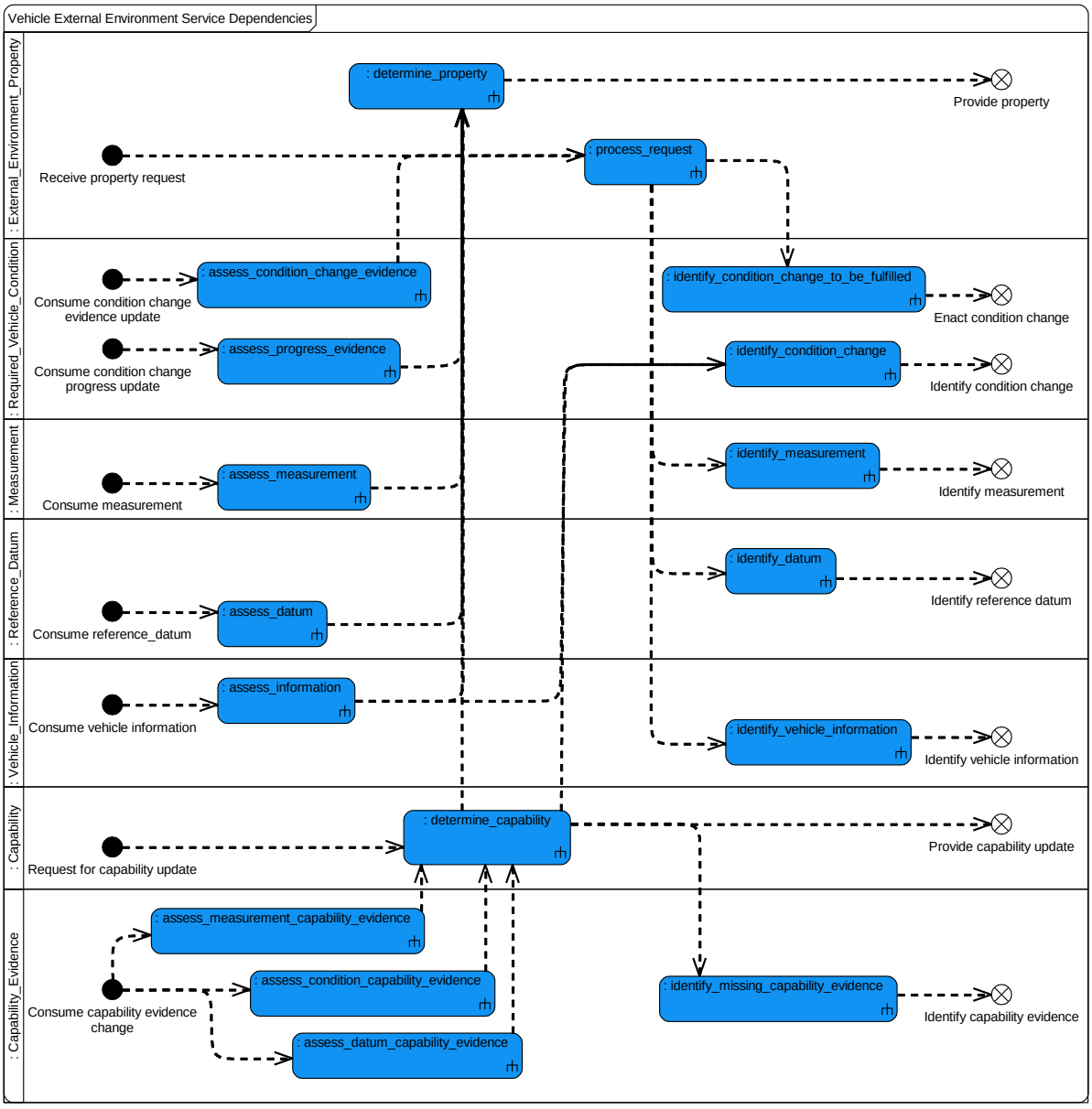

Figure 1252: Vehicle External Environment Service Dependencies

B.2.78 Vehicle Guidance

B.2.78.1 Role

The role of Vehicle Guidance is to determine and execute a vehicle trajectory to fulfil provided trajectory requirements.

B.2.78.2 Overview

Control Architecture

Vehicle Guidance is an action component as defined in the Control Architecture policy.

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. 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 to meet the path in order to correct the movement of the vehicle, if 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.

B.2.78.3 Service Summary

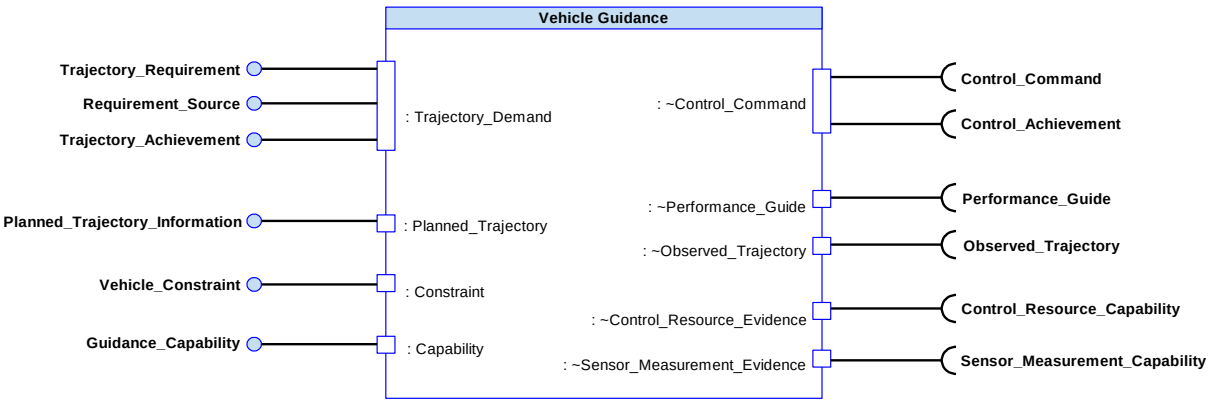

Figure 1253: Vehicle Guidance Service Summary

B.2.78.4 Responsibilities

capture_trajectory_requirements

- To capture given Trajectory_Requirements.

capture_trajectory_control_source

- To capture the Demand_Source of the trajectory currently being executed (i.e. path demand or direct control demand).

capture_vehicle_motion_constraints

- To capture given Vehicle motion Vehicle_Constraints, e.g. bank angle constraint.

determine_capability

- To determine the Guidance_Capability of the Vehicle, taking into account observed anomalies.

predict_vehicle_guidance_capability

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

determine_planned_vehicle_trajectory

- To determine a Planned_Trajectory to achieve Trajectory_Requirements.

issue_vehicle_control_commands

- To execute the selected Planned_Trajectory by issuing Motion_Commands (e.g. attitude and speed or thrust level).

identify_vehicle_trajectory_divergence

- To identify Trajectory_Divergence of the Vehicle.

identify_whether_requirement_remains_achievable

- To identify whether a Trajectory_Requirement is still achievable given current or predicted Guidance_Capability, Performance_Guides and Vehicle_Constraints.

determine_predicted_quality_of_deliverables

- To determine the predicted quality of the Planned_Trajectory against given Measurement_Criterion/criteria.

determine_trajectory_requirement_progress

- To determine the progress of a Vehicle against the Trajectory_Requirement.

provide_vehicle_trajectory

- To provide the Planned_Trajectory of the Vehicle.

identify_missing_information

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

B.2.78.5 Subject Matter Semantics

The subject matter of Vehicle Guidance is the Vehicle's trajectory.

Exclusions

The subject matter of Vehicle Guidance does not include:

- Why a manoeuvre is to be performed or the prioritisation of conflicting demands.
- How Motion_Commands are fulfilled, e.g. the movement of a control surface to fulfil an attitude demand.
- Trajectory prediction.

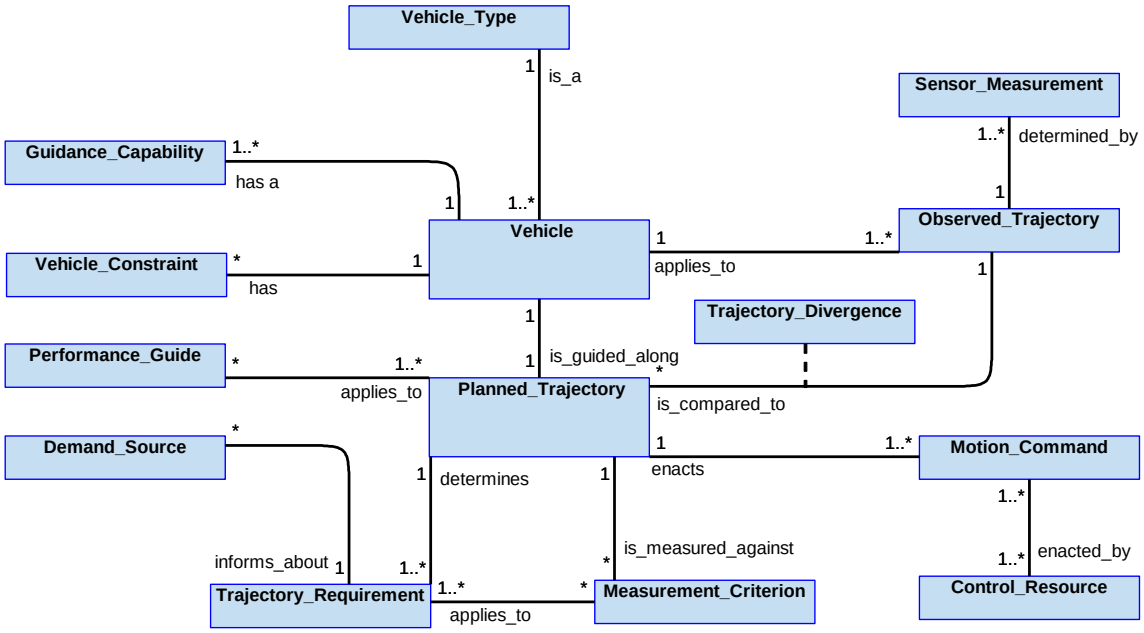

Figure 1254: Vehicle Guidance Semantics

B.2.78.5.1 Entities

Control_Resource

A resource or other part of the system which can be instructed to control the Vehicle.

Demand_Source

An input that controls which source of Trajectory_Requirement should be used, i.e. switches between a path demand and a direct control input from an authorised operator.

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 (e.g. attitude or speed) to be followed in order to enact the required manoeuvre.

Planned_Trajectory

The plotted movement necessary in order to meet the Trajectory_Requirement.

Trajectory_Divergence

Any divergence from the commanded Planned_Trajectory.

Trajectory_Requirement

An input demand for the Vehicle to be moved.

Vehicle

An instance of a moveable object whose movement can be controlled.

Observed_Trajectory

The actual trajectory of the Vehicle.

Measurement_Criterion

Something by which the quality or cost of the trajectory will be measured.

Vehicle_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 due to noise abatement limits).

Sensor_Measurement

The determined values about a Vehicle Observed_Trajectory such as speed or altitude.

Vehicle_Type

A type of Vehicle.

B.2.78.6 Design Rationale

B.2.78.6.1 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.

B.2.78.6.2 Design Considerations

Directly Applicable Policies

These policies were specifically taken into account in defining Vehicle Guidance:

- Data Driving - It is expected that the classes of Vehicle and their guidance behaviour will be data driven.

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 instances 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 Path Demands and Vehicle Stability and Control as part of an Exploiting Platform.
- Vehicle Guidance may receive Trajectory_Requirements from other components providing path demands 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, height, magnetic heading, true heading or track; a speed, Mach Number or climb / descent rate, and direct control inputs from an authorised operator, such as climb/descent rate and turn rate.
- Demand_Source will control which source Vehicle Guidance will use to determine the Vehicle Planned_Trajectory.
- Demand_Source may also indicate that the trajectory of the vehicle is being controlled by other components or systems. For example, stick and throttle inputs direct to the Vehicle Stability and Control component, direct throttle control of an engine or mechanical control of control surfaces. In these cases Vehicle Guidance may be expected to ensure that step changes to Motion_Commands do not occur when sources are changed.

B.2.78.6.3 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 limits of the air vehicle, the path of the air vehicle 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.

B.2.78.6.4 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.

B.2.78.7 Services

B.2.78.7.1 Service Definitions

B.2.78.7.1.1 Trajectory_Demand

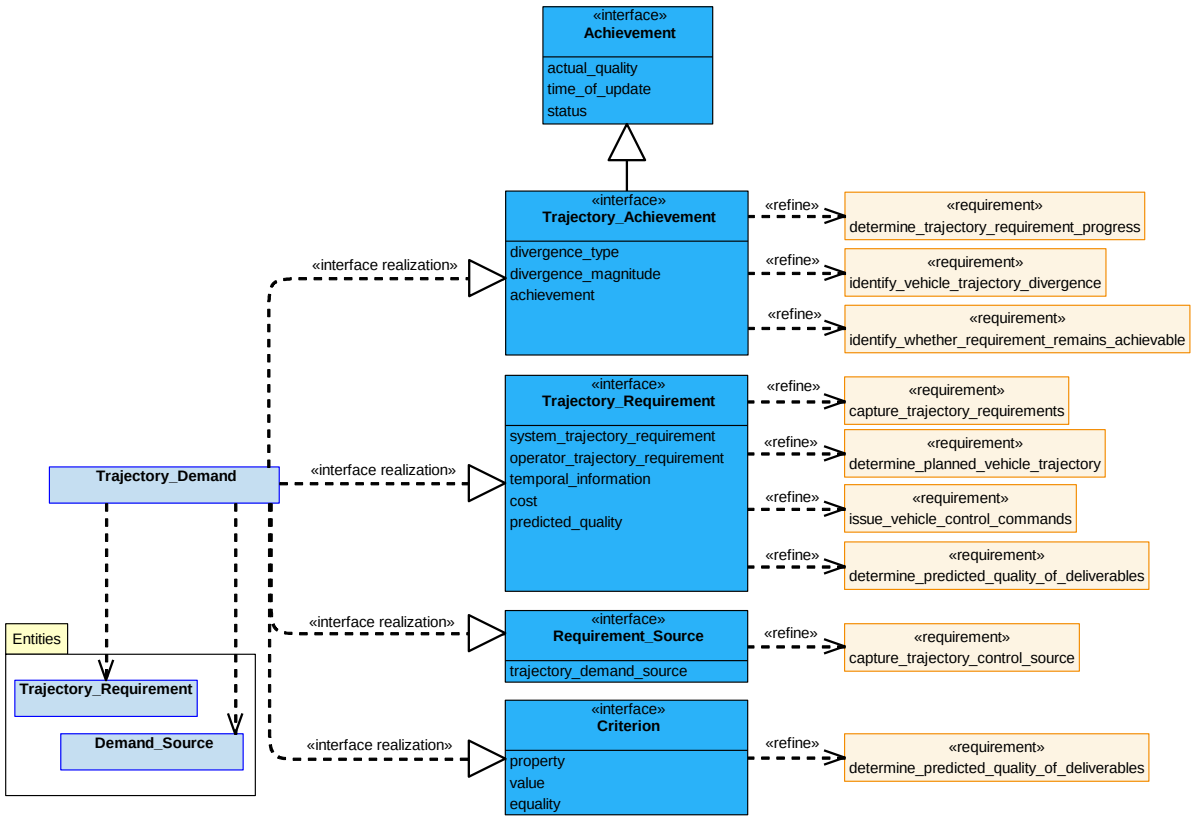

Figure 1255: Trajectory_Demand Service Definition

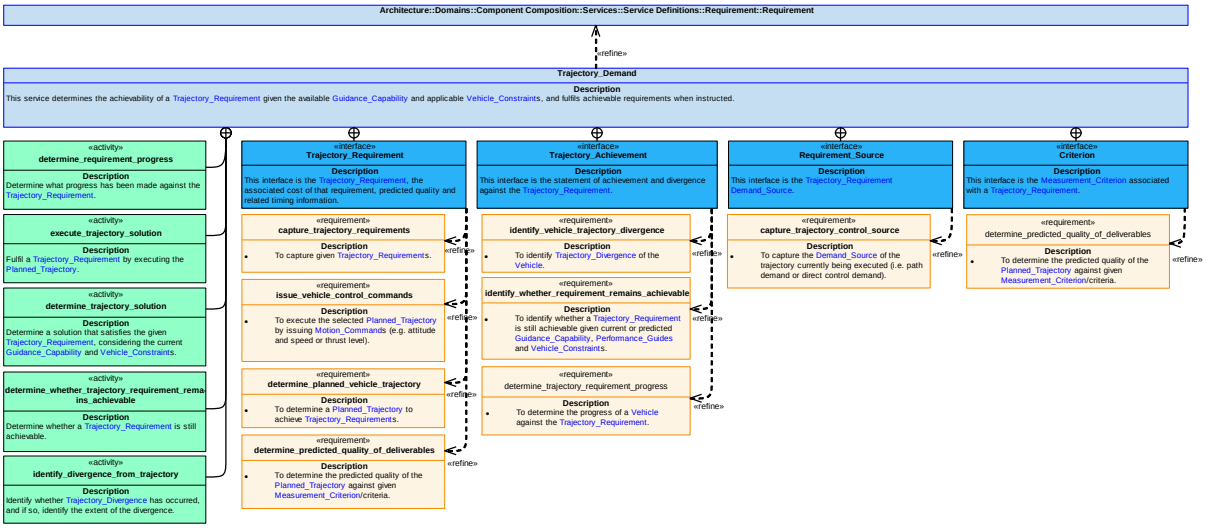

Figure 1256: Trajectory_Demand Service Policy

Trajectory_Demand

This service determines the achievability of a Trajectory_Requirement given the available Guidance_Capability and applicable Vehicle_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

system_trajectory_requirement	The definition of the Trajectory_Requirement received from the system.
operator_trajectory_requirement	The definition of the Trajectory_Requirement received from the 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.

Requirement_Source

This interface is the Trajectory_Requirement Demand_Source.

Attribute

trajectory_demand_source	Information defining the Trajectory_Requirement Demand_Source.

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 Vehicle_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.

B.2.78.7.1.2 Control_Command

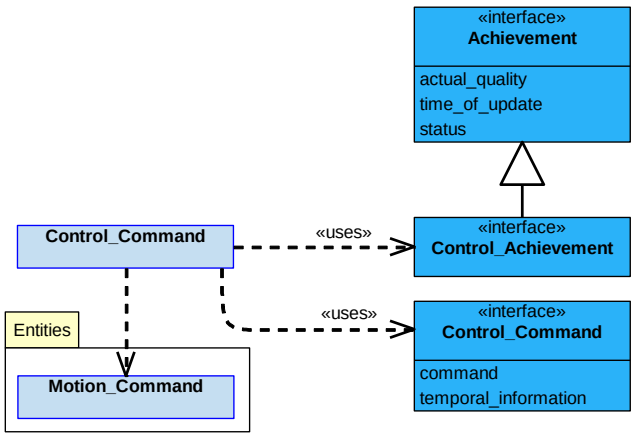

Figure 1257: Control_Command Service Definition

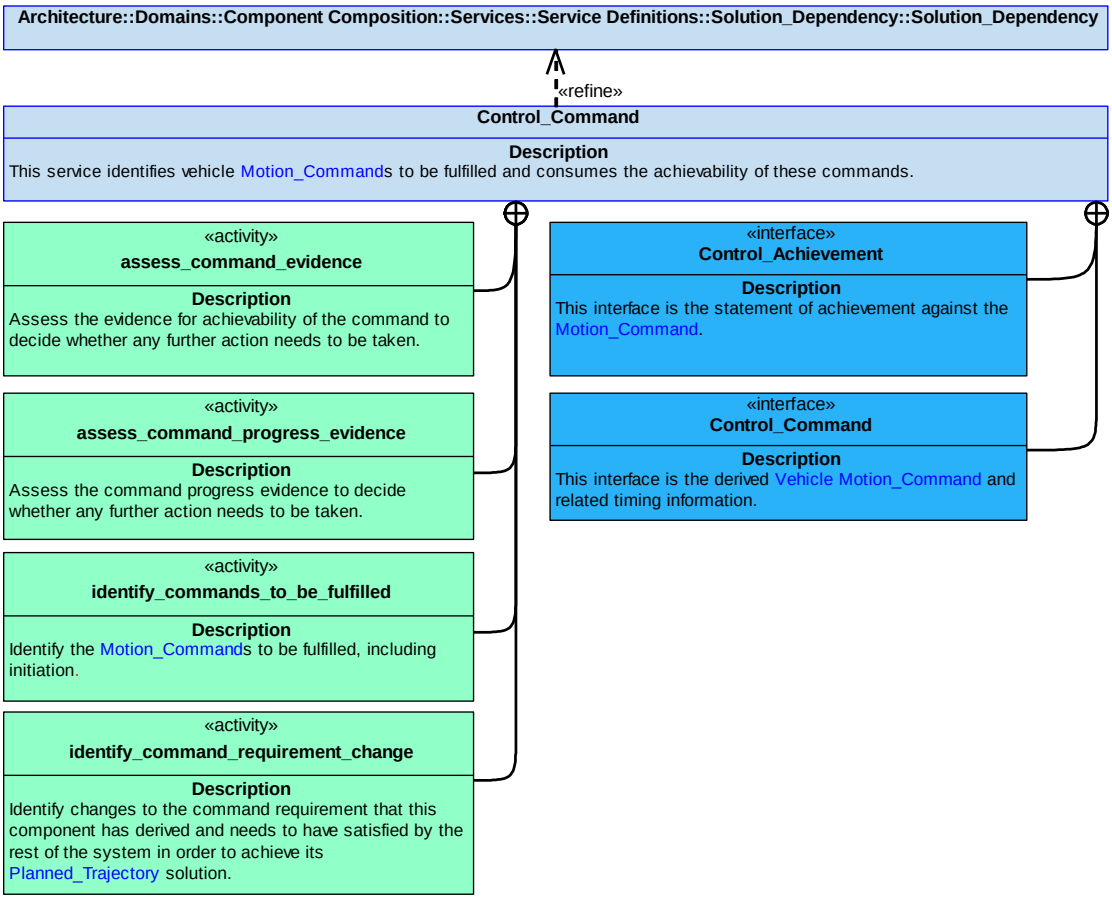

Figure 1258: Control_Command Service Policy

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.

B.2.78.7.1.3 Planned_Trajectory

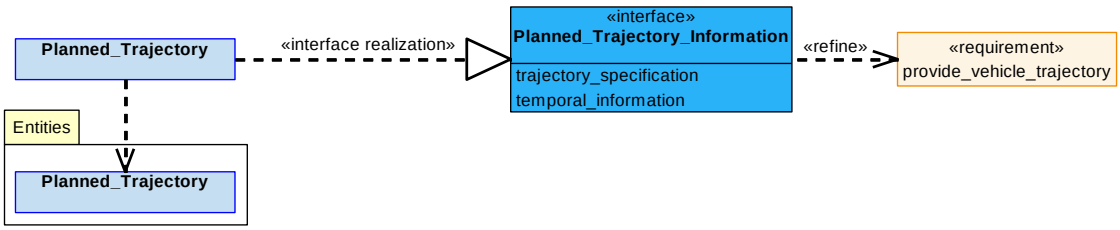

Figure 1259: Planned_Trajectory Service Definition

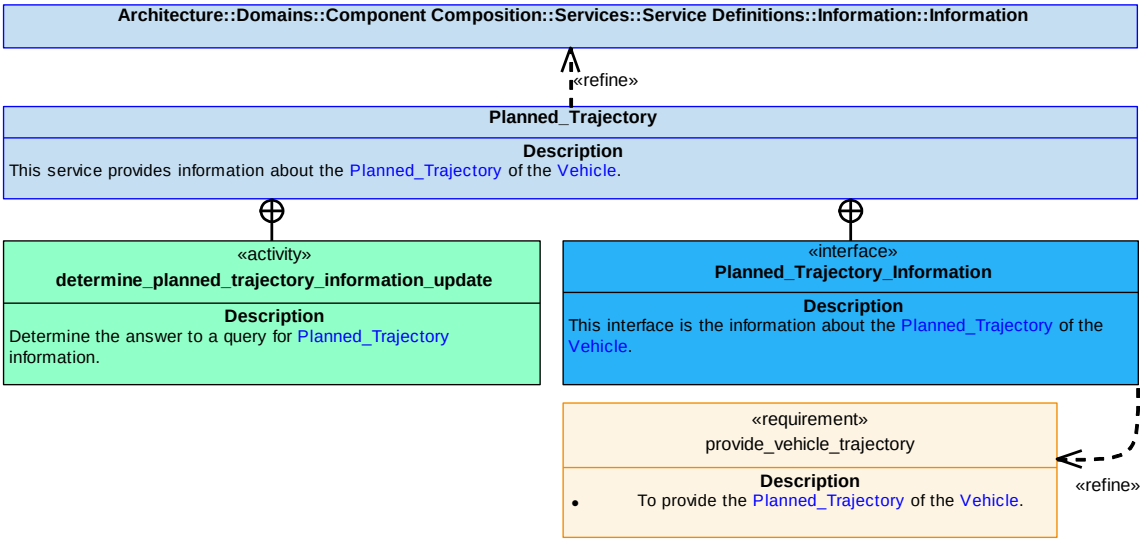

Figure 1260: Planned_Trajectory Service Policy

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.

B.2.78.7.1.4 Performance_Guide

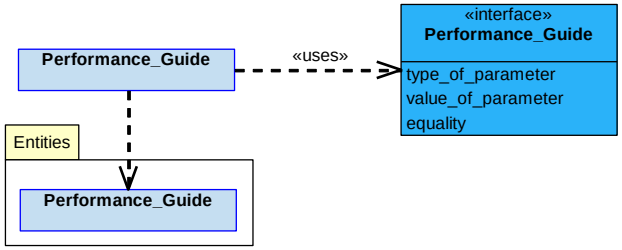

Figure 1261: Performance_Guide Service Definition

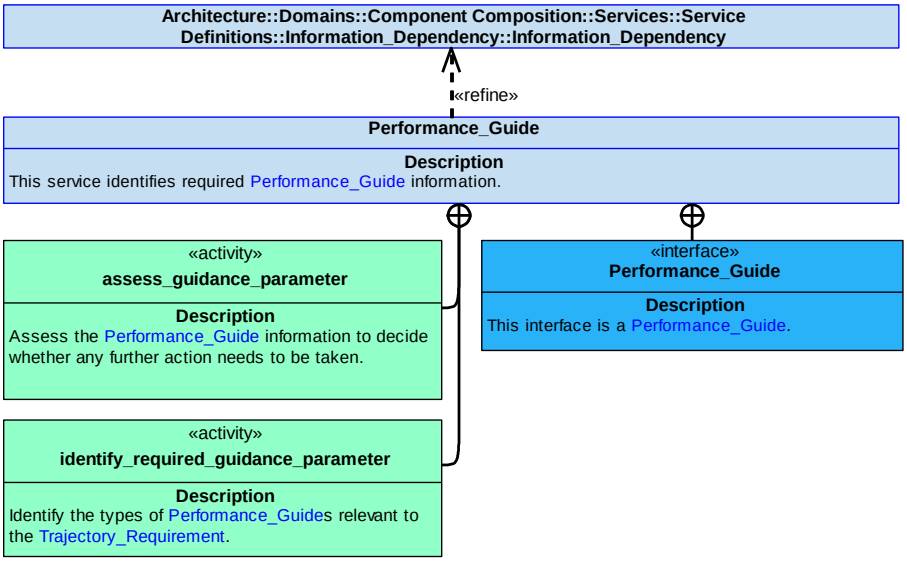

Figure 1262: Performance_Guide Service Policy

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.

B.2.78.7.1.5 Observed_Trajectory

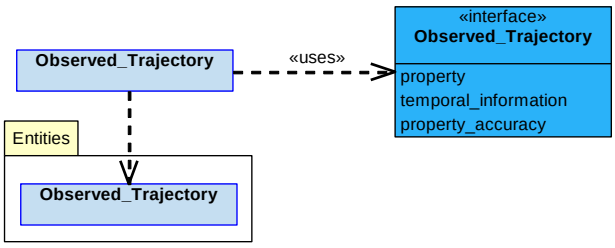

Figure 1263: Observed_Trajectory Service Definition

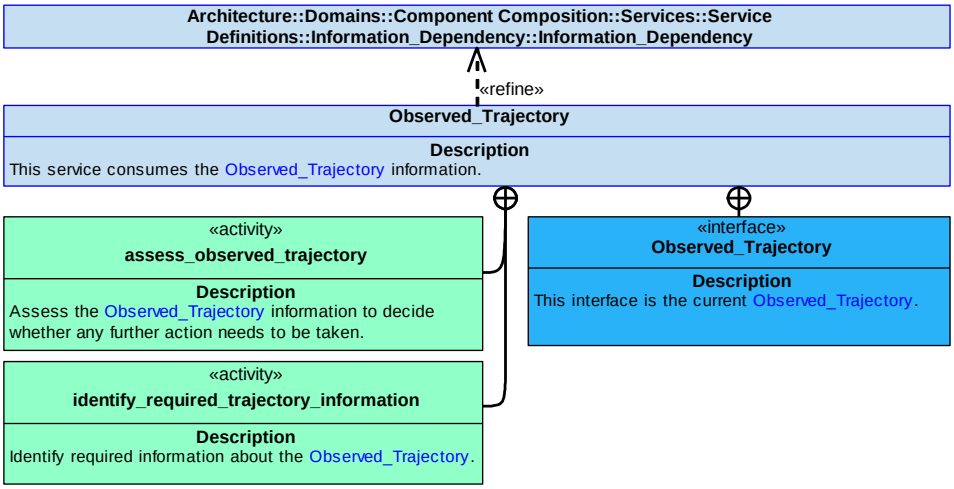

Figure 1264: Observed_Trajectory Service Policy

Observed_Trajectory

This service consumes the Observed_Trajectory information.

Interface

Observed_Trajectory

This interface is the current Observed_Trajectory.

Attributes

property	The property and value of the Observed_Trajectory.
temporal_information	Information covering timing of the Observed_Trajectory information provided.
property_accuracy	The accuracy of the value of a particular Observed_Trajectory property.

Activities

assess_observed_trajectory

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

identify_required_trajectory_information

Identify required information about the Observed_Trajectory.

B.2.78.7.1.6 Constraint

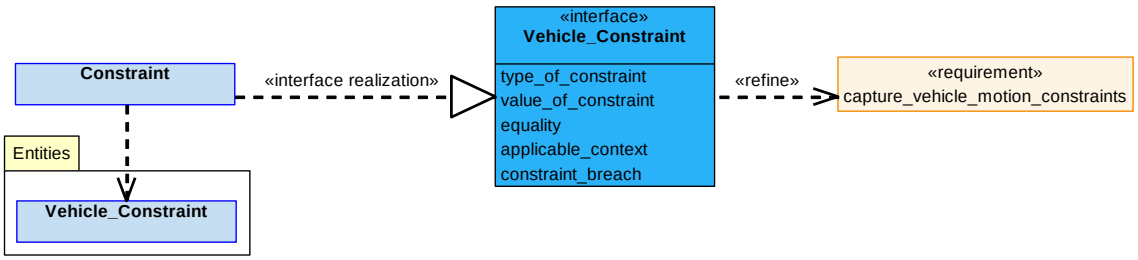

Figure 1265: Constraint Service Definition

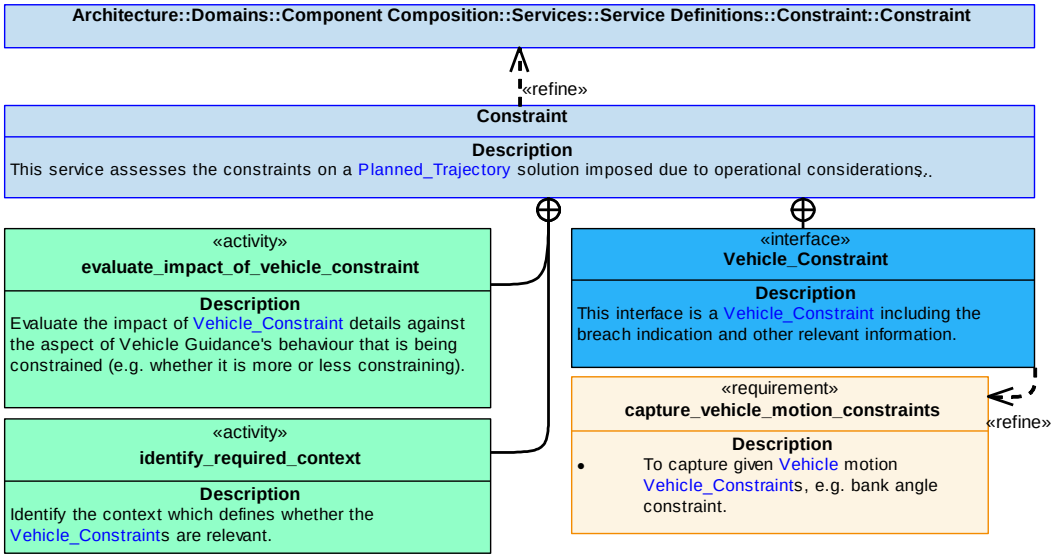

Figure 1266: Constraint Service Policy

Constraint

This service assesses the constraints on a Planned_Trajectory solution imposed due to operational considerations.

Interface

Vehicle_Constraint

This interface is a Vehicle_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 Vehicle_Constraint is applicable.
constraint_breach	A statement that the Vehicle_Constraint has been breached.

Activities

evaluate_impact_of_vehicle_constraint

Evaluate the impact of Vehicle_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 Vehicle_Constraints are relevant.

B.2.78.7.1.7 Capability

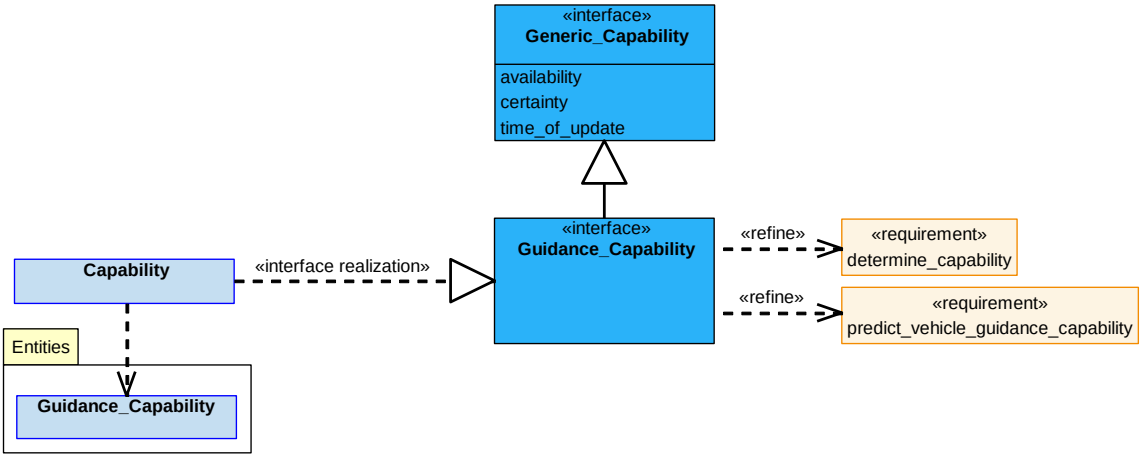

Figure 1267: Capability Service Definition

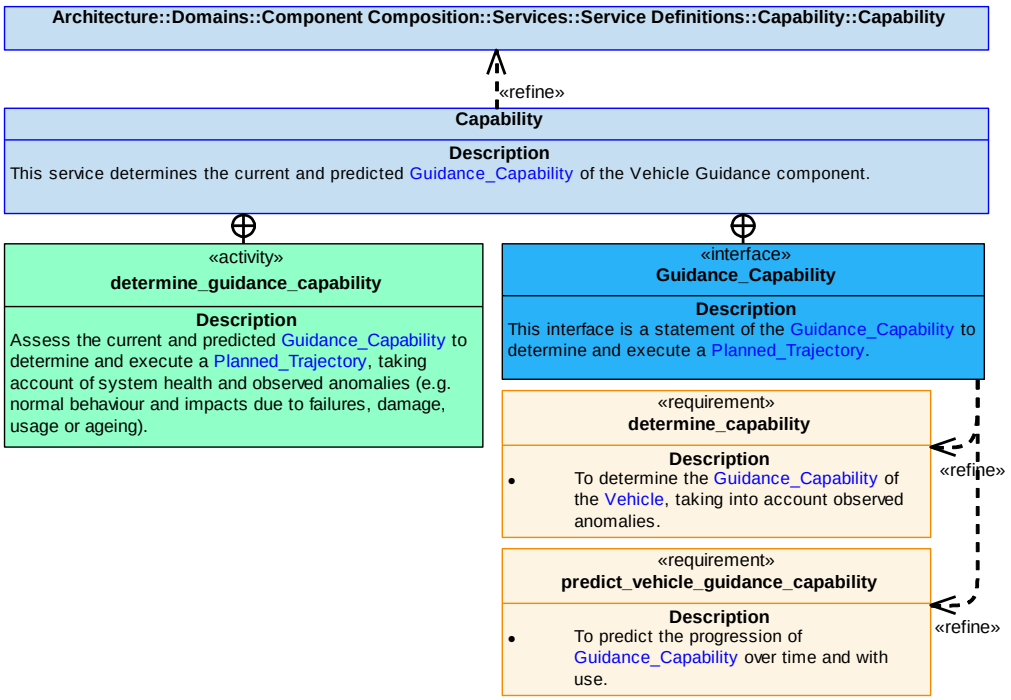

Figure 1268: Capability Service Policy

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).

B.2.78.7.1.8 Control_Resource_Evidence

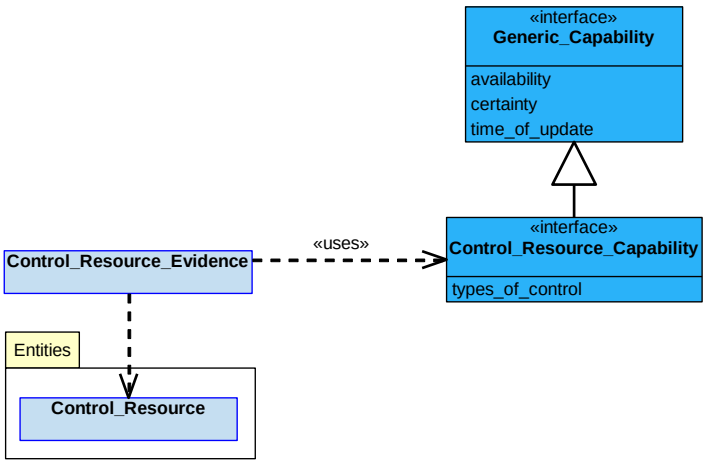

Figure 1269: Control_Resource_Evidence Service Definition

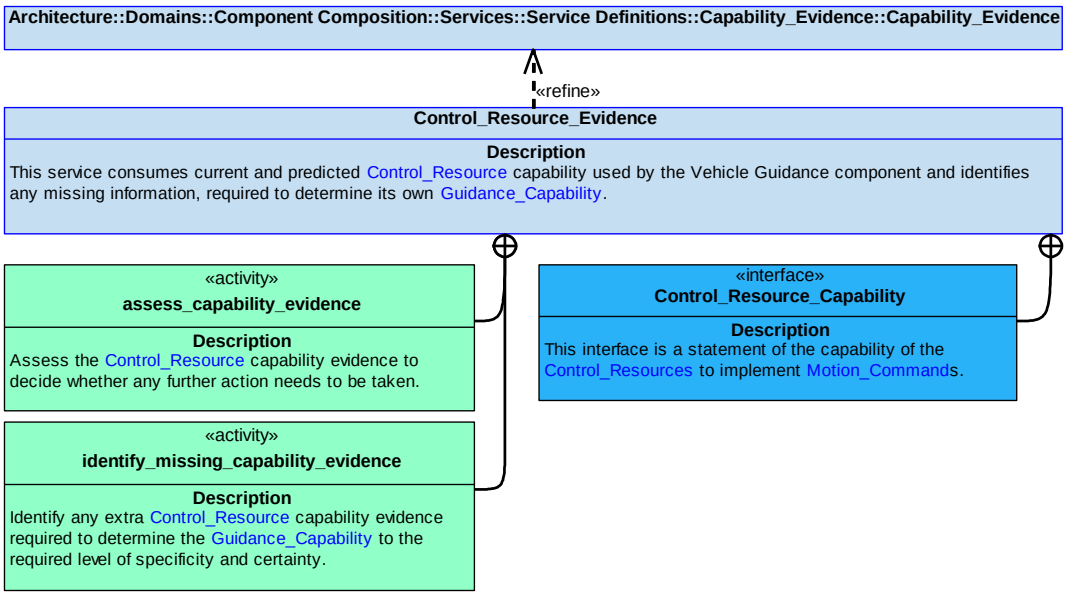

Figure 1270: Control_Resource_Evidence Service Policy

Control_Resource_Evidence

This service consumes current and predicted Control_Resource capability used by the Vehicle Guidance component and identifies any missing information, required to determine its own Guidance_Capability.

Interface

Control_Resource_Capability

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.

Activities

assess_capability_evidence

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

identify_missing_capability_evidence

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

B.2.78.7.1.9 Sensor_Measurement_Evidence

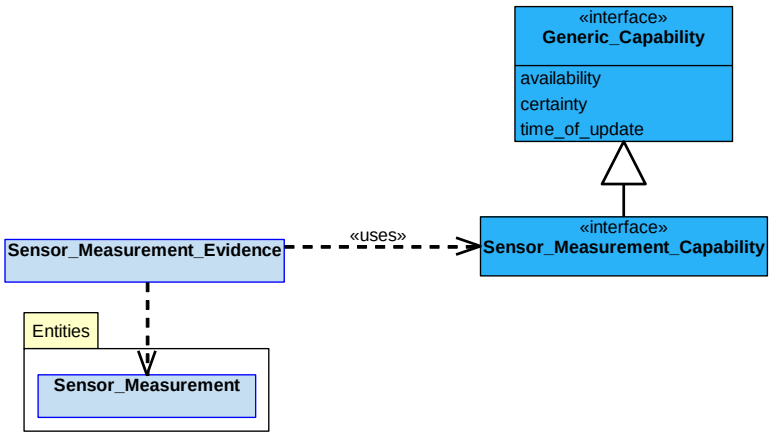

Figure 1271: Sensor_Measurement_Evidence Service Definition

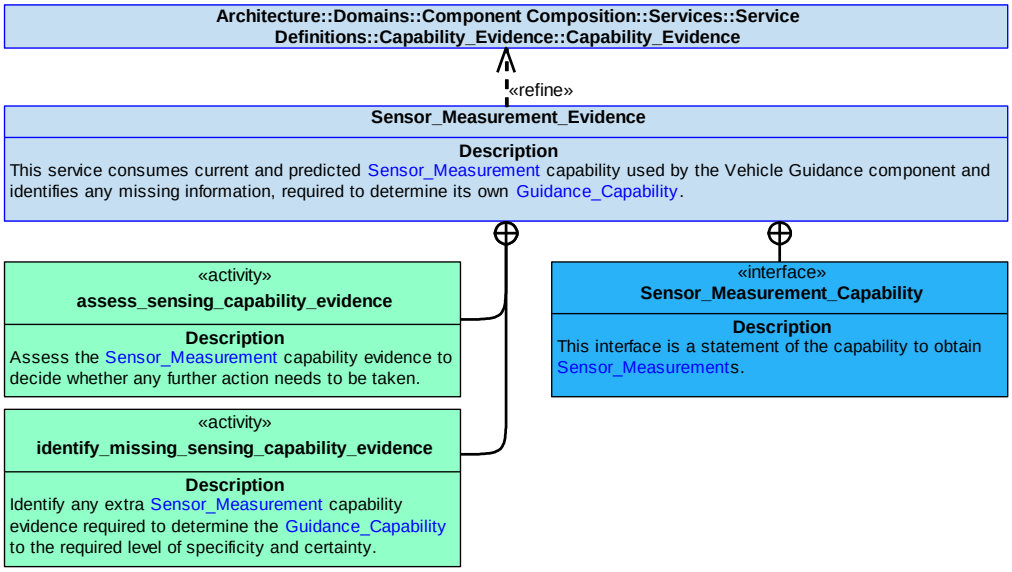

Figure 1272: Sensor_Measurement_Evidence Service Policy

Sensor_Measurement_Evidence

This service consumes current and predicted Sensor_Measurement capability used by the Vehicle Guidance component and identifies any missing information, required to determine its own Guidance_Capability.

Interface

Sensor_Measurement_Capability

This interface is a statement of the capability to obtain Sensor_Measurements.

Activities

identify_missing_sensing_capability_evidence

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

assess_sensing_capability_evidence

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

B.2.78.7.2 Service Dependencies

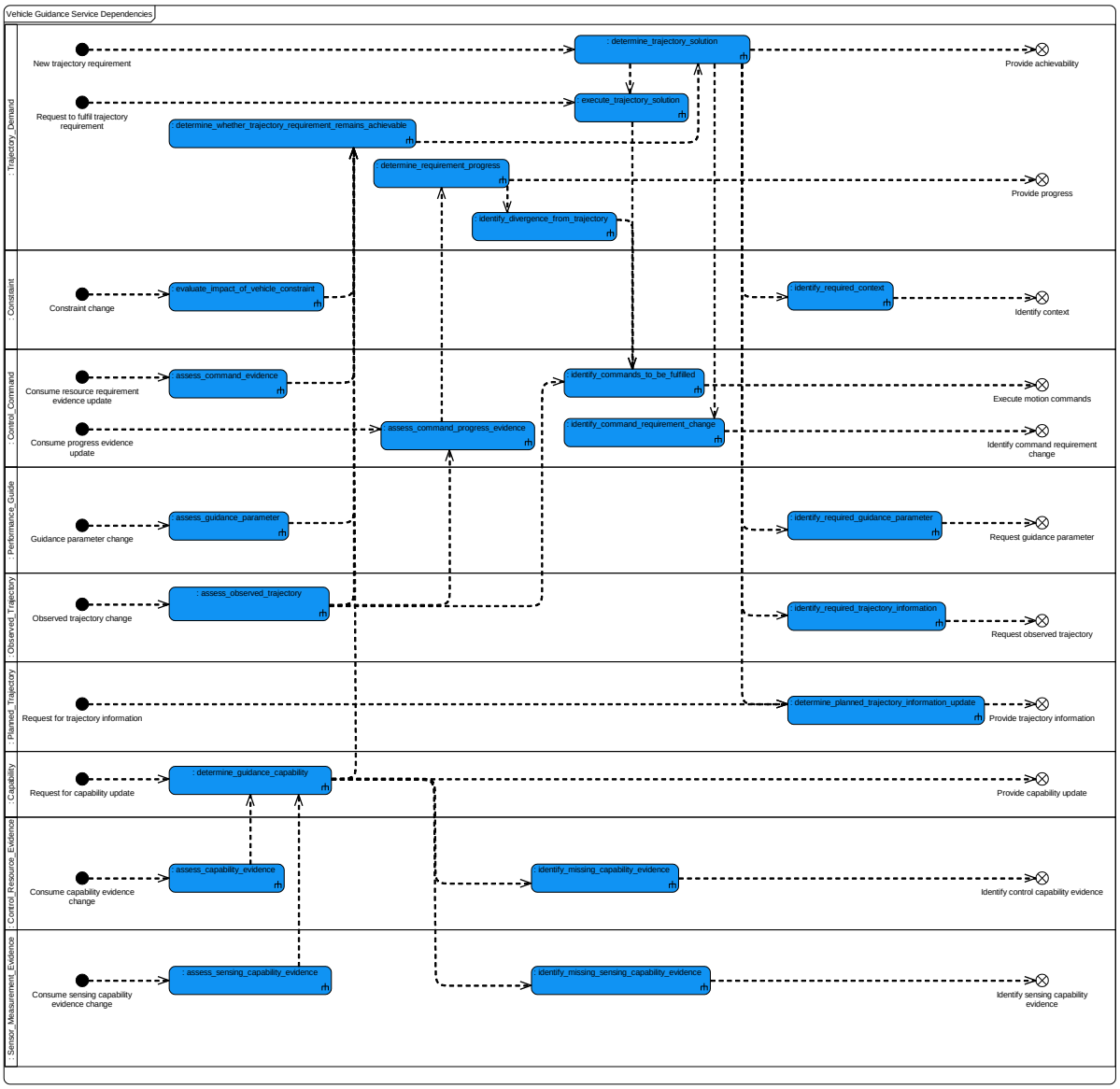

Figure 1273: Vehicle Guidance Service Dependencies

B.2.79 Vehicle Performance

B.2.79.1 Role

The role of Vehicle Performance is to provide performance parameters for different vehicle configurations and performance regimes.

B.2.79.2 Overview

Control Architecture

Vehicle Performance is a service component as defined in the Control Architecture policy.

Standard Pattern of Use

In response to a query about the Vehicle_Configuration or Performance_Parameters, Vehicle Performance determines the appropriate data considering all factors. For example, for a Parameter_Query, the Performance_Regime, External_Conditions and Vehicle_Configuration will be used to determine the maximum speed during banking.

Examples of Use

- Vehicle Performance will be required where allowable vehicle Performance_Parameter values vary, depending on the Vehicle_Configuration and/or Performance_Regime. 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.

B.2.79.3 Service Summary

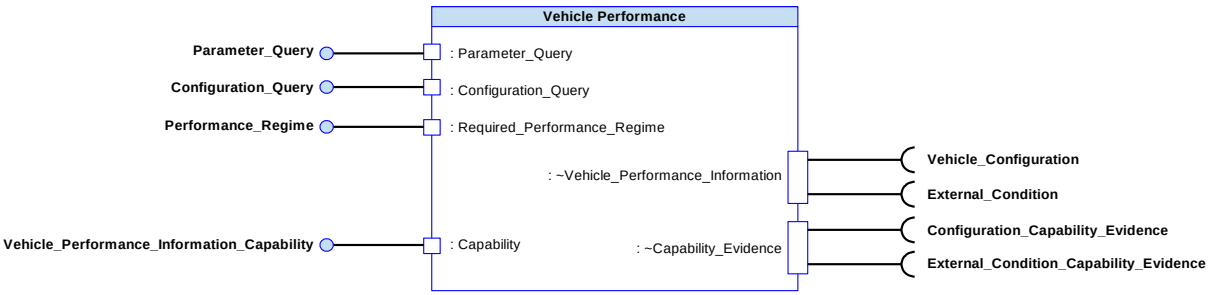

Figure 1274: Vehicle Performance Service Summary

B.2.79.4 Responsibilities

capture_vehicle_configuration

- To capture the Vehicle_Configuration.

determine_applicable_values

- To determine applicable Performance_Parameter values for combinations of External_Conditions, Vehicle_Configuration and Performance_Regimes.

capture_performance_regime

- To capture the vehicle Performance_Regime(s) applicable to a particular query or time period.

capture_conditions

- To capture External_Conditions that affect vehicle performance.

determine_vehicle_configuration

- To determine Vehicle_Configurations for which a combination of Performance_Parameter values is allowable.

assess_capability

- To assess the Capability to answer queries, 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 the component's Capability over time and with use.

identify_missing_information

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

B.2.79.5 Subject Matter Semantics

The subject matter of Vehicle Performance is values for Performance_Parameters.

Exclusions

The subject matter of Vehicle Performance does not include:

- The determination of whether Performance_Parameter limits are complied with or exceeded, Vehicle Performance merely determines those that are applicable.

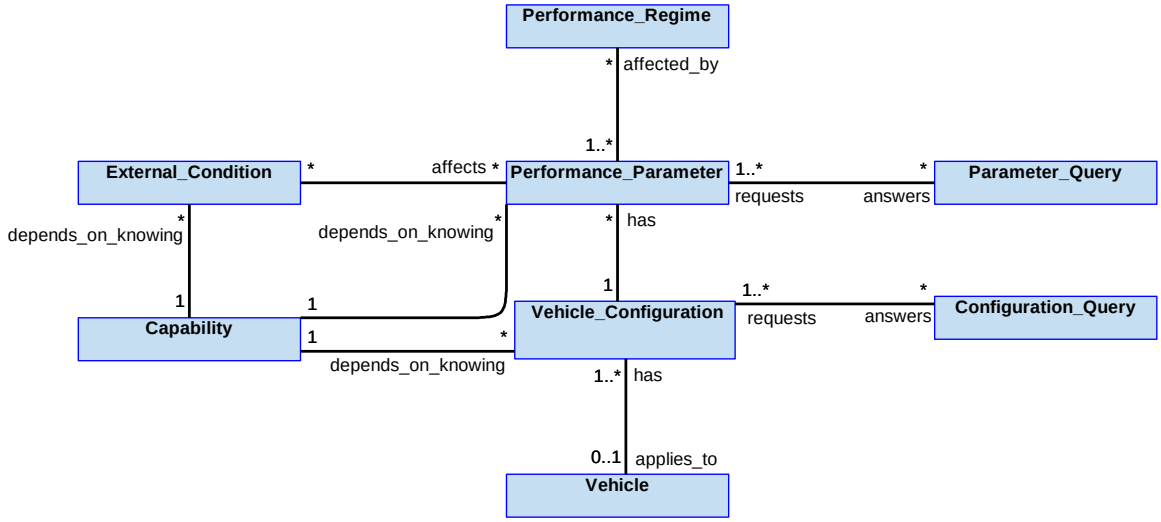

Figure 1275: Vehicle Performance Semantics

B.2.79.5.1 Entities

Configuration_Query

A query to determine the appropriate configuration needed to achieve or maintain Performance_Parameters for a Performance_Regime in certain External_Conditions.

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 in a Performance_Regime in certain External_Conditions.

Performance_Regime

A particular mode of operation that has a set of minimum, maximum or optimum performance characteristics (e.g. for take-off, optimum cruise range or during stores release).

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 parameter value that defines or limits some aspect of vehicle performance, e.g. optimum speed, maximum altitude, or maximum take-off mass.

B.2.79.6 Design Rationale

B.2.79.6.1 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.

B.2.79.6.2 Design Considerations

Directly Applicable Policies

These policies 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.
- It is expected that this component will be told which performance regimes are applicable to a particular requirement and that any conflicts between regimes have been already resolved by other components.
- 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.

B.2.79.6.3 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.

B.2.79.6.4 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.

B.2.79.7 Services

B.2.79.7.1 Service Definitions

B.2.79.7.1.1 Parameter_Query

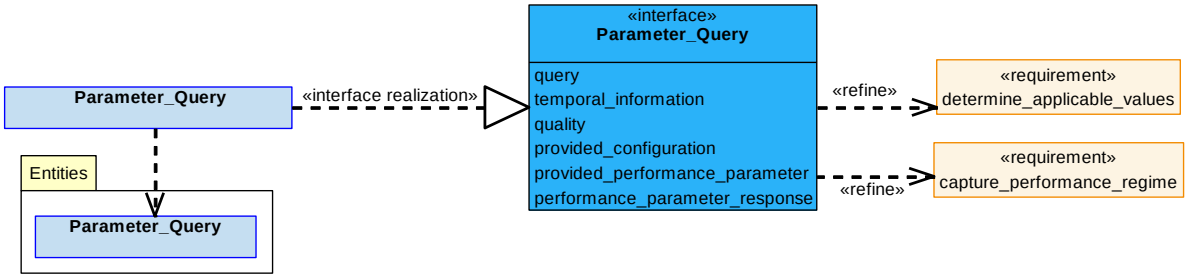

Figure 1276: Parameter_Query Service Definition

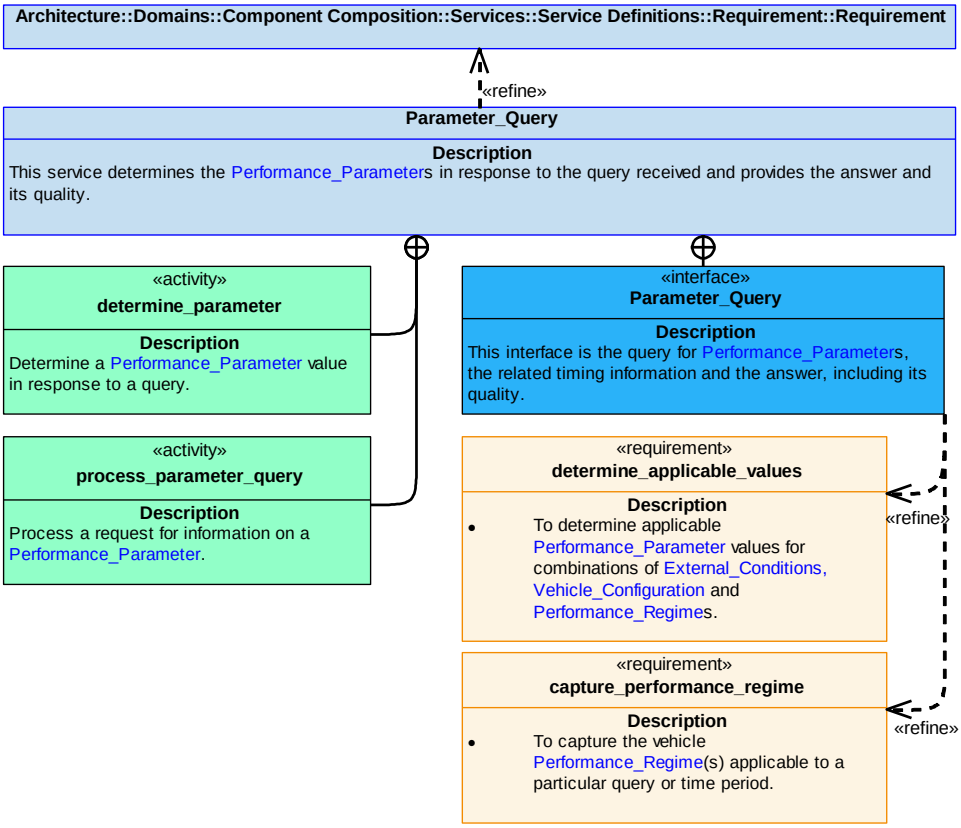

Figure 1277: Parameter_Query Service Policy

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_configuration	The Vehicle_Configuration provided within the query (e.g. flap angle or aperture position).
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.

B.2.79.7.1.2 Configuration_Query

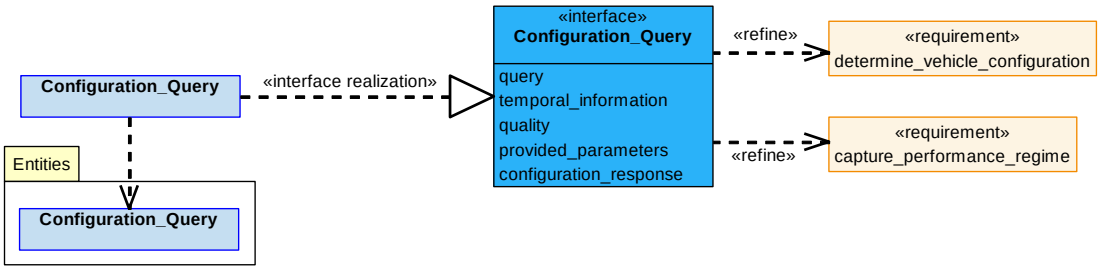

Figure 1278: Configuration_Query Service Definition

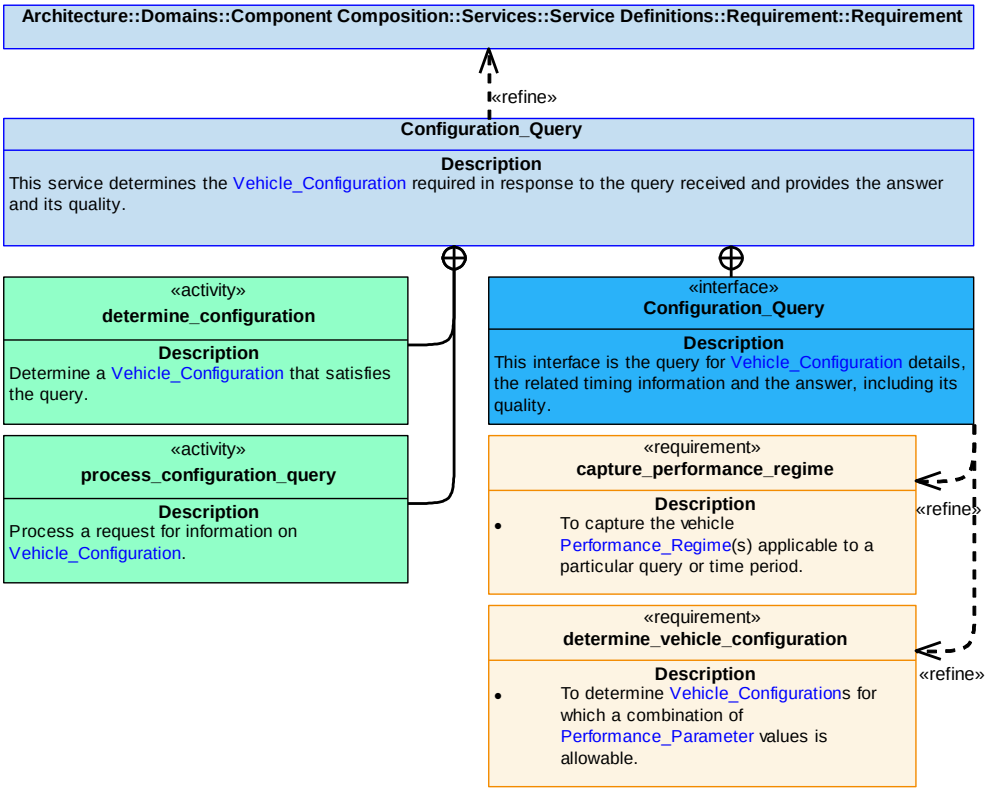

Figure 1279: Configuration_Query Service Policy

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.

B.2.79.7.1.3 Required_Performance_Regime

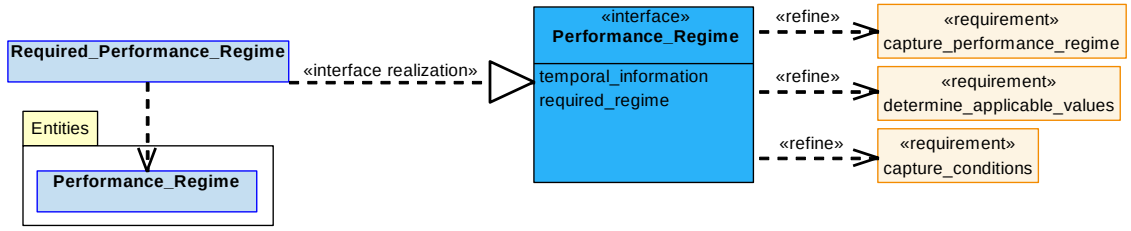

Figure 1280: Required_Performance_Regime Service Definition

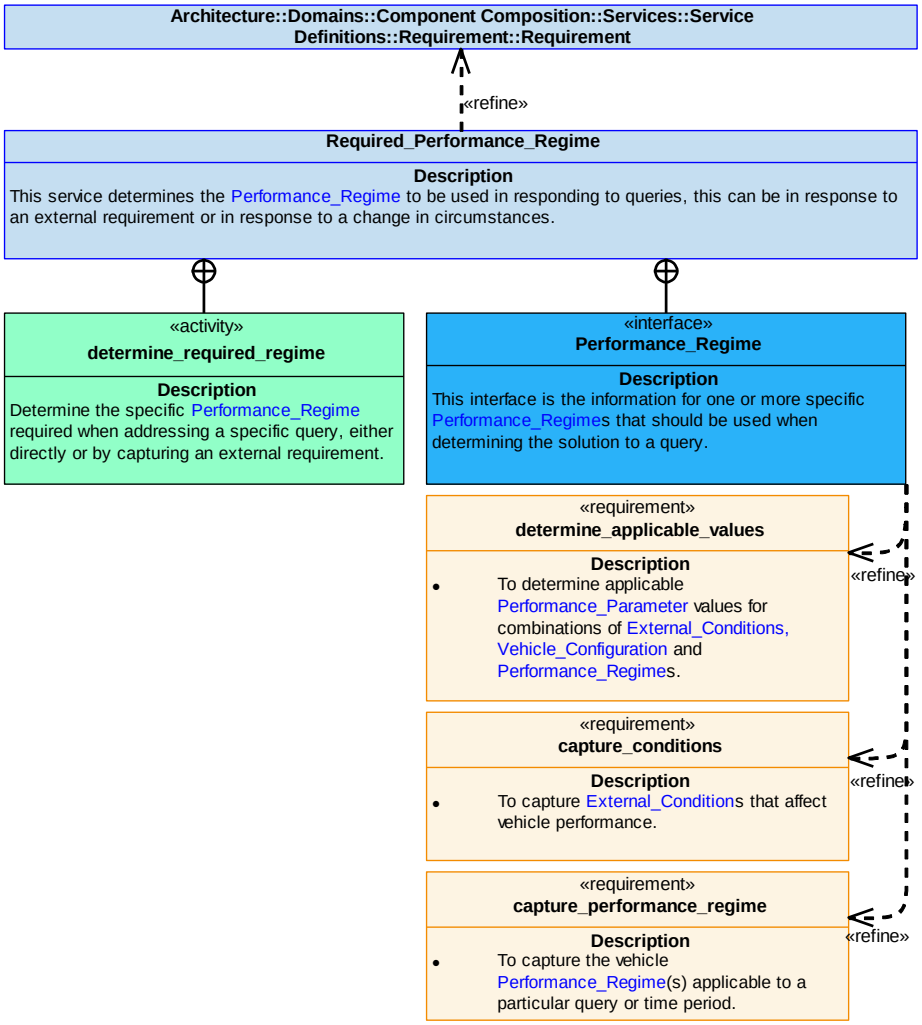

Figure 1281: Required_Performance_Regime Service Policy

Required_Performance_Regime

This service determines the Performance_Regime to be used in responding to queries, this can be in response to an external requirement or in response to a change in circumstances.

Interface

Performance_Regime

This interface is the information for one or more specific Performance_Regimes that should be used when determining the solution to a query.

Attributes

temporal_information	Information covering timing, such as start and end times, e.g. the period over which a Performance_Regime is required to be used in responding to queries.
required_regime	The specific Performance_Regime 'rule set' to be used.

Activity

determine_required_regime

Determine the specific Performance_Regime required when addressing a specific query, either directly or by capturing an external requirement.

B.2.79.7.1.4 Vehicle_Performance_Information

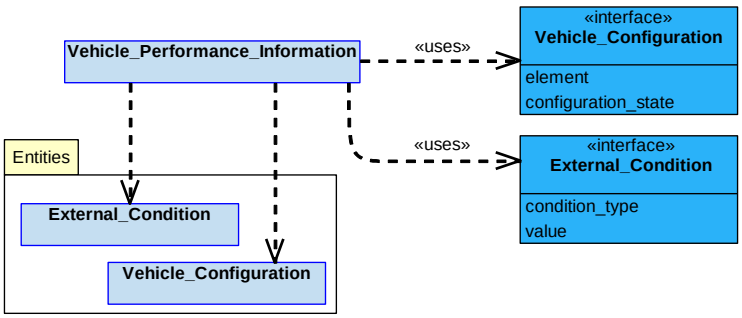

Figure 1282: Vehicle_Performance_Information Service Definition

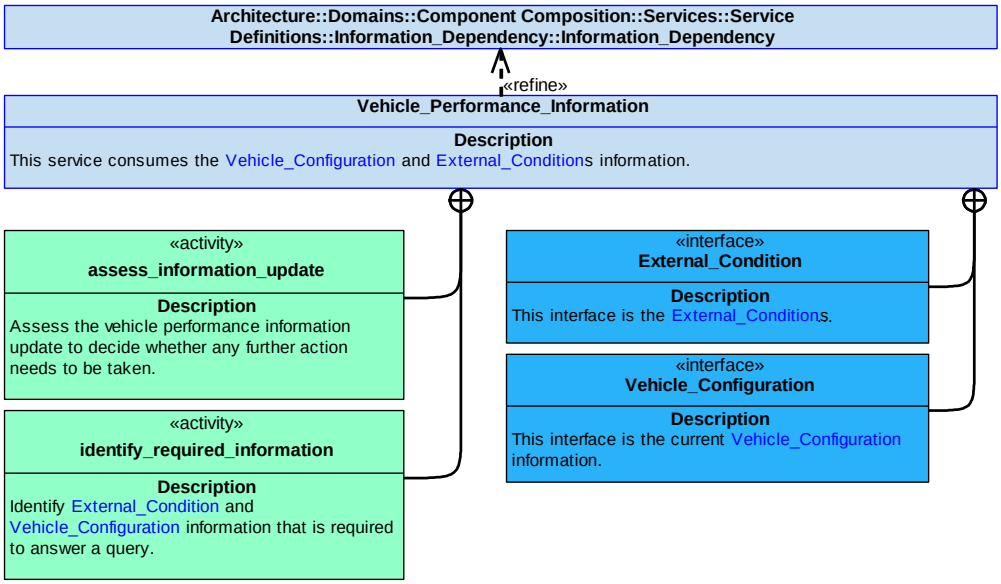

Figure 1283: Vehicle_Performance_Information Service Policy

Vehicle_Performance_Information

This service consumes the Vehicle_Configuration and External_Conditions information.

Interfaces

Vehicle_Configuration

This interface is the current 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).

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_information_update

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

identify_required_information

Identify External_Condition and Vehicle_Configuration information that is required to answer a query.

B.2.79.7.1.5 Capability

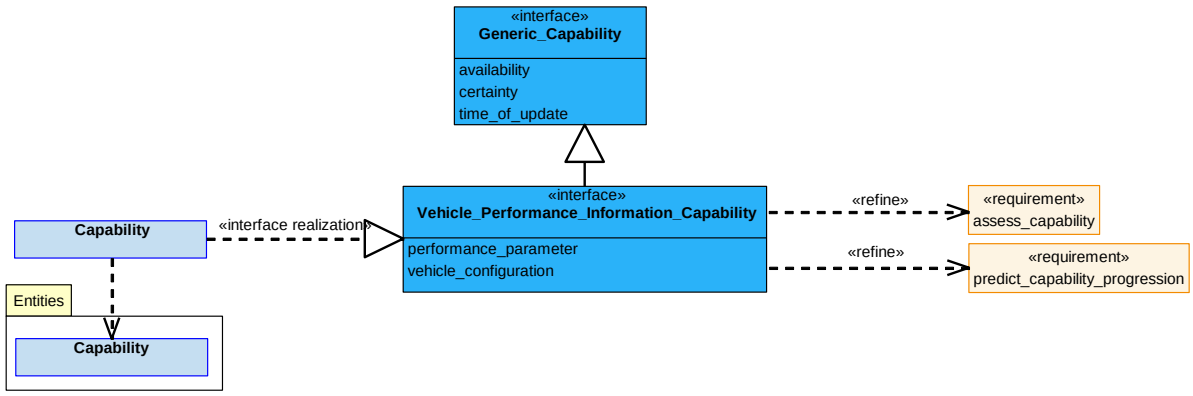

Figure 1284: Capability Service Definition

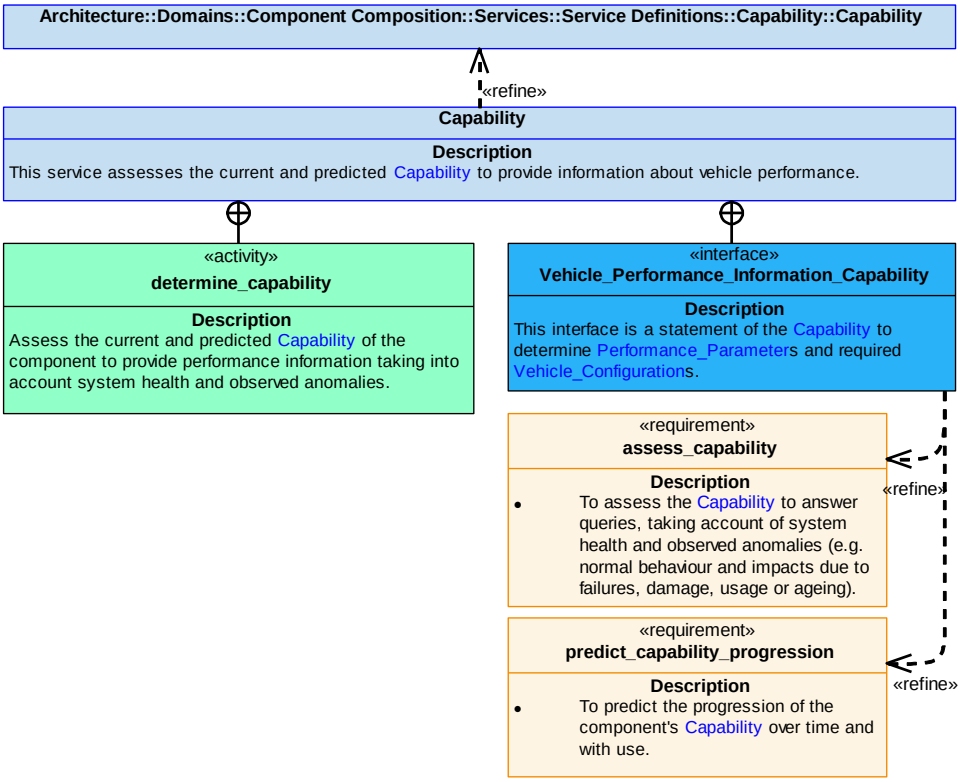

Figure 1285: Capability Service Policy

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. a Performance_Parameter).
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.

B.2.79.7.1.6 Capability_Evidence

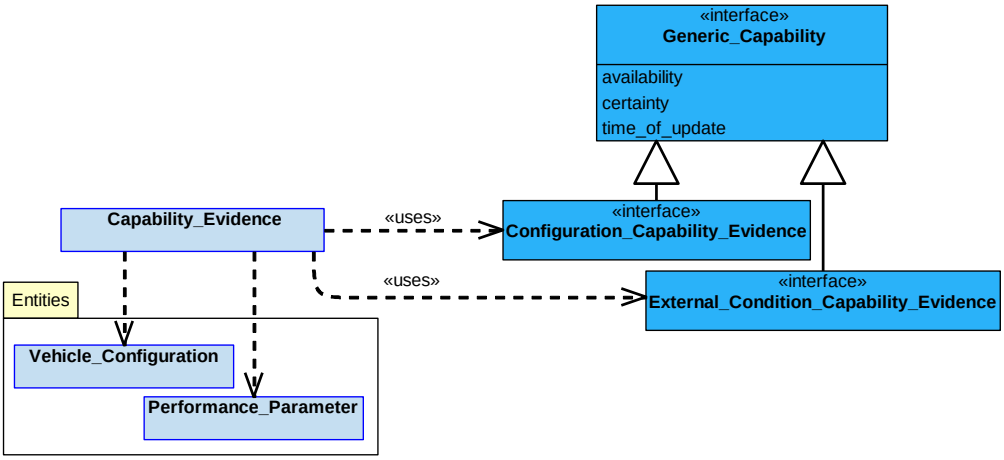

Figure 1286: Capability_Evidence Service Definition

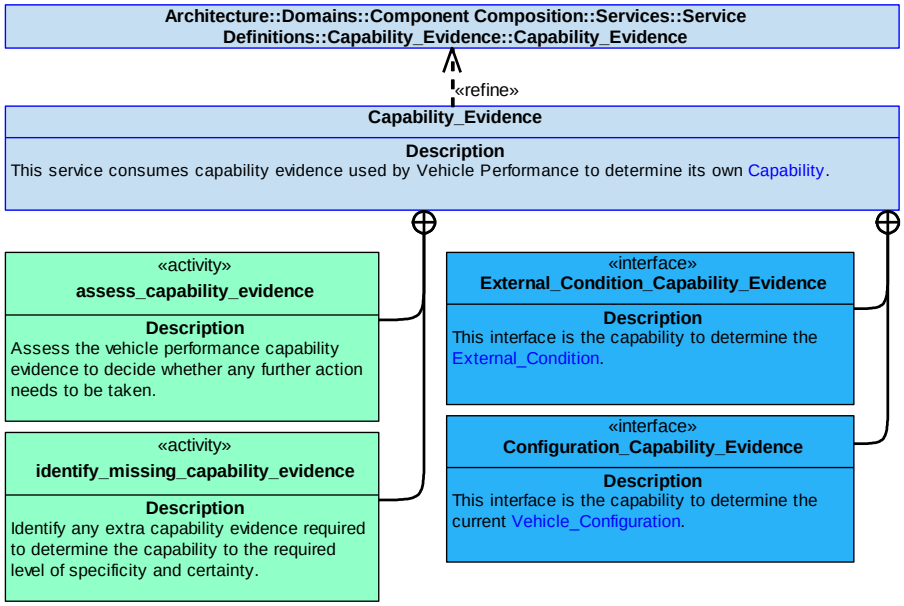

Figure 1287: Capability_Evidence Service Policy

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.

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.

B.2.79.7.2 Service Dependencies

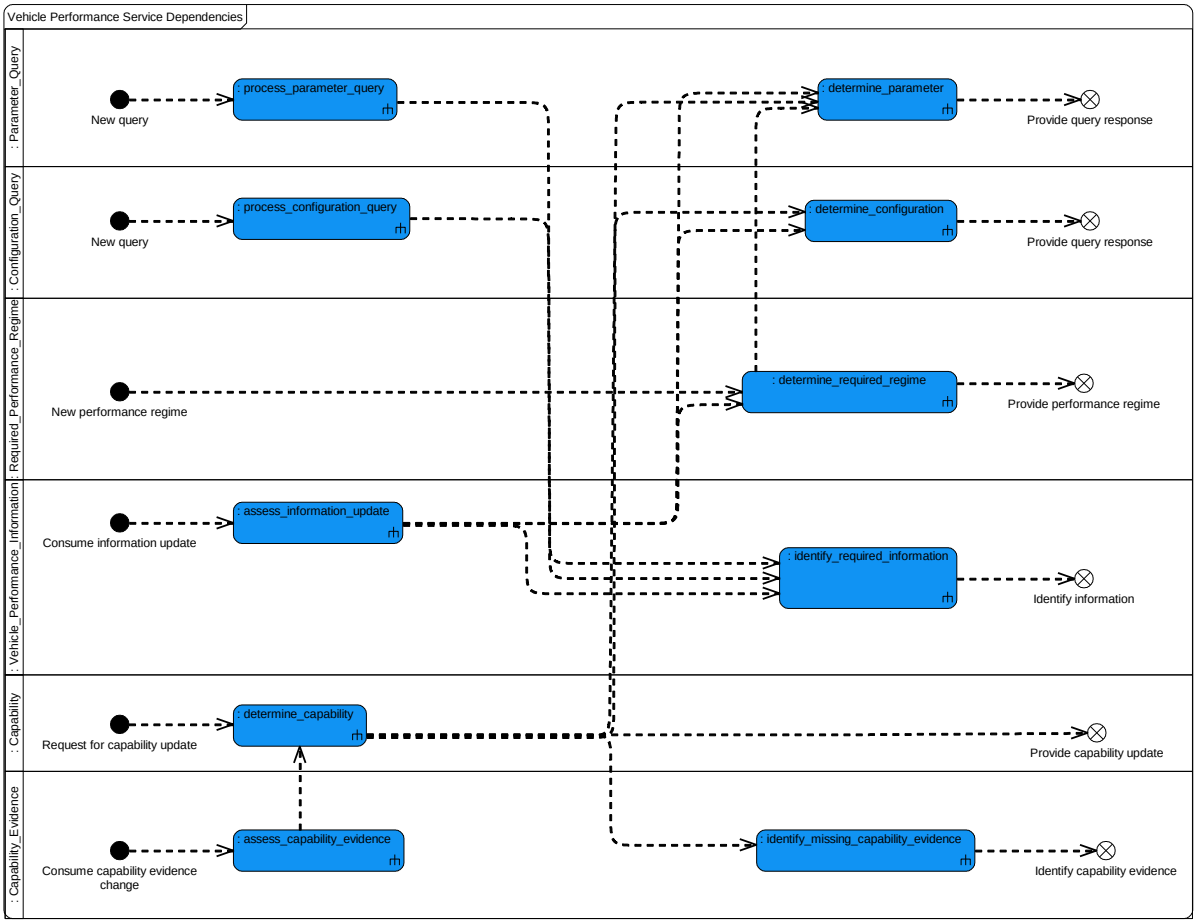

Figure 1288: Vehicle Performance Service Dependencies

B.2.80 Vehicle Stability and Control

B.2.80.1 Role

The role of Vehicle Stability and Control is to determine control effector commands to achieve vehicle control requirements whilst ensuring vehicle stability.

B.2.80.2 Overview

Control Architecture

Vehicle Stability and Control is an action component as defined in the Control Architecture policy.

Standard Pattern of Use

When a request to control the attitude or speed 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. 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.

B.2.80.3 Service Summary

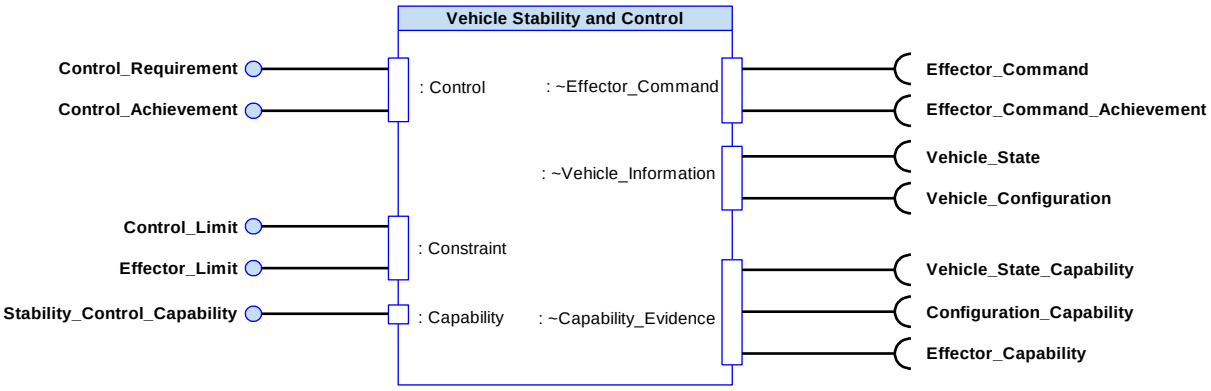

Figure 1289: Vehicle Stability and Control Service Summary

B.2.80.4 Responsibilities

capture_vehicle_control_requirements

- To capture Vehicle_Control_Requirements (e.g. attitude or speed).

determine_control_effector_commands

- To determine Control_Effector_Commands (e.g. control surface movement or thrust demand) to achieve Vehicle_Control_Requirements.

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).

predict_capability_progression

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

identify_missing_information

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

fulfil_requirement

- To fulfil a Vehicle_Control_Requirement by executing the planned Control_Effector_Commands.

identify_whether_requirement_remains_achievable

- To identify whether a Vehicle_Control_Requirement is still achievable given current or predicted Capability and Stability_Control_Limits.

capture_limits

- To capture given Stability_Control_Limits.

B.2.80.5 Subject Matter Semantics

The subject matter of Vehicle Stability and Control is the attitude and speed 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 speed demands, also called the "outer loops".

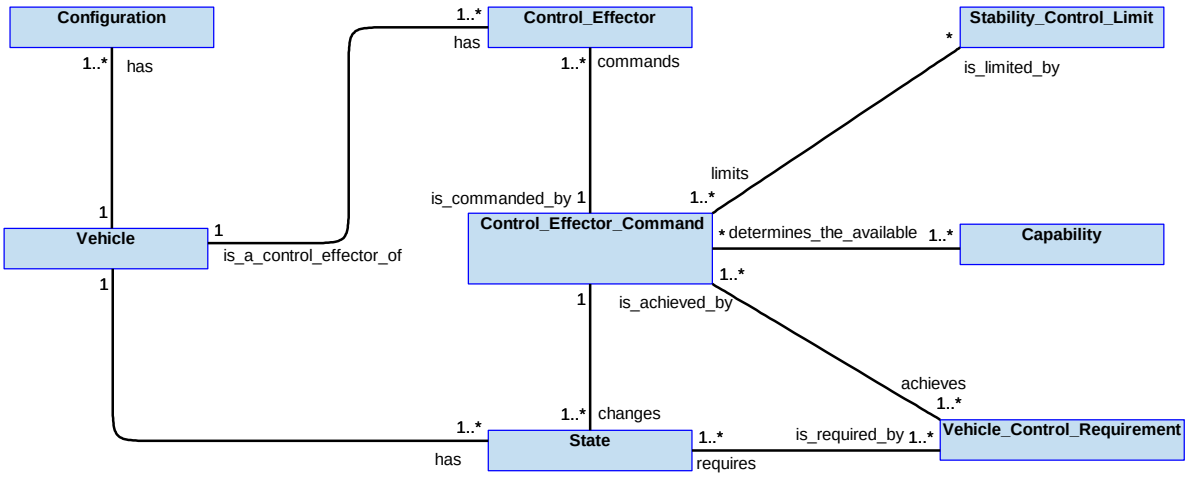

Figure 1290: Vehicle Stability and Control Semantics

B.2.80.5.1 Entities

Capability

The capability of controlling the attitude and speed of the Vehicle taking account of system health and observed anomalies.

Configuration

The configuration of the Vehicle, e.g. bay doors open.

Control_Effector

A control effector on a Vehicle (e.g. control surface or propulsion unit).

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 the Control_Effector_Command has to conform to.

State

The state (e.g. speed or attitude) of the Vehicle.

Vehicle

The object whose attitude and speed are to be controlled using its Control_Effectors.

Vehicle_Control_Requirement

A requirement to control the attitude or speed of the Vehicle.

B.2.80.6 Design Rationale

B.2.80.6.1 Assumptions

None.

B.2.80.6.2 Design Considerations

Directly Applicable Policies

- No policies 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 speed 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.

B.2.80.6.3 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.

B.2.80.6.4 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 component's data would need protecting to ensure it is kept secret (confidentiality), 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.

B.2.80.7 Services

B.2.80.7.1 Service Definitions

B.2.80.7.1.1 Control

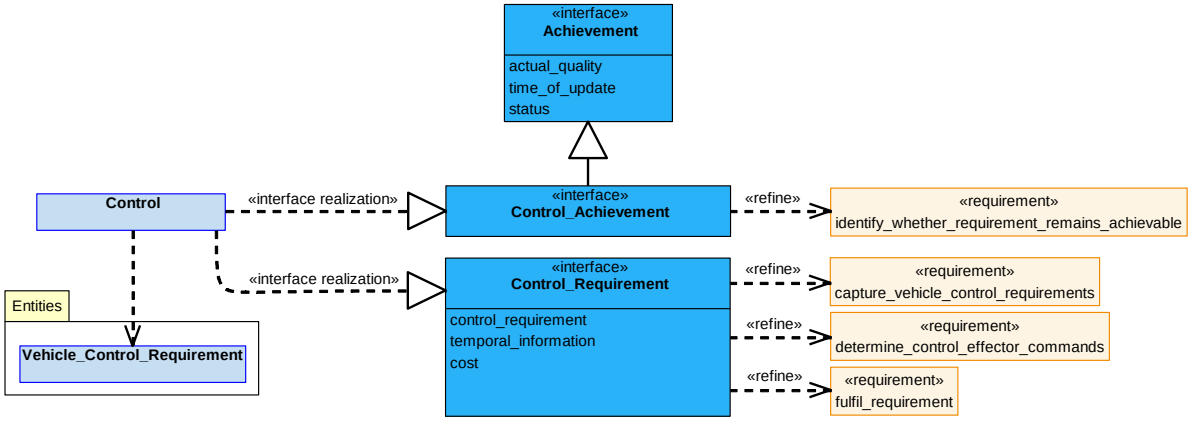

Figure 1291: Control Service Definition

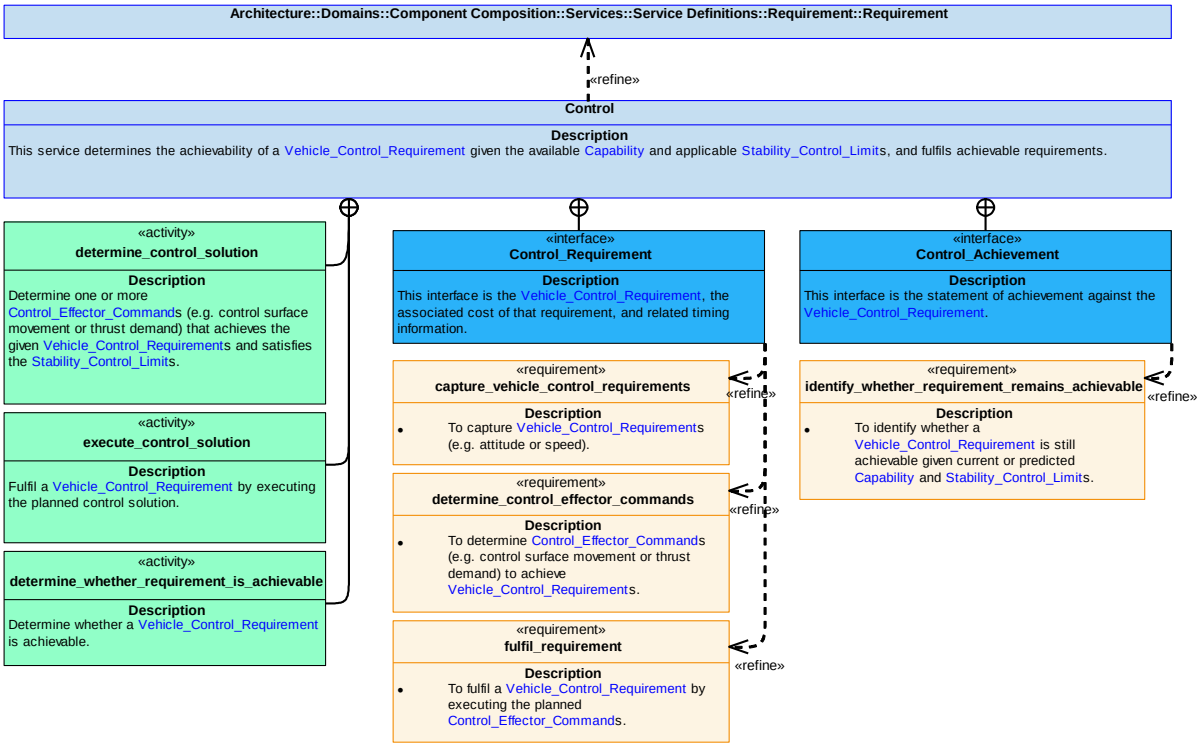

Figure 1292: Control Service Policy

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 speed 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.

B.2.80.7.1.2 Effector_Command

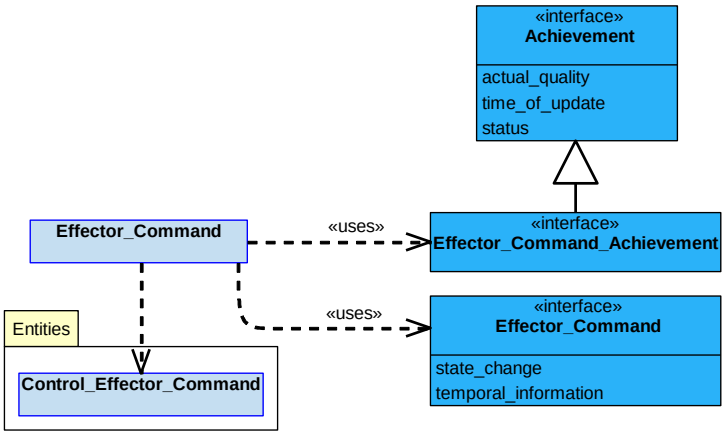

Figure 1293: Effector_Command Service Definition

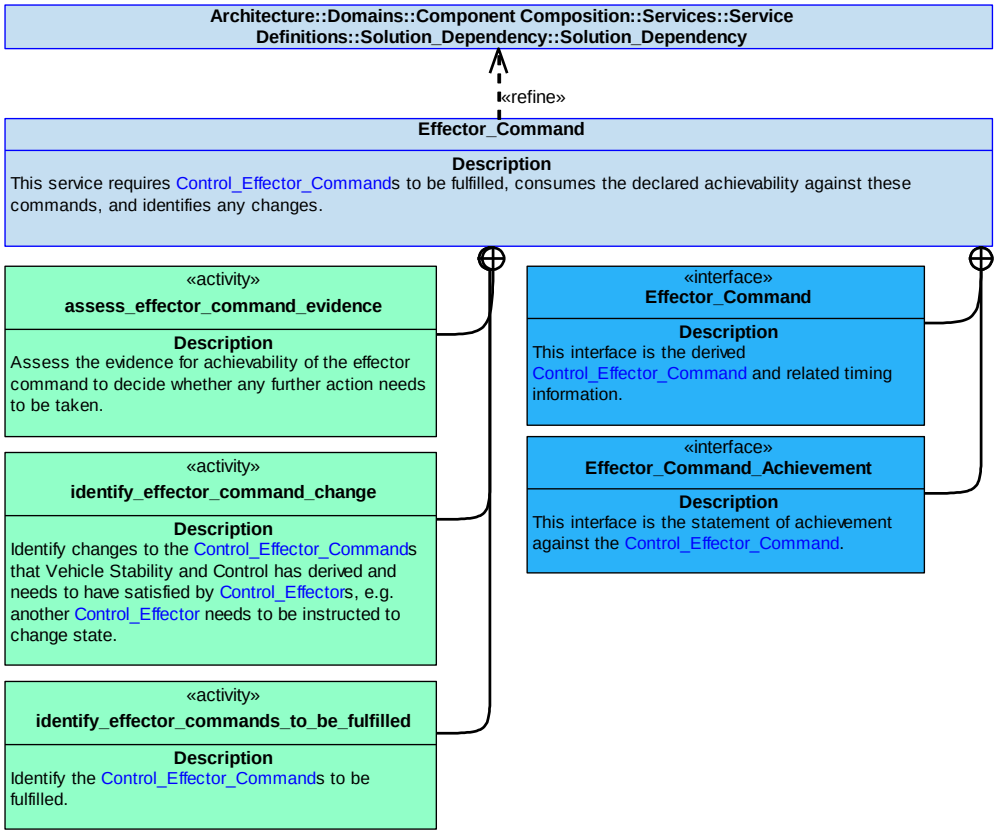

Figure 1294: Effector_Command Service Policy

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.

B.2.80.7.1.3 Constraint

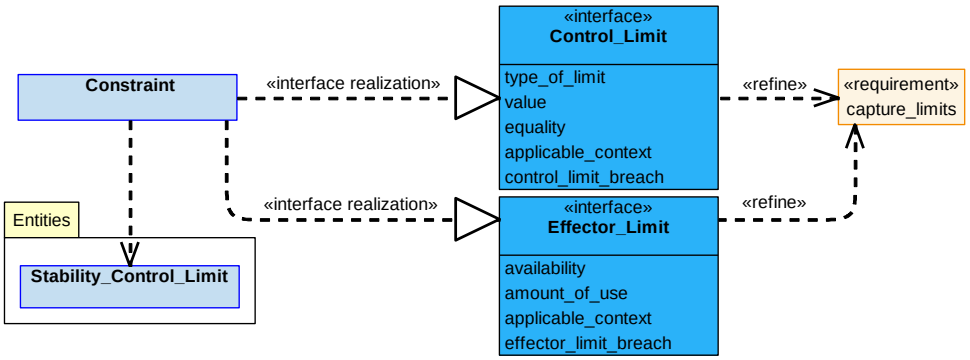

Figure 1295: Constraint Service Definition

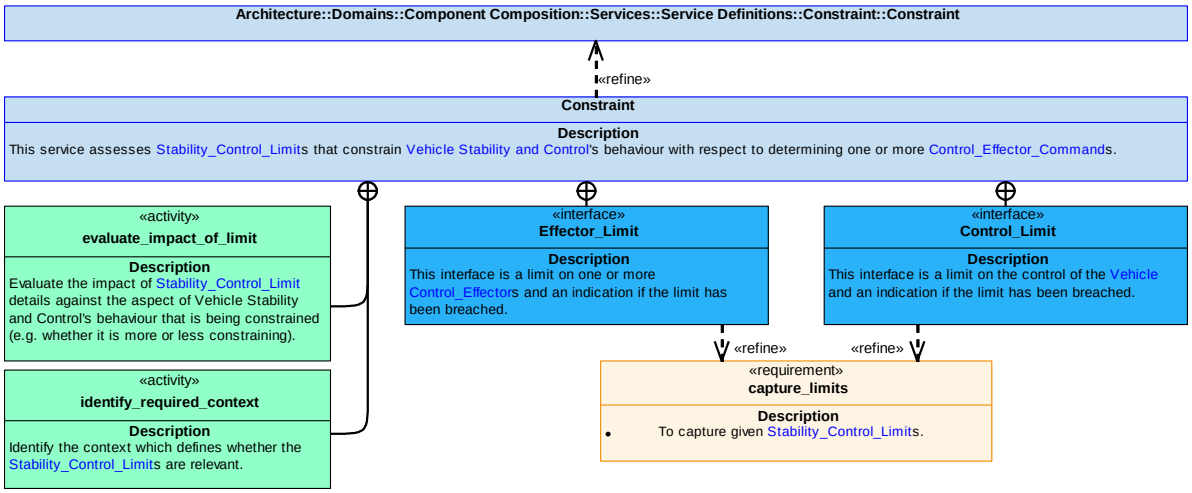

Figure 1296: Constraint Service Policy

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.
equality	The relationship between the value and the type of limit (e.g. less than, or equal to).
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.

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 available to be used or not.
amount_of_use	The utilisation of a particular Control_Effector, e.g. only 50% of a control surface can be moved to minimise observability of the Vehicle.
applicable_context	The context in which the limit is applicable.
effector_limit_breach	A statement that the 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.

B.2.80.7.1.4 Vehicle_Information

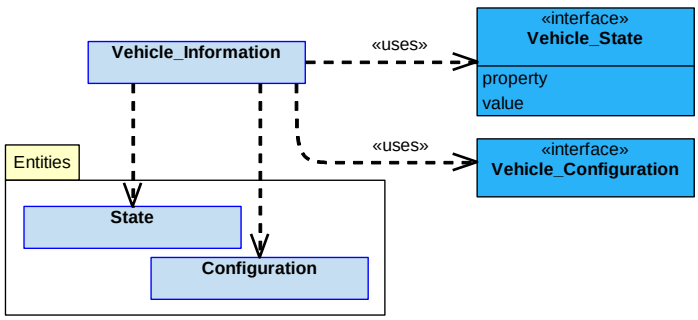

Figure 1297: Vehicle_Information Service Definition

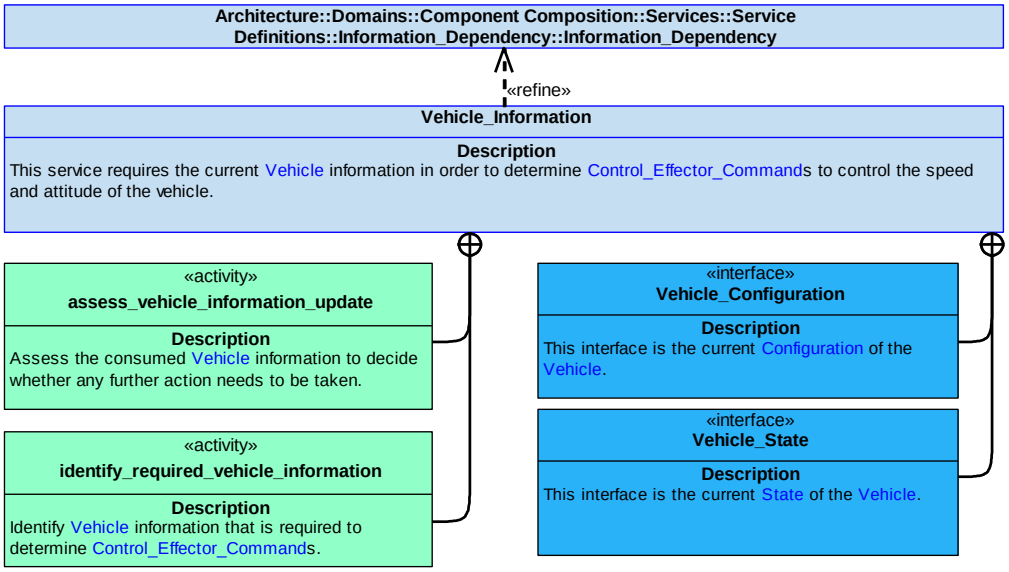

Figure 1298: Vehicle_Information Service Policy

Vehicle_Information

This service requires the current Vehicle information in order to determine Control_Effector_Commands to control the speed and attitude of the vehicle.

Interfaces

Vehicle_State

This interface is the current State of the Vehicle.

Attributes

property	The property relating to the Vehicle State (e.g. speed, pitch, roll, groundspeed or temperature).
value	The value of the property.

Vehicle_Configuration

This interface is the current Configuration of the Vehicle.

Activities

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.

B.2.80.7.1.5 Capability

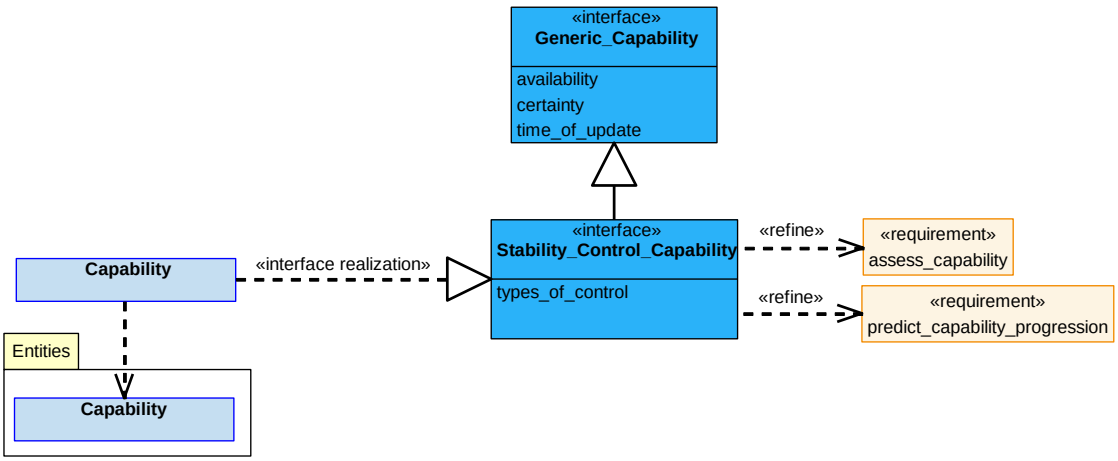

Figure 1299: Capability Service Definition

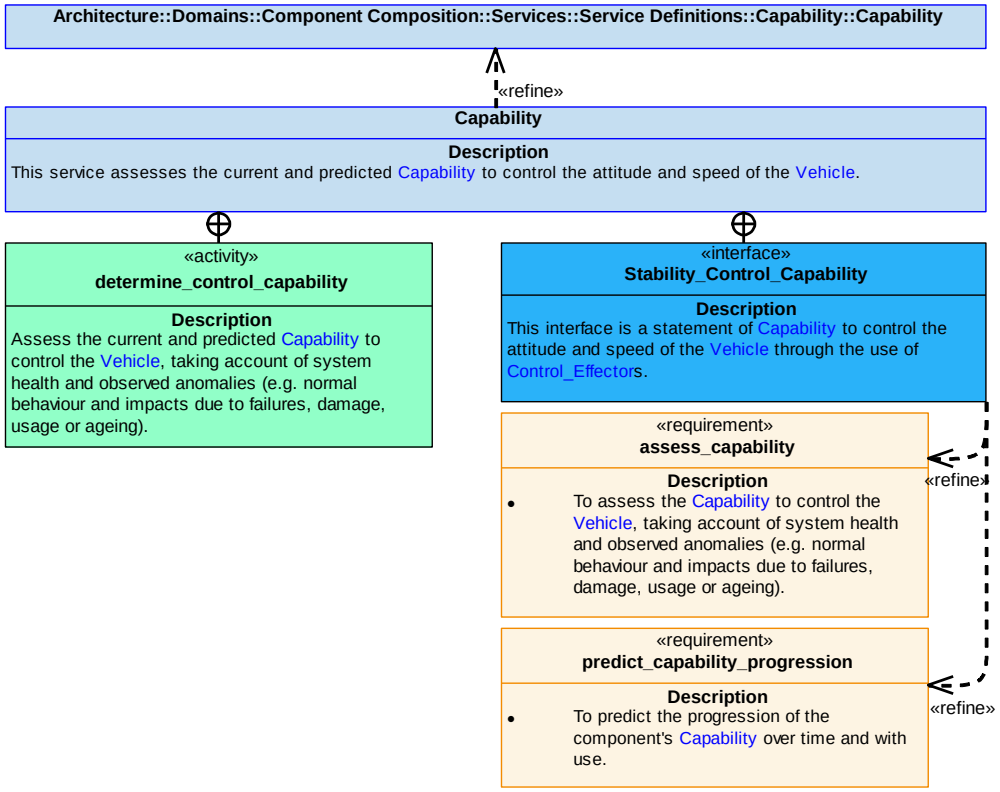

Figure 1300: Capability Service Policy

Capability

This service assesses the current and predicted Capability to control the attitude and speed of the Vehicle.

Interface

Stability_Control_Capability

This interface is a statement of Capability to control the attitude and speed of the Vehicle through the use of Control_Effectors.

Attribute

types_of_control	The different types of control that can be determined and issued, e.g. attitude or speed of the Vehicle.

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).

B.2.80.7.1.6 Capability_Evidence

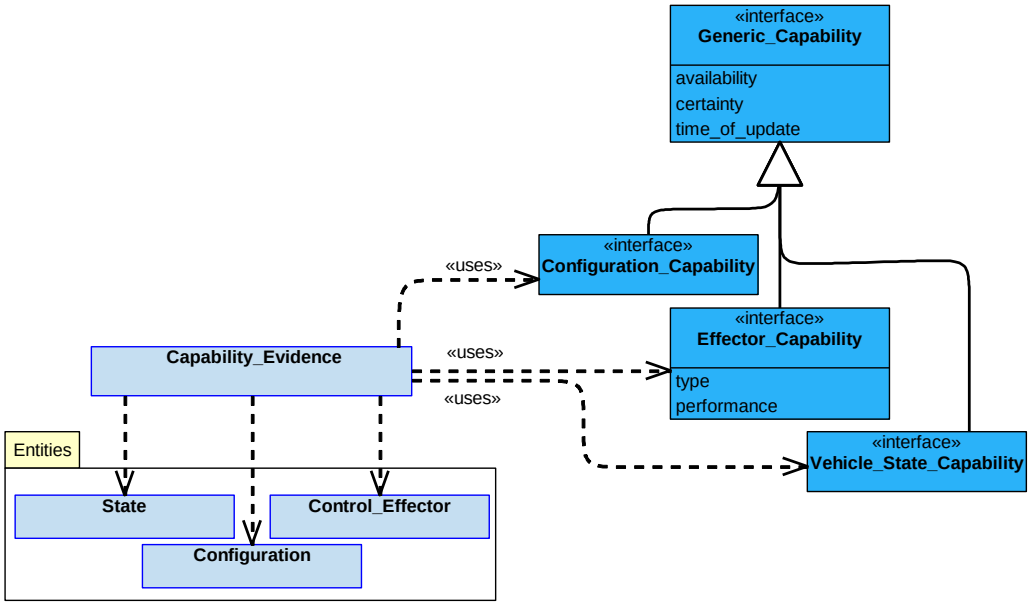

Figure 1301: Capability_Evidence Service Definition

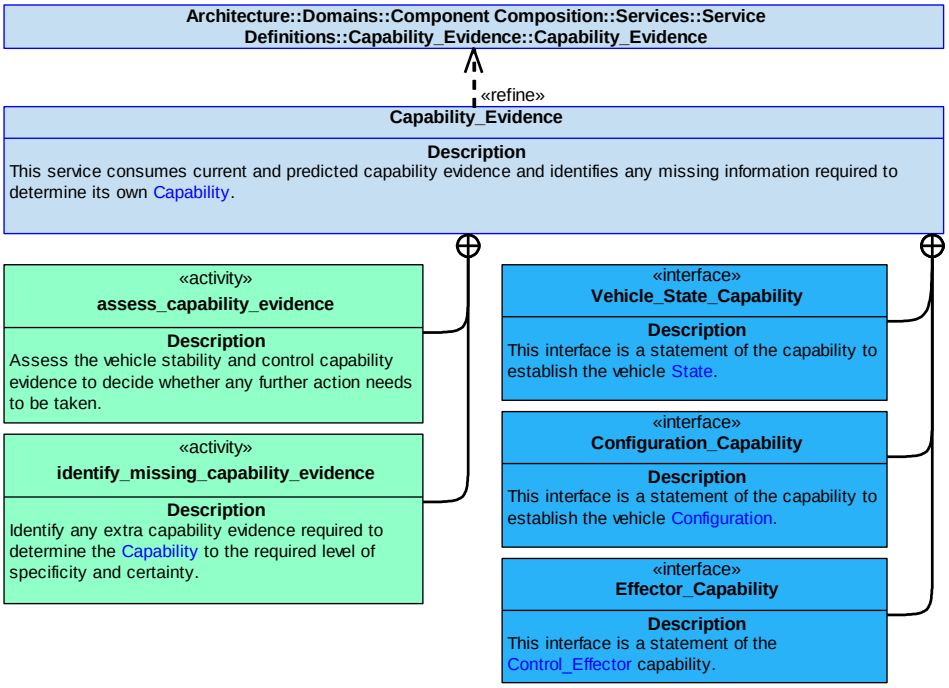

Figure 1302: Capability_Evidence Service Policy

Capability_Evidence

This service consumes current and predicted capability evidence and identifies any missing information required to determine its own Capability.

Interfaces

Configuration_Capability

This interface is a statement of the capability to establish the vehicle Configuration.

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.

Vehicle_State_Capability

This interface is a statement of the capability to establish the vehicle State.

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.

B.2.80.7.2 Service Dependencies

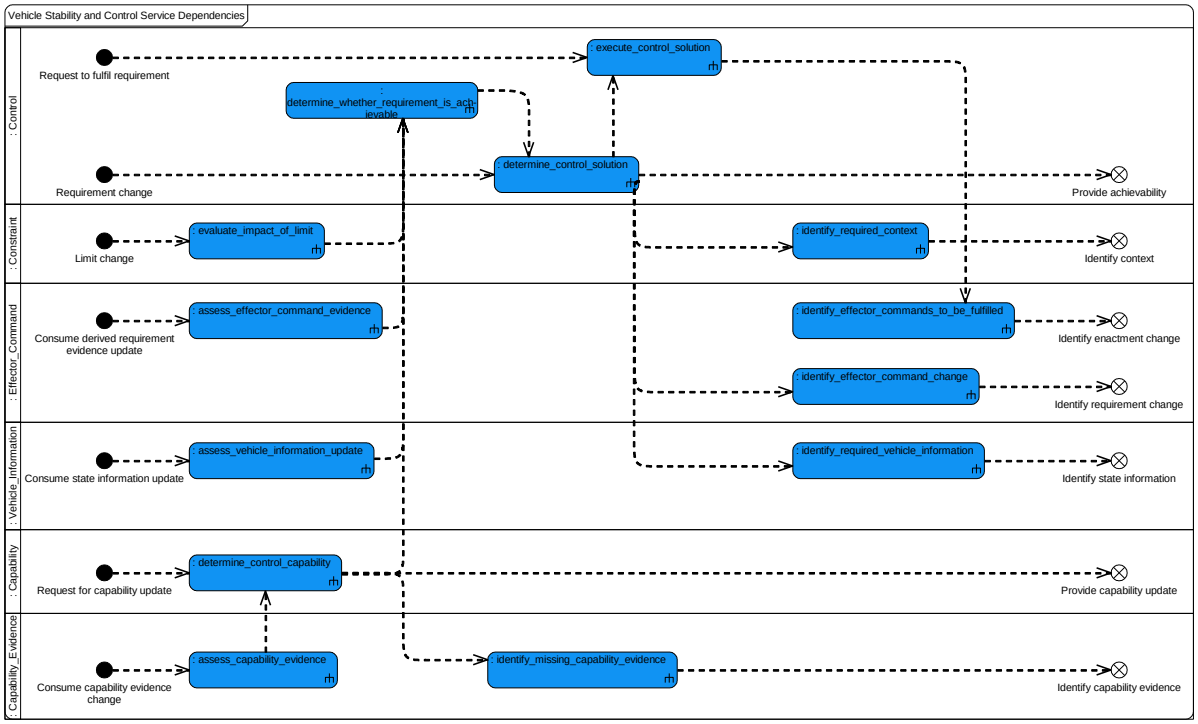

Figure 1303: Vehicle Stability and Control Service Dependencies

B.2.81 Weather

B.2.81.1 Role

The role of Weather is to determine the overall weather picture (including predicted weather based on received forecasts) in the operating environment.

B.2.81.2 Overview

Control Architecture

Weather is a service component as defined in the Control Architecture policy.

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.

B.2.81.3 Service Summary

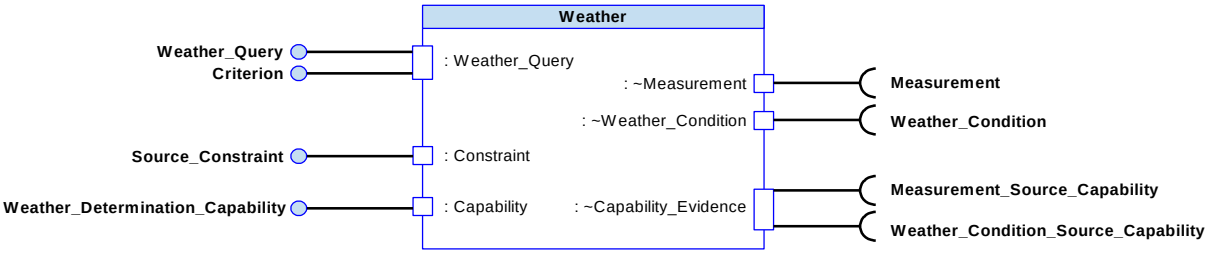

Figure 1304: Weather Service Summary

B.2.81.4 Responsibilities

determine_relevant_weather_information

- To determine the Weather_Picture at particular times (or time windows) and locations based on forecasts and Measurements.

capture_weather_condition_information

- To capture information on Weather_Conditions, e.g. forecasts.

capture_measurements

- To capture Measurements from which Weather_Conditions can be determined, e.g. inputs from sensors.

capture_weather_picture_requirements

- To capture provided Requirements for the provision of a Weather_Picture.

capture_constraints_for_weather_picture

- To capture Constraints relating to which Sources can be used for Weather_Types and Measurement_Types.

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.

determine_quality_of_weather_picture

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

capture_measurement_criteria

- To capture provided Measurement_Criterion/criteria for the Weather_Picture, e.g. confidence of prediction.

predict_capability_progression

- To predict the progression of the Capability to determine the overall Weather_Picture over time and with use.

B.2.81.5 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.

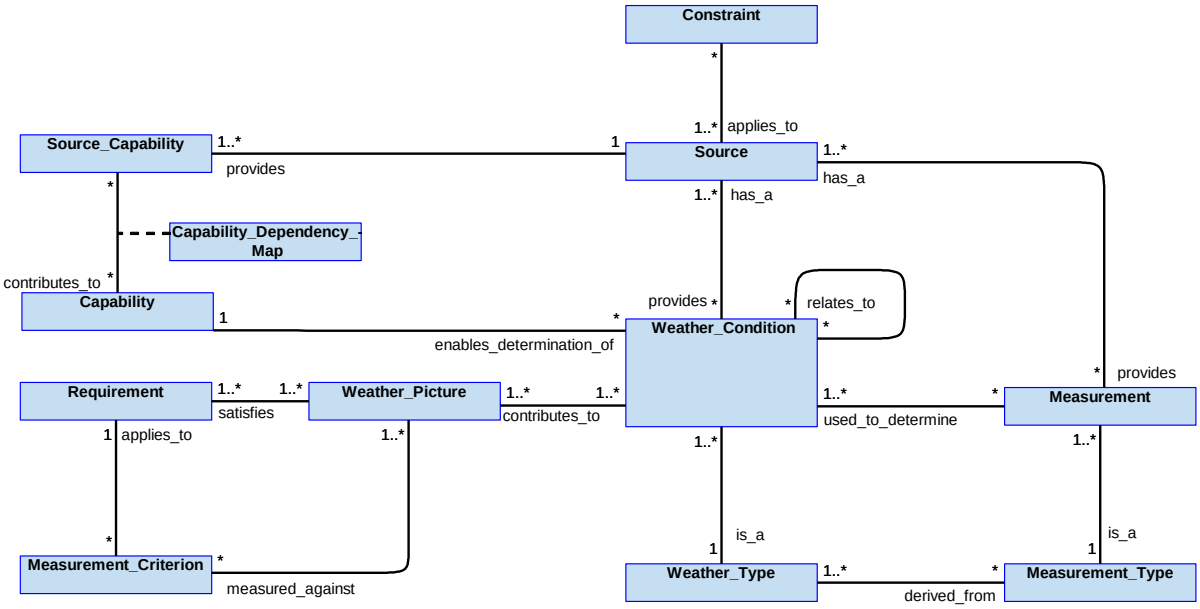

Figure 1305: Weather Semantics

B.2.81.5.1 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.

B.2.81.6 Design Rationale

B.2.81.6.1 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.

B.2.81.6.2 Design Considerations

Directly Applicable Policies

These policies 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.

Although this component provides an evolving view of its own capability the information sources used for this are not expected to be capable of providing multiple layers of capability evidence, therefore no functionality to identify additional required capability evidence is included.

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.

B.2.81.6.3 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.

B.2.81.6.4 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.

B.2.81.7 Services

B.2.81.7.1 Service Definitions

B.2.81.7.1.1 Weather_Query

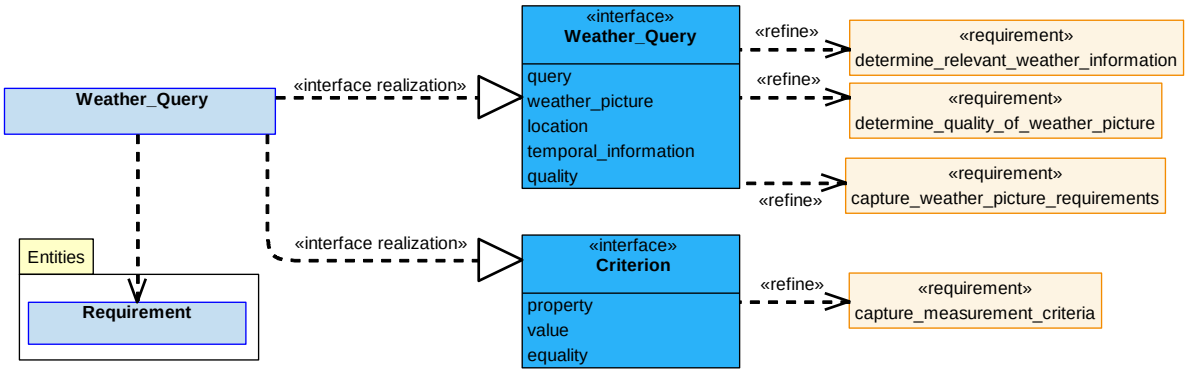

Figure 1306: Weather_Query Service Definition

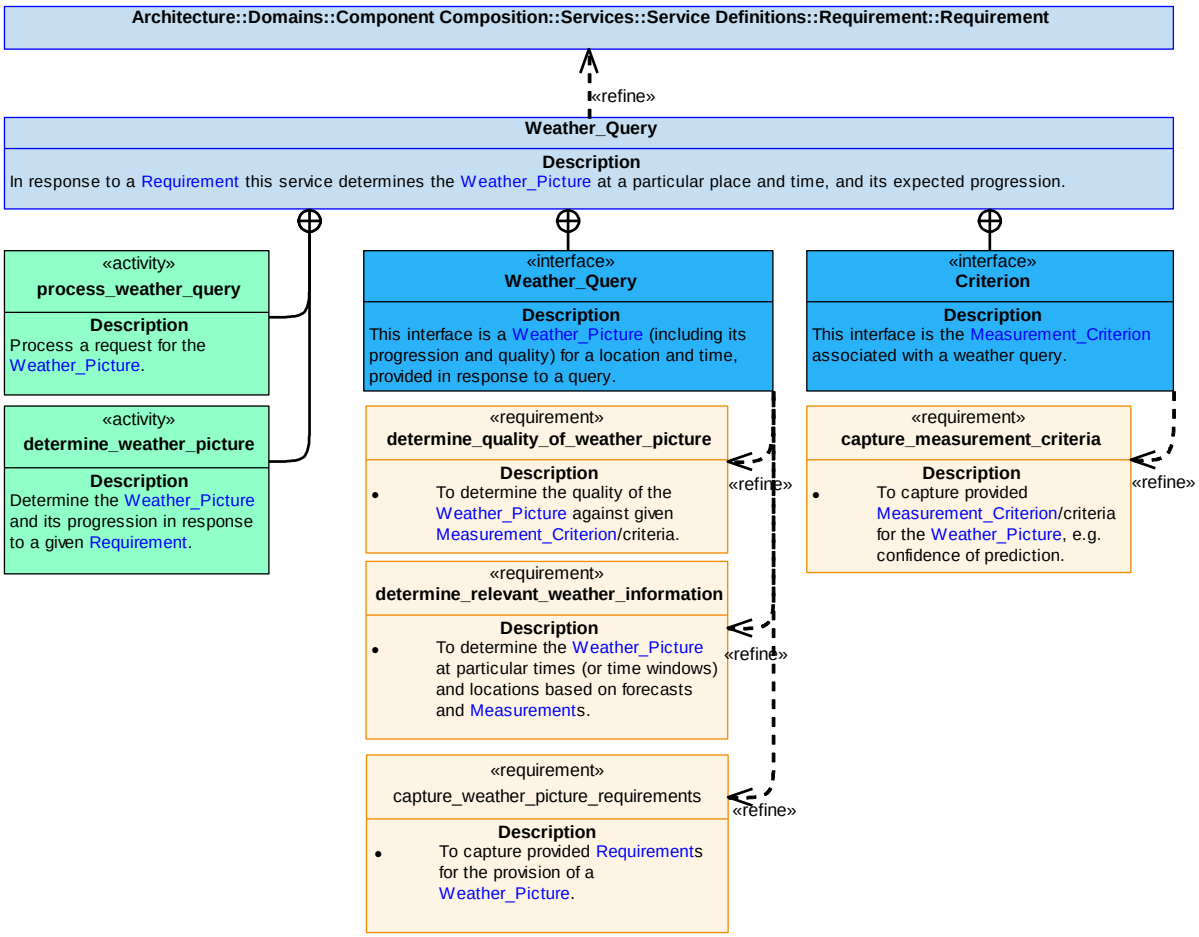

Figure 1307: Weather_Query Service Policy

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.

B.2.81.7.1.2 Measurement

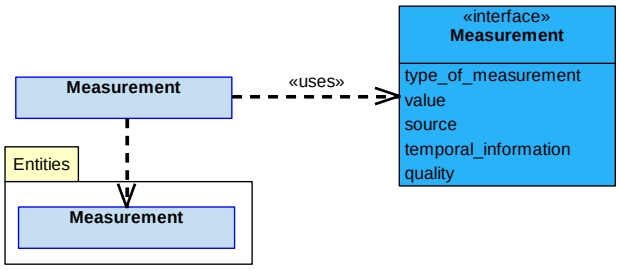

Figure 1308: Measurement Service Definition

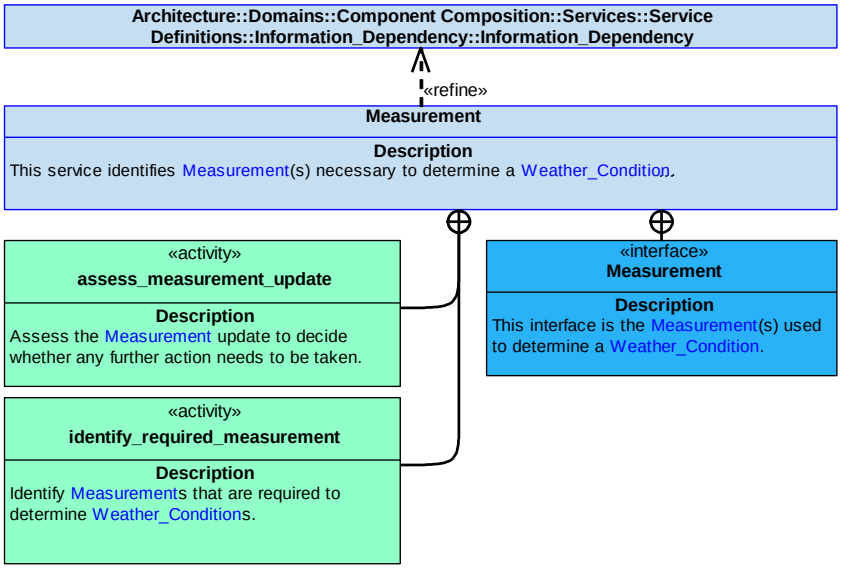

Figure 1309: Measurement Service Policy

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.

B.2.81.7.1.3 Weather_Condition

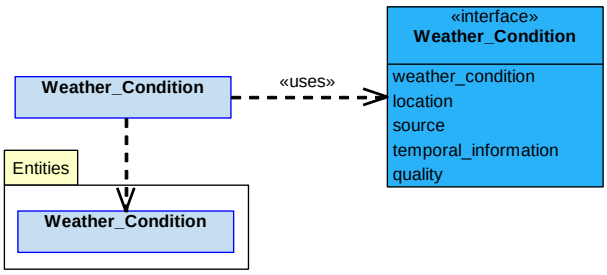

Figure 1310: Weather_Condition Service Definition

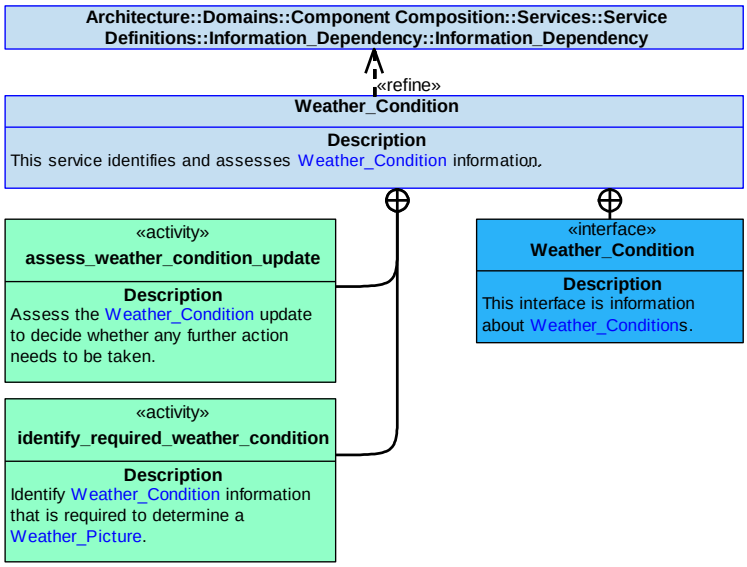

Figure 1311: Weather_Condition Service Policy

Weather_Condition

This service identifies and assesses Weather_Condition information.

Interface

Weather_Condition

This interface is information about Weather_Conditions.

Attributes

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.

B.2.81.7.1.4 Constraint

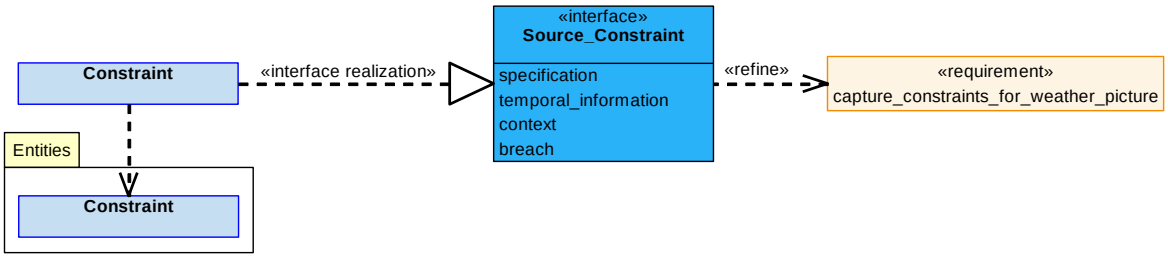

Figure 1312: Constraint Service Definition

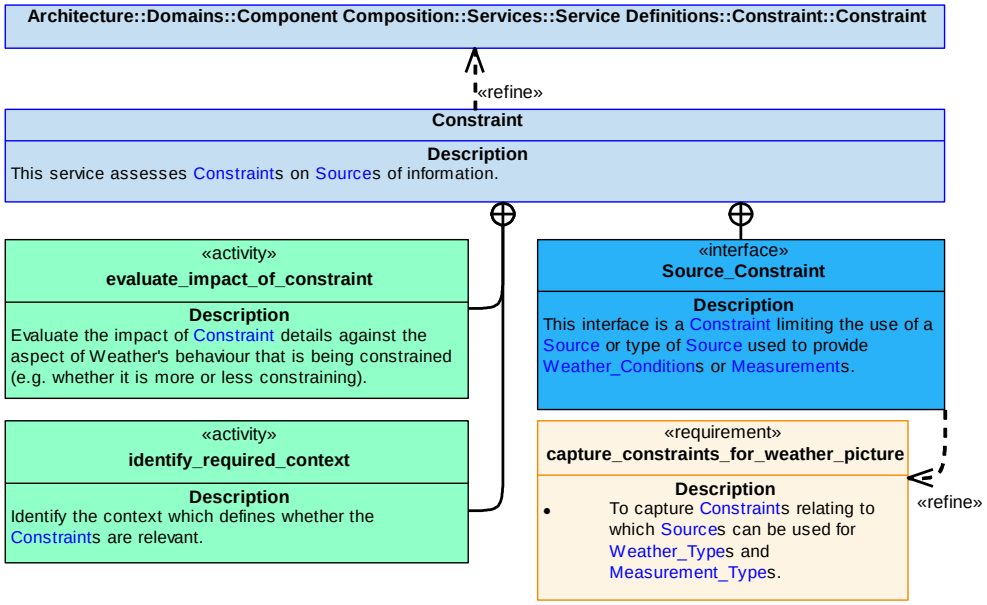

Figure 1313: Constraint Service Policy

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.

B.2.81.7.1.5 Capability

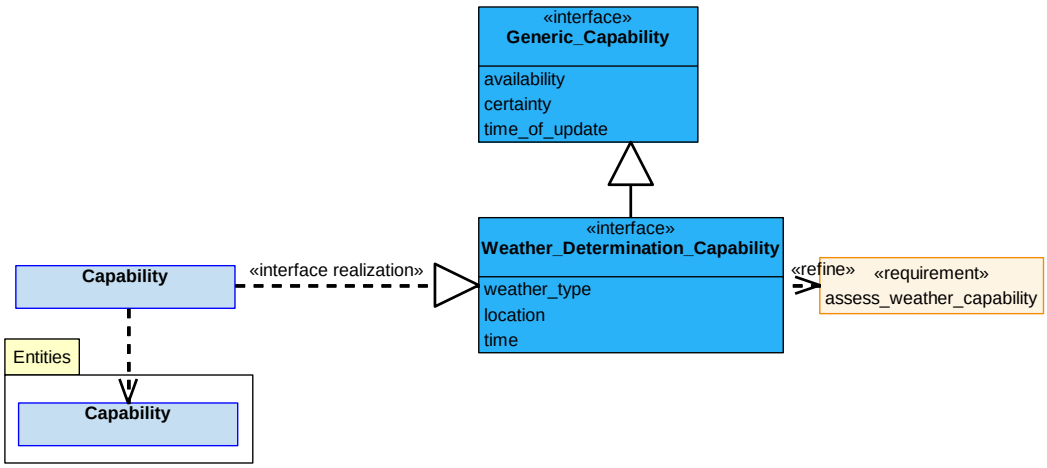

Figure 1314: Capability Service Definition

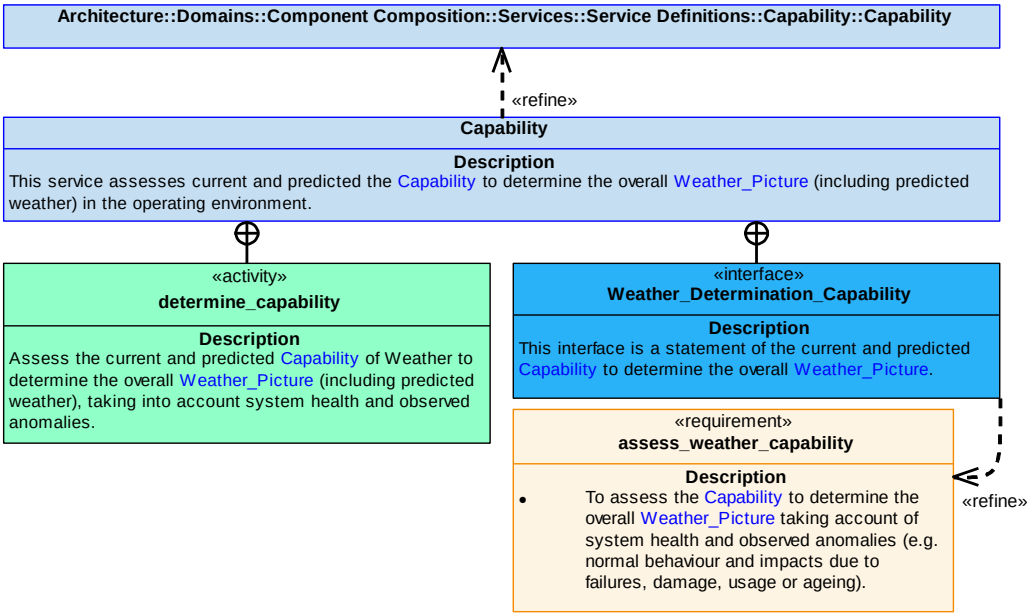

Figure 1315: Capability Service Policy

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.

B.2.81.7.1.6 Capability_Evidence

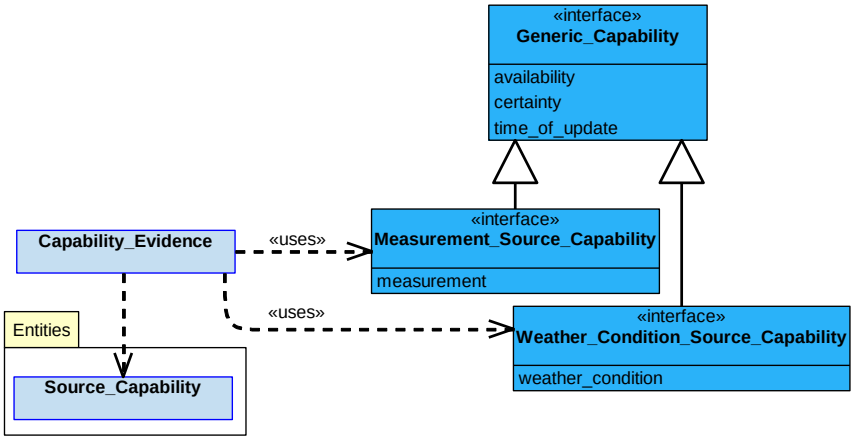

Figure 1316: Capability_Evidence Service Definition

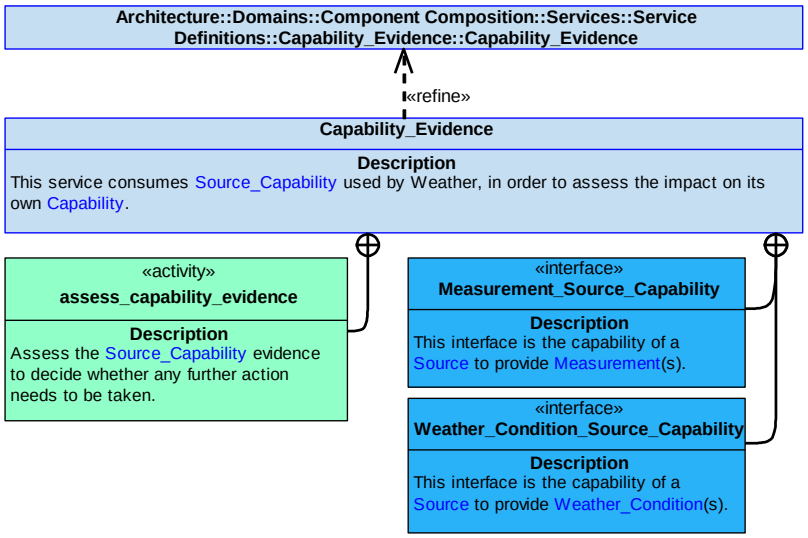

Figure 1317: Capability_Evidence Service Policy

Capability_Evidence

This service consumes Source_Capability 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.

B.2.81.7.2 Service Dependencies

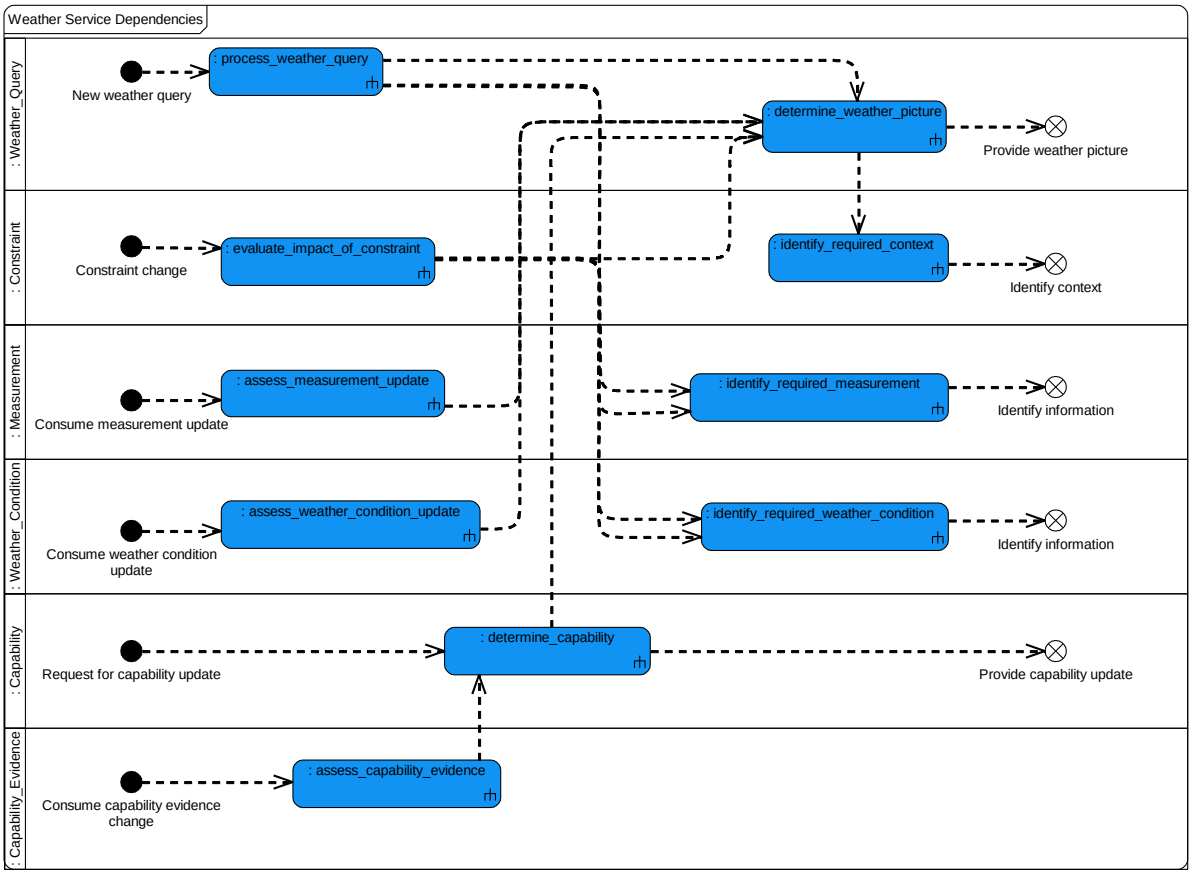

Figure 1318: Weather Service Dependencies

C Appendix C: Interaction Views

This appendix defines the PRA Interaction Views as introduced in section 4.

C.1 Vehicle Path Views

This section contains IVs relating to the path of an air vehicle and how it is controlled.

C.1.1 Path Execution

This IV describes how the execution of the planned path of an air vehicle is managed over different phases of that path (i.e. when the path is controlled by different path demands). The example uses the scenario of an air vehicle taking off, departing an airfield via a standard departure and performing a transit through controlled airspace, to demonstrate the path execution behaviour. The transit includes a search task as part of its requirements, as well as the receipt and implementation of ATC instructions.

It is assumed that:

- The system is capable of interpreting ATC instructions received via datalink and is approved to do so.

The following related areas of functionality are excluded:

- Obtaining the required clearances relating to flight in controlled airspace - obtaining authorisations.
- Coordination and control of the take-off.
- Control of the air vehicle in order to follow the determined flight path.
- Generation of routes.
- Voice communications with an ATC unit.
- Checking that the route and demanded path does not conflict with terrain, obstructions, unauthorised airspace, etc.

C.1.1.1 Pre-Conditions

- Airspace rules for the terminal operation area and transit airspace have been determined.
- Take-off profile requirements have been determined.
- Routing requirements for the departure and transit have been determined (e.g. appropriate SID and ATC instructions).
- Routing requirements for the search have been determined.
- Routes has generated a planned departure route.
- Routes has generated a planned transit/search route.
- Planned sequence of path demands (take-off manoeuvre, departure route and transit/search route) has been checked.
- Take-off has been initiated.

C.1.1.2 View

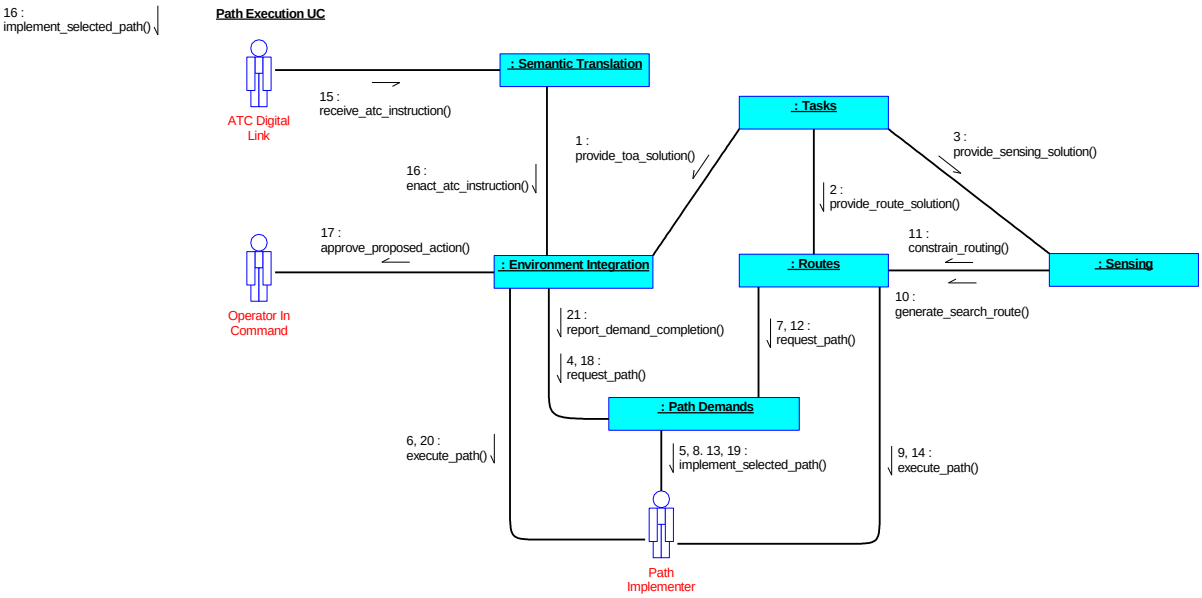

Figure 1319: Path Execution IV

Tasks requests Environment Integration, Routes and Sensing to deliver action solutions relating to take-off, routing (for departure, transit and search) and sensing respectively. Tasks manages the execution of these action solutions via coordination points (e.g. between the completion of take-off and the start of the departure routing, or the start of the sensing actions at the completion point in the transit).

Path Demands approves the execution of a path demand for the take-off profile, requested by Environment Integration. Executed by the Path Implementer, the air vehicle follows the provided path demand, as described in the Figure 1320: Take-Off IV.

On completion of the take-off profile, Path Demands approves the execution of a path demand for the planned departure route, commanded by Routes. Executed by the Path Implementer, the air vehicle follows the provided path demand, as described in the Figure 1323: Vehicle Movement In Air IV.

On completion of the departure route, Path Demands approves the execution of the path demand for the planned transit route, commanded by Routes. Executed by the Path Implementer, the air vehicle follows the provided path demand, as described in the Figure 1323: Vehicle Movement In Air IV.

Routes generates a path demand for the planned search route, using positioning requirements provided by Sensing. Path constraints in support of the sensing actions are provided by Sensing to Routes. On completion of the transit route, Path Demands approves the execution of this path demand for the planned search route, commanded by Routes. The air vehicle follows the provided path demand, as described in the Figure 1323: Vehicle Movement In Air IV. At the appropriate point in the search route, Sensing will terminate the sensing actions.

ATC instructions received via datalink that affect the execution of the air vehicle transit are processed by Semantic Translation, which decodes the instructions and provides them to Environment Integration to interpret and determine any required path changes. Any such path changes will be assessed and authorised by the Operator In Command prior to their enactment by the air vehicle and will include procedural elements such as voice back-up. The Operator In Command approves the proposed action, Path Demands approves the execution of a path demand from the interpreted ATC instructions commanded by Environment Integration. Implemented by the Path Implementer, the air vehicle executes the path changes and Path Demands reports completion of the demand request.

C.1.1.3 Post-Conditions

- Air vehicle has completed the transit through controlled airspace.
- Air vehicle has completed the search task.

C.1.1.4 Actors

ATC Digital Link

A digital link to an ATC unit for two-way communication (e.g. CPDLC). This includes the elements of the system that enable such communications.

Operator In Command

An authorised operator with the command responsibility for an air vehicle.

Path Implementer

A component responsible for implementing selected path demands.

C.1.2 Take-Off and Landing

C.1.2.1 Take-Off

This IV illustrates the process of coordinating and executing the transition of an air vehicle from ground to air. It covers the coordination of the conditions needed for take-off, performing the take-off manoeuvre and controlling the changes needed to the aircraft configuration after take-off.

It is assumed that:

- The Exploiting Platform is a conventional-take-off aircraft that takes off from a runway.
- This is a successful take-off of an air vehicle, with no need to abort.

The following related areas of functionality are excluded:

- The detailed execution of the take-off manoeuvre including provision/acquisition of the information needed to compensate for local weather conditions.
- The later stages of departure, and any interactions with Air Traffic Control that would be associated with this.
- Handover from local control to central control for the purposes of take-off.

C.1.2.1.1 Pre-Conditions

- The air vehicle is powered up and at the end of the runway, ready to take off.
- The air vehicle has received an operator instruction to take off.

C.1.2.1.2 View

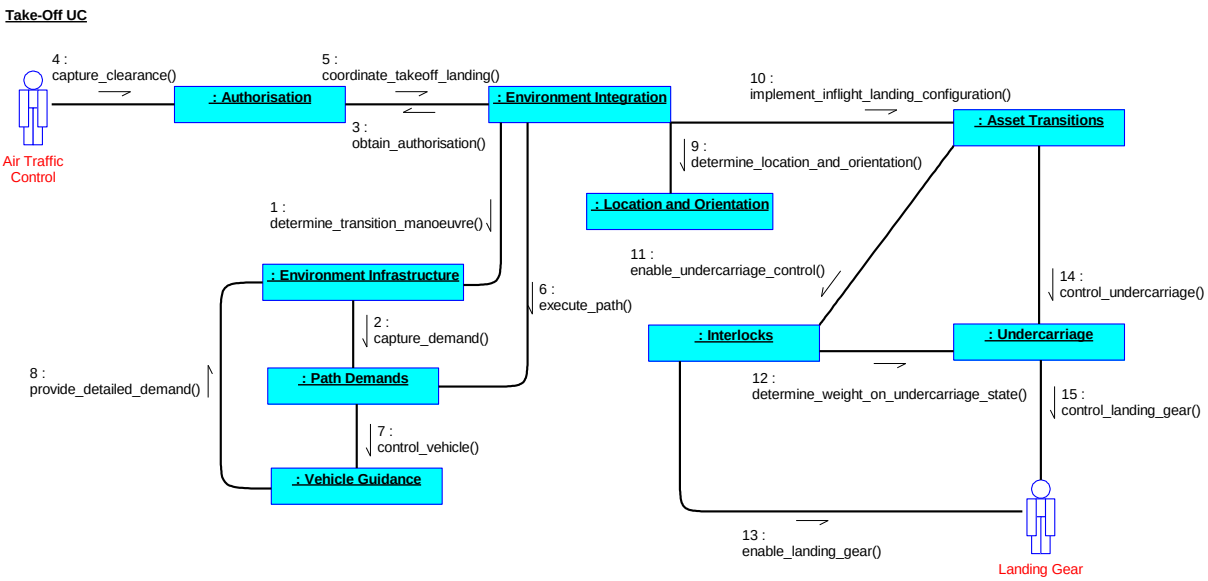

Figure 1320: Take-Off IV

Environment Integration begins coordination of the take-off by requesting Environment Infrastructure to determine the required take-off profile, the identity of which is passed to Path Demands. Environment Integration requests authorisation to commence take-off from Authorisation. Upon receiving take off clearance from Air Traffic Control, Environment Integration asks Path Demands to execute the take-off profile. Path Demands approves the execution of the take-off path and provides the identity of this to Vehicle Guidance. Vehicle Guidance, using detailed demand information provided by Environment Infrastructure, generates the take-off manoeuvre and controls the vehicle.

Once the necessary flight criteria are achieved, e.g. appropriate altitude, Environment Integration requests Asset Transitions to transition the vehicle to an in-flight configuration. Asset Transitions requests Interlocks to enable the control of the undercarriage. Undercarriage determines that the weight is off the undercarriage and Interlocks enables control. Asset Transitions requests that Undercarriage retracts the Landing Gear.

C.1.2.1.3 Post-Conditions

- The aircraft has taken off and is proceeding with departure.

C.1.2.1.4 Actors

Air Traffic Control

The external authority with responsibility for providing take-off and landing clearances.

Landing Gear

The landing gear of the undercarriage.

C.1.2.2 Landing

This IV illustrates the process of coordinating and executing the transition of an air vehicle from air to ground. It covers the coordination of the conditions needed for landing, performing the landing manoeuvre and controlling the changes needed to the aircraft configuration prior to landing.

It is assumed that:

- The Exploiting Platform is a conventional-landing aircraft that lands on a runway.
- This is a successful landing of the aircraft, with no need to abort the landing.

The following related areas of functionality are excluded:

- The detailed execution of the landing manoeuvre including provision/acquisition of the information needed to compensate for local weather conditions.
- Any need for an emergency landing.
- The earlier stages of approach, and any interactions with Air Traffic Control or additional manoeuvres that would be associated with this.
- Handover from central control to local control for the purposes of landing.

C.1.2.2.1 Pre-Conditions

- The aircraft has completed its approach and is ready to commence landing.
- The aircraft has received an operator instruction to land.

C.1.2.2.2 View

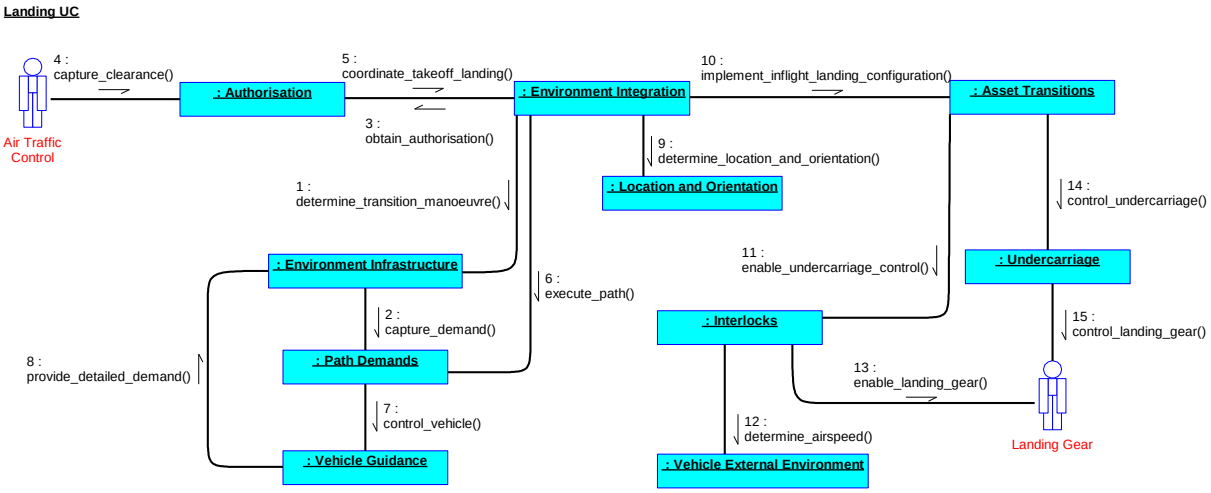

Figure 1321: Landing IV

Environment Integration begins coordination of the landing by requesting Environment Infrastructure to determine the required landing profile, the identity of which is passed to Path Demands. Environment Integration requests authorisation to commence landing from Authorisation. Upon receiving landing clearance from Air Traffic Control, Environment Integration asks Path Demands to execute the landing profile. Path Demands approves the execution of the landing path and provides the identity of this to Vehicle Guidance. Vehicle Guidance, using detailed demand information provided by Environment Infrastructure, generates the landing manoeuvre and controls the vehicle.

Once the necessary flight criteria are achieved, e.g. appropriate altitude, Environment Integration requests Asset Transitions to transition the vehicle to a touch-down configuration. Asset Transitions requests Interlocks to enable the control of the undercarriage. Interlocks uses Vehicle External Environment to determine that the aircraft is travelling at an appropriate speed to extend the undercarriage and enables control. Asset Transitions requests Undercarriage to extend the Landing Gear. Once the Landing Gear is extended, the landing profile completes with the vehicle touching down and reducing speed.

C.1.2.2.3 Post-Conditions

- The aircraft has landed safely and is ready to taxi.

C.1.2.2.4 Actors

Air Traffic Control

The external authority with responsibility for providing take-off and landing clearances.

Landing Gear

The landing gear of the undercarriage.

C.1.3 Routing

The mission requires a task to be performed by a vehicle that requires a route to visit a number of waypoints. Constraints (e.g. air volumes, terrain and speed restrictions) and limits (vehicle capability) apply to the generation of the route.

This scenario assumes a tactically benign environment, i.e. there are no threats present. It also assumes the presence of an independent route checker to provide a high integrity guarantee of no conflict with terrain, obstructions, unauthorised airspace, etc.

The following related areas of functionality are excluded:

- The reason for the route being requested.
- How the route is followed by a vehicle.
- Determination of the fuel cost for traversing the route.

C.1.3.1 Pre-Conditions

- A routing task has been received.

C.1.3.2 View

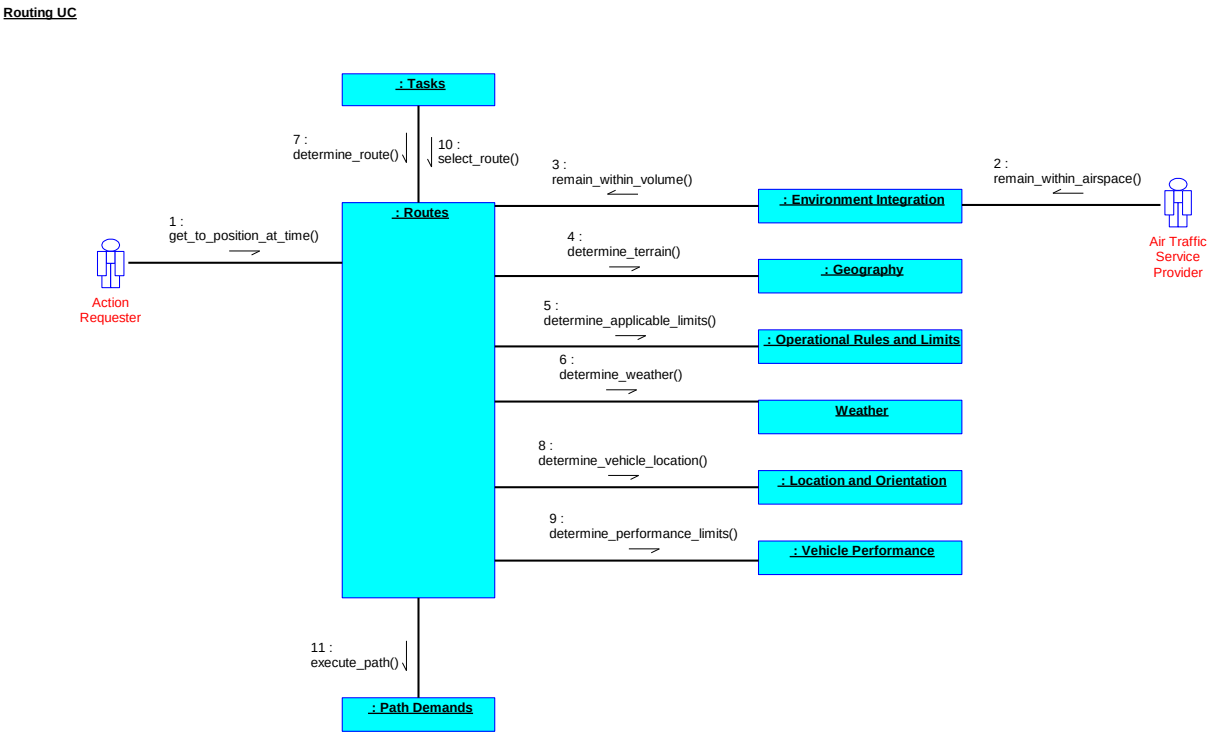

Figure 1322: Routing IV

Requirements are placed upon Routes from a number of sources:

- Tasks places requirements upon Routes to achieve a number of positions.
- Action Requester requests Routes to position the vehicle such that an action can be performed at a particular time.

Routes is made aware of a number of factors that will constrain the route including:

- Air volumes to remain within.
- Regions to avoid.
- Rules of engagement stating that supersonic flight is prohibited within a certain region.
- Weather patterns which may need to be avoided.

Tasks asks Routes to determine whether a route can be generated that meets all requirements within the constraints. Routes generates a route from the current location (as provided by Location and Orientation) to the required position, considering the vehicle movement capability (as provided by Vehicle Performance).

The achievability of the route is reported to Tasks. Tasks accepts the route for execution and the path identity is passed to Path Demands for execution.

C.1.3.3 Post-Conditions

- An achievable route that meets the positional requirements, constraints and is within limits has been selected.

C.1.3.4 Actors

Action Requester

A component that places requirements upon Routes so that actions can be performed at a particular time.

Air Traffic Service Provider

The ATS with responsibility for providing information and instructions required to comply with airspace rules and restrictions.

C.1.4 Vehicle Movement

C.1.4.1 Vehicle Movement in Air

This IV describes how a vehicle's movement is controlled whilst in the air (i.e. aviating), based on the receipt of a path demand or direct control demand.

It is assumed that:

- Air vehicle control is via actuator-driven control surfaces and a propulsion unit.
- All direct control feeds into Vehicle Guidance, rather than some direct control demands (e.g. pitch rate or thrust) feeding directly to Vehicle Stability and Control.
- There are no system health issues.
- In this example the thrust level is controlled by Vehicle Stability and Control. In other scenarios, e.g. climb or descent, thrust levels may be controlled by Vehicle Guidance.

The following related areas of functionality are excluded:

- Transition from flight to taxiing or vice-versa.
- The use of sensing and/or communicators in establishing a path demand.
- Determination of air vehicle location and orientation, and vehicle external environment parameters.
- Determination of vehicle performance limits.
- Mechanical Positioning control of Effectors.
- Collision prediction (this scenario is focussed on a 'time horizon' that does not require any collision prediction functionality, see the Figure 1328: Terrain Avoidance IV for a scenario that includes collision prediction and avoidance functionality).

C.1.4.1.1 Pre-Conditions

- Path Demands has been given appropriate arbitration rules.

C.1.4.1.2 View

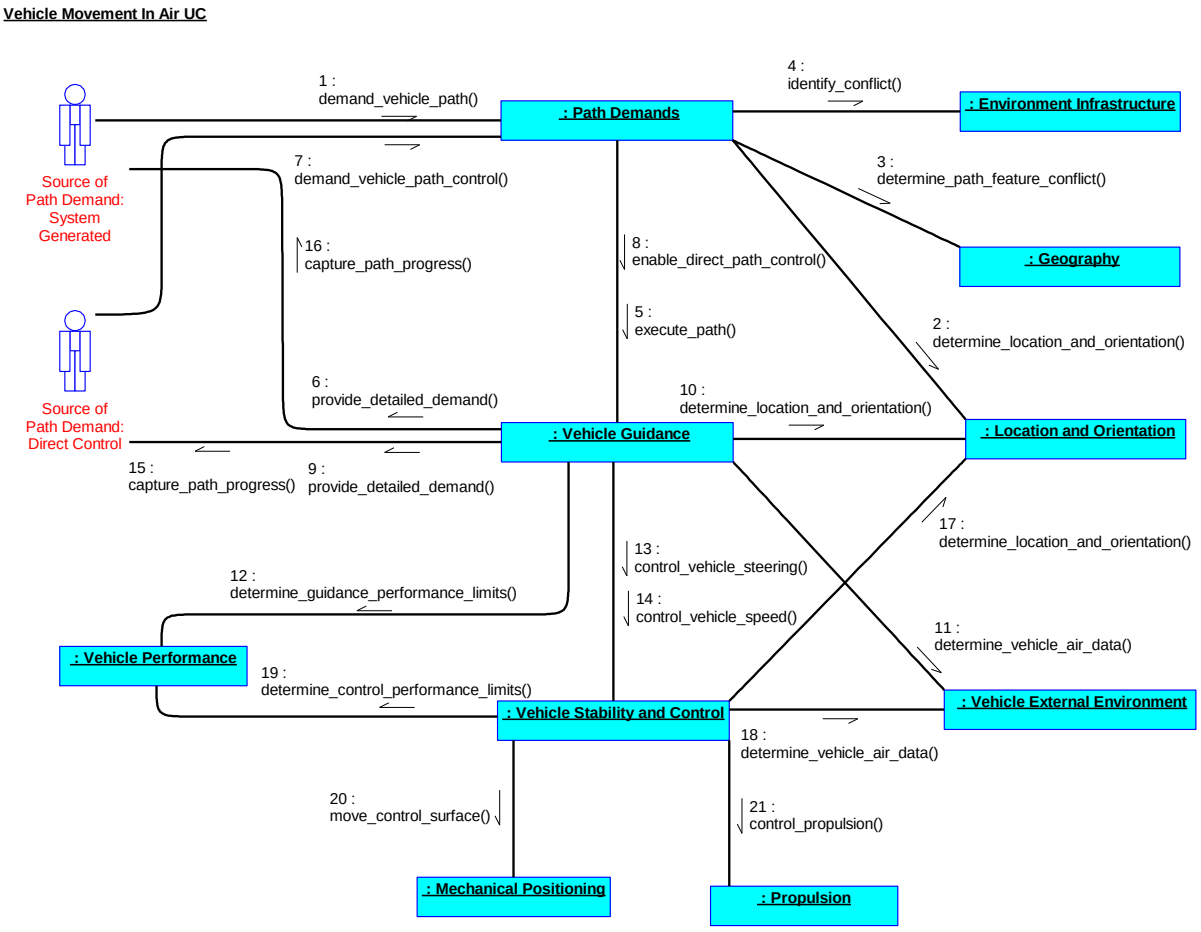

Figure 1323: Vehicle Movement In Air IV

A demand to alter the flight path of the air vehicle is received as either a path demand or a direct control demand, as follows:

If the demand is generated by a Source of Path Demand: System Generated:

- Path Demands asks Geography to determine whether the path demand will conflict with any geographical features using location information provided by Location and Orientation. Path Demands then asks Environment Infrastructure to identify any conflicts with infrastructure features (e.g. no fly zones). If the path demand is acceptable, Path Demands arbitrates between the requested path demand and all other concurrent (acceptable) path demands and any direct control requests to determine the source of the air vehicle trajectory demand to be used by Vehicle Guidance. Path Demands instructs Vehicle Guidance to implement the prioritised demands, passing Vehicle Guidance the identity of the demand. Vehicle Guidance then acquires the detailed path properties from the Source of Path Demand: System Generated.

If the request for direct control of the air vehicle path is generated by a Source of Path Demand: Direct Control:

- Path Demands arbitrates between the received control request and all other concurrent (acceptable) path demands to determine the source of the demand to be used by Vehicle Guidance. Path Demands instructs Vehicle Guidance to enable direct control and the associated demand is provided to Vehicle Guidance from the Source of Path Demand: Direct Control.

In both cases:

- Air vehicle location and orientation and air data parameters are determined by Location and Orientation and Vehicle External Environment respectively.
- Vehicle Guidance determines a trajectory to meet the prioritised demand, checking the current aerodynamic limits of the vehicle (represented by Vehicle Performance). Vehicle Guidance then generates air vehicle steering and speed demands to achieve the required trajectory, passing the instructions to Vehicle Stability and Control. Vehicle Guidance monitors achievement against the required trajectory which is reported to Path Demands and to the originator of the demand (Source of Path Demand: System Generated or Source of Path Demand: Direct Control).
- Vehicle Stability and Control generates air vehicle actuator and propulsion control demands to meet the required steering and speed demands, based on current air vehicle orientation, air data parameters and the current aerodynamic air vehicle control limits.
- Control surface actuators are controlled by Mechanical Positioning in order to execute the actuator control demands; similarly propulsion units are controlled by Propulsion in order to execute the propulsion demands.

C.1.4.1.3 Post-Conditions

- The air vehicle is aviating along the arbitrated path.

C.1.4.1.4 Actors

Source of Path Demand: Direct Control

An authorised operator that can generate a direct control demand via HMI. This demand includes both a request to select or cancel direct path control and the control demand values (which will affect the trajectory and/or speed of the air vehicle).

Source of Path Demand: System Generated

Part of the system that can generate a path demand.

Path demands include:

- Defined routes (planned, contingency, etc.).
- Avoidance manoeuvres (collision, terrain, weather, threats, etc.).
- Manoeuvre profiles (take-off, landing, etc.).
- Autopilot-type manoeuvres (acquire altitude, hold heading, etc.).

C.1.4.2 Vehicle Movement on Ground

This IV describes how movement of an air vehicle whilst on the ground (i.e. taxiing) is controlled, based on the receipt of a path demand or direct control demand.

It is assumed that:

- Air vehicle control is via actuator-driven controls (such as nose-wheel steering) and a propulsion unit (i.e. the vehicle is not being towed).
- The air vehicle is operating in a fixed-wing, wheels-on-ground taxiing environment only.
- The air vehicle is in a "safe-to-transmit" position/state allowing for remote communications.
- The air vehicle is untethered from any physical cabling and restraints and is free to move.
- All direct control feeds into Vehicle Guidance, rather than some direct control demands (e.g. required speed and steering) feeding directly to Vehicle Stability and Control.
- There are no system health issues.

The following related areas of functionality are excluded:

- Transition from flight to taxiing or vice-versa.
- The use of sensing and/or communicators in establishing a path demand.
- Determination of air vehicle location and orientation parameters.
- Determination of vehicle performance limits.
- Mechanical Positioning control of Effectors.
- Collision prediction (this scenario is focussed on a 'time horizon' that does not require any collision prediction functionality, see the Figure 1328: Terrain Avoidance IV for a scenario that includes collision prediction / avoidance functionality).

C.1.4.2.1 Pre-Conditions

- Path Demands has been given appropriate arbitration rules.

C.1.4.2.2 View

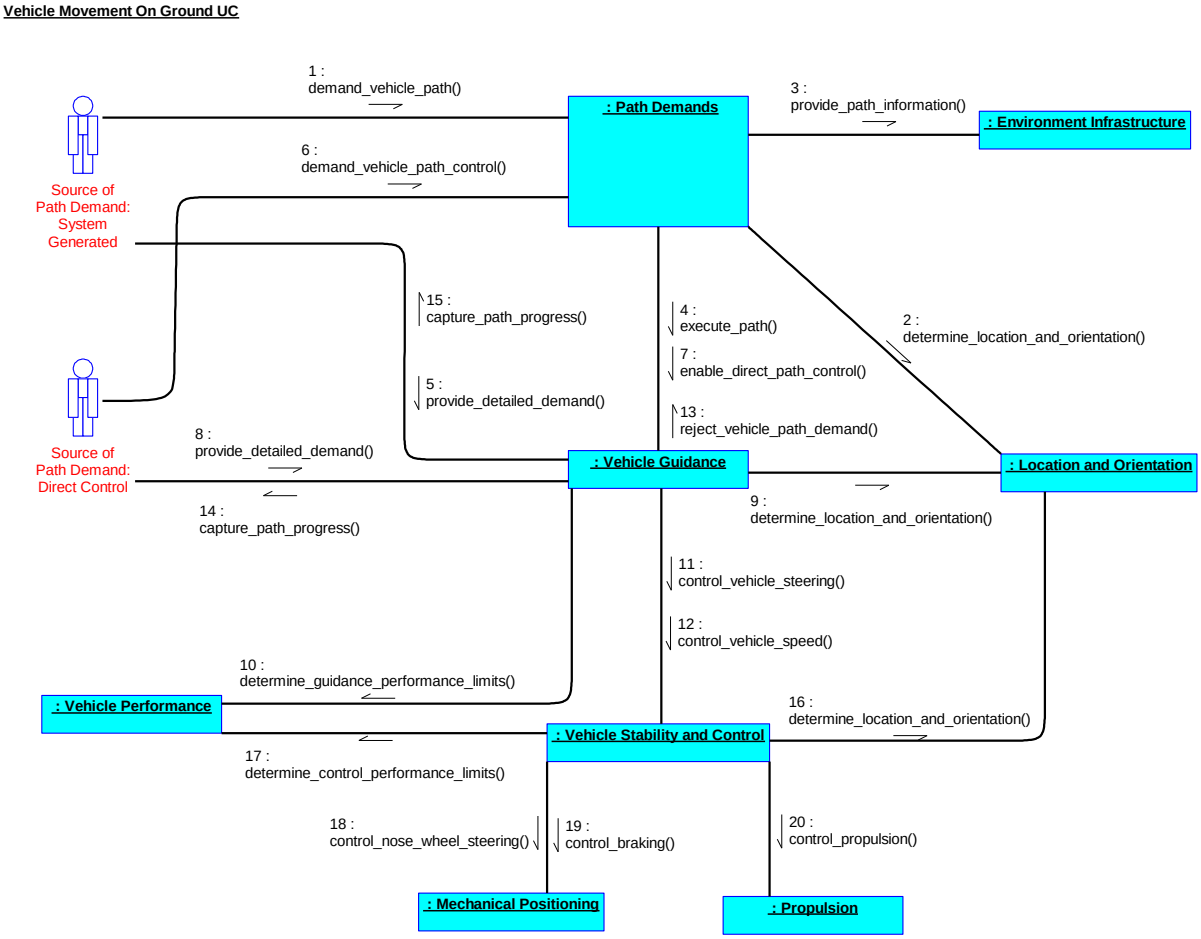

Figure 1324: Vehicle Movement On Ground IV

As with the Vehicle Movement In Air IV, a demand to alter the taxiing movement of the air vehicle is received either as a path demand (from a Source of Path Demand: System Generated) or a direct control demand (from a Source of Path Demand: Direct Control).

Path Demands requests Environment Infrastructure to provide information on the demanded path. Using the provided information, Path Demands determines that the demanded path is a valid path to enact.

Path Demands arbitrates between concurrent (safe) path demands and any direct control requests, as described in the Vehicle Movement In Air IV, and Vehicle Stability and Control and Vehicle Guidance will enact the selected path.

Mechanical Positioning will control taxiing-specific actuators (e.g. nose wheel steering and brakes) and propulsion units are controlled by Propulsion in order to execute the propulsion demands.

C.1.4.2.3 Post-Conditions

- The air vehicle is taxiing along the arbitrated path.

C.1.4.2.4 Actors

Source of Path Demand: Direct Control

An authorised operator that can generate a direct control demand via HMI. This demand includes both a request to select or cancel direct path control and the control demand values (which will affect the trajectory and/or speed of the air vehicle).

Source of Path Demand: System Generated

Part of the system that can generate a path demand.

Path demands include:

- Defined routes (planned, contingency, etc.).
- Avoidance manoeuvres (collision, terrain, weather, threats, etc.).
- Manoeuvre profiles (take-off, landing, etc.).
- Autopilot-type manoeuvres (acquire altitude, hold heading, etc.).

C.1.5 Vehicle Performance

This use case describes how the current vehicle performance limits are determined so that Vehicle Guidance can meet a demand to fly at the maximum airspeed.

It is assumed that:

- Maximum airspeed is a function of altitude, external temperature, vehicle mass, position of the undercarriage and the external role fit configuration. In reality, other inputs may also be required, but this subset is used to describe the possible interactions.

The following related areas of functionality are excluded:

- Consumption of source data for the determination of Vehicle External Environment parameters.
- Controlling the trajectory of the air vehicle.
- Determining the current inventory of the air vehicle.
- Determining a vehicle performance parameter at some point in the future - i.e. when the current conditions do not apply. In this case it is expected that a component requesting this information would define the conditions applicable to the determination of the particular performance parameter (e.g. undercarriage position, vehicle mass or temperature).

C.1.5.1 Pre-Conditions

- The air vehicle is airborne.

C.1.5.2 View

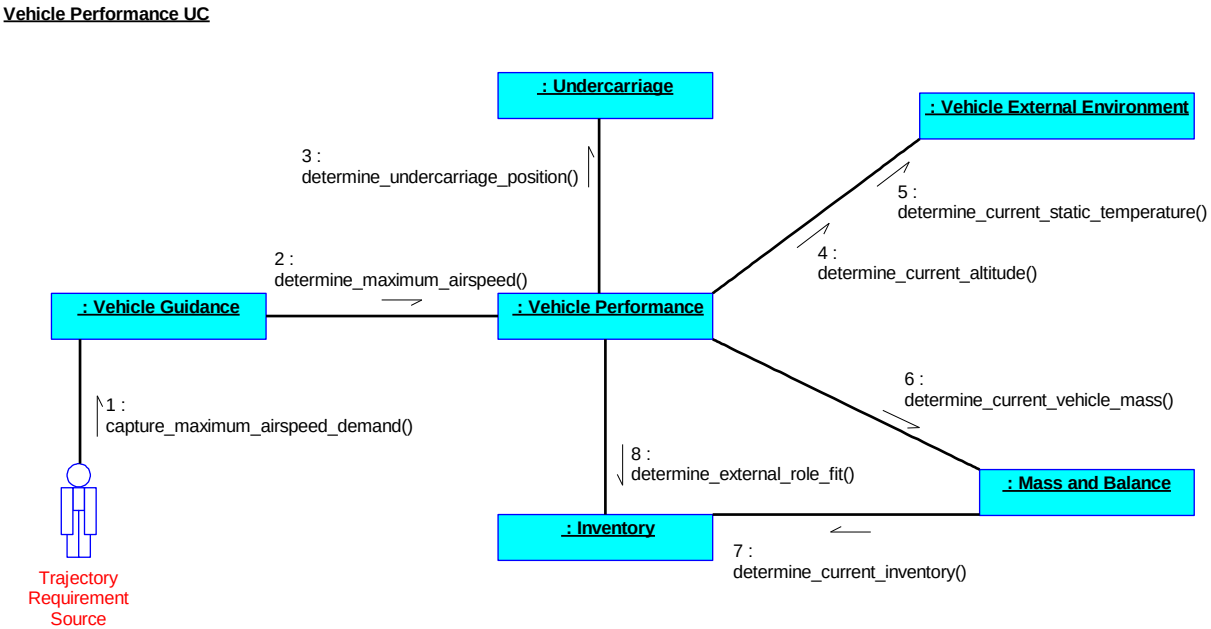

Figure 1325: Vehicle Performance IV

A demand to fly at the maximum airspeed of the air vehicle is received from the Trajectory Requirement Source by Vehicle Guidance.

Vehicle Performance determines the maximum airspeed for the current conditions based on undercarriage position (determined by Undercarriage), the current altitude and static temperature (determined by Vehicle External Environment), the current vehicle mass (determined by Mass and Balance based on the current inventory (determined by Inventory)) and the current external role fit items (determined by Inventory).

C.1.5.3 Post-Conditions

- The Vehicle Guidance component knows the maximum airspeed for the current conditions and so can maintain the air vehicle within the vehicle performance limits.

C.1.5.4 Actors

Trajectory Requirement Source

The source of a trajectory requirement.

C.2 Vehicle Environment Views

This section contains IVs relating to the environment in which an air vehicle is operating.

C.2.1 Airspace Integration

This IV illustrates activities for integrating into controlled airspace:

- The use of IFF and SSR transponders to provide identification and location to other airspace users and ATS.
- Communicating with ATS to request and obtain clearances, in order to comply with airspace restrictions, and to receive instructions upon approaching a controlled airspace boundary.
- Implementing changes to communications and identification when transitioning between ATS controllers.

This IV assumes:

- ATS will initiate the transfer from one air traffic control unit to another when the vehicle approaches the boundary between control areas.
- There is a need to squawk a new identification code upon crossing the controlled airspace boundary into a new airspace zone.

The following considerations are excluded:

- Detection of potential collisions and subsequent determination of an avoidance manoeuvre.
- Determining and following a route.
- Communications between the authorised operator and ATS.
- Authorised operator interactions with the HMI.
- Set-up of requirements to facilitate a new communications channel and determining that the new channel is permitted.
- Initial processing and semantic translation of ATS instructions.

C.2.1.1 Pre-Conditions

- The system is pre-loaded with transponder codes and ATS communications frequencies.

C.2.1.2 View

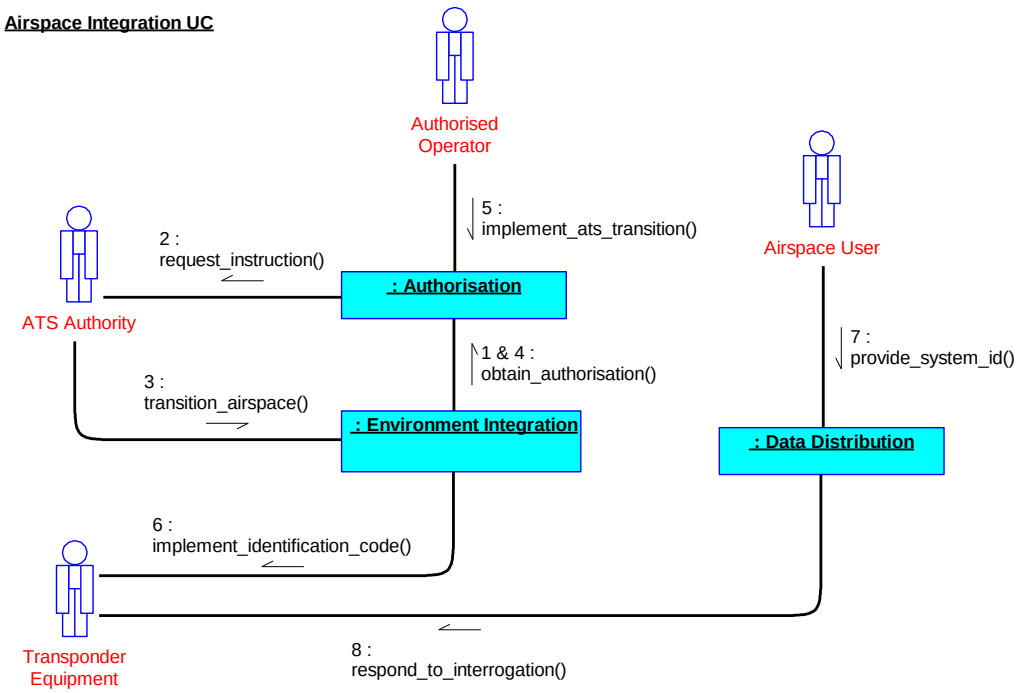

Figure 1326: Airspace Integration IV

Whilst transiting through controlled airspace, Environment Integration requests authorisation for required deviations from the cleared route (e.g. due to weather), which Authorisation obtains by requesting clearance from ATS.

Upon approaching the boundary between control areas, ATS initiates a handover of the air vehicle from the releasing air traffic control unit to a receiving air traffic control unit. Environment Integration interprets the handover instructions from ATS to determine the system setup (e.g. new radio channel or IFF codes) required to comply with them. Environment Integration requests authorisation to enact the communications and identification changes received from ATS for the transition, which Authorisation obtains by requesting approval from an Authorised Operator with an appropriate role. Environment Integration requests the transponder equipment to implement the new identification code received from ATS. The transponder uses this code when responding to interrogations from an Airspace User received by Data Distribution.

C.2.1.3 Post-Conditions

- The system has transitioned from one area of controlled airspace to another.

C.2.1.4 Actors

Airspace User

Another airspace user to whom identification information must be provided. This includes other aircraft and ATS.

ATS Authority

The external authority with responsibility for providing information and instructions required to comply with airspace rules and restrictions.

Authorised Operator

An authorised operator within a system with the authority to approve ATS instructions.

Transponder Equipment

A transponder.

C.2.2 Avoidance

C.2.2.1 Cooperative Air Collision Avoidance

This IV describes co-operative collision avoidance between ownship and another air vehicle. Collision Prediction uses vehicle trajectories to identify potential conflicts. In addition, Environment Integration coordinates the provision of information to other airspace users about the location, heading and speed of ownship. When a potential conflict is identified, Collision Avoidance determines an appropriate manoeuvre to avoid the collision.

In the case of ACAS, following a Resolution Advisory (RA), Collision Avoidance generates an avoidance manoeuvre, coordinating with the intruder according to published ACAS logic. Where the intruder has a transponder but no ACAS, only ownship will manoeuvre.

This use case describes how the flight path of the vehicle is adapted to cater for the presence of an intruder in the protected volume. The use case described in this IV is for an airborne example. Similar principles would apply for ground based avoidance.

It is assumed that:

- Collision Prediction and Collision Avoidance are compliant with the ACAS requirement (minimum ACAS II), and avoidance is achieved entirely through the use of ACAS.
- The avoiding action is taken autonomously.
- Received transponder information will not need tactical processing for collision avoidance purposes.
- The Path Demand Arbitrator will arbitrate all path demands placed upon it, taking into account task priorities provided by Tasks. It will therefore be possible for the Path Demand Arbitrator to de-prioritise or ignore the demand for manoeuvre to avoid the potential infringement. It is assumed that this example is occurring during peace time and that an immediate collision avoidance manoeuvre has priority.

The following considerations are excluded from this Use Case:

- Tactical interactions.
- Sensing interactions.
- HMI interactions.
- Use of the information to create or update tracks for purposes other than collision avoidance.
- Human control of the air vehicle and human interactions with Air Traffic Services (ATS) or proximate traffic, etc.
- Recovery to the required route and clearance following the avoidance manoeuvre.
- The use of the Trajectory Prediction component since avoidance is achieved, in this scenario, through the use of ACAS.

C.2.2.1.1 Pre-Conditions

- The air vehicle is airborne and allocated to an air traffic controller.
- The EMCON restrictions under which the air vehicle is operating permit the appropriate hardware to transmit.

C.2.2.1.2 View

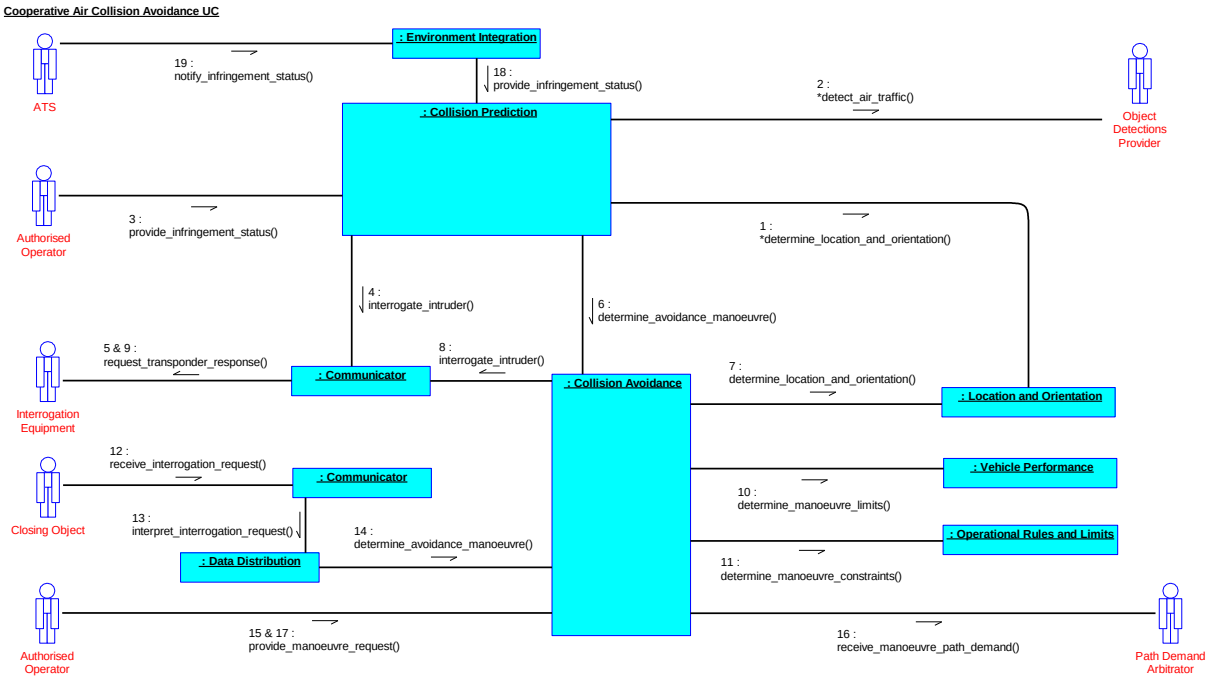

Figure 1327: Cooperative Air Collision Avoidance IV

Collision Prediction constantly monitors ownship position and predicted movement in relation to other objects in the environment. Collision Prediction acquires ownship position and velocity vector from Location and Orientation. Collision Prediction receives notification of other objects in the environment from the Object Detections Provider. As a result, Collision Prediction determines that a Closing Object will infringe the TA region of the protected volume, thus becoming an intruder. Collision Prediction therefore alerts the Authorised Operator to the presence of the Closing Object via a TA.

When the intruder enters the RA region, Collision Avoidance interrogates the intruder and generates an RA alert to coordinate an avoidance manoeuvre with the intruder. This manoeuvre is based on ownship information from Location and Orientation, with the manoeuvre (climb or descend) being provided in the interrogation via Communicator (based on the available position information of both the aircraft, including height above ground, speed and heading). Applicable constraints from Vehicle Performance and Operational Rules and Limits (e.g. to avoid breaching limits) may influence the ownship manoeuvre.

If the intruder initiates the manoeuvre first, this information is received by Communicator via an interrogation request, then interpreted by Data Distribution and passed to Collision Avoidance to trigger the coordinated manoeuvre.

The Authorised Operator is alerted to the proximity of the aircraft via an RA, from Collision Avoidance.

Collision Avoidance informs Path Demand Arbitrator of the manoeuvre requirement. The manoeuvre is performed, resulting in collision avoidance.

Collision Avoidance notifies the Authorised Operator that the conflict has cleared and an autonomous avoidance manoeuvre has been conducted.

Data from Collision Prediction is recorded to support the reporting of an Airprox incident to ATS; where appropriate, Environment Integration notifies the ATS of the incident and if the prescribed flight level has been contravened by the avoidance manoeuvre.

C.2.2.1.3 Post-Conditions

- Collision avoided, air vehicle continues to be airborne and monitored by an air traffic controller.

C.2.2.1.4 Actors

ATS

The ATS controlling the airspace.

Authorised Operator

The authorised operator (e.g. the pilot) of the ownship, located within the air vehicle or otherwise.

Closing Object

The object closing with the protected volume and therefore posing the collision risk. The Closing Object will become the intruder whilst within the protected volume.

Object Detections Provider

Source of object detections, e.g. Sensors or Tactical Objects.

Path Demand Arbitrator

The arbitrator for path demands that can be generated by the system or by an Authorised Operator via HMI.

Path demands include:

- Defined routes (planned, contingency, etc.).
- Avoidance manoeuvres (collision, terrain, weather, threats, etc.).
- Manoeuvre profiles (take-off, landing, etc.).
- Autopilot-type manoeuvres (acquire altitude, hold heading, etc.).

Interrogation Equipment

An IFF interrogator.

C.2.2.2 Terrain Avoidance

This IV describes a specific example of potential infringement of the MSD between an air vehicle and the ground being identified by Collision Prediction, based on the current vehicle location and dynamics. Once identified, Collision Avoidance determines an appropriate manoeuvre to avoid the collision.

It is assumed that:

- Terrain avoidance is performed autonomously.
- The route or intended path of the air vehicle, as planned, remains outside of the MSD.
- The Path Demand Arbitrator will arbitrate all path demands placed upon it, taking into account task priorities provided by Tasks. It will therefore be possible for the Path Demand Arbitrator to de-prioritise or ignore the demand for manoeuvre to avoid the potential infringement, during landing or CTT for example. In this example it is assumed that the collision avoidance manoeuvre has priority.
- The air vehicle is not performing terrain following.

The following considerations are excluded:

- HMI interactions.
- Terrain avoidance used to detect divergence from landing aids glideslope.
- Recovery to the required route or clearance following the avoidance manoeuvre.

C.2.2.2.1 Pre-Conditions

- The air vehicle is airborne and flying within the operating envelope.
- The air vehicle is not preparing to land.
- The air vehicle has deviated from the intended path.

C.2.2.2.2 View

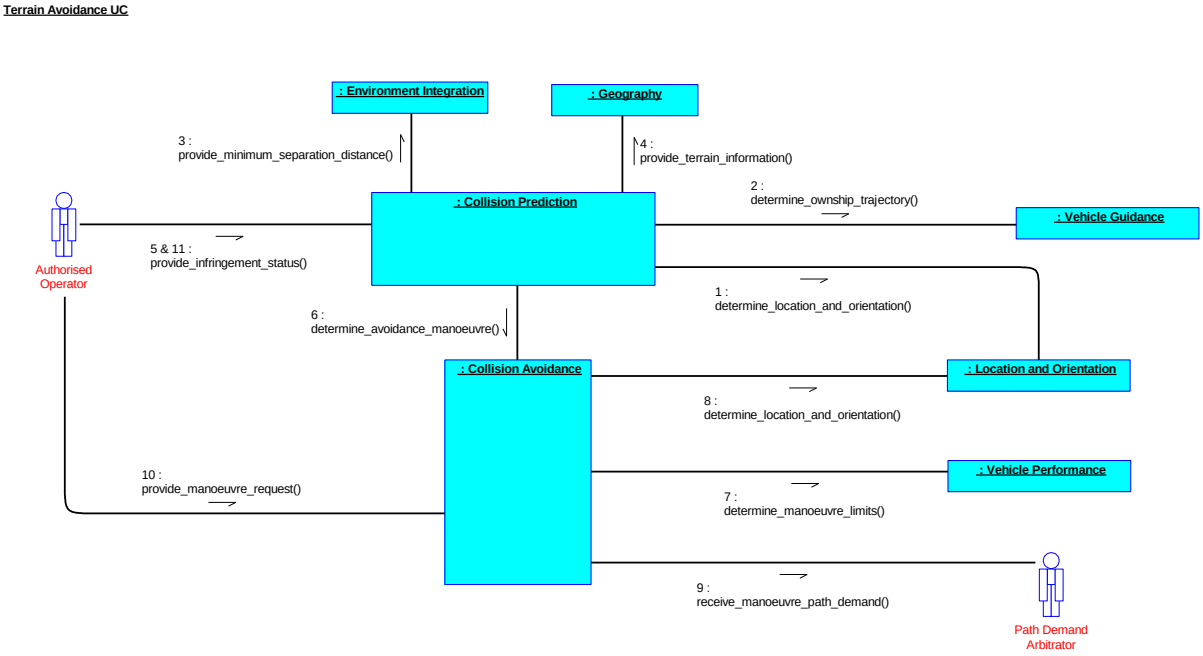

Figure 1328: Terrain Avoidance IV

Using ownship information received from Location and Orientation and Vehicle Guidance, Collision Prediction determines that the air vehicle will infringe the MSD between the air vehicle and ground terrain. The MSD between the air vehicle and ground terrain is provided by Environment Integration; with the terrain information provided by Geography.

Collision Prediction alerts the Authorised Operator of the MSD infringement.

Collision Prediction notifies Collision Avoidance of the need to perform a collision avoidance manoeuvre. Collision Avoidance determines the appropriate manoeuvre, applying manoeuvre limits from Vehicle Performance and taking into account ownship information received from Location and Orientation. Collision Avoidance informs Path Demand Arbitrator of the manoeuvre requirement. Path Demand Arbitrator considers all vehicle path demands put upon it and arbitrates to determine the path to be commanded. Collision Avoidance notifies the Authorised Operator that a manoeuvre has been requested. The manoeuvre is performed, the Authorised Operator is informed and safe separation is recovered.

When Collision Prediction no longer predicts an infringement of the MSD, the avoidance demand on Collision Avoidance will be cancelled and the vehicle path demand removed. Collision Prediction will notify the Authorised Operator that the collision risk is cleared.

C.2.2.2.3 Post-Conditions

- Infringement of the MSD (terrain or obstacle) has been avoided and the air vehicle continues to fly within the operating envelope.

C.2.2.2.4 Actors

Authorised Operator

The authorised operator (e.g. the pilot) of the ownship, located within the air vehicle or otherwise.

Path Demand Arbitrator

The arbitrator for path demands that can be generated by the system or by an Authorised Operator via HMI.

Path demands include:

- Defined routes (planned, contingency, etc.).
- Avoidance manoeuvres (collision, terrain, weather, threats, etc.).
- Manoeuvre profiles (take-off, landing, etc.).
- Autopilot-type manoeuvres (acquire altitude, hold heading, etc.).

C.2.3 Navigation

This IV illustrates an example of the use of navigation sensors to establish a navigation solution that provides the location, orientation and rates of change of the air vehicle. The current navigation solution accuracy does not meet the new situational needs so a new demand to determine a navigation solution to a specified accuracy is made.

It is assumed that:

- Human interaction to accept or manually update position fixing is not required.
- EMCON does not preclude the use of any available sensor.
- The specific location and orientation parameters and references, and the methods used to derive them lie within the Location and Orientation component itself.
- The location of all possible geo-located navigation aids (e.g. beacons) is known to the system.
- Maps relating magnetic variation or true north to locations on an Earth model are loaded within the system.

The following considerations are excluded:

- Inputs due to sensor detection processing (e.g. terrain map matching) - the scenario uses inputs provided by sensors only.
- Resource contention and resource allocation.
- The impact of the navigation solution using sensors that may be needed for other tasks including reporting the cost of using these sensors.

C.2.3.1 Pre-Conditions

- There is an existing navigation solution that provides the current air vehicle location.

C.2.3.2 View

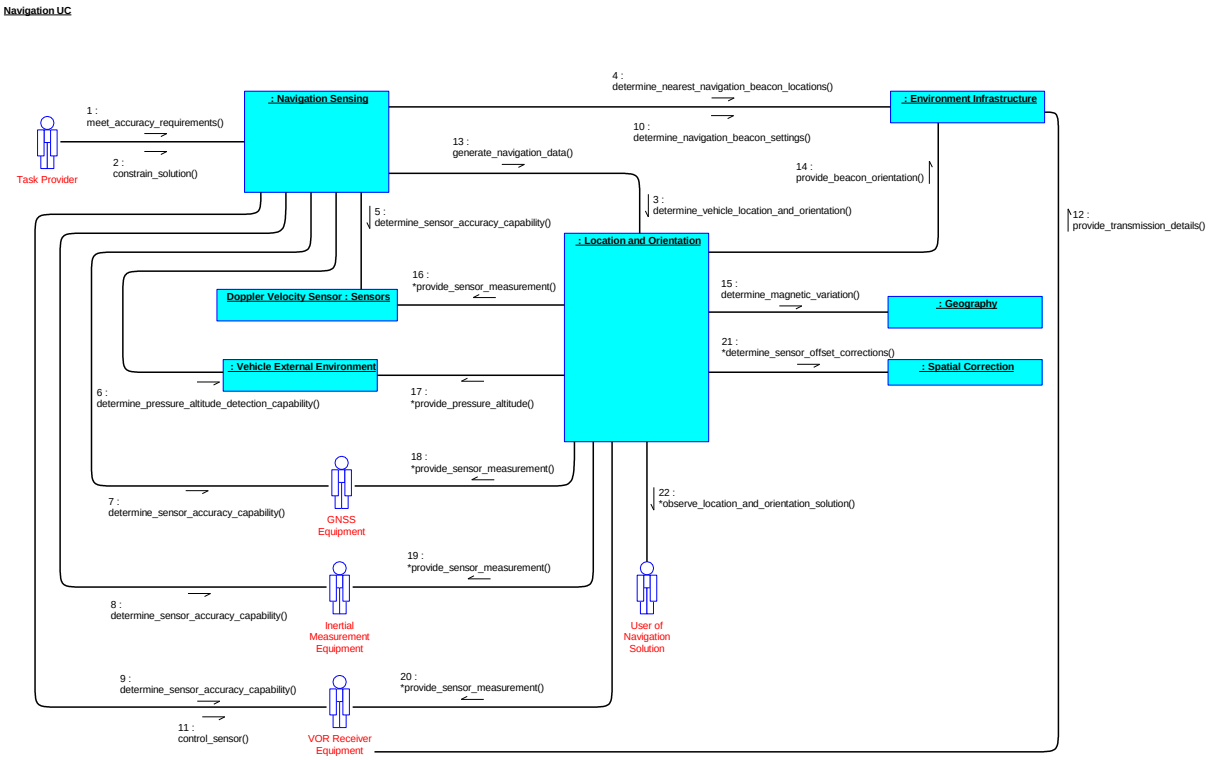

Figure 1329: Navigation IV

Navigation Sensing receives a demand from the Task Provider to establish a navigation solution that provides the location, orientation and rates of change of the air vehicle to a specified accuracy. Navigation Sensing also receives a constraint from the Task Provider to not require any specific air vehicle positioning or manoeuvres in support of generating a navigation solution.

To aid in determining the options available at the current location, Navigation Sensing requests the current air vehicle position from Location and Orientation, which Location and Orientation then provides based on the existing navigation solution.

Navigation Sensing requests Environment Infrastructure to provide information about any TACAN or VOR beacons within a specified region (centred on the current air vehicle location). Environment Infrastructure identifies two VOR beacons matching the criteria and provides the requested information including their coverage.

Navigation Sensing requests the potential navigation sensor sources to provide a statement of their capability.

Navigation Sensing determines a combination of available sensor capabilities which when combined has the capability to meet the required navigation solution accuracy and constraints. In this scenario, this includes the following: Doppler Velocity Sensors, GNSS, Inertial Sensor, pressure based altitude and VOR receiver.

The previous requirements placed on the Doppler Velocity Sensor, GNSS, Inertial Sensor and Vehicle External Environment (for pressure altitude) are determined by Navigation Sensing as still being valid to achieve their contribution to the required navigation solution accuracy and constraints. Note that there may be several requirements that can be satisfied simultaneously by a single navigation solution. Navigation Sensing will balance the needs of the different requirements within its solution.

Navigation Sensing requests Environment Infrastructure to determine the properties of the VOR beacons identified as available navigational aids (e.g. frequency and channel). Once these have been determined, Navigation Sensing places requirements on the air vehicle VOR Receiver Equipment to obtain the necessary transmission details from Environment Infrastructure, and use the settings to configure itself to detect these beacons and provide the air vehicle position relative to the VOR beacons.

Once all of the sensor settings have been established, Navigation Sensing provides requirements to Location and Orientation to produce the navigation solution information. This includes a specification of which information sources should be used and if they should be incorporated into the solution or used as checks to ensure that the solution remains within the expected bounds.

Information about the reference orientation of the beacons is provided to Location and Orientation (i.e. whether the beacon is aligned to true or magnetic North). Location and Orientation requests Geography to provide the magnetic variation at the beacon locations to allow the VOR measurements to be adjusted if necessary.

Location and Orientation continuously obtains the information from each source that is needed to produce the navigation solution.

Spatial Correction is used to determine corrections to offsets between the sensors producing measurement information where they are affected by aero-elastic deformation (e.g. when mounted on the wings).

Location and Orientation continuously refines the air vehicle navigation solution using the relevant data from different sources.

Location and Orientation continuously makes available the air vehicle navigation solution information to any User of Navigation Solution that requires it.

In response to its requirement to produce the navigation solution, Location and Orientation continuously reports the achieved accuracy of the solution to Navigation Sensing.

Navigation Sensing reports the status of achieving the required navigation solution to the Task Provider. Navigation Sensing continuously monitors the performance achieved in meeting the requirement and identifies if any further changes to the utilised sensors, their settings and how these should be integrated into a navigation solution are required to maintain the achievement of the requirement over time.

C.2.3.3 Post-Conditions

- Navigation sensors have been allocated and settings established.
- Air vehicle navigation solution(s) are providing the required location, orientation and rates of change information to the required accuracy.

C.2.3.4 Actors

GNSS Equipment

The interface to the equipment hardware for the GNSS receiver.

Inertial Measurement Equipment

The interface to the equipment hardware for the inertial measurement system.

Task Provider

The source of a demand to generate a navigation solution that provides the location, orientation and rates of change of the air vehicle to a specific accuracy.

User of Navigation Solution

A user of the navigation solution location, orientation and rates of change information (which may be either a person observing the output using a HMI, or another component within the architecture).

VOR Receiver Equipment

The interface to the equipment hardware for the VOR receiver.

C.2.4 Weather

The purpose of this IV is to show how the system obtains weather data through sensing and reports and then determines which weather is threatening. It also shows how the system determines the steps to take to mitigate that weather threat.

It is assumed that:

- The Weather Provider updates the system throughout the duration of the mission.

The following related areas of functionality are excluded:

- The communication between external actors and the system.
- Pre-mission planning and showing the loading of mission related data (including weather).
- The components and external equipment that provides the Sensor Products component with data.

C.2.4.1 Pre-Conditions

Tasks has placed a requirement on Susceptibility to assess the vehicle's susceptibility to particular weather elements.

C.2.4.2 View

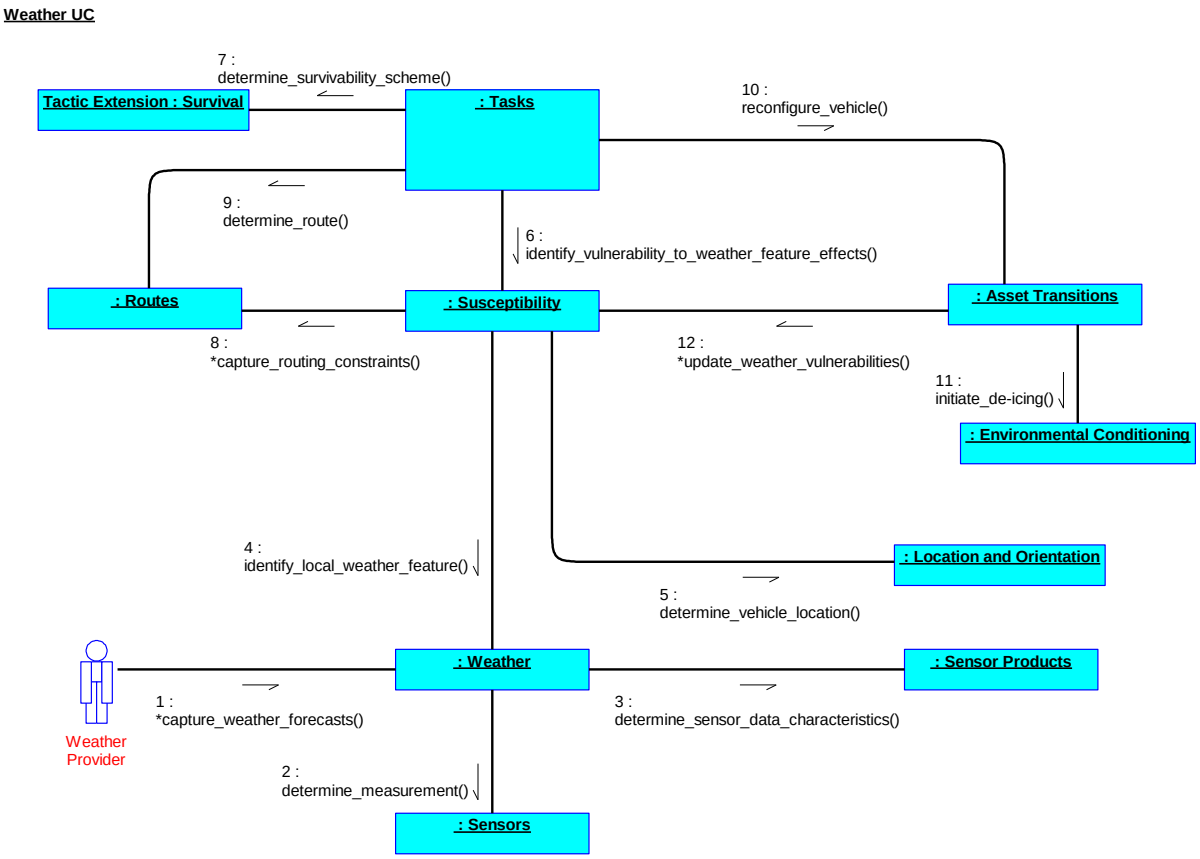

Figure 1330: Weather IV

Weather obtains data from various different sources; the Weather Provider provides weather reports throughout the mission, Sensors gathers real-time temperature data around the vehicle, Sensor Products collates more complex sensor data, such as cloud formations, which is provided to Weather. Once a weather feature has been identified, it is passed to Susceptibility which assesses the vehicle's susceptibility to particular weather elements. This assessment will be based on knowledge of ownship vulnerabilities in its current state (e.g. icing) to the particular weather element.

As part of this determination of susceptibility the component, with the aid of Location and Orientation, will determine a number of regions in which the vehicle is susceptible and the level of susceptibility it has in these regions. The susceptibility risks are passed up to Tasks which in conjunction with its survival extension plans or executes actions to reduce the susceptibility of the vehicle. In this case constraints can be given to Routes to avoid a particular region (with this region being updated continuously by Susceptibility). Tasks also places a requirement on Asset Transitions to reconfigure the vehicle to get it into a state which makes it less susceptible; this is facilitated by the use of Environmental Conditioning (e.g. start de-icing). Asset Transitions informs Susceptibility of the change in asset state such that the route can be reassessed based on new vulnerability advice from Susceptibility.

C.2.4.3 Post-Conditions

- A weather threat has been identified and the steps to reduce the risk of the threat have been enacted by the system.

C.2.4.4 Actors

Weather Provider

A provider of both pre-flight and in-flight information from various sources, including forecast and live weather information.

C.2.5 Air Data

This use case describes how air data is determined. Air data includes:

- Calibrated airspeed.
- Mach number.
- Outside air temperature.
- True airspeed.
- Angle of attack (alpha).
- Sideslip (beta).
- Pressure Altitude (1013.25mb reference).
- Barometric reference altitude (to a specific pressure reference 1013.25mb, QNH, QFE).

It is assumed that:

- Monitoring of the air external to the air vehicle is accomplished using pressure, temperature and flow angle sensors only.

The following considerations are excluded:

- Determination of the fluid state using laser sensing systems.
- Determination of angle of attack or sideslip using inertial sensors.
- Pressure error corrections reliant on inputs other than pressure, temperature and flow angle.
- Inhibiting and modifying use of particular sensor inputs during firing of weapons and guns where these would adversely affect the sensor.
- Sensor and probe heating.

C.2.5.1 Pre-Conditions

- The air system is powered-up and sensors are available.

C.2.5.2 View

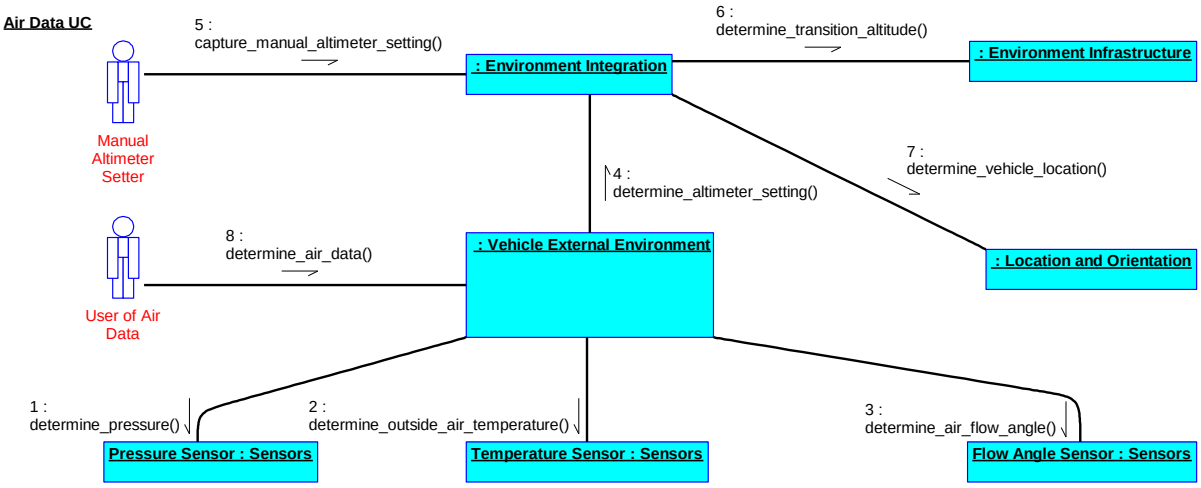

Figure 1331: Air Data IV

Vehicle External Environment captures values of pressure, temperature and flow angle from instances of Sensors (pressure sensor, temperature sensor and flow angle sensor) to determine air data parameters.

To determine a barometric reference altitude (e.g. referenced to QNH), Vehicle External Environment requests the altimeter setting from Environment Integration. Environment Integration determines the altimeter setting from either a direct input (from Manual Altimeter Setter) or automatically (for an unmanned vehicle that has lost communications). The automatically determined altimeter setting may be based on the last manual altimeter setting input or transition altitude at the current location (using inputs from Environment Infrastructure and Location and Orientation).

Air data, including accuracy, is provided to the User of Air Data. This could include HMI components for crew displays.

C.2.5.3 Post-Conditions

- Up-to-date air data is available.

C.2.5.4 Actors

Manual Altimeter Setter

A user directly inputting the altimeter setting, most likely the crew of the vehicle.

The altimeter setting is used to offset pressure altitude so it represents altitude with respect to a particular reference (e.g. 1013.25mb, QNH or QFE).

User of Air Data

Another part of the system, or crew, which uses air data.

C.3 Vehicle Stores Views

This section contains IVs relating to the management of stores carried by an air vehicle.

C.3.1 Role Fit Discovery

This IV shows how role fit equipment fitted to the vehicle is discovered and validated. This scenario describes a system which has 3 stations with the following role fit:

- Station1 - Store X (MIL-STD-1760) Ref. [15].
- Station2 - Store Y (Dumb).
- Station3 - empty.

Each station has a number of attachment mechanisms. The electrical connections at these attachment mechanisms are used to monitor whether a store is connected.

It is assumed that:

- Semantic Translation is configured with MIL-STD-1760 protocols.
- Inventory is provided with the allowable role fit configurations for the vehicle stations.
- The planned and fitted role fit configurations are correct.
- Store X (MIL-STD-1760) communicates via MIL-STD-1760.
- Store Y (Dumb) does not communicate.
- The inventory needs to be authorised after passing checks.

The following considerations are excluded:

- HMI interactions between the Planner, Fitter, the Inventory Authoriser and certain Inventory Receivers.
- The startup sequence associated with the provision of power.

C.3.1.1 Pre-Conditions

- Stores are fitted to stations.

C.3.1.2 View

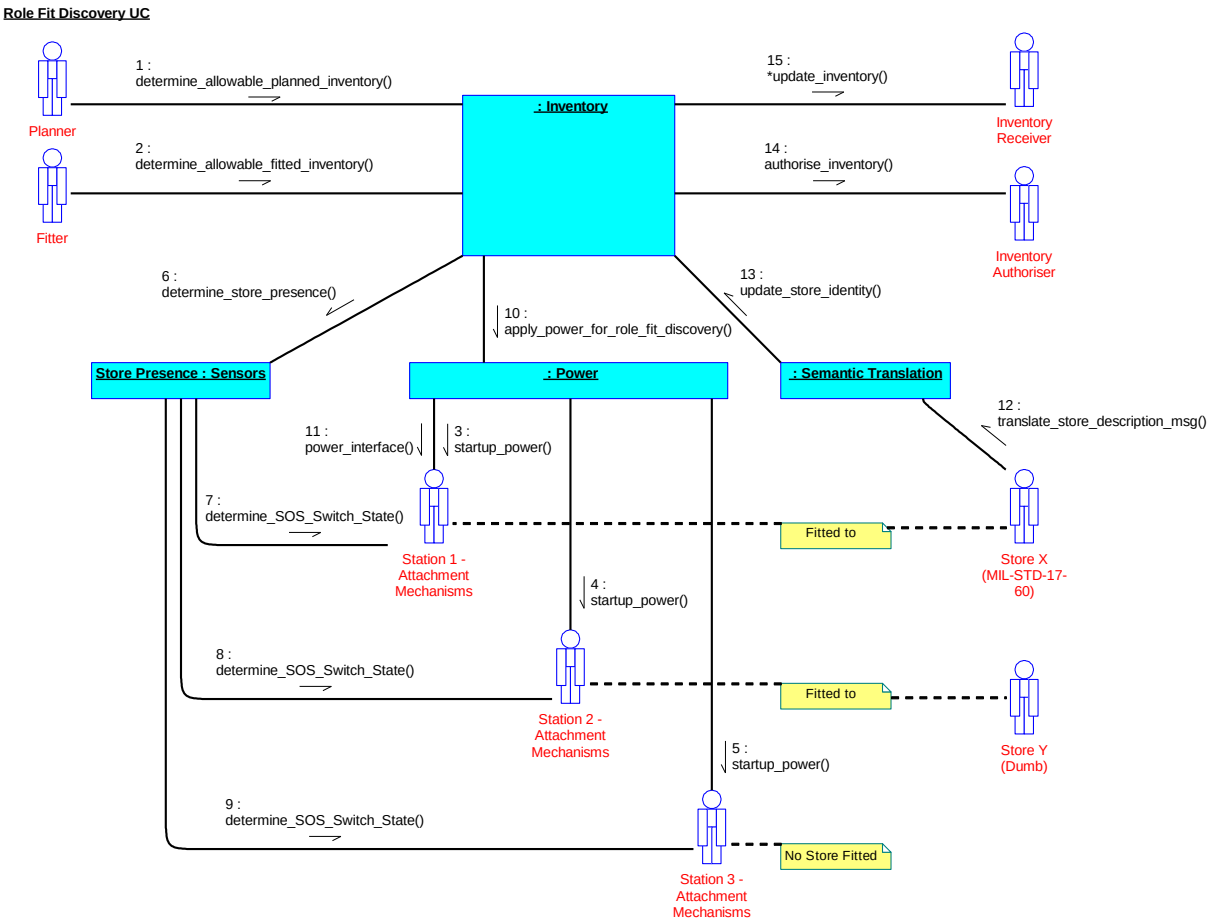

Figure 1332: Role Fit Discovery IV

Upon receipt of the planned and fitted inventories, Inventory determines that both are allowable and that what is stated as fitted matches the planned inventory.

As part of startup, Power will provide limited power to store attachment mechanisms in accordance with the planned and fitted inventory.

Store Presence monitors the state of store on station switches in each attachment mechanism, in this case Station 1 - Attachment Mechanisms and Station 2 - Attachment Mechanisms have stores present and Station 3 - Attachment Mechanisms do not. Inventory then verifies that station 1 and station 2 having items present and station 3 not is correct in line with the planned and fitted inventory.

Inventory will request Power to supply specific stores with power to generate electrical signals (in this case store description messaging).

When power is applied to the store, Semantic Translation receives a store description message, as it understands MIL-STD-1760 protocols it converts this into information on the store identity. Once translated, this information is then passed on to Inventory which confirms that the identity of the store at station 1 Store X (MIL-STD-1760) is correct in line with planned and fitted configurations and the store presence indicators. A store description message is not received for station 2 as the store fitted is dumb, and on station 3 no store is fitted.

Now that Inventory has identified all stores it can, it knows which stations have stores present and verified against the fitted, planned and allowable inventories, it presents the verified inventory to the Inventory

Authoriser for authorisation. Once the inventory is authorised Inventory will then continuously provide updates on the current inventory to any Inventory Receiver.

C.3.1.3 Post-Conditions

- Inventory Receivers are informed of the current inventory.

C.3.1.4 Actors

Fitter

Responsible for defining which inventory was fitted to the system.

Planner

Responsible for defining the planned inventory and inputting it into the system.

Inventory Authoriser

Responsible for authorising the inventory once the Inventory component has performed the necessary checks. This is likely to be a user/operator of the system and it may be possible for them to override the details of some of the inventory before authorisation.

Inventory Receiver

A generalisation of users of the inventory which require knowledge of the current inventory in order to fulfil their responsibilities.

Examples of users which may need to know the current inventory are:

- For releasable inventory, Stores Release determines what release packages are currently available.
- For external tanks, Fluids determines an increase in the tanks it has and the total contents.
- For a sensor such as a rangefinder, Sensors determines an increase in capability.

Station 1 - Attachment Mechanisms

The set of attachment mechanisms (e.g. clamps) for station 1.

Station 2 - Attachment Mechanisms

The set of attachment mechanisms (e.g. clamps) for station 2.

Station 3 - Attachment Mechanisms

The set of attachment mechanisms (e.g. clamps) for station 3.

Store X (MIL-STD-1760)

A particular store which can communicate via MIL-STD-1760 Ref. [15] protocols.

Store Y (Dumb)

A particular store which cannot communicate (is "Dumb").

C.3.2 Releasing

This IV illustrates the armed release of a package of stores when required.

Potential store types include weapons (e.g. AMRAAM, Paveway IV or Brimstone), fuel tanks, countermeasures (e.g. chaff or flares), sensor pods, deployable sensors (e.g. sonobuoys) or gun rounds. This IV covers a rail launched weapon, to illustrate the possible interactions. Other stores would involve different interactions, mostly with the same set of components. A jettison would involve mostly the same set of components, though the physical release mechanism may be different (e.g. jettison of rail launched store may be achieved by downward ejection of the launcher), release intervals may be different and weapon arming would not be activated (or be positively de-activated for enhanced safety).

It is assumed that:

- The weapons bay doors are open.

The following considerations are excluded:

- Controlling the air vehicle to fly within the required flight envelope for release is excluded as this is covered by the Vehicle Movement In Air IV.
- Selection of a store package (types and quantity) that is suitable for a particular task.
- Determining the precise location of the release and the routing to arrive at that location.
- Configuring the air vehicle for release or jettison (e.g. commanding weapon bay door opening).
- Applying logic power and priming of the weapon with, for example, target location.

C.3.2.1 Pre-Conditions

- The hardware Master Safety Switch (MSS) (or Master Armaments Safety Switch (MASS) per Def Stan 00-970 Ref. [16] terminology) has already been set to LIVE.

C.3.2.2 View

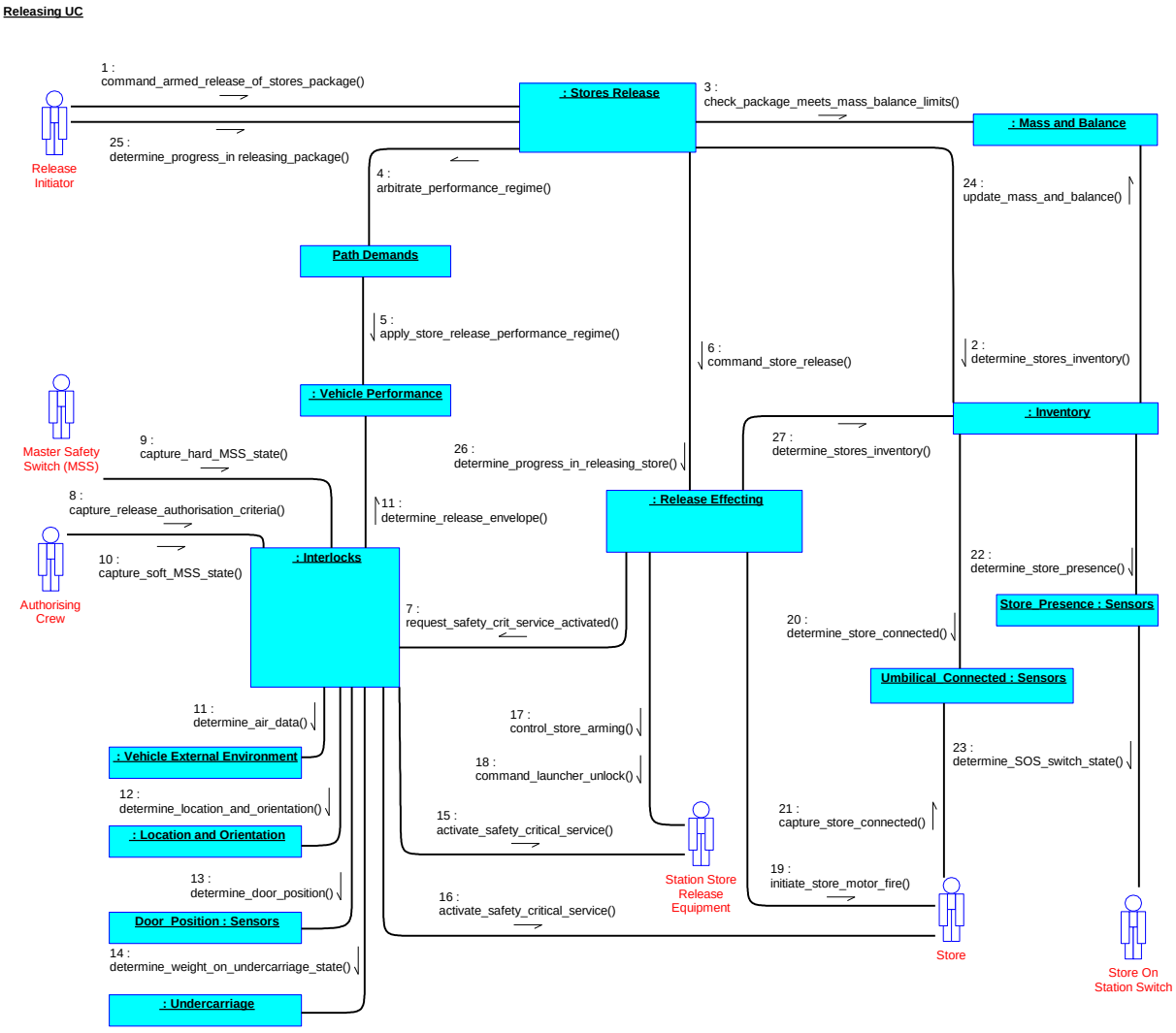

Figure 1333: Releasing IV

When the need for armed release of a store type has been determined, the Release Initiator commands a release. Stores Release confirms which stores are present with Inventory and selects a release package that conforms to Mass and Balance constraints. Whilst the release is in progress Stores Release will inform Path Demands that a store release performance regime is required, which provided it is valid for the current path demand will be passed to Vehicle Performance to limit the flight envelope to the stores release performance regime. Stores Release commands Release Effecting to release the individual store in the package.

Release Effecting requests Interlocks to activate safety critical services for the resources needed to perform the release. Interlocks has knowledge of the interlocks required for a particular safety critical service and captures the authorisation criteria (from Authorising Crew) and release envelopes (from Vehicle Performance). Authorisation criteria could be in the form of a binary "armed/safe" input, such as that provided by a "Late Arm" switch. Interlocks activates the requested safety critical services provided the appropriate interlocks, release envelope and authorisation criteria are satisfied using inputs from Master Safety Switch (MSS) (hard MSS state), Authorising Crew (for "soft" MSS), Vehicle External Environment (for airspeed, etc.), Location and Orientation (for vehicle position and attitude data), Undercarriage (for weight on undercarriage state), and Door_Position: Sensors (for weapon bay door position).

Release Effecting commands Station Store Release Equipment to unlock the rail launched missile and to activate the arming discrete. Once Release Effecting has received confirmation that the missile is unlocked, it commands the stores rocket motor to fire, which will result in the store leaving the aircraft.

During the release sequence Inventory will update the role fit inventory based on inputs from Umbilical_Connected: Sensors (store umbilical connected) and Store_Presence: Sensors which monitors Store On Station Switch. Mass and Balance will reflect the revised stores configuration.

Stores Release provides progress on the requirement to release the stores to Release Initiator based on progress reports of individual releases from Release Effecting.

C.3.2.3 Post-Conditions

- Store has been released.
- The inventory, mass and balance have been updated.

C.3.2.4 Actors

Authorising Crew

The crew that are authorising the release and controlling the "soft" Master Safety Switch (MSS).

Master Safety Switch (MSS)

The hardware switch located on the air vehicle. It is normally LIVE throughout take-off, flight and landing.

Release Initiator

The Release Initiator determines the store type (e.g. ASRAAM) to be released and initiates the release. Initiation could be from Action components (e.g. Target Engagement) or from the crew (via HMI).

Station Store Release Equipment

The equipment that directly outputs electrical discretes (release, fusing, etc.) to S&RE or stores if commanded to do so and the appropriate safety critical service is enabled. Typically this equipment will include power drivers and relays.

Store

A store fitted to a particular station.

Store On Station Switch

A Store On Station (SOS) switch.

C.4 Communications Views

This section contains IVs relating to the communications capability of a system.

C.4.1 Network Initialisation

This IV illustrates how the system satisfies a requirement to set up a series of communication links to form a new network (either for use immediately or in the future) to support one or more tasks.

It is assumed that:

- No existing network route exists which can provide the support required.

The following considerations are excluded:

- Use of the network once established.
- The details of data transfer over the network.
- The detailed control of a Communicator.
- Harmonisation of antennae in flight via Spatial Correction.
- Coordination of multiple activities across the EM spectrum.
- Detailed use of cryptography.
- The brokering of power.

C.4.1.1 Pre-Conditions

- A task that requires communications over a network exists.

C.4.1.2 View

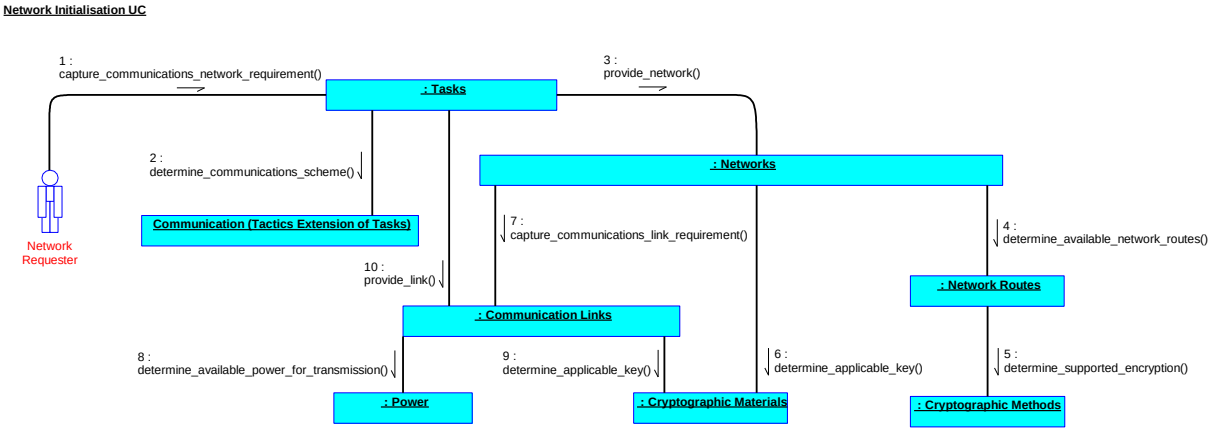

Figure 1334: Network Initialisation IV

Tasks, supported by the Communication tactics extension, captures a requirement for communications from a Network Requester, and instructs Networks to provide the required communications network. Networks, Network Routes and Cryptographic Methods determine that the need for communications cannot be met by existing network routes, or cannot be met to a required cryptographic standard, and hence a new communications link is required.

Networks determines the requirements for this new link, and the requirements are captured by Communication Links. Communication Links confirms with Power and Cryptographic Materials the availability of the required resources.

Tasks instructs Communication Links to establish the required communications link. Communication Links establishes the new communications link receiving confirmation from the other involved node, which makes it available for use by Networks.

C.4.1.3 Post-Conditions

- A communications link is fully planned and agreed by both the transmitting and receiving communications capabilities, and can either be used immediately or at a later point in time to support the tasks illustrated in the Human Communications and Data Transfer IVs.

C.4.1.4 Actors

Network Requester

A component that has determined there is a need for additional communications infrastructure to be provided.

C.4.2 Connection Management

This IV shows how communications can be planned, and later established, to meet a given communication requirement. This includes showing how continual feedback from network resources is used to aid prediction of communication achievability.

A Planner is aware of a communication requirement and needs to assess whether the resources will be available to meet the demand; taking into consideration reported performance data. It is determined that the communication requirement is achievable at a planned time. When the communications are needed, a Requestor makes a demand to establish communications.

The interactions shown are applicable whether or not the Planner and Requestor are the same individual or entity. Likewise, it remains the same, when the communication requirement is ad-hoc for immediate use or planned for a later time.

The following considerations are excluded:

- Control of communications infrastructure.
- Control of the Communicator.
- Observability analysis.
- Use of cryptography.
- Permitted transmission capabilities.
- Use of the communication to transfer data.

C.4.2.1 Pre-Conditions

- The nodes making up the network are known; either entered during prior planning or by data configuration.
- The preferred transfer solutions for different types of data for different recipients have been preloaded in the relevant components.

C.4.2.2 View

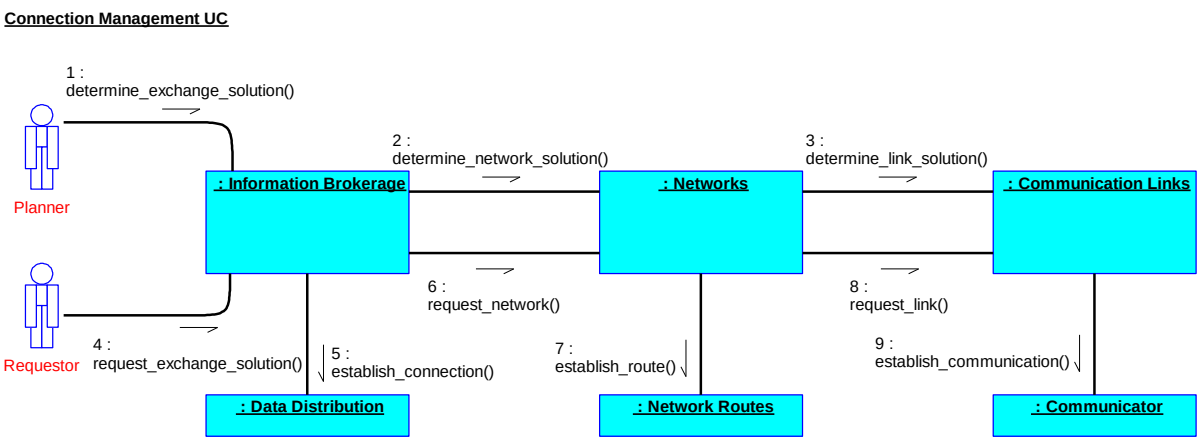

Figure 1335: Connection Management IV

A Planner asks Information Brokerage to determine an information exchange solution.

Information Brokerage determines that the exchange is allowed and as this is a request for a new data exchange of a known type, checks with Networks for the expected reachability across the network.

Networks evaluates possible solutions, including determining the quality of service required to meet the connectivity preferences. As there is currently no network in place Networks requests recommended link solutions from Communication Links.

Based on the request from Networks, Communication Links checks for the best link solution to meet the latency, reliability and integrity requirements. It also considers the real-world constraints on communication between vehicles, this includes support from Observability (not shown).

Communication Links determines that the link requirement is achievable, with this information Networks can choose a network solution, informing Information Brokerage that communication requirement is achievable, based on this feedback Information Brokerage determines that the information exchange is achievable and informs the Planner.

Throughout this process, and in support of all communications, communication resources provide continuous feedback which is analysed by the communications components; this allows achievability of communication requirements to be continuously assessed and remedial actions to be performed to inform achievability determination in future service requests.

At a later time a Requestor requests that Information Brokerage establish the previously planned information exchange service. Information Brokerage asks Data Distribution to facilitate data exchange, in line with the service requirements, and requests establishment of the network from Networks.

Networks in turn asks Network Routes to configure routes, in line with the routing requirements, and requests Communication Links to establish links.

Communication Links asks the Communicator to establish communication, in-line with the communication requirements.

C.4.2.3 Post-Conditions

- Communications resources have been configured to meet the communication requirements.

C.4.2.4 Actors

Planner

A component that has communication requirements which need to be planned, either for immediate or later use.

Requestor

A component that has communication requirements which need to be established, for immediate or later use.

C.4.3 Data Transfer

This IV shows how data is transferred to a recipient across a communications link, including both transmission and reception. The monitoring of the data transfer service is also shown, to highlight the management components' responsibilities for maintaining the quality of the communication service.

Note: Cryptographic Methods is shown as a service used by Network Routes; however at a platform level, for security reasons, Cryptographic Methods may be in-line, between Network Routes and Communicator.

It is assumed that:

- Only packet encryption is required for the data transfer.

The following considerations are excluded:

- How the data transfer request is made.
- How data is passed between the transmitting and receiving nodes.
- How the service provision dynamically changes in response to quality of service monitoring.

C.4.3.1 Pre-Conditions

- All necessary communications services have been set up and established, in order to support interactions between data senders and communications transport.
- Transmission protocol has been determined.
- Encryption requirements have been determined.
- The data to be transferred exists within the system.
- The transfer of data has been initiated.

C.4.3.2 View

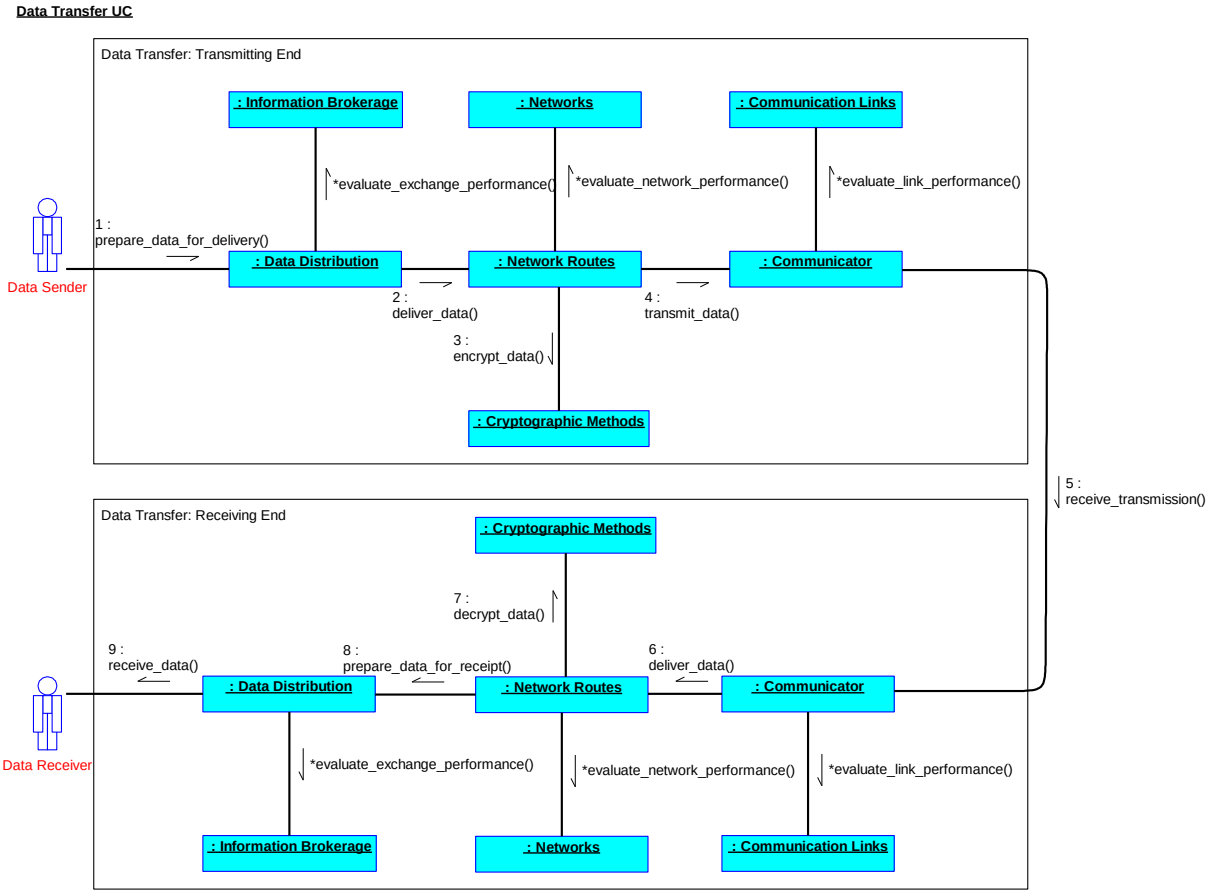

Figure 1336: Data Transfer IV

Transmission

Data Distribution prepares the data for delivery by formatting the data according to the transmission protocol (determined during communications planning); this may involve splitting the data into multiple packets. Network Routes directs the prepared packets for forward data delivery. Network Routes determines which route the data will take. This may include going via Cryptographic Methods to encrypt the packets prior to transmission, depending upon the security policy determined during communications planning. Actual transmission of data is performed by Communicator.

Reception

Communicator receives incoming data which it passes, as packets, to Network Routes for determination of the next route to take (in this case to report to the local system, as the data has arrived at its destination). Network Routes may use Cryptographic Methods to decrypt the packets prior to delivery, depending upon the communications setup. The packets are given to Data Distribution in order that they can be prepared for use. This involves collating the packets into a single data source and reformatting the data. When the data has been prepared, a receive request is issued to pass the completed data to its intended recipient.

At all times, the resource layer components (Data Distribution, Network Routes and Communicator) report on the performance of the service they provide, in order that the action layer components can evaluate performance and determine whether resources need reconfiguring to maintain the optimum level of service.

C.4.3.3 Post-Conditions

- The intended end user has the transmitted data.

C.4.3.4 Actors

Data Receiver

The intended recipient of the data.

Data Sender

The provider of the data to be transferred. This could be a human initiated request, or an automated transfer.

C.4.4 Tactical Exchange

C.4.4.1 Track Distribution

This IV shows how a track is distributed to an external system via a tactical datalink. This IV is specific to exchanging tracks via a tactical datalink.

It is assumed that:

- The deployment is set up so the components can access the track when required.

The following related areas of functionality are excluded:

- How tracks are produced from sensor data.
- How tactical datalink connections are managed.
- Non-track exchanges via tactical datalink.

C.4.4.1.1 Pre-Conditions

- There is a need to report tracks over a tactical datalink.
- The latest version of the track list has been provided to Semantic Translation.
- Information Brokerage has been loaded with the rules around what is allowed to be distributed outside of the system.
- The rules on how to declassify a track are loaded to Information Brokerage.
- The connection type that Data Distribution needs to use for the external system has been previously loaded.
- Tactical datalink structure rules are loaded to Semantic Translation.
- Tactical datalink delivery rules are loaded to Data Distribution.

C.4.4.1.2 View

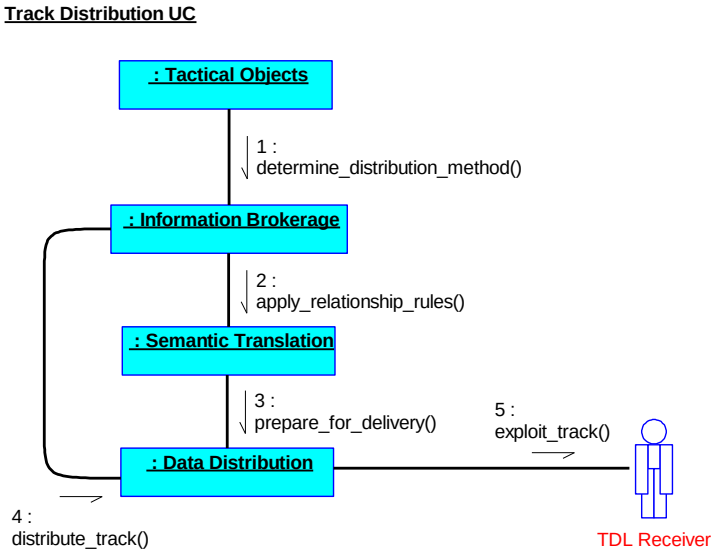

Figure 1337: Track Distribution IV

Tactical Objects produces a track set. Information Brokerage determines that a track within the track set meets a distribution requirement of an external system.

Information Brokerage understands that to distribute the track to the specific external system it will need to be exchanged via a tactical datalink. Information Brokerage also determines that the system has reporting responsibility for the track, and therefore is required to distribute it. Information Brokerage determines that only some of the track attributes should be shared; ensuring that the distribution is allowable by limiting which attributes are included.

Once the shareable attributes have been identified, Information Brokerage informs Semantic Translation that the track information needs to be reformatted before it can be distributed to the external system.

Semantic Translation applies the necessary translation rules to the track data and reformats it into the tactical datalink structure, ensuring the correct structure is in place before the distribution is initiated.

Once the track data has been reformatted, Information Brokerage requests Data Distribution to distribute it to the TDL Receiver. Data Distribution determines the communication protocol and packages data into messages as defined by the protocol. Data Distribution then distributes the track to the TDL Receiver for exploitation.

C.4.4.1.3 Post-Conditions

- A track has been distributed via a tactical datalink.

C.4.4.1.4 Actors

TDL Receiver

A receiver capable of receiving tracks from a tactical datalink system.

C.4.4.2 TDL Receipt

This IV shows how a message is received from an external system that follows a different logic to the local system. In this scenario a TDL system sends a message, which is an overloaded message type and is referencing different subject matters. This is resolved by Semantic Translation which converts the TDL specific logic into something understandable by the local system.

It is assumed that:

- The deployment is set up so the components can access the related information.

The following related areas of functionality are excluded:

- How tactical datalink connections are managed.
- Further analysis (including data fusion) of received TDL information.
- Low level data exchange using the Communicator component.

C.4.4.2.1 Pre-Conditions

- The connection type that Semantic Translation needs to use for the external system has been previously loaded.
- Tactical datalink structure rules are loaded to Semantic Translation.
- Tactical datalink receipt rules are loaded to Data Distribution.
- Ownship has conducted an engagement of a target using remote track data received over Link-16 tactical datalink.

C.4.4.2.2 View

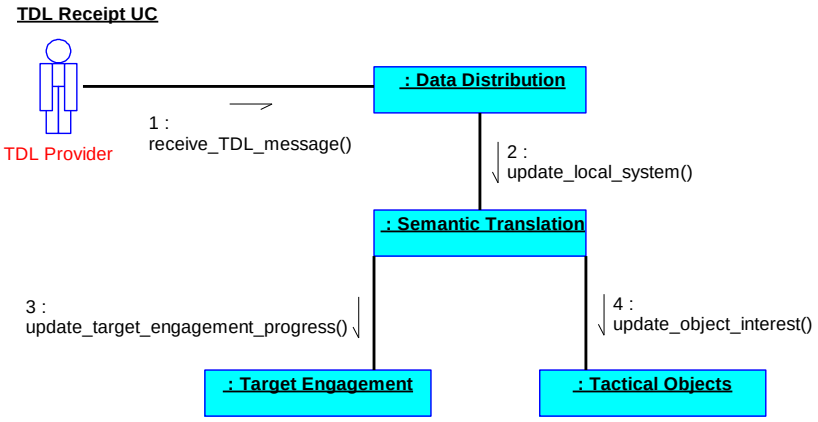

Figure 1338: TDL Receipt IV

TDL Provider provides a '*target sorting'* message (J12.6) from a Link-16 tactical datalink to Data Distribution. Data Distribution then unpacks the message into the individual status information discretes. As these are tactical datalink specific they are passed to Semantic Translation for conversion for use by PYRAMID components.

Semantic Translation analyses the information from the external system and identifies that it is a destruction of target confirmation. Using its mapping rules Semantic Translation determines the related internal events.

Target Engagement is informed of the updated engagement result, and Tactical Objects is informed of the change in reported status of the referenced object.

C.4.4.2.3 Post-Conditions

- The relevant engagement update has been captured.
- The relevant object has been deleted.

C.4.4.2.4 Actors

TDL Provider

A provider that has non-C2 reporting responsibility within a tactical datalink system.

C.4.5 Link Selection

This IV shows how the most appropriate communication solutions can be utilised as circumstances change. In this case, it is shown how a HF communications link with a Communication Target is established and then, as the communications distance reduces, a UHF link is established as a more effective solution.

It is assumed that:

- The radio supports a signal lock capability, allowing local control of the antenna to maintain the best signal.

The following considerations are excluded:

- Mechanical steering.
- Determination of own position.
- Determination of Communication Target position (this varies depending upon the method used to determine the position).
- Flow of data for communication.
- Translation between target location and pointing directions.
- EM Interoperability, e.g. blanking.
- Harmonisation via Spatial Correction.
- Planning of communications, e.g. detailed considerations of how and when to make selection changes.
- Measurement of communications quality.

C.4.5.1 Pre-Conditions

- All necessary communications services have been set up and established sufficiently to support link management.
- The location of the Communication Target is known.
- HF and UHF resources are available at both ends.

C.4.5.2 View

Link Selection UC

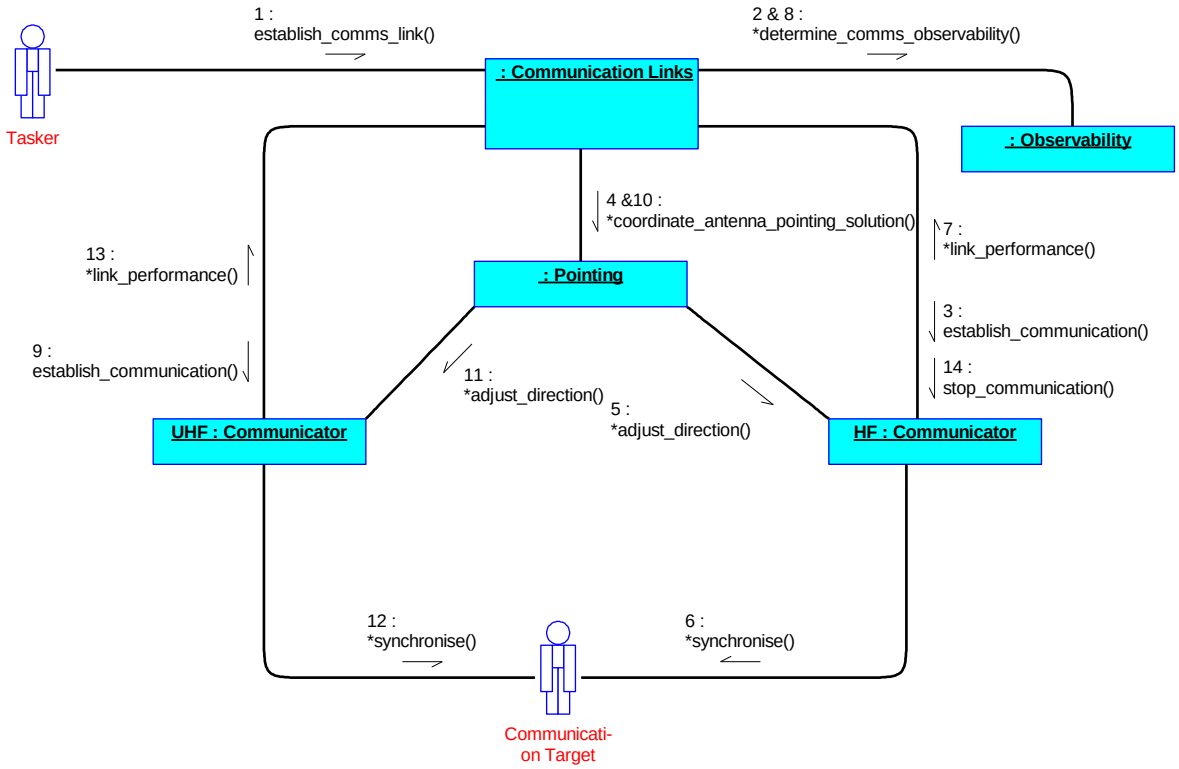

Figure 1339: Link Selection IV

Tasker requests the establishment of communications from Communication Links. Communication Links determines the costs of possible links based on observability information from Observability. Using this, alongside knowledge of the supported options for connection to the Communication Target, Communication Links determines that a HF radio link is the best solution as the range is greater than can be achieved via UHF communications.

Communication Links commands the HF Communicator to start establishing communications with the Communication Target. Communication Links commands Pointing to point the HF Communicator at the known location of the Communication Target; Pointing directs the HF Communicator to electronically course steer by providing directional instructions.

The HF Communicator synchronises its connection with that of the Communication Target, setting up the communications channel. This synchronisation continues throughout communications, in order to maintain the connection in accordance with the link protocol, with HF Communicator automatically performing fine tuning to maintain the best signal.

Throughout communication, the Communicator reports link performance information to Communication Links. This information, along with Observability results, is used to continually assess if, and when, an alternative communication solution should be selected.

In this case, it is observed that the distance to the Communication Target has reduced, and it is determined that an UHF connection will now better satisfy the communication quality requirements than the existing HF link.

Communication Links, commands UHF Communicator to start establishing communications with the Communication Target. Communication Links commands Pointing to point the UHF Communicator at the known location of the Communication Target; Pointing directs the UHF Communicator to electronically course steer by providing directional instructions.

The UHF Communicator synchronises its connection with that of the Communication Target, setting up the communications channel. Once the alternative link is established Communication Links commands the HF Communicator to stop communicating.

C.4.5.3 Post-Conditions

- Communications are established.

C.4.5.4 Actors

Communication Target

An external entity with reception capabilities that is the target for communications (such as another instance of Communicator).

Tasker

Higher level system component responsible for determining if, and when, a communication link with a Communication Target is required (e.g. the Tasks or Networks component).

C.5 Sensing Views

This section contains IVs relating to the sensing capability of a system.

C.5.1 Search

A task has been generated that requires the system to search for a tank within a specified area. This IV illustrates how the task is planned and coordinated using the Task and Action layers, detailed in the Control Architecture policy.

It is assumed that:

- Resources are available when required.
- A single air vehicle is being used.
- A range of sensors are fitted onto the air vehicle.
- There are no limits on the mission (e.g. RoE, EMCON or restricted airspace).

The following considerations are excluded:

- The operation of the sensor to perform the sensing actions.
- The interpretation of captured sensor data.
- The coordination of the routing solution.

C.5.1.1 Pre-Conditions

- The air vehicle is en route to an area within its sensors range of the target.
- A task has been issued to search for a tank in a specific region.

C.5.1.2 View

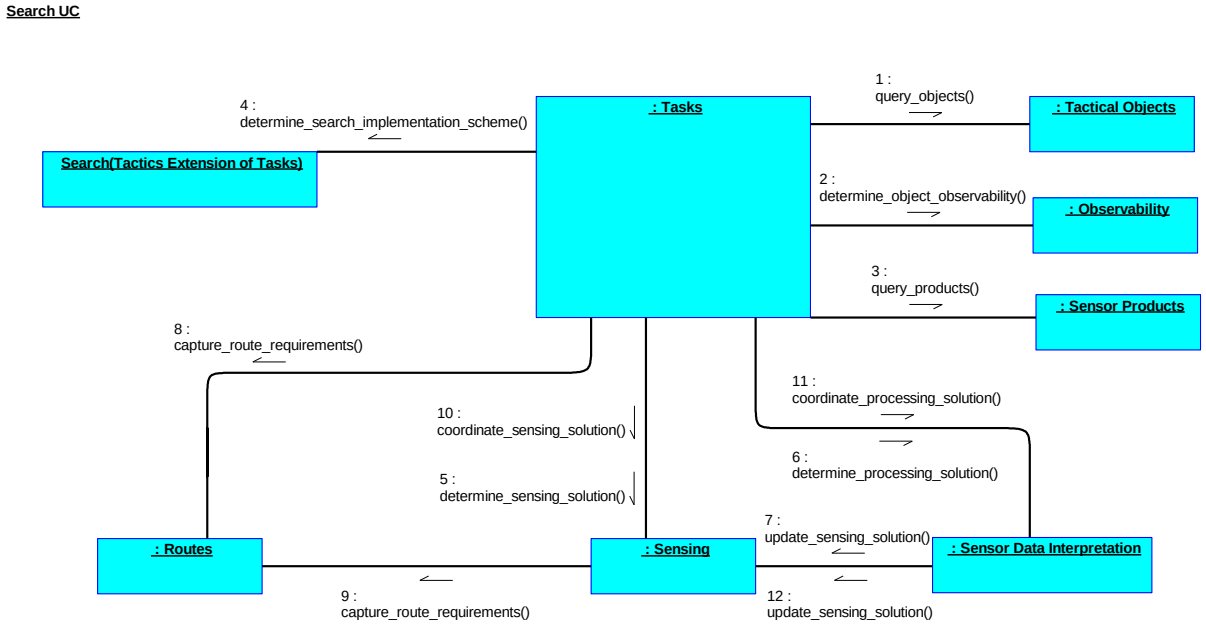

Figure 1340: Search IV

Having received the task to search for a tank in a region, Tasks utilises its Search extension to determine if the task can be satisfied using any pre-existing data. If the task cannot be satisfied using existing data, Search needs to determine the best tactic to use to gather the necessary data. In order for it to make these determinations, it queries a number of service components which provide relevant information as follows:

- Tactical Objects is queried for objects with a type that equals tank and location within the specified area, currently there are no objects known which match that criteria.
- Observability is asked for probabilities of making observations of tanks in that region based on the current environmental conditions (e.g. weather conditions). This provides the observation types (e.g. visual, electromagnetic, and thermal) which could be made of a tank and the probabilities of each.
- Sensor Products is then queried for sensor products matching those observation types within that region: there are currently no matching products.

In addition to this information the capabilities of action components are provided to Tasks. The Search extension uses all this information to determine the tactic to utilise and creates an implementation scheme for this task. A number of actions are defined to facilitate the task along with relevant constraints:

- Sensing is told to generate sensor products of the most probable observation type within the region, and Sensor Data Interpretation is given an action to process these products to identify tanks.
- Sensing and Sensor Data Interpretation then determine the solutions to their actions.
- Sensor Data Interpretation places requirements on Sensing, which refines the requirement given by Tasks.
- Tasks places a requirement on Routes for the air vehicle to operate within the defined search region.
- Sensing places a requirement on Routes for the required positioning of the air vehicle to carry out the search, which refines the requirement given by Tasks.
- With an achievable routing solution and achievable Sensing actions, Tasks commands the execution of the task.

During execution, Sensor Data Interpretation determines that the sensor product is not of sufficient quality to satisfy the requirements of its processing solution (e.g. an image is not of sufficient resolution to find tanks). To deal with this Sensor Data Interpretation updates the requirements on Sensing to increase the quality of the sensor product it produces. Sensing then refines the solution it provides (e.g. chooses another sensor mode) within the constraints given by Tasks.

C.5.1.3 Post-Conditions

- The search task is complete.

C.5.2 Tactical Sensing

This IV illustrates the enactment of a sensing task in order to produce a sensor product from a single Tactical Sensor. This includes the application of power, and the manoeuvring of the sensor in order to maintain an effective pointing angle.

It is assumed that:

- The Tactical Sensor is externally mounted to the vehicle on a gimbal.
- Application of power to the Tactical Sensor is a pre-requisite to enable access to the capability of the sensor rather than being part of the sensor capability itself. Hence the power demands are made from the action layer component (Sensing) rather than from the associated resource layer component (Sensors).

The following considerations are excluded:

- The planning of a sensing task.
- The processing of captured sensor data.
- The brokering of power.

C.5.2.1 Pre-Conditions

- The air vehicle is on route to be within its sensor range of the target.

C.5.2.2 View

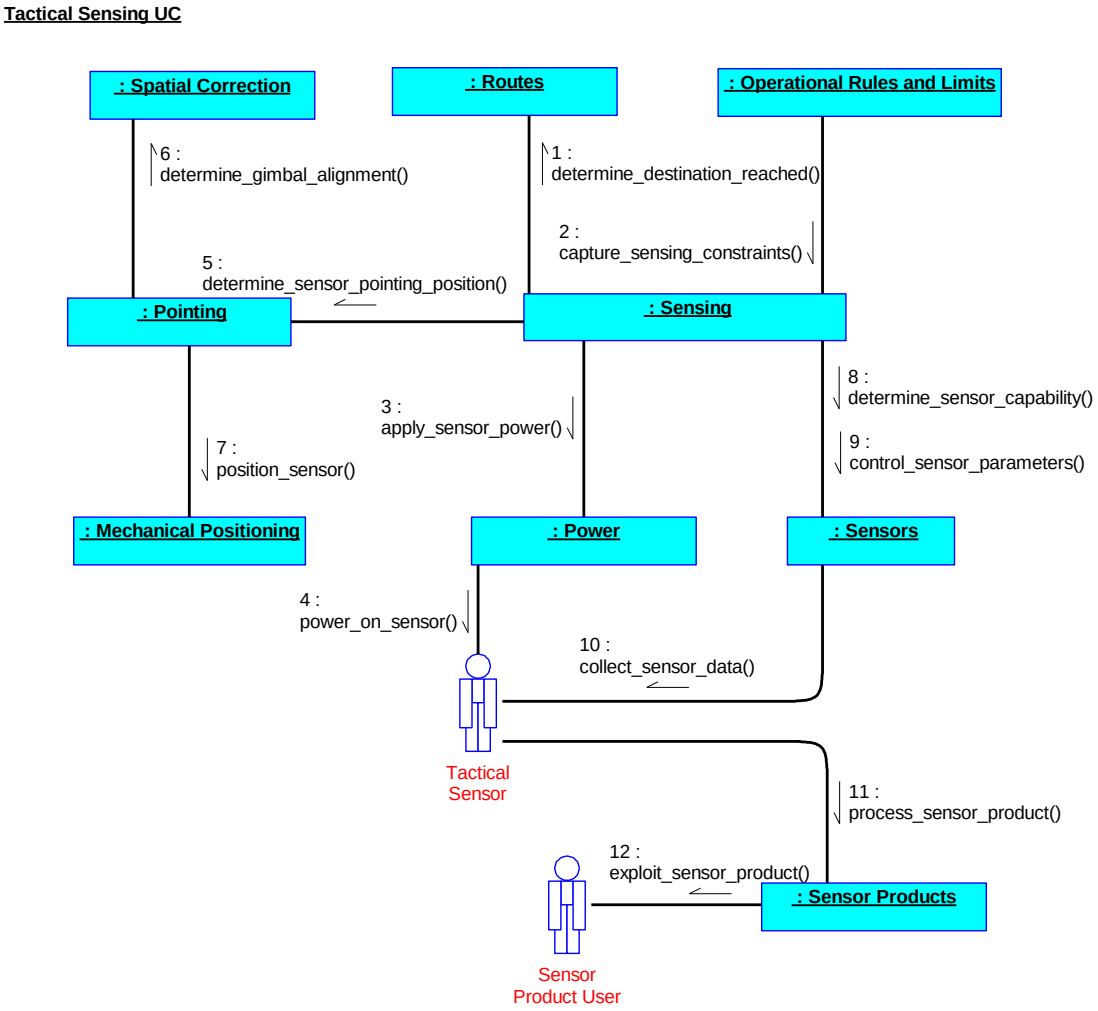

Figure 1341: Tactical Sensing IV

Sensing checks with Routes that the location for the sensing task has been reached, triggering Sensing to begin enacting the coordination of sensing activities.

Operating within the constraints set by Operational Rules and Limits, Sensing ensures power is applied to the Tactical Sensor via Power, and that the Tactical Sensor is positioned correctly via Pointing, Spatial Correction and Mechanical Positioning. Sensing commands Sensors to apply the relevant modes, settings and demands onto the Tactical Sensor in order to execute the sensing actions.

Sensor data is provided to Sensor Products for processing, until the sensing requirement has been fulfilled, and the resulting sensor product is distributed to any relevant Sensor Product Users.

C.5.2.3 Post-Conditions

- A sensor product has been generated for exploitation by a Sensor Product User.

C.5.2.4 Actors

Sensor Product User

Any element, either internal or external to the system, that utilises sensor products.

Tactical Sensor

The equipment of any sensor used for tactical aims (e.g. radar, IRST, or EO camera).

C.5.3 Sensor Data Interpretation

A task has been planned to maintain situation awareness of objects within the immediate battlespace around the ownship. As part of planning, Tasks has not set up coordination between components executing the task, so any refinements of this task solution are coordinated through Tasks. This IV starts with the solution to the task being executed.

It is assumed that:

- The sensor selected as part of the sensing solution is only being used for this task.

The following related areas of functionality are excluded:

- Exchange of tactical information with an external system.
- The execution of sensing activities to generate sensor products.
- The planning of a sensing task.
- Determination of own air vehicle location and kinematics.
- Sanitisation of an area (interpretation of data to determine confidence of a lack of objects in an area).

C.5.3.1 Pre-Conditions

- The Sensor Products and Data Fusion components are configured with necessary algorithms.
- The ownship location is known.

C.5.3.2 View

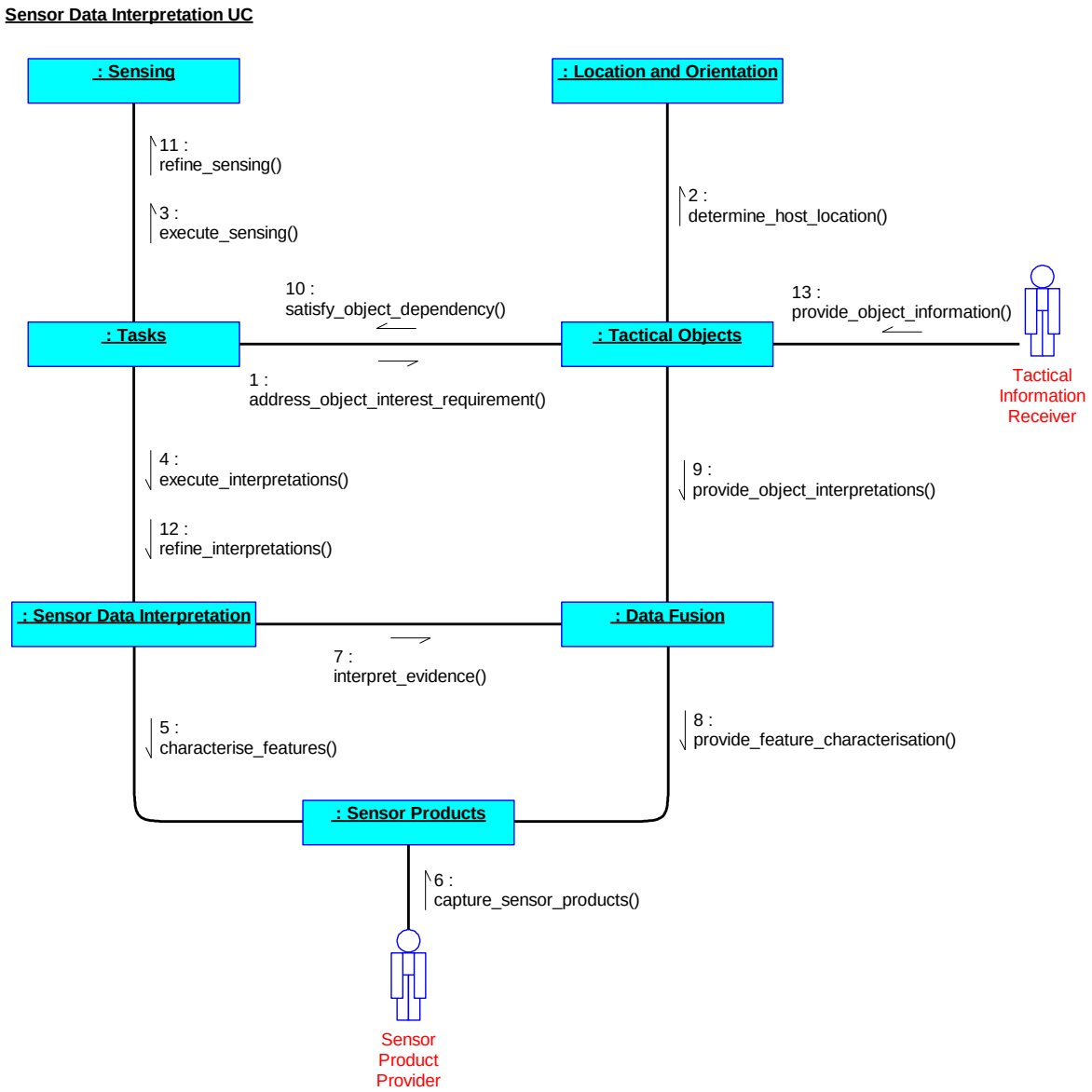

Figure 1342: Sensor Data Interpretation IV

To initiate execution of the task, Tasks registers an interest in objects within a region surrounding the air vehicle by placing an object interest requirement on Tactical Objects. As part of this, the characteristics of the objects that are of interest (e.g. position and classification) and the desired level of quality of these characteristics are defined. Tactical Objects uses Location and Orientation to determine the host location. Tasks also commands the execution of the planned sensing and interpretation solutions (in line with the exclusions the sensing execution is not shown in detail here).

For the interpretation solution, Sensor Data Interpretation commands Sensor Products to identify and characterise features, and Data Fusion to interpret and combine those detections to generate interpretations of objects.

Sensor Data Interpretation continuously monitors progress against the requirement and initiates any remedial action.

These processes lead to the generation of objects within Tactical Objects. Based on the interest requirements it has been given, Tactical Objects will identify dependencies to Tasks in order to maintain and gather more information about particular objects. Two examples where this could occur are:

- If an object of unknown classification is identified within the region of interest, Tactical Objects will identify a dependency to determine the classification of that object.
- If an object which has been deemed of interest is not of the required quality or begins dropping in quality, Tactical Objects will identify a dependency to increase the quality of that object.

Based on the dependencies identified, Tasks will refine sensing actions (e.g. use more sensor resources, or focus sensing onto a particular object) and interpretations actions (e.g. process a new type of sensor product, apply new processes) to maintain information or gather information about the objects specified in those dependencies.

Throughout this flow, object information will be provided by Tactical Objects to Tactical Information Receivers.

C.5.3.3 Post-Conditions

- Tactical Information Receivers have knowledge of objects surrounding the vehicle.

C.5.3.4 Actors

Sensor Product Provider

An entity which can provide sensor products to the system, this could be a particular tactical sensor (e.g. radar).

Tactical Information Receiver

An entity which has an interest in tactical objects and the information about them to fulfil its own responsibilities. Examples of this could be other tactical information components (e.g. Threats or Susceptibility) or entities which need to make decisions which are based on tactical information such as an authorised operator or the Tasks component.

C.6 Support Functions Views

This section contains Interaction Views relating to functions that support the operational use of a system.

C.6.1 Startup and Shutdown

C.6.1.1 Startup

This IV illustrates the coordination of startup via Asset Transitions, with support from Interlocks. The startup process includes activating the internal power, pumping fuel to the engines and triggering the engine ignition once appropriate authorisation has been obtained.

This IV assumes:

- The Exploiting Platform is starting up to an operational mode.
- The Exploiting Platform has been provided with limited (potentially external) power as required to power up and allow built-in tests to be run.
- The Exploiting Platform has been provided with HMI and/or communications to allow Authorisation to confirm required permissions for startup.
- The Exploiting Platform startup can be completed normally.

This IV excludes:

- Initial power up (possibly by the application of external power) to bring the system to the level of operation required to control safe startup.
- Any required Network Initialisation to support communications as part of startup.
- Required communication support between the Exploiting Platform and external actors.
- Internal capability assessment by a component.
- Pre-mission data load and the loading of physical stores.
- Carrying out post-startup role fit discovery.

C.6.1.1.1 Pre-Conditions

- The vehicle is in a configuration ready for transfer to internal power and engine start.
- The vehicle internal power and propulsion are inactive.

C.6.1.1.2 View

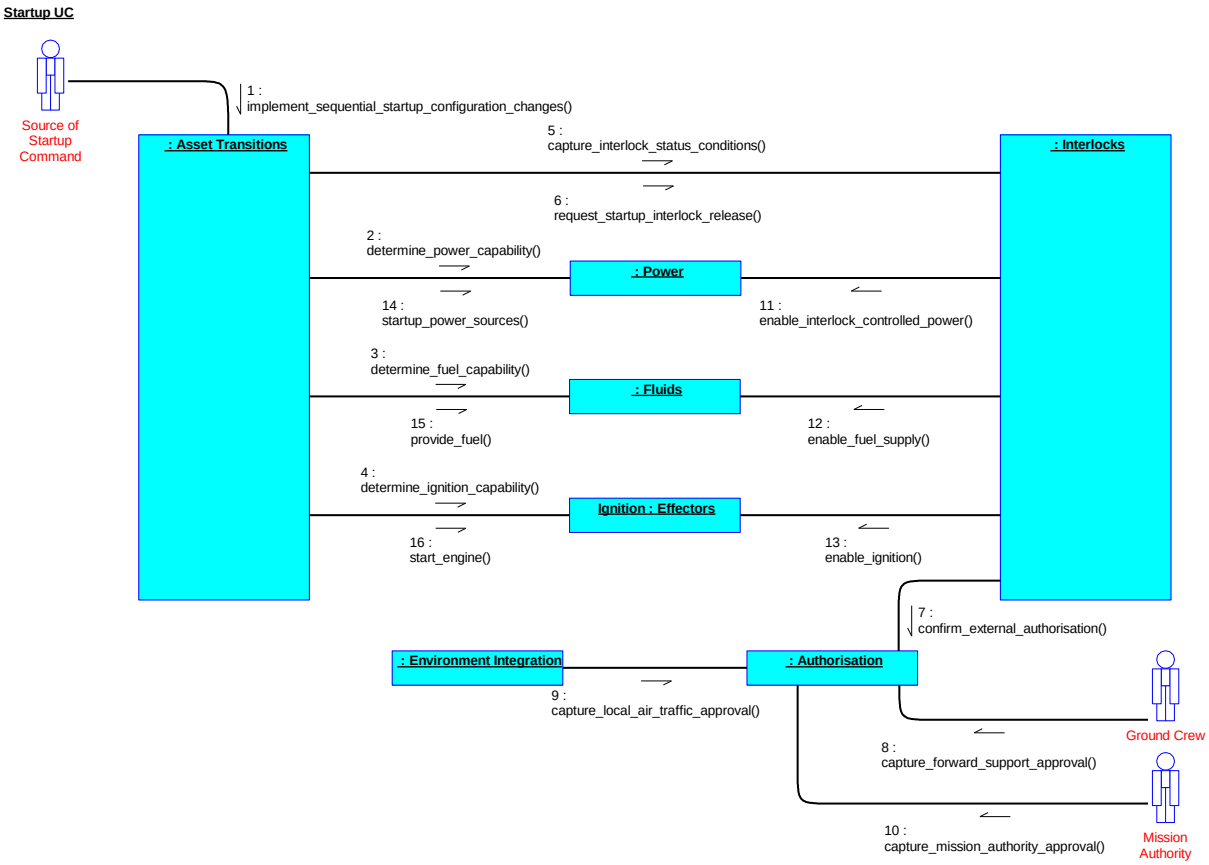

Figure 1343: Startup IV

The Source of Startup Command sends an instruction to carry out vehicle startup to Asset Transitions.

Power, Fluids and Effectors report their capability to Asset Transitions in accordance with the Capability Assessment policy.

Interlock status conditions are captured by Interlocks, along with conditions (such as external authorisation) required to ensure that the vehicle can safely startup.

Authorisation confirms approval from the Ground Crew attending the vehicle (such as safe distances to prevent hazards from radiating communications), local traffic approval (if not provided directly to a Mission Authority) via Environment Integration and that final approval from the Mission Authority has been received for each stage of startup.

During this process, the vehicle may be required to provide a ground crew communications relay between Ground Crew and the Mission Authority in accordance with the Human Communications IV.

Asset Transitions implements the startup configuration changes. Power and Fluids move to their active states then the engine is started via the Effectors.

C.6.1.1.3 Post-Conditions

- The vehicle power and engine are now in an active state.
- The Power component is able to apply power to equipment as required.

C.6.1.1.4 Actors

Ground Crew

The ground crew (and attendant ground support equipment) responsible for the vehicle, and who may be required to give approval for the startup or shutdown of vehicle (either that the vehicle is in a safe state or that the ground crew is safely clear of the vehicle prior to startup).

Mission Authority

The external authority responsible for authorising the vehicle startup or shutdown.

Source of Startup Command

The source of the instruction to Asset Transitions to implement a startup configuration. This may represent a direct instruction from an external entity or an instruction from a component higher in the control architecture.

C.6.1.2 Shutdown

This IV illustrates the coordination of a shutdown via Asset Transitions, with support from Interlocks. The shutdown process confirms that shutdown is authorised and then deactivates the Exploiting Platform's engine and power supply.

This IV assumes:

- The Exploiting Platform's engine and internal power is active, though the engine is at an idle state.
- The Exploiting Platform has reached its final position and taxiing activities are completed.
- External power is available if required to support shutdown of internal power.
- The Exploiting Platform shutdown can be completed normally.

This IV excludes:

- Post-mission data extraction and the removal of physical stores.
- Internal capability assessment by a component in accordance with the Capability Assessment policy.
- Required communication support between the Exploiting Platform and external actors.
- Sanitisation of the Exploiting Platform (for example by Cryptographic Materials and Storage) to make the Exploiting Platform secure from a data perspective.

C.6.1.2.1 Pre-Conditions

- The vehicle is in a configuration ready for shutdown.

C.6.1.2.2 View

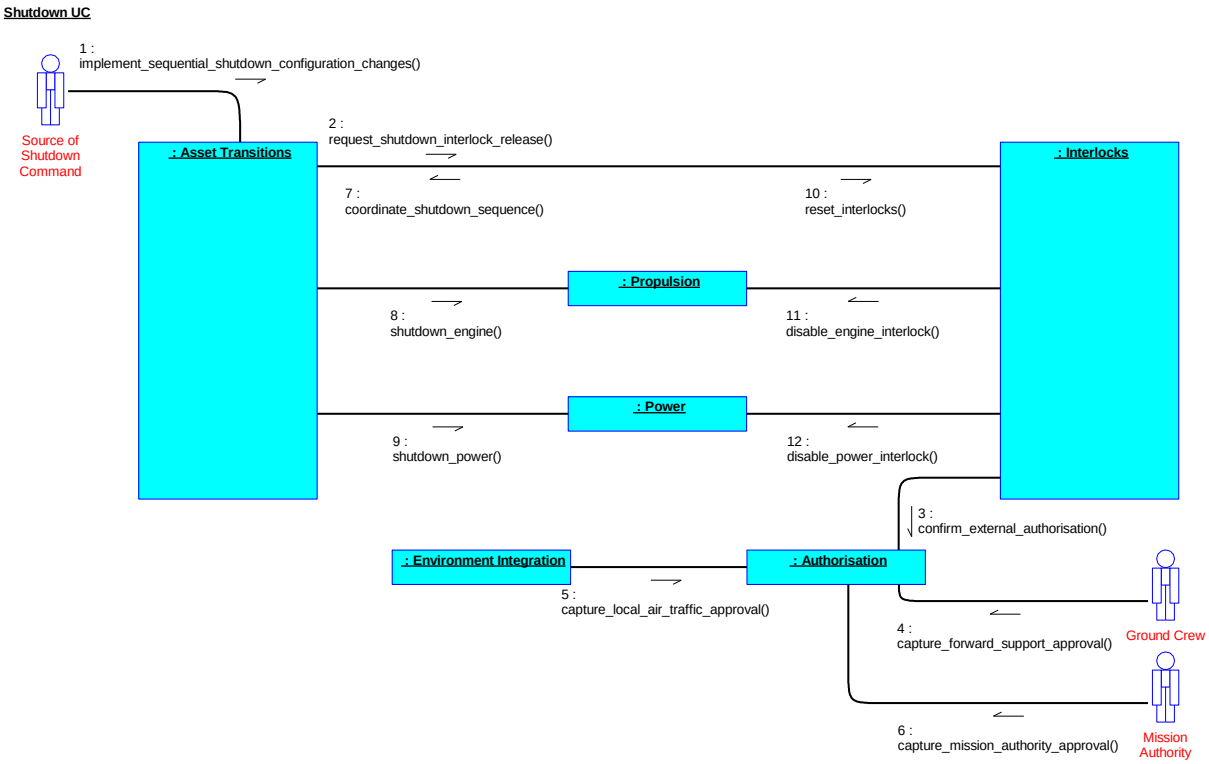

Figure 1344: Shutdown IV

The Source of Shutdown Command sends an instruction to carry out vehicle shutdown to Asset Transitions.

Asset Transitions confirms through Interlocks that a shutdown is allowed.

Interlocks performs any conditional checks and requests shutdown approval through Authorisation to ensure that the vehicle can safely shutdown.

Authorisation confirms that shutdown approval has been obtained from the required sources:

- Ground Crew attending the vehicle.
- Local traffic approval (if not provided directly to a Mission Authority).
- Final approval has been received from the Mission Authority (as relevant) for each stage of shutdown.

During this process, the vehicle may be required to provide a ground crew communications relay between Ground Crew and the Mission Authority in accordance with the Human Communications IV.

Asset Transitions commands the shutdown configuration changes. Interlocks restores the interlock states to prevent a restart without proper approval and then Power and Propulsion move to their inactive states.

C.6.1.2.3 Post-Conditions

- All power has been removed from the aircraft.

C.6.1.2.4 Actors

Ground Crew

The ground crew (and attendant ground support equipment) responsible for the vehicle, and who may be required to give approval for the startup or shutdown of vehicle (either that the vehicle is in a safe state or that the ground crew is safely clear of the vehicle prior to startup).

Mission Authority

The external authority responsible for authorising the vehicle startup or shutdown.

Source of Shutdown Command

The source of the instruction to Asset Transitions to implement a shutdown configuration. This may represent a direct instruction from an external entity or an instruction from a component higher in the control architecture.

C.6.2 Health Management

C.6.2.1 Fault Investigation

This IV demonstrates how a hardware fault is identified by using an anomaly as a trigger event; an anomaly is detected, investigated, a fault is then identified, characterised and published. This particular IV describes a health change associated with an effector, which was detected by a strategically positioned sensor.

In this scenario the Sensor Monitoring Effector (Hardware) is depicted as being a separate entity in its own right. In some instances the Sensor Monitoring Effector (Hardware) and the Effector (Hardware) may be embodied in the same item of equipment. Any item of equipment that provides information which is relevant to health can be regarded as a health sensor.

It is assumed that:

- The system has no existing faults.

The following considerations are excluded:

- This IV does not demonstrate Health Assessment components working and communicating with other Health Assessment components in a hierarchy, which would be relevant in more complex health scenarios.
- HMI interactions between Health Assessment and the Maintenance Planner.
- Behaviour associated with how Cyber Defence analyses an anomaly.
- Behaviour associated with Test.
- Behaviour associated with how system capability is determined based on a published anomaly or health assessment.
- Problem resolution is excluded from this IV. It does not cover any of the following:
  - Replacing/repairing the fault candidate.
  - Fault accommodation at a hardware level.
  - Replanning the capability.

C.6.2.1.1 Pre-Conditions

- No pre-conditions are identified.

C.6.2.1.2 View

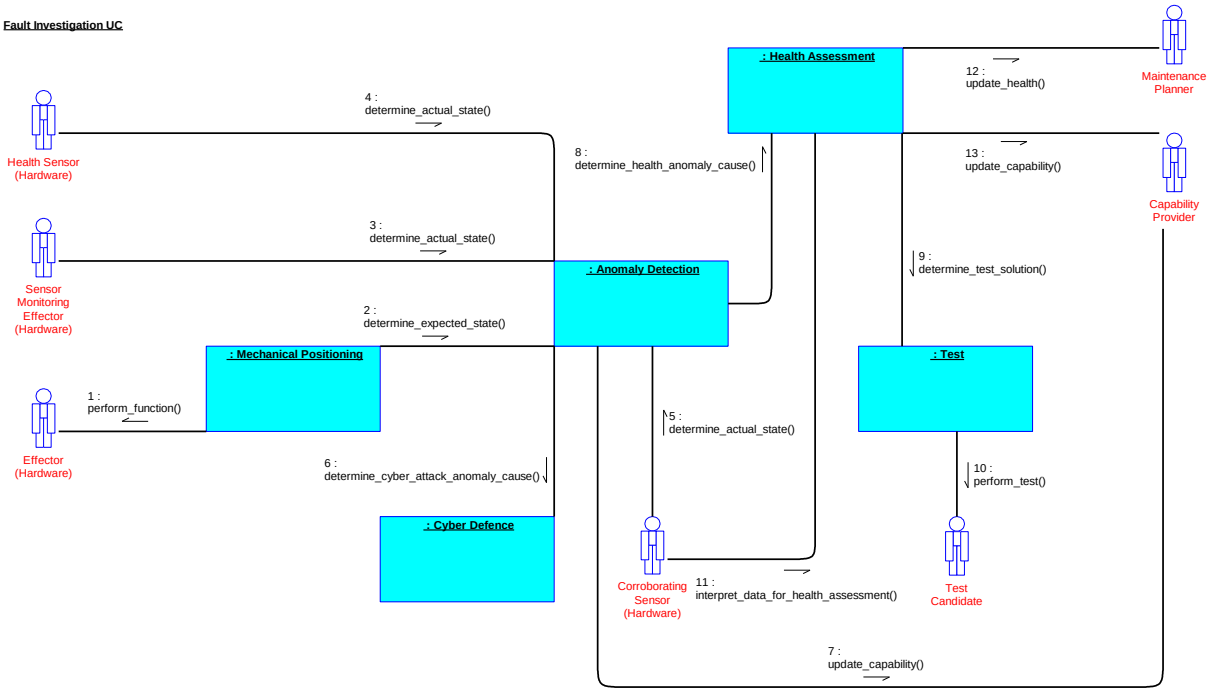

Figure 1345: Fault Investigation IV

Anomaly Detection monitors interactions between PYRAMID components in order to identify interactions that would result in a change of state of the system. If Anomaly Detection identifies an interaction that would result in a change, it will create an expected state (the state the system is expected to transition into when the command is fulfilled).

When the Effector (Hardware) performs a function, Anomaly Detection will determine if the Effector (Hardware) has transitioned into the expected state. It does so by monitoring any relevant Sensor Monitoring Effector (Hardware) to determine the Effector (Hardware) actual state. This can include monitoring any relevant Health Sensor (Hardware) and Corroborating Sensor (Hardware). Once the Effector (Hardware) actual state has been determined it is cross referenced against the expected state that was created. If the actual state and expected state do not concur, this is classified as an observed anomaly.

Anomaly Detection publishes the observed anomaly to Cyber Defence, any relevant Capability Providers and Health Assessment.

Health Assessment analyses the observed anomaly in order to determine if a health issue is the root cause of the anomaly. If the root cause cannot be ascertained simply by analysing the observed anomaly, Health Assessment can attempt to gather further information. This can be achieved by looking at information provided by Corroborating Sensor (Hardware), or by requesting Test to initiate any relevant built in tests. Initiating a test is covered in more detail in the Test policy. Any information provided by the test becomes available to Health Assessment and Anomaly Detection via a Corroborating Sensor (Hardware).

Health Assessment analyses the additional information to attempt to determine the root cause of the anomaly. It may continue to request further information to refine its assessment and increase the certainty of its final answer.

Once Health Assessment has determined the root cause of the anomaly, it will notify the Maintenance Planner so that a maintenance solution can be identified and also notify the Capability Provider so that it can assess how the failure affects its capabilities.

C.6.2.1.3 Post-Conditions

- The Maintenance Planner and relevant Capability Providers are aware of health changes.

C.6.2.1.4 Actors

Capability Provider

A PYRAMID software component that provides one or more capabilities to the overall system. This capability may be affected by one or more health changes.

Corroborating Sensor (Hardware)

A sensor that provides additional information to aid health assessment in its root cause analysis (e.g. test results).

Effector (Hardware)

An item of equipment that is responsible for performing a function.

Health Sensor (Hardware)

A sensor that generates health data.

Maintenance Planner

The person responsible for analysing the health data and deciding on the most appropriate maintenance action.

Sensor Monitoring Effector (Hardware)

A sensor that is strategically positioned to monitor a specific property of an effector.

Test Candidate

The item of equipment responsible for performing the test.

C.6.2.2 Life and Usage

This IV demonstrates how life and usage information is collected, analysed and reported.

This scenario considers a health specific sensor whose role is purely to collect health data about a discrete element of a system so that Health Assessment can calculate total amount of usage consumed by that discrete element, calculate total remaining usage and then identify if this information needs reporting. Life monitoring uses a similar mechanism to monitor the duration an element is in operation.

It is assumed that:

- The system has no existing life and usage exceedances.

The following considerations are excluded:

- Behaviour associated with abnormal life or usage consumption. This would be treated as an anomalous event and be handled in the normal way by Anomaly Detection.
- HMI interactions between Health Assessment and the Maintenance Planner.

C.6.2.2.1 Pre-Conditions

- Life and usage thresholds are pre-defined and supplied to Health Assessment prior to initialisation.

C.6.2.2.2 View

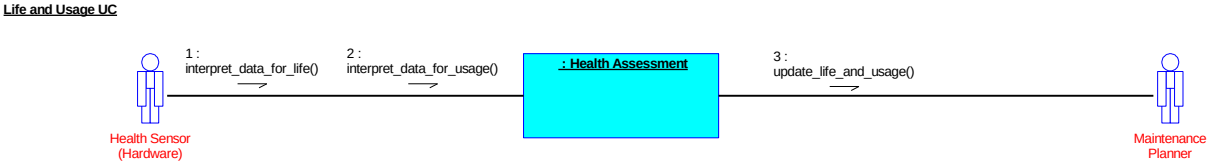

Figure 1346: Life and Usage IV

Health Sensor (Hardware) are strategically placed in order to collect health information on certain aspects of discrete elements within a system. A Health Sensor (Hardware) provides health data to Health Assessment. Health Assessment analyses the health data to determine the total amount of life or usage consumed by a discrete element and then determines the total amount of remaining life or usage.

Health Assessment compares the total remaining life or usage against a predefined threshold and determines that the threshold has been exceeded.

Health Assessment provides the life and usage information to the Maintenance Planner so that they can update their life and usage records and so that a maintenance solution can be identified.

C.6.2.2.3 Post-Conditions

- The Maintenance Planner is aware that the usage threshold has been exceeded.

C.6.2.2.4 Actors

Health Sensor (Hardware)

A sensor that generates health data.

Maintenance Planner

The person responsible for analysing the health data and deciding on the most appropriate maintenance action.

C.6.3 EM Interoperability

The system has to manage electromagnetic emissions in order to achieve EM interoperability. This IV illustrates how the EM emissions are managed to enable interoperability by using the blanking interoperability mechanism.

Even though this use case only considers blanking and sensor processing, there are other interoperability mechanisms that could have been used in its place. Example EM interoperability mechanisms are:

- Blanking - Informing a receiver to not receive anything.
- Rejection - Reject any data received.
- Suppression - Denial of transmission.
- Processing - Eliminating spurious data by post-processing.

This use case only uses Effectors and Sensors. Other types of EM resources (e.g. Communicator) can be managed in the same manner at the same time.

The numerical bandwidths used in this use case are fictitious and do not represent the bandwidths at which these types of resource would function.

It is assumed that:

- Use of the resources has been planned in adherence with the Resource Management policy.

The following considerations are excluded:

- Influencing the direction or zones of transmission and reception associated with EM interoperability. I.e. the equipment is assumed to spatially and chronologically interfere, and methods such as repositioning the transmitters or receivers, beamforming, or rescheduling other activities (in accordance with the Resource Management policy) are not possible in this IV.
- The mechanisms used for detecting and reporting threats to Countermeasures.

C.6.3.1 Pre-Conditions

- The conditions for the use of EM transmissions have been loaded into Operational Rules and Limits.
- Spectrum has been loaded with the EM Interoperability mechanisms.
- No countermeasures are active.

C.6.3.2 View

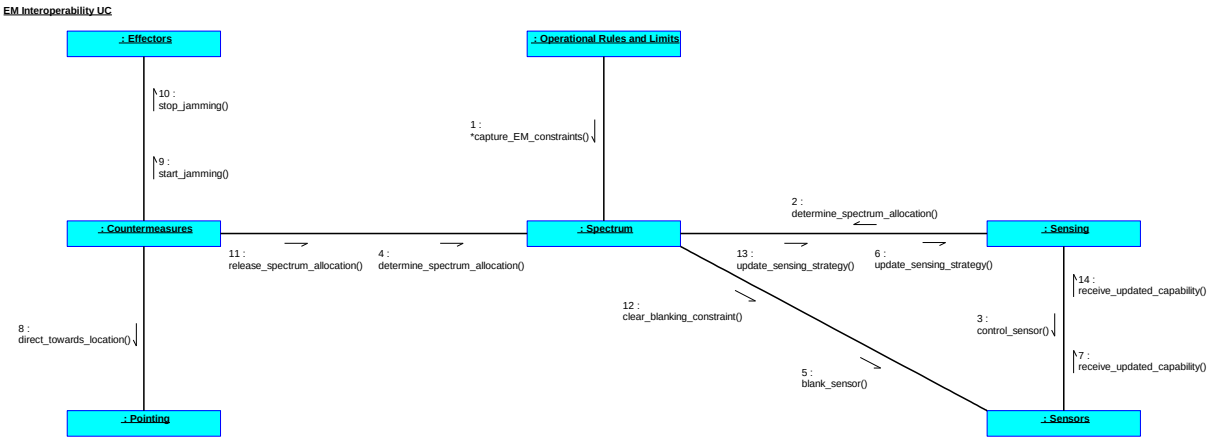

Figure 1347: EM Interoperability IV

Operational Rules and Limits provides limits to Spectrum that affect spectrum usage. Sensing requests use of the 2 GHz frequency from Spectrum. Once allocated, Sensing directs the Sensors to perform sensing actions.

During the operation, Countermeasures is made aware of a threat and requests a spectrum allocation of 2 GHz frequency from Spectrum.

Spectrum will arbitrate between the demands from Countermeasures and the existing Sensing allocation. Spectrum restricts use of the 2 GHz frequency by blanking Sensors. Spectrum also informs Sensing of that restriction. Sensors reports to Sensing its loss of capability.

The 2 GHz frequency is allocated to Countermeasures as requested. Countermeasures instructs Pointing to direct the effects of the jamming pod towards the threat location, and instructs Effectors to activate jamming.

Once the threat is avoided, Countermeasures instructs Effectors to de-activate jamming, and informs Spectrum to release the allocation.

Spectrum removes the blanking restriction from Sensors and allocates the use of the 2 GHz frequency to Sensing. Sensors reports to Sensing that its capability has been restored.

C.6.3.3 Post-Conditions

- Countermeasure emissions have completed.
- Sensor equipment is operating at 2 GHz frequency.

C.6.4 Cryptographic Management

C.6.4.1 Cryptographic Device Management

This IV illustrates the configuration of encryption in response to a request for integrity protection for data-in-transit. Other permutations of confidentiality, integrity and availability protection configuration for other uses follow the same pattern.

It is assumed that:

- Cross domain behaviour will not change if multiple security domains are involved.
- If cryptographic material is distributed encrypted (e.g. black keys) then the appropriate cryptographic material for decryption has already been distributed. Note: at some point in the hierarchy of keys, unencrypted cryptographic material has to be supplied (e.g. red keys) otherwise encrypted material can never be decrypted.

The following considerations are excluded:

- The encryption of data-in-transit is not shown, only the setting-up of encryption.

C.6.4.1.1 Pre-Conditions

- The plan of which cryptographic devices are to be used for which task in the mission is already loaded in the system.
- Cryptographic material is already loaded in the system in an appropriate cryptographic store.
- The data exchange requirements are pre-configured within the relevant components.

C.6.4.1.2 View

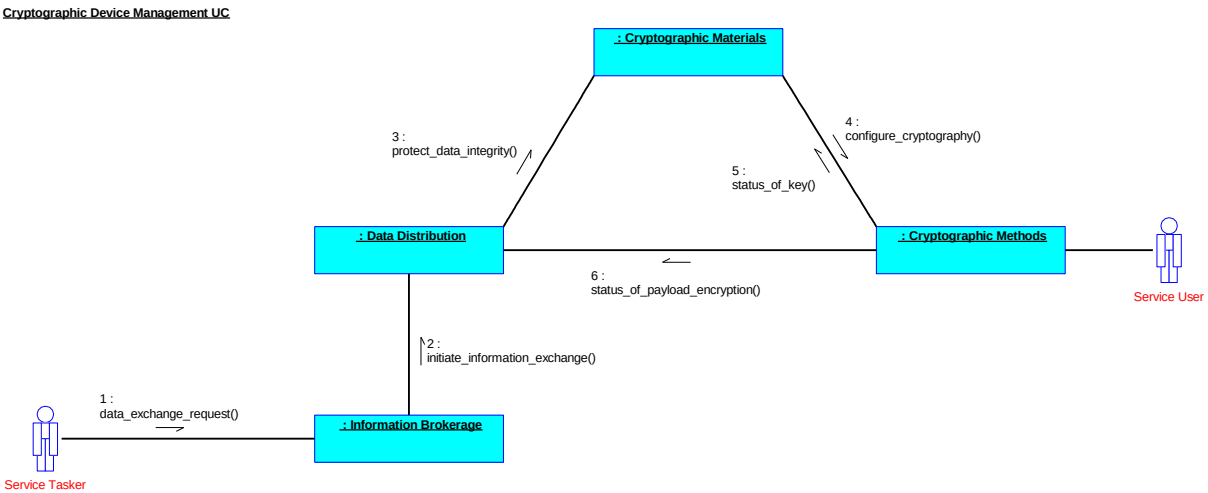

Figure 1348: Cryptographic Device Management IV

Following a request for an information exchange from the Service Tasker, Information Brokerage determines that integrity protection is required. This requirement to carry out a secure information exchange is placed onto Data Distribution and Data Distribution requests a protection solution from the Cryptographic Materials component. Cryptographic Materials determines what protection is required and provides the required information to Cryptographic Methods, including any cryptographic key material needed by the Cryptographic Methods component.

Cryptographic Methods may be used by components as a resource, without involvement of action components or Cryptographic Materials.

Cryptographic Methods reports security related information (status of keys) back to Cryptographic Materials and non-classified performance related information (status of payload protection) back to the Data Distribution component.

C.6.4.1.3 Post-Conditions

- The appropriate encryption has been prepared to preserve the integrity of data-in-transit.

C.6.4.1.4 Actors

Service Tasker

The user or internal component that requests a service that indirectly requires cryptographic support.

Service User

The user of the cryptography service.

C.6.4.2 Cryptographic Material Revocation

This IV illustrates the revocation of a cryptographic certificate that has been identified as compromised, and is currently used in data-at-rest protection. Multiple instances of Cryptographic Methods are involved in the deployment, and are using shared cryptographic material. Other permutations of revocation will follow the same pattern.

It is assumed that:

- Cross-domain behaviour will not change if multiple security domains are involved.

The following considerations are excluded:

- The use of cryptography is not included in this interaction diagram only the revocation of cryptographic material.
- Actions triggered by Storage as a result of the certificate being compromised, which do not relate to cryptography are not shown.

C.6.4.2.1 Pre-Conditions

- The plan of which cryptographic devices are to be used for which task in the mission is already loaded in the system.
- Cryptographic material is already loaded in Crypto Devices.

C.6.4.2.2 View

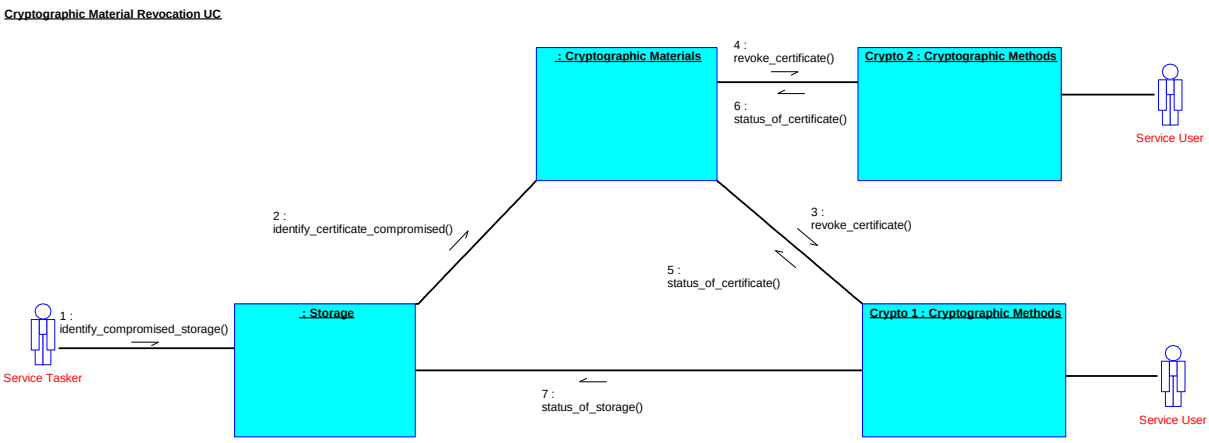

Figure 1349: Cryptographic Material Revocation IV

The Service Tasker will inform the Storage component that the store protection has been compromised. The Storage component, in consultation with other components (not shown) determines the impact on the system of the reduced trust in the storage. Storage informs Cryptographic Materials that a store has been compromised. Cryptographic Materials then determines which certificate has been compromised.

As the certificate has been compromised the certificates must not be used by any device. The Cryptographic Materials component is responsible for keeping track of keys, algorithms and certificates in the system, so will determine which devices are using the certificate and request certificate revocation from all relevant Cryptographic Methods instances. As multiple Cryptographic Methods instances are to be updated Cryptographic Materials will determine the best order to perform this request.

Cryptographic Methods instances will inform the Cryptographic Materials component that the certificate has been revoked (as this component is responsible for security related information) and will inform Storage of the status of the compromised store (i.e. protection service no longer available for this store).

C.6.4.2.3 Post-Conditions

- Cryptographic material has been successfully revoked from multiple devices.

C.6.4.2.4 Actors

Service Tasker

The user or internal component that requests a service that indirectly requires cryptographic support.

Service User

The user of the cryptography service.

C.6.4.3 Cryptographic Device Sanitisation

This IV illustrates the sanitisation of a communication cryptographic device as it is no longer required for the mission. Other permutations of sanitisation, including emergency sanitisation follow a similar pattern.

It is assumed that:

- Only the single device needs sanitising, if other devices needed sanitising this would be managed by the Cryptographic Materials component.

The following considerations are excluded:

- Non-cryptographic actions not shown in this IV, for examples of this behaviour see the Communications Views.

C.6.4.3.1 Pre-Conditions

- The plan of which cryptographic devices are to be used for which task in the mission is already loaded in the system.
- Cryptographic material is already loaded in cryptographic devices.

C.6.4.3.2 View

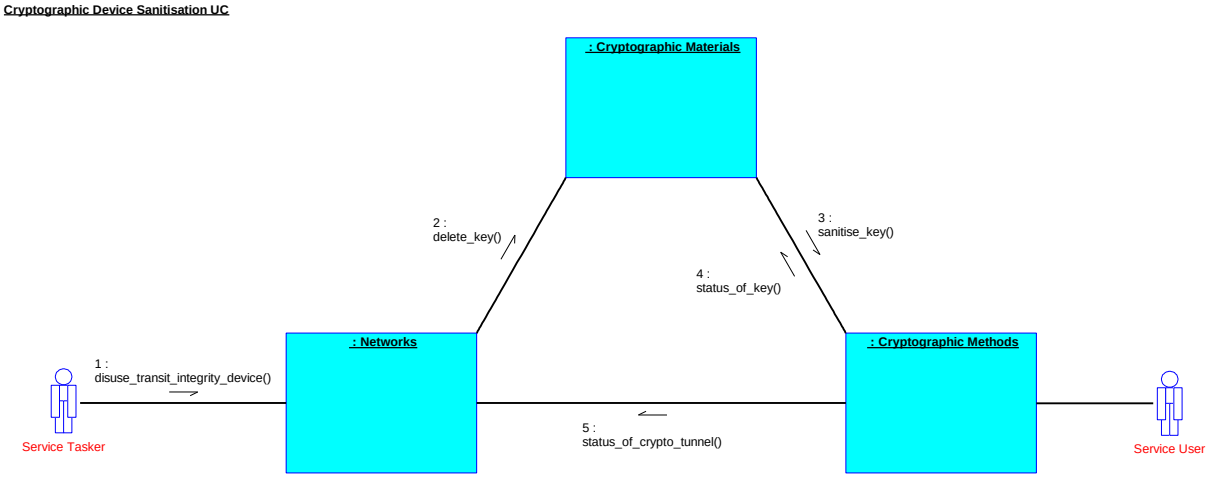

Figure 1350: Cryptographic Device Sanitisation IV

The Service Tasker will inform the Networks component that certain parts of the communications infrastructure are no longer required for the mission. Networks informs Cryptographic Materials that as part of the changes a crypto device is no longer needed. Cryptographic Materials determines that the redundant device can be cleared.

The Cryptographic Materials component determines which cryptographic material can be removed and requests sanitisation of the related Cryptographic Methods component.

On completion of the sanitisation Cryptographic Methods will inform the Cryptographic Materials component. Cryptographic Materials will mark the key material as no longer being used on that device. Cryptographic Methods will also inform the Networks component that the encryption service it was providing is no longer available. Based on this information the Networks component will determine that the encrypted tunnel is no longer able to pass traffic, which was the expected outcome.

C.6.4.3.3 Post-Conditions

- Cryptographic material has been successfully sanitised.

C.6.4.3.4 Actors

Service Tasker

The user or internal component that requests a service that indirectly requires cryptographic support.

Service User

The user of the cryptography service.

C.6.5 Defence Against Cyber Attack

This IV covers a cyber attack scenario whereby a vulnerability has been exploited by an adversary and a malicious payload activated.

Detection of an exploitation is achieved within the PRA by a multi-component analysis of multiple event sources that together can be used to identify a problem (e.g. a process gaining extra privileges combined with unusually large amounts of data being accessed). The "effect" is detected by monitoring the capability and health of the system and identifying any reduced or unexpected performance, analysing the cause and reacting appropriately.

This IV depicts a single cyber attack which contributes to a larger cyber attack. Here, an adversary has successfully delivered a malicious payload which manipulates sensor data resulting in spoofed tracks being generated. In response Anomaly Detection, Cyber Defence and Health Assessment work together to identify and, if possible, mitigate the attack.

This section follows the Health Management, Control Architecture and Cyber Defence policies. On the IV the Anomaly Detection component is a single instance associated with Data Fusion. If multiple components were displayed in Figure 1351: Defence Against Cyber Attack IV there would be multiple instances, which is aligned to the Health Management policy.

This IV only considers spoofed track(s) as an example. If other components have been attacked or other types of cyber attack methods have been deployed, the general interactions between the relevant components wouldn’t change.

It is assumed that:

- Defence against cyber attack will be fulfilled using a "Defence in Depth" approach, comprising personnel, physical, procedural and technical controls against cyber threats.

The following considerations are excluded:

- How an adversary successfully delivered the malicious payload.
- Personnel, physical and procedural controls are outside the scope of the PRA, as are technical controls residing below the application layer of the computing stack.
- Interactions between HMI components are not shown as these are shown elsewhere.
- Generation of a security log.

C.6.5.1 Pre-Conditions

- The cyber attack contingency tactics have been loaded to Contingency.
- Cyber Defence is loaded with and has knowledge of cyber attacks and their methods, including those associated with data fusion.
- An adversary has successfully delivered the malicious payload that manipulates sensor data to the air vehicle.

C.6.5.2 View

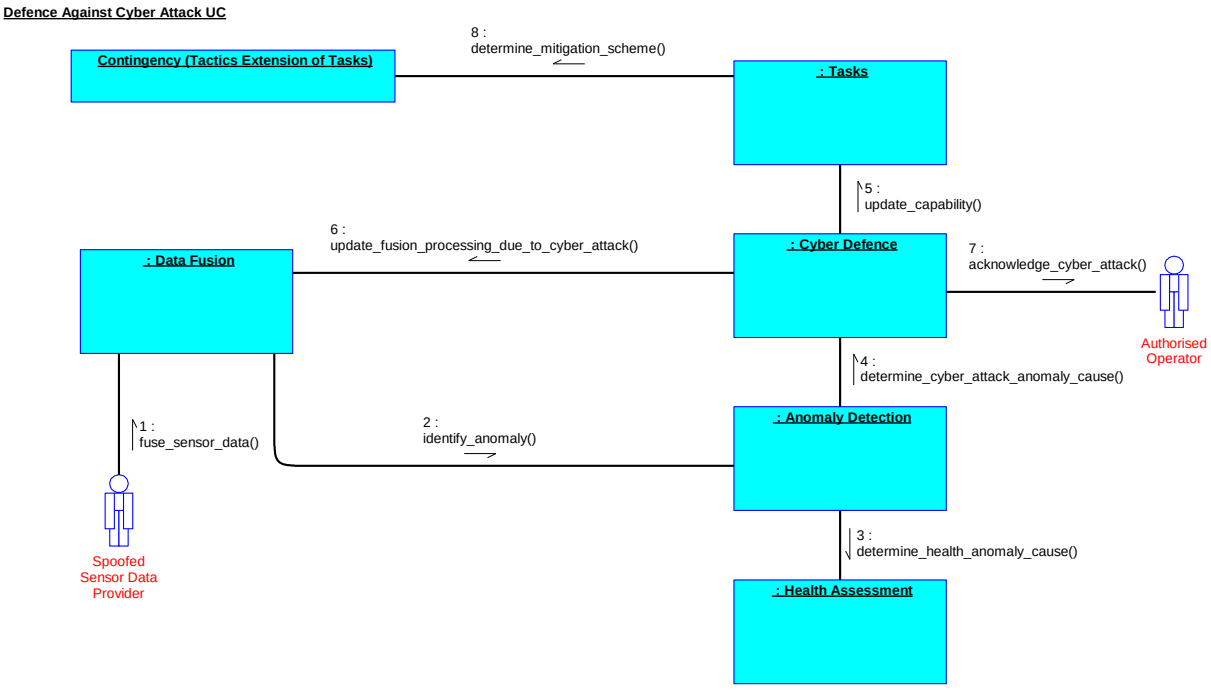

Figure 1351: Defence Against Cyber Attack IV

The Spoofed Sensor Data Provider provides manipulated sensor data to Data Fusion which then uses that data to maintain the interpretations.

The interpretations and other health related data are then collected by Anomaly Detection so that it can be interpreted and the anomalies can be identified.

Anomaly Detection identifies that there are anomalous Data Fusion results based on interpreting current results against a baseline of previous results. Anomaly Detection provides the anomalies to Data Fusion, Cyber Defence and Health Assessment. Cyber Defence and Health Assessment simultaneously assess the anomaly. Health Assessment does not identify a hardware issue and Cyber Defence determines that a cyber attack has taken place, in this case by comparing the results against a knowledge base of known cyber attack types, malware and threats.

Cyber Defence identifies the possible actions in response to the cyber attack and informs the affected components, as well as Tasks. As a result, Data Fusion determines that any sensor data received from the Spoofed Sensor Data Provider should be ignored. Additionally, where a pre-determined notification threshold has been exceeded, the Authorised Operator is informed that a cyber attack has been successful.

Tasks then produces a mitigation scheme using the Contingency Tactics Extension.

C.6.5.3 Post-Conditions

- The Contingency tactics extension has determined a mitigation scheme.

C.6.5.4 Actors

Authorised Operator

Any person with validated credentials allowed to interact with the system to carry out a system role.

Spoofed Sensor Data Provider

A software component of the system, such as Sensor Products, which has been corrupted into providing spoofed sensor data.

C.6.6 Generation of Reports

C.6.6.1 Generation of Handover Briefing

This IV shows how a handover briefing is created to support handover from one Authorised Operator to another. The briefing provides the incoming Authorised Operator with a summary of the mission status at the time of handover. The briefing will follow a general format, but the precise details will depend on the mission.

It is assumed that all required data has been successfully recorded.

The HMI and communication components required for the interface with the operators are excluded from this IV.

C.6.6.1.1 Pre-Conditions

None.

C.6.6.1.2 View

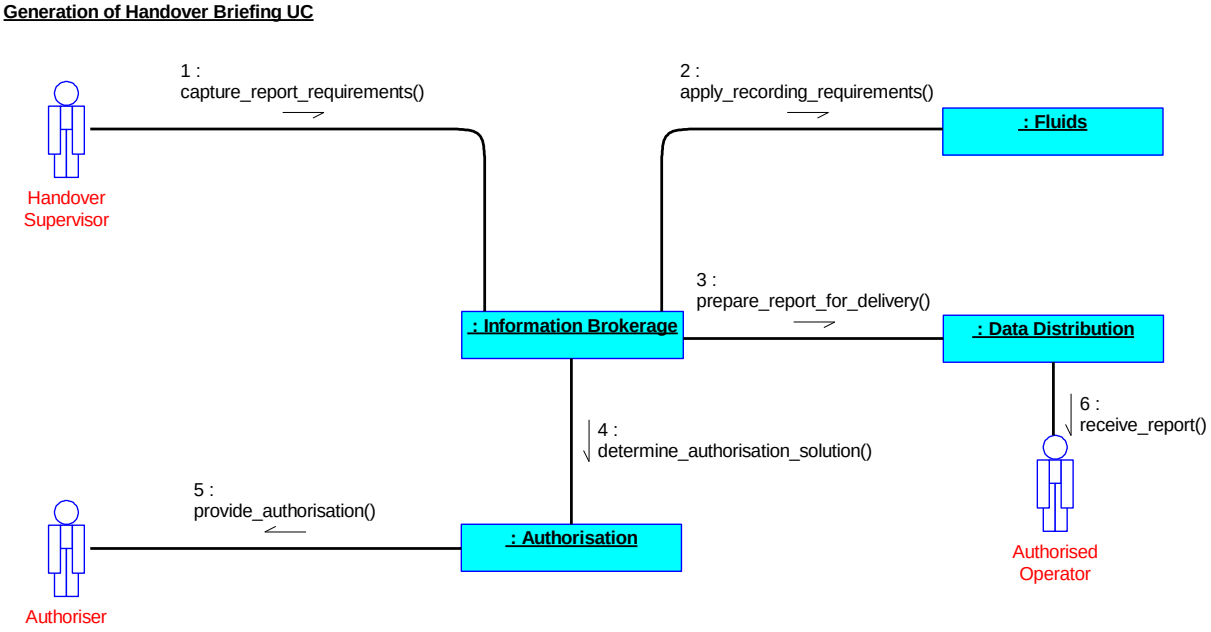

Figure 1352: Generation of Handover Briefing IV

The Handover Supervisor defines the required handover briefing. The precise contents of the briefing will depend on the mission details, so the briefing is defined in terms of rules for the data items that are to be included, based on events and mission conditions. Information Brokerage captures these requirements for the handover briefing and imposes requirements on the relevant components (represented here by Fluids) to record the necessary data.

During the mission, Fluids writes the fuel contents data to storage hardware by using the middleware.

When the time comes for handover, Information Brokerage applies the briefing rules and, based on mission events and conditions, instructs Data Distribution to collate the fuel contents records and other data items required. Once the report content has been collated, Information Brokerage requests Authorisation to attain release authorisation from the Authoriser.

The Authoriser fulfils the request to authorise release of the briefing to the Authorised Operator.

Information Presentation (not shown) then presents the briefing report.

C.6.6.1.3 Post-Conditions

- The handover briefing has been released to the Authorised Operator who is taking over control.

C.6.6.1.4 Actors

Authorised Operator

An operator whose control role will be handed over to another operator during the course of the planned mission.

Authoriser

An authorised operator of the system who is able to authorise the release of a handover report, or mission data for PMDH.

Handover Supervisor

An operator who will supervise the handover of control roles between operators during the course of a mission.

C.6.6.2 Generation of PMDH Report

This IV shows how a set of mission data is generated for Post Mission Data Handling (PMDH) by an Authorised External User. The data set will include detailed technical information about the conduct of the mission. The data set will follow a general format, but the precise details will depend on the mission.

It is assumed that all required data has been successfully recorded.

The HMI and communication components required for the interface with the operators are excluded from this IV.

C.6.6.2.1 Pre-Conditions

The mission has completed and all necessary data has been recorded.

C.6.6.2.2 View

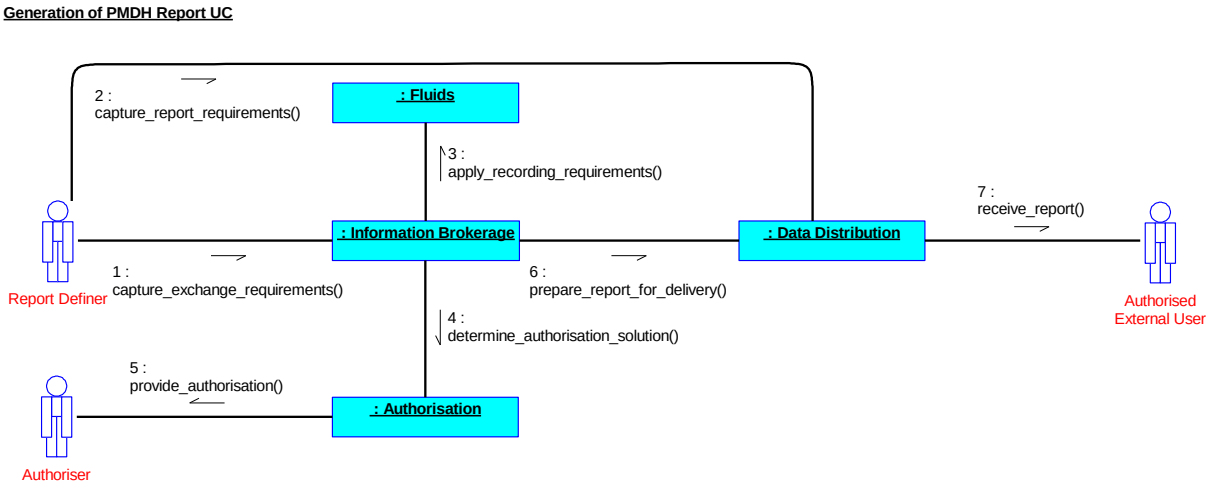

Figure 1353: Generation of PMDH Report IV

The Report Definer defines the PMDH data set. The precise contents of the data set will depend on the mission details, so it is defined in terms of rules for the data items that are to be included, based on events and mission conditions. Information Brokerage captures the required exchanges to collate the PMDH data set, and Data Distribution captures the requirement to collate the report. Information Brokerage imposes requirements on the relevant components (represented here by Fluids) to record the necessary data.

During the mission, Fluids writes the fuel contents data to storage hardware by using the middleware. During the mission, an error condition arises which causes Fluids to create a software error statement.

After the mission, Information Brokerage identifies the data items required by the PMDH data set rules based on the events and mission conditions. This includes the fuel contents and the software error statement. Information Brokerage requests Authorisation to obtain release authorisation from the Authoriser. The Authoriser fulfils the request for authorisation to release the data set. Data Distribution collates the PMDH data set from the relevant components, which is released to the Authorised External User for analysis.

C.6.6.2.3 Post-Conditions

- The PMDH data set has been released to the Authorised External User for analysis.

C.6.6.2.4 Actors

Authorised External User

An operator external to the operational system who will receive mission data for analysis once the mission is complete.

Report Definer

An operator of the system who is authorised to define PMDH data sets.

Authoriser

An authorised operator of the system who is able to authorise the release of a handover report, or mission data for PMDH.

C.6.7 Mission Data Load

This IV shows how a PYRAMID compliant deployment can derive and assemble the information that an operational system requires (the mission data load) to support a mission. The IV does not exclude the possibility that the generation of the mission data load could be executed in an independent mission planning system (see Operational Support policy).

It is assumed that:

- All necessary data libraries are available to the mission planning system.

The following considerations are excluded:

- The mechanism for physical transfer of the data load to the operational system.

C.6.7.1 Pre-Conditions

- A template has been created in Data Distribution which specifies the information that the mission data load could contain. The template defines the format for any category of data that could be required in the mission data load.
- The authorisation process for mission data loads has been defined.

C.6.7.2 View

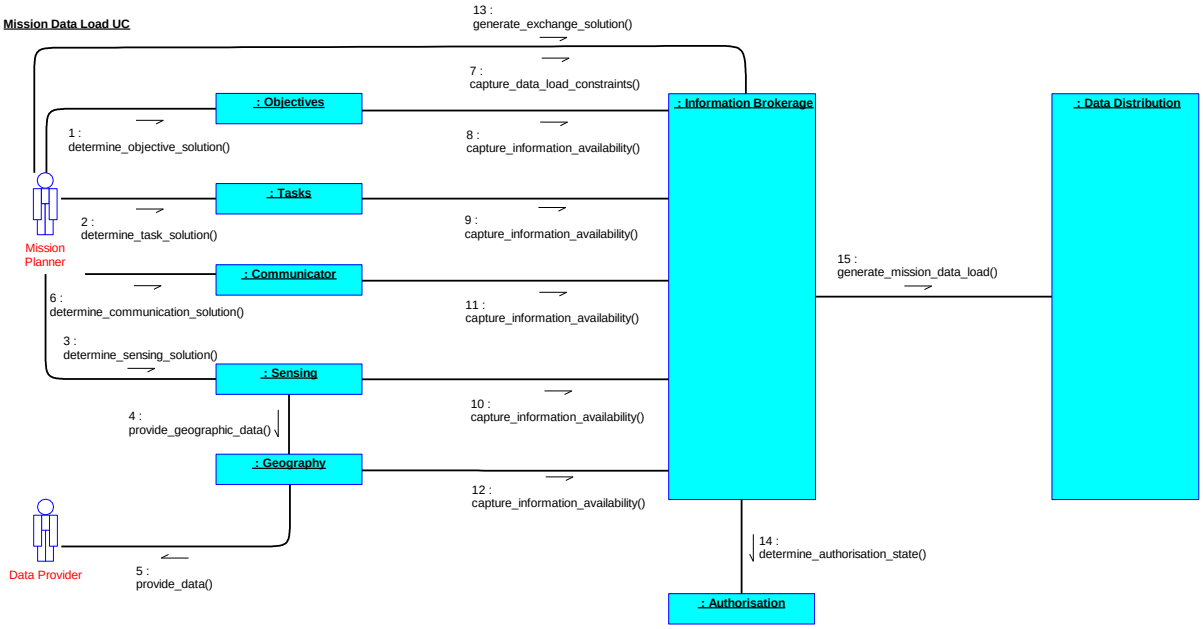

Figure 1354: Mission Data Load IV

The Mission Planner interacts with the system to create a mission plan. The Mission Planner interacts with components in the Objective, Task, Action and Resource Layers, as required (see the Control Architecture policy). In this example, Sensing is a component in the Action Layer and Communicator is a resource component. The Mission Planner refines the requirements for the data load, working within a template defined by the design authority (e.g. by constraining the data for distribution to a specific platform only).

When required, the Mission Planner triggers the generation of the data load. The Information Brokerage component determines the information exchanges making up the required data load solution, based on the template and the details of the mission plan. This includes supporting data identified by components (a need for data in Geography to support Sensing is shown as an example).

The Authorisation component checks the authorisation state. Once authorisation is confirmed Information Brokerage requests Data Distribution to compile the mission data load in preparation for delivery.

C.6.7.3 Post-Conditions

- A complete Data Load exists and is ready for transfer to and use by the operational system.

C.6.7.4 Actors

Data Provider

The provider of data (e.g. geographical data) that is required to support the mission plan.

Mission Planner

An authorised operator of the mission planning system who is responsible for developing the mission plan.

C.6.8 Request for Information

An Operator makes a request for information (RFI). This request is for any EO or IR images of a specific physical location. This IV shows how the system is used to search for the requested images and then distribute them to meet the requirements of the RFI. This includes ensuring the format is correct and the clearance of the Image Receiver is appropriate for the classification of the images produced.

This example IV can be used to represent the retrieval of any type of mission generated data, by replacing the represented instance of the Sensor Products component with the appropriate component(s) responsible for the subject matter involved with the query, as each component is responsible for managing its own data including its associated storage.

It is assumed that:

- Images of the requested location have been produced.
- The Image Receiver is already known to the system, including aspects such as communication methods, clearance level, etc.
- The HMI Dialogue and Information Presentation have been configured for the Operator's role. This enables them to interact appropriately with the system.
- The system has been configured so that Information Brokerage is informed of the properties of images whenever the new image data are created (e.g. classification and time created).
- Authorisation to release images to the Image Receiver has been given as long as specific criteria are met.

The following considerations are excluded:

- How the images are distributed using the communications hardware.
- Interactions with storage.

C.6.8.1 Pre-Conditions

- A request for image data has been received

C.6.8.2 View

Request for Information UC

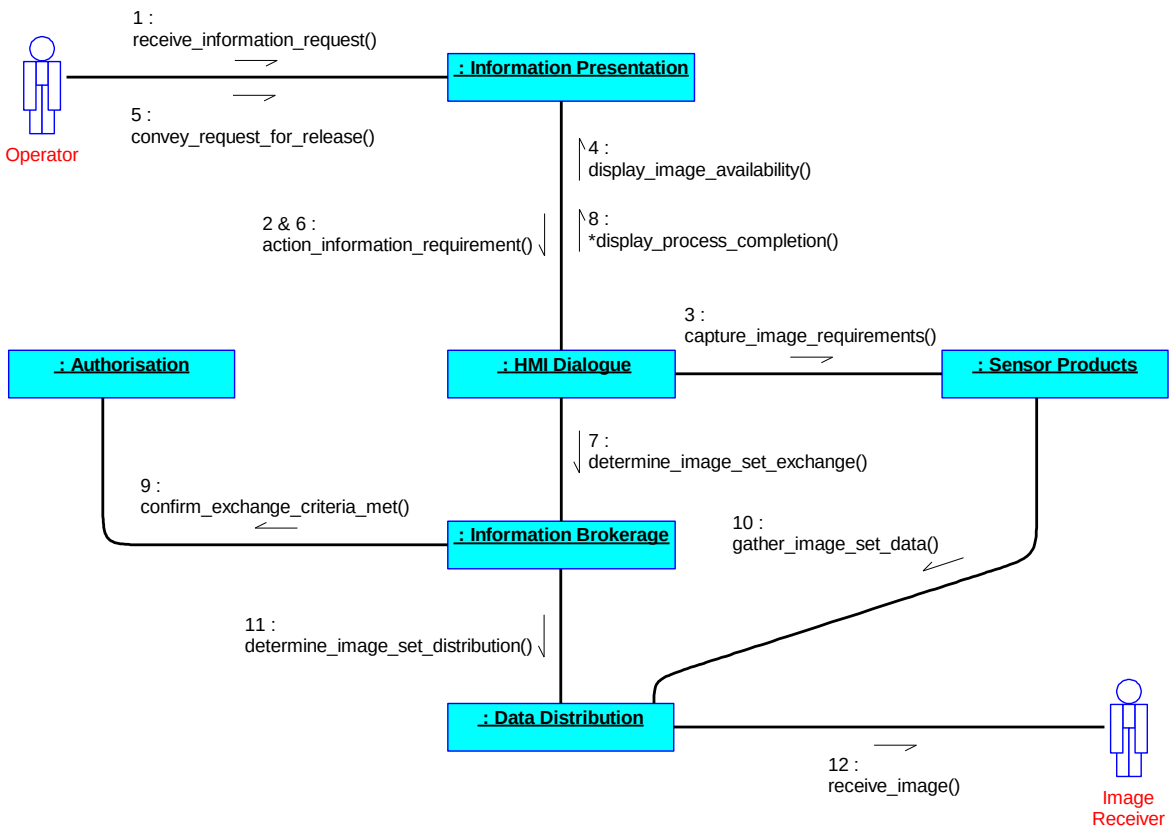

Figure 1355: Request for Information IV

To determine if the RFI can be met, the Operator will query the system to see if any stored images match the specified location. The Operator interacts to insert the search request via Information Presentation. HMI Dialogue then places a requirement on Sensor Products to provide an image of the specified location: this is only pulling on existing data. Sensor Products determines it holds images that meet the requested criteria. The Operator is then informed that the images exist, therefore the RFI can be met.

The Operator then decides that the images need to be released to the Image Receiver and inputs this request into the HMI via Information Presentation. HMI Dialogue responds to this request by requesting Information Brokerage to determine the exchange solution, to ensure the exchange is allowable and determine if any reclassification of data is required due to multiple data sources. Information Brokerage requests Authorisation to determine if the necessary criteria have been met and the exchange is allowable. Once authorisation is confirmed, Information Brokerage instigates the release of the image set by informing Data Distribution that the image set requires distribution to the Image Receiver. Data Distribution gathers the required data and determines a distribution solution, taking into account the communication resources as appropriate.

C.6.8.3 Post-Conditions

- The images have been released to the external system.

C.6.8.4 Actors

Operator

A user of the system, who has been given a role that allows them to search the system for data and instigate the release of the data.

Image Receiver

A system that is able to receive images.

C.6.9 Time Synchronisation between Nodes

This IV describes how time values may be synchronised between different nodes, each with access to a local time source. The nodes may be within an Exploiting Platform, or may be separate Exploiting Platforms.

It is assumed that:

- Transmission latency exists for communications between the nodes, though the latency is not large.

The following considerations are excluded:

- Alternative synchronisation strategies. The preferred strategy will depend on numerous factors, such as the relationship between nodes (e.g. connection type, distance, and interference), accuracy required, available time sources, and clock drift.
- Synchronisation between nodes where none are considered particularly reliable. In such cases a weighted average of principal times may be used to determine a reference time to be synchronised against.
- Mechanisms for communication of time information between nodes. See the Data Transfer IV for details.
- Identification of system health problems that may lead to a requirement for synchronisation.
- The setting of rules and limits concerning time sources or any allowable inconsistencies in time values.

C.6.9.1 Pre-Conditions

- A requirement for synchronisation between Node B and Node A has been identified.

C.6.9.2 View

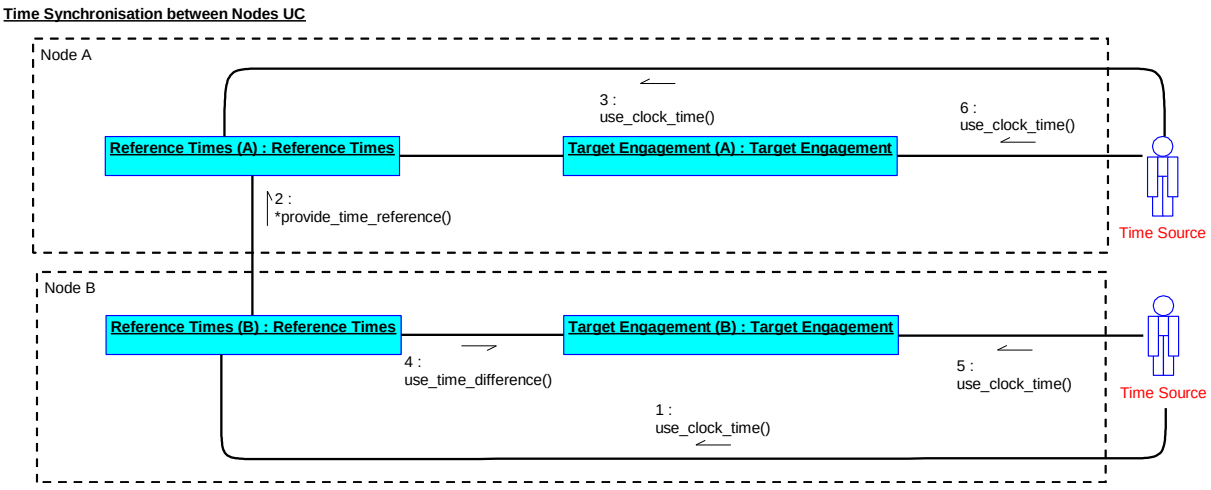

Figure 1356: Time Synchronisation between Nodes IV

Reference Times (B) gets the local clock time from the local Time Source. Reference Times (B) then requests the current clock time Difference from Reference Times (A) - i.e. the difference between its provided local clock time instance and Node A's local clock time.

Reference Times (A) gets the local clock time from its local Time Source. Reference Times (A) determines time Difference between this value of clock time and that provided in the query by Reference Times (B) and returns the time Difference to Reference Times (B).

Reference Times (B) uses the provided time Difference, and any latencies in transactions and processing, to adjust (synchronise) its understanding of the equivalent Node A reference time.

Reference Times (B) provides the time Difference, now synchronised to Reference Times (A), to Target Engagement (B).

Target Engagement (B) consumes local clock times from its local Time Source, adjusted according to the time Difference information provided by Reference Times (B).

Target Engagement (A) consumes local clock times from its local Time Source.

C.6.9.3 Post-Conditions

- Use of time in Node B is synchronised to that of Node A.

C.6.9.4 Actors

Time Source

A time source, for example supplied by the local infrastructure.

C.6.10 Middleware Error

This group of Interaction Views illustrates how the PRA components can interact with middleware functionality in the context of health management. Middleware is software that provides a set of general purpose services for the PRA components, in a layered infrastructure architecture the middleware sits in between the application layer and the processing hardware layer.

The complexity of the middleware is dependent on the precise implementation, however it is expected that the middleware will be responsible for managing faults in the underlying processing hardware. The middleware will be able to take pre-defined actions in response to hardware faults, e.g. by restarting hardware or using backup resources.

The PRA health management components will interact with the middleware to perform fault handling functionality and will insulate other PRA components from such concerns. Other PRA components may have a limited role in managing the middleware, for example the Asset Transitions component triggering software configuration changes.

The PRA components will be responsible for understanding any change in their capability due to middleware changes.

C.6.10.1 Middleware Handled Error

This IV demonstrates how a hardware fault is identified and handled by the infrastructure. The PYRAMID components are not involved in the diagnosis and corrective activity; they are only informed of the loss of the resource, which may lead to a change in capability and a reduction in system redundancy. Information Presentation is used as an example of a component making use of a resource.

It is assumed that:

- The Middleware is able to handle certain types of fault arising from the underlying resource hardware.
- The Middleware understands the allowable software configurations, i.e. which components are permitted to use which resource.

The following considerations are excluded:

- Health Assessment components working and communicating with other Health Assessment components in a hierarchy, which would be relevant in more complex health scenarios.
- What Information Presentation is using the resource for.
- Any maintenance action to address the faults identified.
- Behaviour associated with running tests (e.g. CBIT).
- How system capability is determined based on a published anomaly or health assessment.

C.6.10.1.1 Pre-Conditions

- No pre-conditions identified.

C.6.10.1.2 View

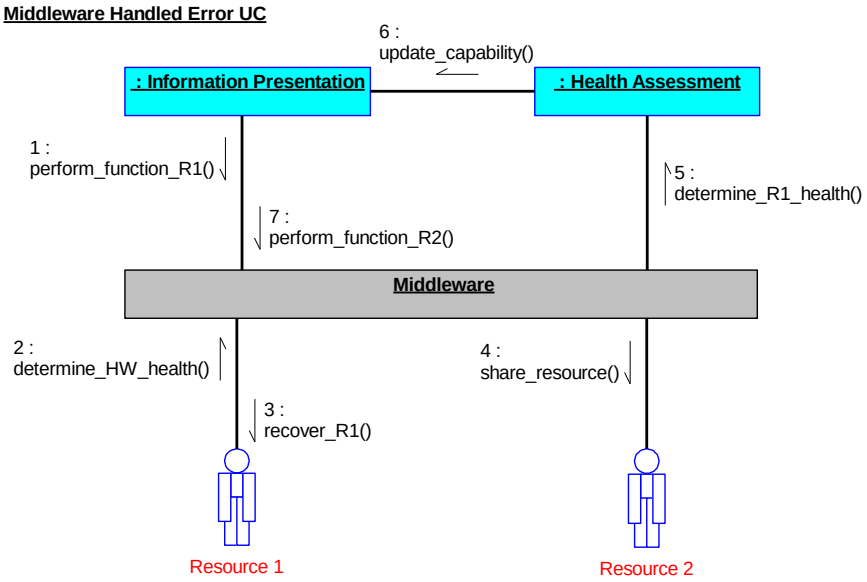

Figure 1357: Middleware Handled Error IV

Information Presentation is using Resource 1 and performing its function when Resource 1 identifies a fault during CBIT and reports it to the Middleware. The fault falls within the remit of what the Middleware is permitted to address and the Middleware attempts to recover Resource 1, e.g. by performing a warm start. The recovery action is not successful (i.e. the fault is still present) therefore the Middleware updates the configuration to allow Information Presentation to use Resource 2 (albeit on a shared basis) instead of Resource 1.

The Middleware reports the appropriate details of the fault and resource remapping to Health Assessment, which logs the information to assist subsequent maintenance. Health Assessment then informs Information Presentation that there may be a change to its capability due to the increase in response times, this is due to Resource 2 being a shared resource; the fact that the resource has changed is invisible to Information Presentation, which continues to use the available resource in the conduct of its activities.

C.6.10.1.3 Post-Conditions

- The Middleware has remapped the resource allocation from Resource 1 to Resource 2.
- Health Assessment has logged the fault identified by Resource 1 for subsequent maintenance activity.

C.6.10.1.4 Actors

Resource 1

A hardware processing resource used by the system.

Resource 2

A hardware processing resource used by the system.

C.6.10.2 PRA Handled Error

This IV demonstrates how a hardware fault is identified and handled by the PYRAMID components which then request the Middleware to reconfigure to allow access to an alternative resource. The PYRAMID components also deal with the change in capability and reduction in system redundancy. Information Presentation is used as an example of a component making use of a resource.

It is assumed that:

- The Middleware understands the allowable software configurations, i.e. which components are permitted to use which resource.
- The Middleware is unable to handle the resource failure.
- BIT can be run on the hardware resource with no adverse impact on ongoing activities.

The following considerations are excluded:

- What Information Presentation is using the resource for.
- Any maintenance action to address the faults identified.
- How system capability is determined based on a published anomaly or health assessment.
- Cyber attack assessment using Cyber Defence.

C.6.10.2.1 Pre-Conditions

- No pre-conditions identified.

C.6.10.2.2 View

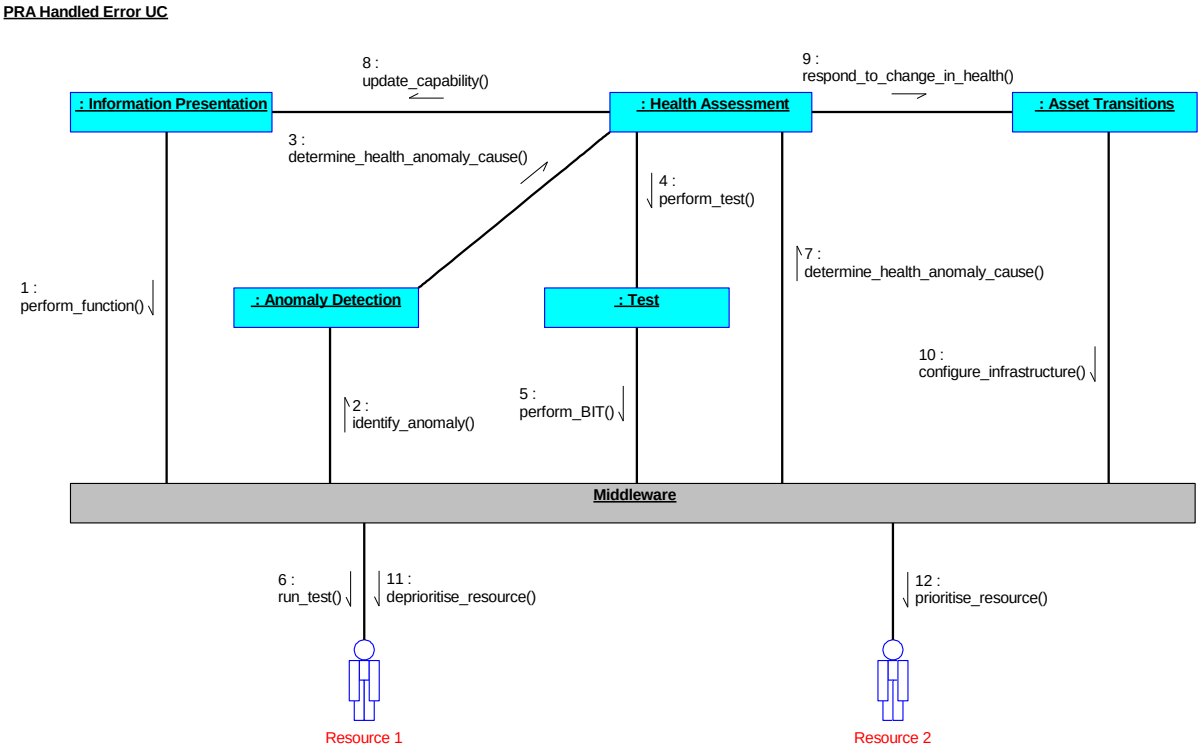

Figure 1358: PRA Handled Error IV

Information Presentation is running on Resource 1 and performing its function. Anomaly Detection identifies a difference between the expected and actual state of the system and flags this as an anomaly to Health Assessment (also sent to Cyber Defence for investigation of whether this is indicative of a cyber attack - not shown here). Health Assessment identifies that there is insufficient information available to determine the cause of the anomaly and therefore directs Test to gather more information. Test instructs the middleware to run a BIT on Resource 1 and the test results indicate that there is a fault. The test results are passed to Health Assessment which in turn communicates this health change to Information Presentation, Information Presentation determines that it now has a reduced capability. The change in health identified by Health Assessment is also provided to Asset Transitions, which instructs the Middleware to use Resource 2 instead of Resource 1. Information Presentation no longer has a reduced capability.

C.6.10.2.3 Post-Conditions

- Information Presentation is running on Resource 2.
- Health Assessment has logged the fault identified in Resource 1 for subsequent maintenance activity.

C.6.10.2.4 Actors

Resource 1

A hardware processing resource used by the system.

Resource 2

A hardware processing resource used by the system.

C.6.10.3 PRA Handled Error - Distributed Hardware

This IV demonstrates how a hardware fault in a primary resource is identified by the infrastructure, resulting in a handover to a secondary resource commanded by the PYRAMID components, which also handle any change in capability and reduction in system redundancy. Information Presentation is used as an example of a component making use of a resource.

It is assumed that:

- The Middleware is able to handle certain types of fault arising from the underlying resource hardware.
- The Middleware understands the allowable software configurations, i.e. which components are permitted to use which resource.

The following considerations are excluded:

- What Information Presentation is using the resource for.
- Any maintenance action to address the faults identified.
- Behaviour associated with running tests (e.g. CBIT).
- How system capability is determined based on a published anomaly or health assessment.
- Cyber attack assessment using Cyber Defence.

C.6.10.3.1 Pre-Conditions

- No pre-conditions identified.

C.6.10.3.2 View

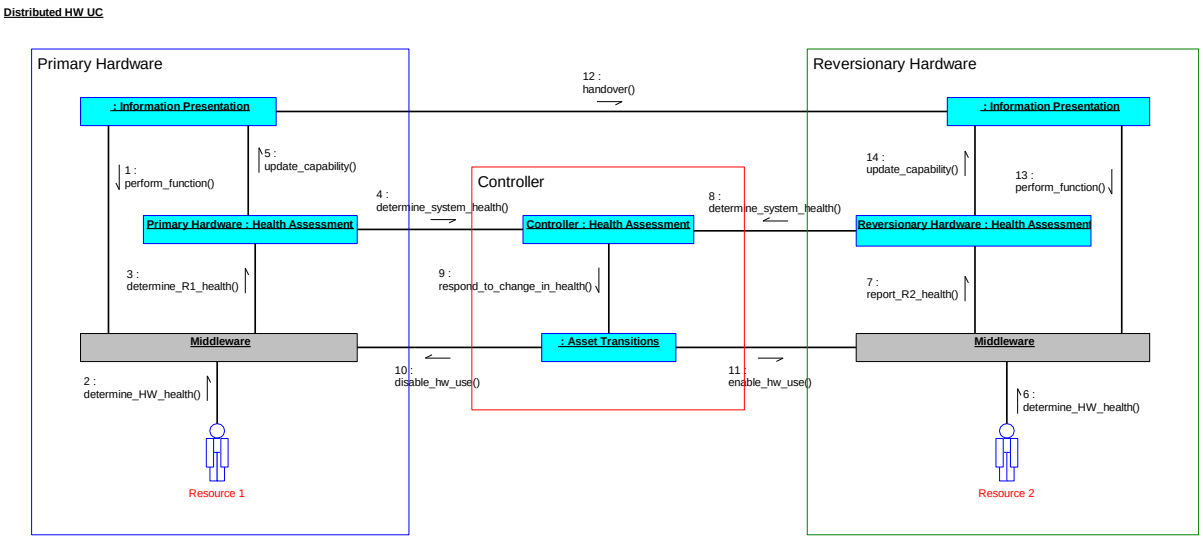

Figure 1359: Distributed HW IV

Information Presentation is running on Resource 1 and performing its function when Resource 1 identifies a fault during CBIT and reports it to the Middleware. The Middleware reports the appropriate details of the fault to the Primary Hardware instance of Health Assessment, which logs the information to assist subsequent maintenance. The change in health is available to the Controller instance of Health Assessment.

The Controller instance of Health Assessment also has the health status reported by the Reversionary Hardware instance of Health Assessment. The Controller instance of Health Assessment reports the combined health status of Resource 1 and Resource 2 to Asset Transitions. Asset Transitions then coordinates the switch over from the Primary Hardware to the Reversionary Hardware by communicating with the related Middleware, which disables the use of Resource 1 and enables the use of Resource 2. The two instances of Information Presentation share information which allows them to perform a 'hot swap' handover when the switch in hardware is triggered. The Information Presentation running on the Reversionary Hardware is now able to function with the same capability prior to the fault on Resource 1.

C.6.10.3.3 Post-Conditions

- Information Presentation is running on Reversionary Hardware (Resource 2).
- Health Assessment has logged the fault identified by Resource 1 for subsequent maintenance activity.

C.6.10.3.4 Actors

Resource 1

A hardware processing resource used by the system.

Resource 2

A hardware processing resource used by the system.

C.7 Operator Interactions Views

This section contains IVs relating to the interaction of a system with human operators.

C.7.1 Human Communications

This covers the initiation and making of a phone call between two humans via the system. The behaviour is the same if it is between two authorised operators or an authorised operator and an external entity.

Note: While usually a phone call consists of four unidirectional streams (audio out, audio back, call stats out, and call stats back), only one is shown in Figure 1360: Human Communications IV; the understanding of the relationship between the streams is in Information Brokerage, not in Data Distribution (which just sees four channels).

Similar methods can support other real-time communication, such as video streams. Where real-time communication is not needed, transfers can be handled in the same way as other machine-to-machine communication and do not require any special behaviour.

The closing down of a communication session can be processed as the reverse of opening a session.

It is assumed that:

- The correct devices (e.g. headsets) are available to both call parties and the state of these are known.
- All call parties (initiating and receiving users) are logged on and are available.
- The location and registration details have been captured by Human Interaction, as provided by other components.
- All call parties are authorised to communicate with each other.

The following considerations are excluded:

- The setup, and use of, the communications infrastructure to achieve the call.
- Data transfer for inter-node interactions, between component instances at the initiating end and receiving end.
- Conference calls, as the build set is the same, except with multiple Callers and Call Receivers. Data Distribution also provides a "consolidated output", instead of a simple "output".
- Some steps have been removed to reduce cluttering, such as confirmation events (like the Receiving Human Interaction informing the Initiating Human Interaction that the call has been accepted).

C.7.1.1 Pre-Conditions

- Connectivity between the users is in place.

C.7.1.2 View

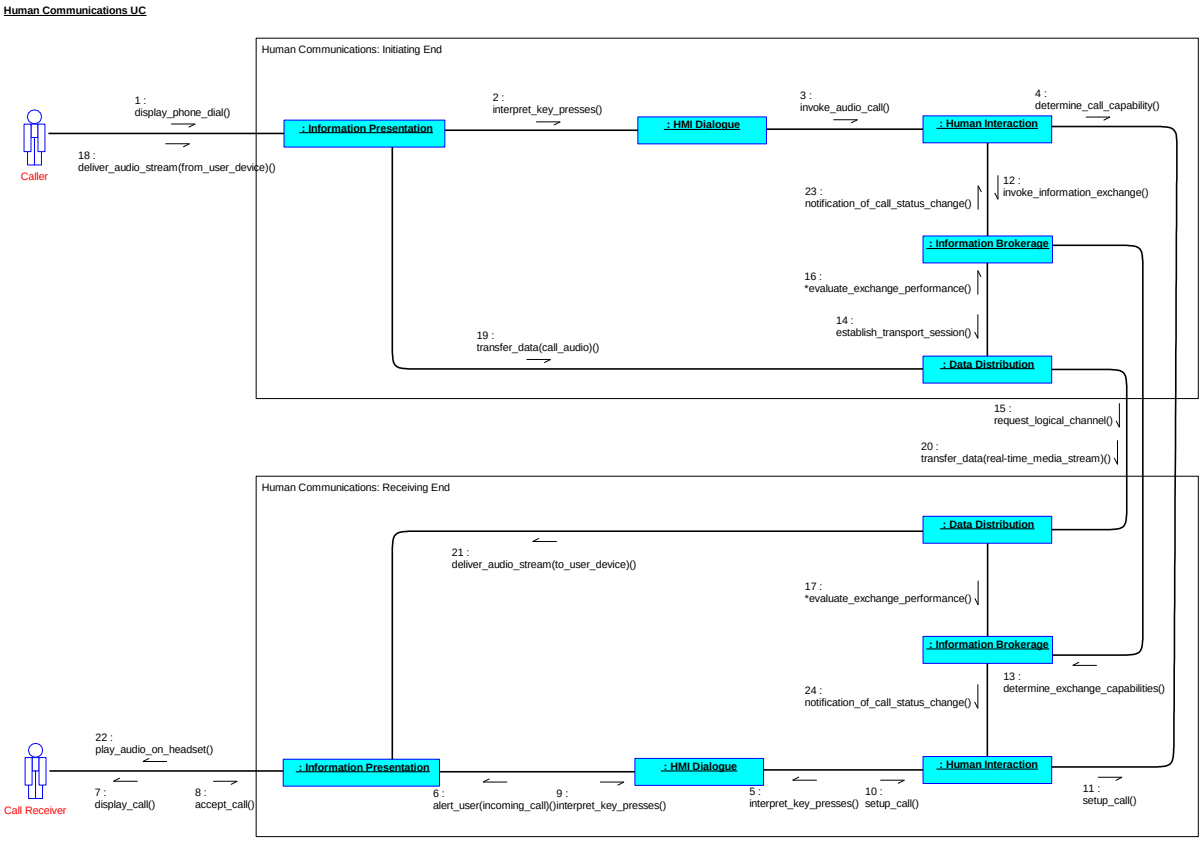

Figure 1360: Human Communications IV

A Caller requests an audio call with a Call Receiver via Information Presentation. Information Presentation interprets the key presses to be a number sequence and passes the information onto HMI Dialogue. HMI Dialogue interprets the sequence from the Caller as a request to dial a Call Receiver at the other end.

HMI Dialogue connects the request to Human Interaction via a request for an audio call.

Human Interaction determines the location and availability of the Call Receiver. Human Interaction (initiating end) then requests an audio call with Human Interaction (receiving end) after checking that the required devices exist at both call endpoints. This is a negotiation to start a call, between the two identified users and includes any agreement that the equipment required for a communication of that type is available. Human Interaction (receiving end) requests a call action from HMI Dialogue to connect call devices to the communication delivery system.

A call alert is sent to the Call Receiver, making the phone ring to alert them that someone wants to interact with them. When the call is accepted via a key press received by Information Presentation, and understood by HMI Dialogue, HMI Dialogue requests the set-up of the audio call.

Human Interaction (initiating end) requests that Information Brokerage sets-up the call (a real-time interaction based service level agreement). The two instances of Information Brokerage determine the available exchange mechanisms between the two locations. This includes how the audio will be digitally represented (encoded) and what exchange method will be used. Having decided on how the exchange is to be established, Information Brokerage requests Data Distribution to establish an exchange protocol and start a session for the audio. Data Distribution requests a logical channel to be opened from the relevant comms components (not shown) between the two locations, allowing the call to take place end-to-end.

Audio from the user is digitally encoded into a data stream by the user's device and Information Presentation, and connected to Data Distribution for delivery between each end.

Note: This is shown unidirectionally, but in reality will be bi-directional.

During the call Data Distribution reports traffic statistics to Information Brokerage which monitors the quality of the service against the requested characteristics; if the quality falls below the acceptable levels Human Interaction is notified; other than that Human Interaction is not involved again until call teardown.

C.7.1.3 Post-Conditions

- Call Receiver and Caller are in a call with each other and able to communicate directly.

C.7.1.4 Actors

Call Receiver

A user that has access to appropriate devices for the purpose of communicating with another user, and another user wishes to contact them.

Caller

An authorised user that has access to appropriate devices for the purpose of communicating with another user, and wishes to start communication.

C.7.2 User Management

The User needs to log into their operator workstation in order to perform their operational role. The operator workstation is configured with two parts in this example. Information Presentation is used to enter a username and password. When the User logs in, Information Presentation (Login) becomes hidden and Information Presentation (Flight Reference) is enabled. This will show display elements (e.g. altitude) in line with the assigned User role. HMI Dialogue is configured by the assignment of the User's role to provide the appropriate data.

In this scenario, user login is handled through a username and password combination. Similar methods can be used to support other mechanisms of login (e.g. voice recognition, or using a hardware token).

It is assumed that:

- The User must first log into the operating system, then log into the PYRAMID system separately. A deployment may use the operating system to provide User Accounts functionality.
- This scenario is a self-contained system, therefore no credentials are transmitted, and so encryption prior to interaction with User Accounts is not needed.

The following considerations are excluded:

- User account administration such as creation and deletion of accounts, assignment of administrator privileges or recovery of passwords.
- Logging into the operating system; this scenario is only concerned with logging into the PYRAMID system.
- Dynamic role allocation.
- Presentation of information.
- User input processing.

C.7.2.1 Pre-Conditions

- The User is not logged in to the PYRAMID system.
- The User has a user account configured in User Accounts.
- The User has initial roles configured in User Roles.

C.7.2.2 View

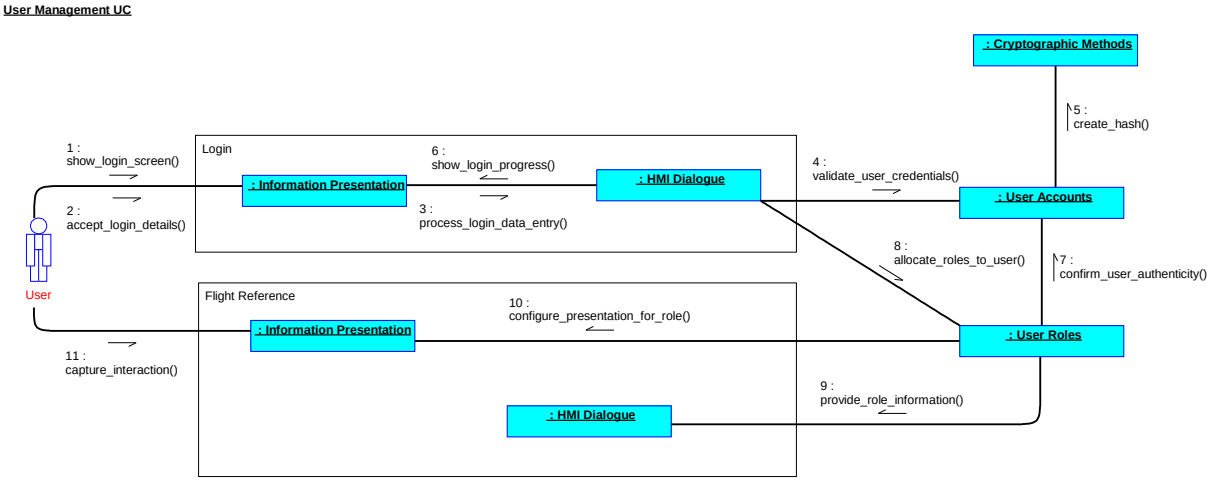

Figure 1361: User Management IV

The User requests display of the login screen and Information Presentation (Login) makes it visible.

The User enters their username and password to request login, which is accepted by Information Presentation (Login).

HMI Dialogue (Login) creates a login request using the username, operator workstation identifier and password, then passes it to User Accounts.

User Accounts validates the credentials in the login request and informs the HMI Dialogue (Login) that the User has successfully logged in. User Accounts requests that Cryptographic Methods generate a hash of the entered password information.

HMI Dialogue (Login) requests User Roles to assign the User's the initial roles.

User Roles requests User Accounts to confirm the User's authenticity. Once confirmed, User Roles determines the User's initial roles.

User Roles informs the HMI Dialogue (Flight Reference) and Information Presentation (Flight Reference) of the roles associated with the User at the operator workstation.

Information Presentation (Flight Reference) updates the presentation to tailor it to the allocated roles (e.g. enabling, disabling, showing or hiding presentation elements).

HMI Dialogue (Flight Reference) is configured based on the User's role to provide the appropriate data.

The User interacts with the system to select the information they want to see and Information Presentation (Flight Reference) determines the appropriate elements to display.

Note: Within this IV HMI Dialogue and Information Presentation have been shown split apart to enable clarity of the interactions between those required for the Login process and those for the tailoring of the display and required data for a specific role (e.g. Flight Reference).

C.7.2.3 Post-Conditions

- The User is logged into the operator workstation and has been allocated initial roles (they are now an authorised operator).
- The Information Presentation and HMI Dialogue components have been configured for the User's allocated roles.

C.7.2.4 Actors

User

A person with an existing user account who needs to log into the system.

C.8 Decision Making Views

This section contains IVs relating to high-level decision making and authorisation within a system.

C.8.1 Action Authorisation

The vehicle is following a planned route which has adverse weather along it. Whilst executing the route, the vehicle encounters the adverse weather. This means the current routing solution meets avoidance criterion and a new route, which crosses restricted airspace, is required. This IV illustrates how authorisation is requested and granted for a route that crosses restricted airspace.

It is assumed that:

- The deployment has a level of autonomy that allows Routes to determine a routing solution.
- Routes that cross restricted airspace require authorisation by an authorised operator.
- Routes considers all relevant aspects of the SA picture when determining that a new route crossing restricted airspace is necessary to avoid the hazardous weather

The following considerations are excluded:

- How routes are enacted.
- The way that weather data is detected and processed.
- How the authorised operator's situation awareness is facilitated via the HMI.
- How authorisation requests are presented to the authorised operator via the HMI.
- How authorisation is granted by the authorised operator via the HMI.

C.8.1.1 Pre-Conditions

- A planned route is being executed.
- Operational Rules and Limits has been loaded with the rules for routing in different airspaces.
- Authorisation has been loaded with an authorisation policy, which contains rules for the authoriser roles allowed to give authorisations for crossing a restricted airspace.

C.8.1.2 View

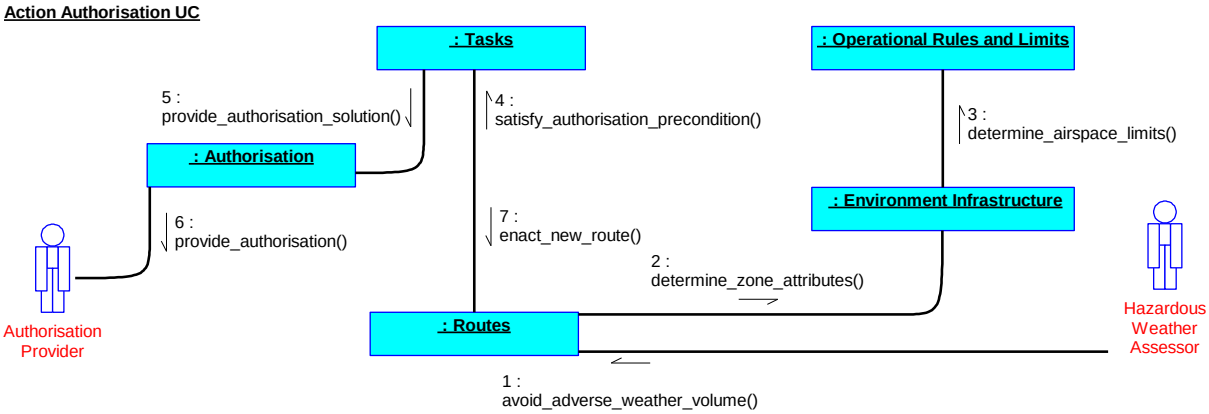

Figure 1362: Action Authorisation IV

The Hazardous Weather Assessor recognises an adverse weather event and demands that Routes avoids the adverse weather volume.

Routes subsequently has to change the current routing solution as a constraint states it must avoid flying in a particular volume. Environment Infrastructure identifies that the new route will take the vehicle over a restricted airspace. Environment Infrastructure uses Operational Rules and Limits to determine that authorisation must be granted to fly over a restricted airspace and therefore Routes identifies an authorisation pre-condition.

Routes informs Tasks that it requires authorisation in order to enact the new routing solution. Tasks requests that Authorisation determines how to acquire the authorisation for this new route. Authorisation determines that this route has to be authorised and requests authorisation from the Authorisation Provider. The Authorisation Provider authorises the new route. This fulfils the pre-condition and allows Routes to enact the new route.

C.8.1.3 Post-Conditions

- The new route has been authorised.

C.8.1.4 Actors

Authorisation Provider

An authorised operator of the system who is able to authorise a route change through restricted airspace.

Hazardous Weather Assessor

The Hazardous Weather Assessor is responsible for detecting weather events and determining their threat to the survival of the aircraft.

C.8.2 Rules

This IV illustrates how a PRA based system could interpret rules in order to determine the zones within which it is not allowed to fly, and which therefore must be identified and the appropriate environmental rules determined before any routing activity can be performed.

There are a number of different categories of rules which an Exploiting Platform might need to comply with. Included within these are Rules of Engagement, EMCON policies and behavioural rules, however other rule sources may exist. This IV will focus on rules which limit where the system can fly.

It is assumed that:

- The system is operating in a benign environment with guaranteed communications between the system and the authorised operator.

The following considerations are excluded:

- Operating the system in accordance with the calculated limits.
- Restrictions on where the system can operate which are not derived from rules.
- Replacement of rules and subsequent re-calculation of limits.

C.8.2.1 Pre-Conditions

- The relationship between rules and system limits has been defined.
- A set of rules with conditions defining when they apply has been defined.

C.8.2.2 View

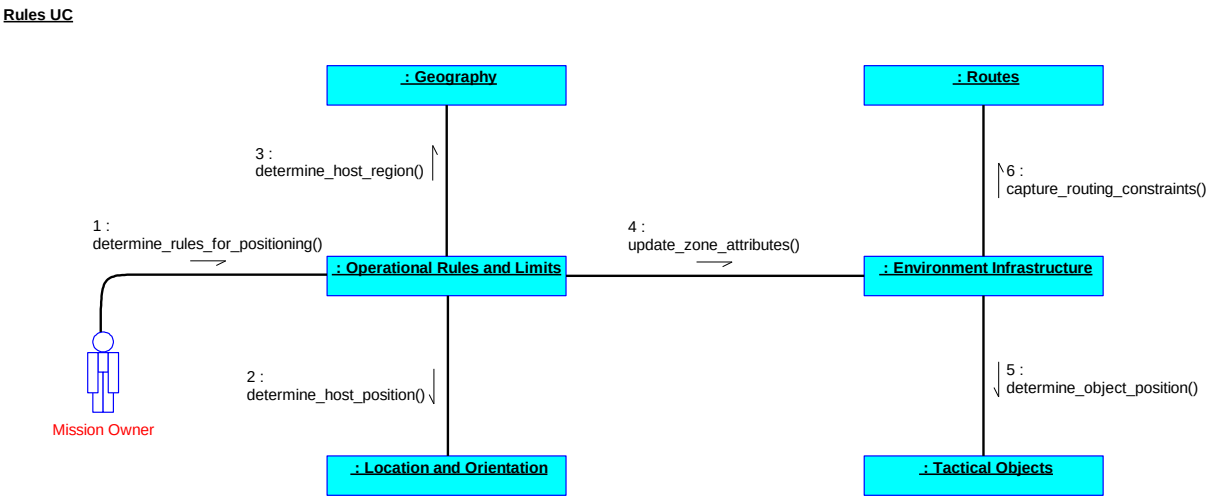

Figure 1363: Rules IV

The Mission Owner sets the context for the mission, Operational Rules and Limits obtains ownship position from Location and Orientation and then requires Geography to identify the geographical region ownship is currently within. With the context and the region Operational Rules and Limits determines the currently applicable rules and thus the current limits on the system.

These limits are communicated to Environment Infrastructure which uses them to update the attributes of environmental regions it understands (e.g. active regions of restricted airspace). The position of regions within Geography are not affected by the limits.

For limits associated with tactical objects (e.g. the missile engagement zone around a SAM system), Environment Infrastructure will obtain the object positions from Tactical Objects and using the limits from Operational Rules and Limits will determine appropriate regions to implement these limits.

Environment Infrastructure will provide all regions as constraints to Routes which will account for them when determining the route.

C.8.2.3 Post-Conditions

- The limits on where the system is permitted to operate are understood.

C.8.2.4 Actors

Mission Owner

The 'owner' of the mission who sets the context within which the system is operating.

C.9 Offensive Actions Views

This section contains IVs relating to the offensive capability of a system and how it is used.

C.9.1 Plan For A/S Engagement

This IV illustrates the planning of a single engagement, which may include a combination of different weapons. The target has already been identified and selected for attack. This IV shows determination of the weapon package and the route to the release location, and activities to improve the target position accuracy. This release location is dependent on a number of factors including the weapon type and weather.

Planning for the deployment of a non-weapon store, such as a deployable sensor or cargo store, would be extremely similar and use largely the same components (aside from a different Tactics extension and not using Destructive Effects or Susceptibility).

It is assumed that:

- The weapons available for the A/S engagement are unpowered and have no routing capability.

The following considerations are excluded:

- All activities relating to the actual execution of the A/S engagement, as shown in the Kinetic Attack IV, such as weapon power up, dynamic LAR refinement during engagement execution, or execution of a route to a launch position.
- Setting weapon parameters, including weapon sensor parameters.
- Determination of the target in accordance with extant rules (e.g. RoE or targeting directive).
- Full determination of weapon availability and serviceability.
- Full reservation process of weapons as resources.
- Full interactions required to determine the weapon settings necessary to achieve the required weapon effect.
- Communication components, shown in the Data Transfer IV, since different weapons require different messaging systems.
- Power requirements, shown in the Power Management IV.
- Authorisation to commence planning the attack.

C.9.1.1 Pre-Conditions

- Weapon systems are fitted and able to communicate with the platform as applicable.
- The target has been identified and selected for attack.
- All necessary engagement tasking parameters (e.g. constraints, budgets, and criteria) are known.

C.9.1.2 View

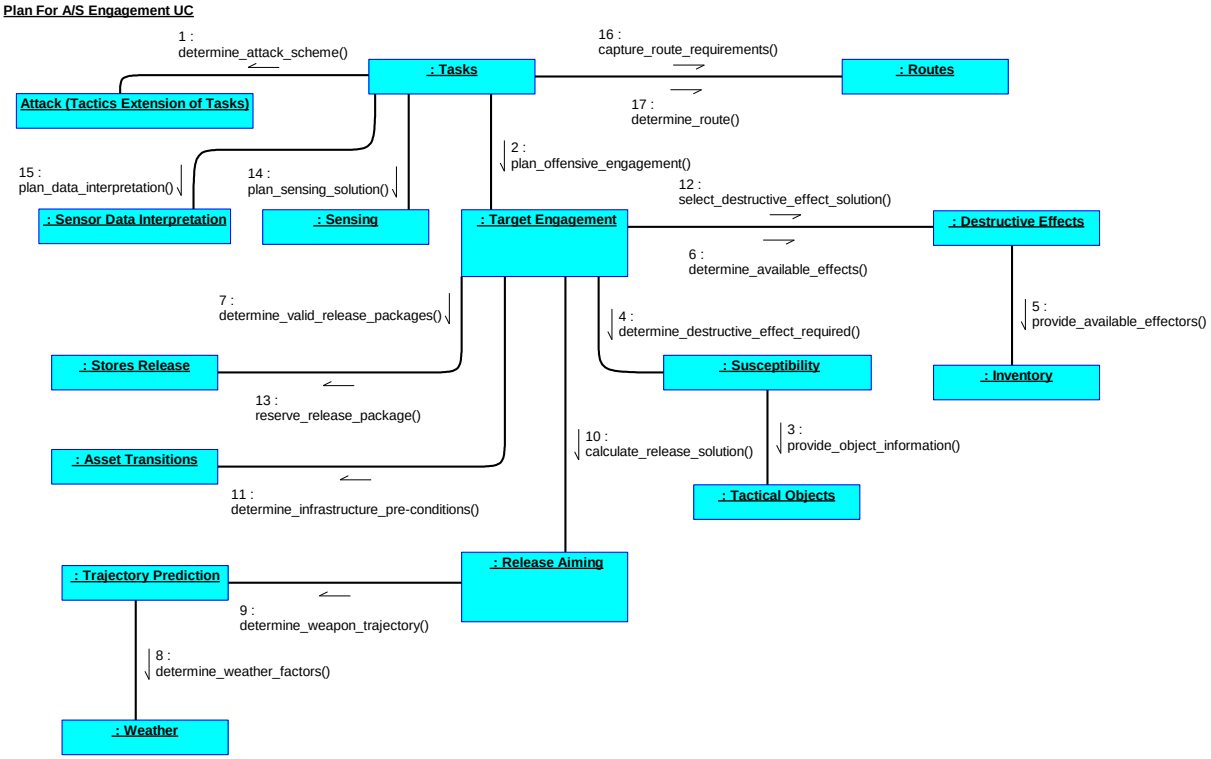

Figure 1364: Plan For A/S Engagement IV

The Attack extension to Tasks is used to determine the appropriate scheme for prosecuting targets by coordinating dissimilar but mutually supporting actions.

Tasks requests that Target Engagement plan the engagement (this is one of the actions that needs to be planned to prosecute the attack task). Target Engagement then begins to plan the offensive engagement.

Susceptibility gathers information about the target using information provided by Tactical Objects and determines the required effects needed to exploit the target's vulnerabilities. Target Engagement requests Destructive Effects to determine what destructive effects the air vehicle has at its disposal to satisfy these effect requirements. Using the available weapon information provided by Inventory, Destructive Effects determines options that could achieve the required destructive effect (in this case where each effect package option is comprised of a set of weapon effectors).

Target Engagement then determines the feasibility of each effect package option and selects the most suitable one. To do this for each effect package option it establishes:

- If the weapons associated with a package will be releasable (determined by Stores Release which constructs a potential release package).
- If the weapons associated with a package are sufficiently serviceable to prosecute a mission (the interactions associated with this are not shown).
- If there is an acceptable release region (determined by Release Aiming).

Since the release aiming solution is influenced by the weapon settings required to achieve the desired effect (such as impact angle), Destructive Effects provides the settings applicable to each effects package option. (The interactions involving information that Destructive Effects needs to establish this are not shown.)

Trajectory Prediction determines the predicted trajectory of the weapon release, taking into account atmospheric conditions provided by Weather, and provides this to Release Aiming. Using the predicted trajectory information, Release Aiming calculates the aiming solution.

Asset Transitions provides pre-conditions related to the activities planned by Target Engagement (such as determining weapon activation actions).

Once Target Engagement has determined the best solution the selected plan is disseminated to the relevant components satisfying the different parts of the solution:

- The chosen destructive effects option is indicated to Destructive Effects so that it can apply the associated settings when the plan is enacted.
- Stores Release is requested to select the required release package and Stores Release then arranges for the relevant weapons to be reserved (the reservation interactions are not shown). Destructive Effects is informed of the chosen weapons package so that it can apply the settings to the appropriate weapons.

Target Engagement informs Tasks whether it has a solution for the engagement that meets the requirements and indicates the solution costs and any pre-conditions needed to achieve the solution. If a solution cannot be found, or if Tasks determines that the costs are too high or a pre-condition cannot be met, then Tasks will adjust the requirements until an overall solution that meets the objective can be found. Tasks makes this determination for all of the action components that it issues action requirements to.

In this case the action requirement can be met and the cost is acceptable; however, Target Engagement has identified a pre-condition to establish and maintain the position of the target to a required accuracy. Tasks therefore provides action requirements to Sensing to plan the necessary sensing activities and Sensor Data Interpretation to plan the sensor data interpretation activities. Sensing and Sensor Data Interpretation report their successful determination of a solution to Tasks with the associated costs and pre-conditions.

The pre-conditions from Sensing and Target Engagement include the need to get the air vehicle to the necessary locations for target detection and for release. Therefore, Tasks provides action requirements to Routes to provide a suitable route. Routes reports its successful determination of a solution to Tasks with the associated costs and pre-conditions.

Tasks determines that the requirements of the overall task can be met with acceptable costs.

C.9.1.3 Post-Conditions

- Weapon(s) have been selected for the engagement.
- The engagement is appropriately planned, including sequences of weapon deployment and associated dependencies.

C.9.2 Kinetic Attack

This IV illustrates an example of the components that interact to allow the system to execute an attack up to the point of weapon release. The attack is generated by an air-surface engagement task on a potentially mobile ground target, with a guided bomb under ownship self-designation.

It is assumed that:

- An agreed attack plan has been generated.
- Generating and securing authorisation for the attack plan has been previously carried out.
- Weapons are provided with targeting data (e.g. designator pulse repetition frequency codes and initial target coordinates) as well as airburst and impact settings as part of an approved attack plan.
- Weapons are not in a final armed state.
- Weapons have no routing capability.

The following considerations are excluded:

- Engaging with multiple weapon types or multiple weapons (either as a salvo release or as a separate repeated attack). This IV covers a single attack with a single weapon type only.
- All elements of the attack planning, as shown in the Plan For A/S Engagement IV, such as weapon type selection or LAR calculation during engagement planning. Only dynamic LAR refinement during engagement execution is covered.
- Details of the release process, shown in the Releasing IV.
- Communications components, shown in the Data Transfer IV, as different weapons require different messaging systems.
- Other post-launch actions including activation of laser designation and collateral damage assessment.
- Refinement of predicted target movement.

C.9.2.1 Pre-Conditions

- Tasks has satisfied the pre-conditions for the attack by setting up coordination mechanisms between Target Engagement, Routes and Sensing as well as a feedback channel between the components to enable execution of the attack.
- Authorisation has been given to execute the attack plan.

C.9.2.2 View

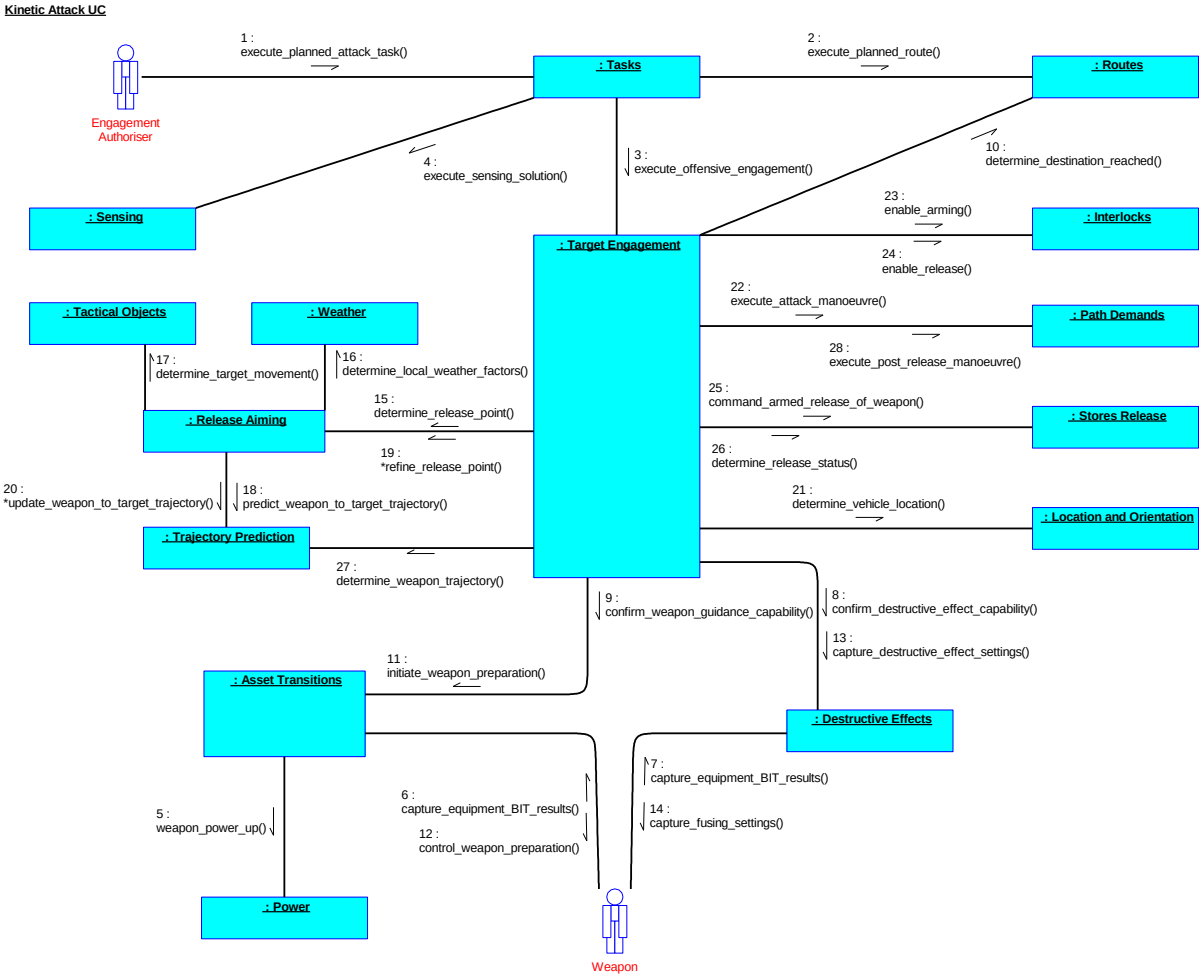

Figure 1365: Kinetic Attack IV

Tasks receives a request from the Engagement Authoriser to initiate a previously planned attack. Tasks holds the scheme for the previously planned solution which is used as the basis of the action requirements which are issued to the components containing the attack plan with details consisting of: a weapons use and designation plan (held by Target Engagement), routes to acquire the target and to the release point (held by Routes) and sensing of the target (held by Sensing).

Asset Transitions requests Power to apply power to the Weapon equipment, which in turns performs its own PBIT.

Target Engagement confirms the capability of the weapon to be guided, supported by Asset Transitions and Destructive Effects which capture the BIT results from the weapon equipment.

Routes executes the planned route and reports when the initiation point has been reached; Sensing executes actions that enable the detection of the target. Once the vehicle is correctly positioned, Target Engagement initiates preparation of the weapon for release and confirms the destructive effect settings (e.g. impact fusing). Asset Transitions transitions the weapon to the correct state for a release (e.g. battery activation). Destructive Effects transfers the required fusing settings to the weapon.

During manoeuvring to the planned release point, Release Aiming refines the planned release point by requesting Trajectory Prediction to continuously update the predicted weapon trajectory required for the weapon to reach the target. If required to support the attack, Target Engagement will provide direct demands to Path Demands supported by Location and Orientation monitoring the vehicle position (for example to define specific manoeuvres as part of a post-release manoeuvre or to 'loft' a weapon in an extended-range attack).

Target Engagement initiates the operational release of the selected weapon by requesting Stores Release to perform the release in accordance with the release point defined by Release Aiming. The weapon is released in line with the Releasing IV.

The Exploiting Platform will need to provide weapon guidance at the appropriate phase of the weapon's flight (which is not part of this IV). To do this, the Exploiting Platform must be manoeuvred appropriately. Target Engagement requests Trajectory Prediction to determine the weapon flight path to allow the support manoeuvres to be determined. Target Engagement requests Path Demands to commence the support manoeuvres.

C.9.2.3 Post-Conditions

- A fully prepared and armed weapon has been safely released from the vehicle.
- The need for any post-release manoeuvres (either safe separation or to provide post release support) has been determined.

C.9.2.4 Actors

Engagement Authoriser

The authorised operator that authorises the engagement.

Weapon

The weapon chosen for engagement when generating the engagement plan.

C.10 Survivability Views

This section contains IVs relating to the survival capability of a system and how it is used.

C.10.1 Survival

This IV illustrates how to determine, select and trigger a pre-planned mitigation strategy in response to a specific Threat.

In order to focus the scope of the IV, the context has been limited to initially pre-planning a survival mitigation strategy based on knowledge of an expected type of Threat, threat behaviour and threat condition (i.e. identifying a worst case scenario where a missile is launched towards the air vehicle). Then implementing the pre-planned survival strategy using the Action Layer when the actual event occurs. This IV illustrates an example of how the Task and Action Layers of the Control Architecture, with support from the tactical service based components, can coordinate together to achieve a survival goal.

It is assumed that:

- The engagement authority has pre-authorised survival tactics for use against a particular Threat.

The following related areas of functionality are excluded:

- Determining situation awareness requirements and planning the actual collection of tactical information to support threat identification or behaviour assessment.
- The management of spectrum interoperability, mission constraints or mission re-planning changes as part of a mitigation strategy in response to a Threat.
- Obtaining authorisation for the Countermeasures implementation scheme.

C.10.1.1 Pre-Conditions

- Tactical Objects holds a list of expected Tactical_Objects (e.g. the SAM system types, their locations, and the missile types associated with them) found within the planned area of operation.
- All necessary survival tasking parameters (e.g. constraints, budgets, and criteria) are known and predefined within the Survival tactics extension of Tasks.
- The autonomy level for Countermeasures enactment has been defined.
- There is a feedback link set up between Countermeasures and Threats.

C.10.1.2 View

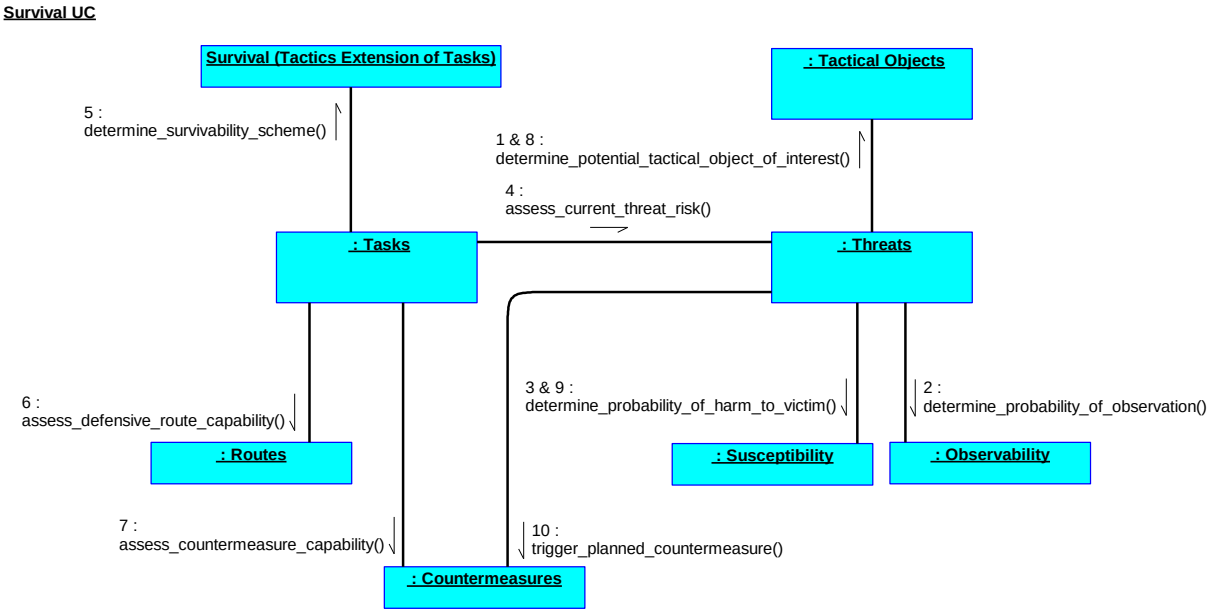

Figure 1366: Survival IV

Threats identifies from Tactical Objects, Observability and Susceptibility that there is a high probability that a potential Threat for a given threatening condition is likely to cause significant harm to the mission. Tactical Objects identifies the SAM system and its potential behaviours (i.e. search, tracking and locked-on, etc.). Observability determines that the SAM system RADAR's probability of observing the air vehicle is high. Susceptibility determines the SAM system has a high probability of harming the air vehicle if the air vehicle is observed. Threats provides the details of the Threat and threatening conditions to Tasks.

Tasks requests the Survival extension to determine a mitigation implementation scheme and provides relevant details of the threatening object. Tasks places a constraint on Survival by specifying that the only Threat mitigation capabilities available are routing or countermeasures. The Survival extension determines this would require an immediate mitigation response. Routes assesses whether it can determine an achievable route that could mitigate the Threat. Countermeasures likewise determines its capability. Survival examines all the pre-authorised tactics and determines that a countermeasure response (e.g. defensive manoeuvres, jamming and releasing of flares) which enables the air vehicle to continue its existing task is preferable to a defensive routing response (i.e. fly outside the weapon engagement zone). Survival provides a countermeasure implementation scheme to Tasks.

Tasks creates a single mitigation action containing the pre-authorised implementation scheme for Countermeasures. Tasks sets a feedback link between Countermeasures and Threats where Threats is responsible for reporting to Countermeasures when the specific Threat exceeds the threat risk threshold. Countermeasures determines the required countermeasure strategy for enactment. Tasks requires Threats to notify when the Threat has exceeded the threat risk threshold and when it returns below the threshold in order to monitor the progress of the task.

Tactical Objects reports a new SAM system has been detected and its behaviour (i.e. locked on to the air vehicle) to Threats. Threats performs a threat risk assessment with support from Susceptibility. Susceptibility determines the probability of the SAM system harming the air vehicle is high and the effect would result in significant damage to the air vehicle. Threats concludes the Threat has exceeded the threat risk threshold and informs Countermeasures. Countermeasures determines the enactment criteria has been fulfilled and executes the pre-determined countermeasure strategy. Tactical Objects reports the SAM system is no longer locked on. Threats re-assesses the threat risk and deems the Threat has been reduced to an acceptable threat risk threshold. Threats passes the updated Threat risk to Countermeasures. Countermeasures stops executing the mitigation plan.

C.10.1.3 Post-Conditions

- The survival mitigation response has successfully reduced the threat risk of the Threat to an acceptable level.

C.10.2 Threat Detection

C.10.2.1 Threat Detection Using Sensor Products

This IV demonstrates how a threat can be identified using sensor data. This type of threat identification may be used in situations where a threat is expected to be present but Tactical Objects doesn't identify any relevant objects. Threats utilises Sensor Products to provide processed sensor information to inform the threat situation for a region of interest.

No assumptions are made for this IV.

The following related areas of functionality are excluded from this IV:

- Coordinating threat detection with other ongoing tasks.
- The planning and coordination of an intelligence gathering task.
- The action coordination and execution of the tactical sensing solution.
- The determination and classification of an object.
- The planning and execution of threat mitigation.
- The countermeasure execution to a threat.

C.10.2.1.1 Pre-Conditions

- A threat identification library has been loaded.
- Threats has been tasked to assess for threats in a region of the battlespace.

C.10.2.1.2 View

Threat Detection Using Sensor Products UC

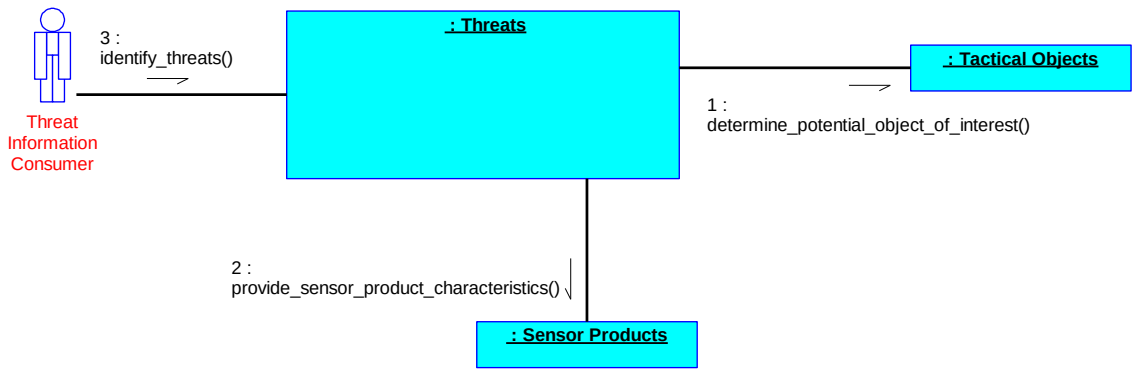

Figure 1367: Threat Detection Using Sensor Products IV

Threats requests battlespace entity information from Tactical Objects against threat criteria, this results in a situation where a threat is expected to be present but Tactical Objects doesn't identify any objects within the battlespace region of interest.

Threats requests Sensor Products to provide processed sensor data for the battlespace region of interest. Sensor Products provides the processed sensor data and Threats compares the sensor product characteristics against its internal threat library to identify a possible match against a potential threat. Threats identifies a match classifying it as a credible threat and reports the threat to the Threat Information Consumer.

C.10.2.1.3 Post-Conditions

- The Threat Information Consumer is aware of the threat against the air vehicle.

C.10.2.1.4 Actors

Threat Information Consumer

An entity which has an interest in actual threats which are required in order to fulfil its own responsibilities or required to make decisions such as an authorised operator or the Tasks component.

C.10.2.2 Threat Detection Using Tactical Objects

This IV demonstrates how a reported tactical object is identified as a threat to the ownship air vehicle. In this case, the scenario focuses on performing a threat identification and risk assessment on a newly observed tactical system, which comprises of a SAM early warning radar, SAM missile system and SAM fire control radar. As part of reporting the tactical system as a threat to the air vehicle, a threat assessment is performed to establish the potential of the SAM missile system (including its early warning radar, fire control radar and missiles) to engage the air vehicle based on the current planned route.

It is assumed that:

- Sensing has been configured during the pre-mission planning phase to prioritise sensing activities within frequency spectrums where more threats are expected to exist, to optimise tactical object identification.
- Observability and Susceptibility are already aware of the capabilities of any known threat types.
- A feedback loop has been established as part of the threat detection task between Sensing and the relevant components to improve the sensing solution.

The following related areas of functionality are excluded from this IV:

- Coordinating threat detection with other ongoing tasks.
- The planning and coordination of an intelligence gathering task.
- The action coordination and execution of the tactical sensing solution.
- The determination and classification of an object.
- The planning and execution of threat mitigation.
- The countermeasure execution to a threat.

C.10.2.2.1 Pre-Conditions

- A new tactical object has been added to the existing tactical object list.

C.10.2.2.2 View

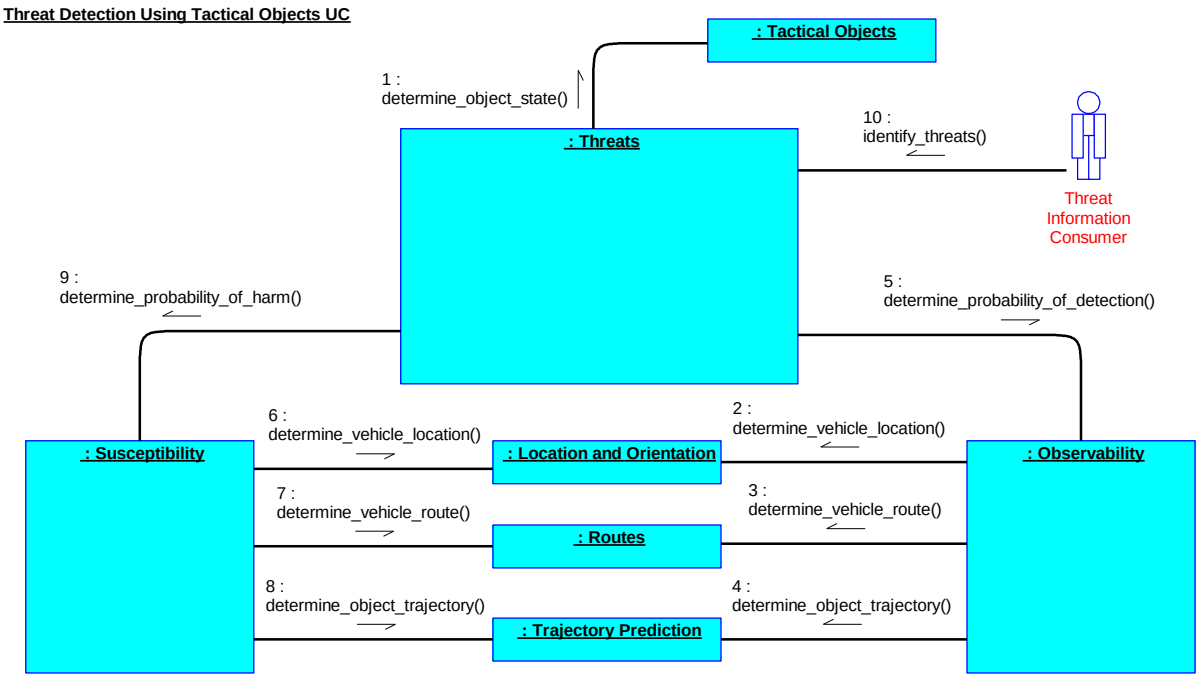

Figure 1368: Threat Detection Using Tactical Objects IV

Tactical Objects provides a new tactical object to Threats which includes information on the object type (i.e. SAM early warning radar), object state ("Search"), and object characteristics (i.e. "RF agile emitter", "Stable PRI", etc.). Tactical Objects also provides details of two other detected objects, one being a SAM missile system with an additional associated weapon, and the other being a tracking capability (SAM fire control radar). Tactical Objects identifies that these are three individual objects that form part of the same tactical system. Threats performs an initial threat assessment against the new tactical objects and determines that they constitute a potential threat.

Once the potential threat has been identified, Threats requests Observability to determine the probability of the air vehicle being detected. Observability needs to understand whether the potential threat is within the boundaries of the vehicles planned route. Observability requests the current vehicle location from Location and Orientation, the current route from Routes and the predicted trajectory of the tactical objects from Trajectory Prediction. Using this information, Observability reports to Threats that the probability of being detected is high.

Threats needs to understand whether the potential threat can cause significant harm to the air vehicle if the air vehicle continues its current planned route. Threats requests Susceptibility to determine the probability of the air vehicle being harmed. Using relevant information from Location and Orientation, Routes and Trajectory Prediction, Susceptibility determines the probability of harm to the air vehicle and reports it to Threats.

Threats processes all of the information through the threat assessment algorithm and determines that the sensed tactical system is a credible threat to the air vehicle if the air vehicle continues its planned route. Threats reports the credible threat to the Threat Information Consumer.

C.10.2.2.3 Post-Conditions

- The Threat Information Consumer is aware of the threat against the air vehicle.

C.10.2.2.4 Actors

Threat Information Consumer

An entity which has an interest in actual threats which are required in order to fulfil its own responsibilities or required to make decisions such as an authorised operator or the Tasks component.

C.10.3 Countermeasure Coordination

This IV describes how to coordinate and execute a planned countermeasure strategy in response to a specific threat.

In order to focus the scope of the IV, the context has been limited to coordinating and executing an existing countermeasure plan in response to a specific threat. The aim of the scenario is to disrupt a threat radar's "tracking" signal which is used to track the air vehicle. The countermeasure response involves initially executing a defensive manoeuvre and then performing an ECM program. The last countermeasure response is to release chaff which causes significant tracking errors to the threat radar and completes the disruption. The scenario will determine the success of the countermeasure response when observing a change in the threat's behaviour (i.e. changes from "Tracking to Search").

This IV illustrates an example of how the Action and Resource Layers of the Control Architecture with support from the tactical service based components can adaptively coordinate together to achieve a countermeasure plan.

It is assumed that:

- Countermeasure tactics are preauthorised against particular threats.
- The survival tactics specify the type of defensive manoeuvre including the associated trigger point to execute for a predefined threat.
- The ECM program parameters will be predefined for a specific threat.
- The threat has the ability to change its waveform transmissions in attempt to mitigate interference from ECM jamming.
- The adaptive ECM waveform generation is considered as an inherent capability of the ECM Jamming Pod however the ECM programmable parameters (e.g. beam direction and time to completion) will be input requirements to the ECM Jamming Pod.
- There are no resource limitations with consumable expendables for survivability purposes.
- There are no operational limits on the countermeasure response to a threat.

The following related areas of functionality are excluded:

- Determining situational awareness requirements and planning the actual collection of tactical information to support threat identification or behaviour assessment.
- The identification and selection of a survival mitigation strategy for a specific threat.
- The management of spectrum interoperability.
- The details of releasing ECM expendables.
- The interlocks required to support the enactment.
- The execution of a path demand.
- Determination of atmospheric conditions.
- Predicting the trajectory of ECM expendables post-release.

C.10.3.1 Pre-Conditions

- A survival plan has been agreed.
- Countermeasures has been preauthorised to enact the countermeasure response to a particular type of threat.
- A feedback link has been established between Countermeasures and Threats.
- A threat is known to the system and is being tracked.

C.10.3.2 View

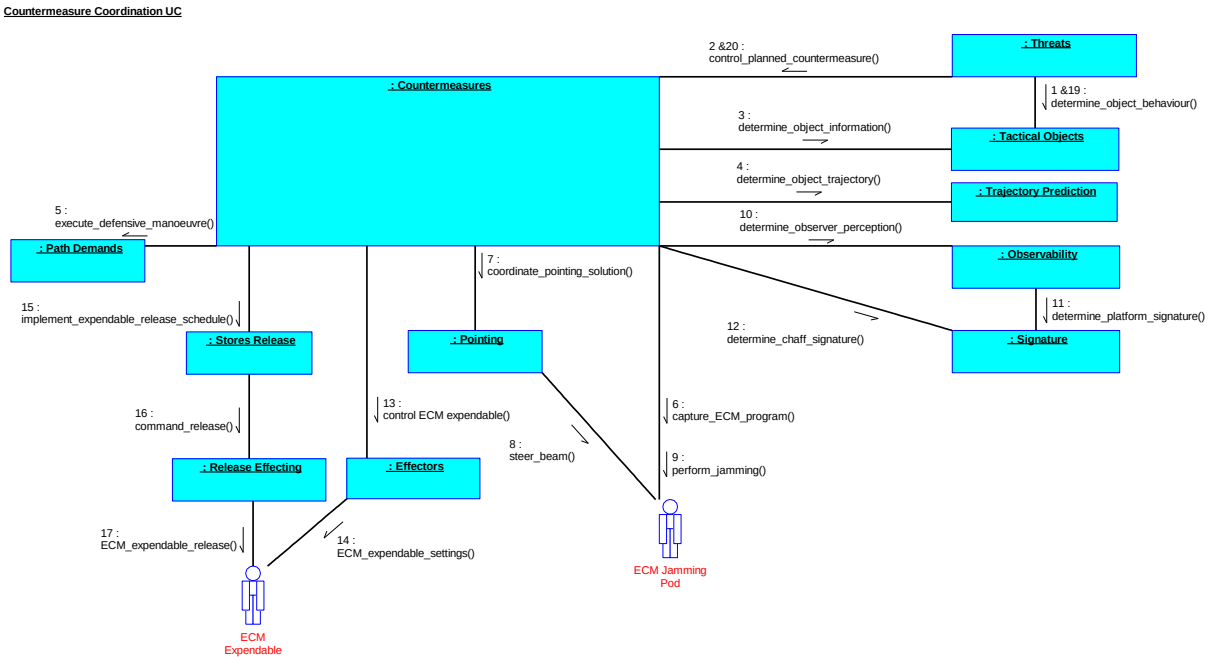

Figure 1369: Countermeasure Coordination IV

Tactical Objects determines the object is in "tracking mode". Threats reports to Countermeasures when a threat has exceeded the threat risk threshold. Countermeasures directs Tactical Objects and Trajectory Prediction to provide additional information about the threat which needs a countermeasure response.

In accordance with the agreed survival plan, Countermeasures begins by tasking Path Demands with a specific defensive manoeuvre to perform. Whilst the defensive manoeuvre is being performed, Countermeasures determines the ECM and chaff countermeasure solutions in parallel. Countermeasures then coordinates the enactment of the two solutions as part of its countermeasure response.

Countermeasures provides the ECM Jamming Pod with a predefined ECM program based on the information provided by Tactical Objects, Trajectory Prediction and countermeasure enactment criteria. In addition, Countermeasures specifies to the ECM Jamming Pod the ECM program run time. Countermeasures tasks Pointing to position the ECM Jamming Pod towards the intended target.

The execution of the ECM program is performed by the ECM Jamming Pod. Pointing commands the ECM Jamming Pod to direct the electronic beam towards the intended target. The ECM Jamming Pod steers the direction of the electronic beam in order to jam the main lobe of the intended target. In addition, the ECM Jamming Pod translates the ECM program into jamming transmissions. The ECM Jamming Pod adapts its jamming waveform in response to the agile threat.

Countermeasures actions Observability to estimate how the object currently perceives the air vehicle. To do this, Observability directs Signature to calculate the current radar cross section of the air vehicle. Countermeasures then actions Signature to calculate the required chaff radar cross section for the chaff countermeasure program.

Countermeasures determines the chaff program properties (e.g. bloom rate, decay rate, dispense rate, polarisation and chaff length), with any internal settings of the chaff program being controlled via Effectors. Countermeasures provides the chaff countermeasure program and enactment criteria to Stores Release. Stores Release commands Release Effecting to execute the dispensing of the chaff.

Tactical Objects determines the object has returned to "search mode". Threats performs a threat risk assessment and determines the threat risk threshold has decreased. Countermeasures stops the countermeasure execution and has fulfilled its requirement.

C.10.3.3 Post-Conditions

- The countermeasure solution has successfully reduced the threat risk of the threat to an acceptable level.

C.10.3.4 Actors

ECM Expendable

A consumable item that can be released from the vehicle for defensive or survivability purposes.

ECM Jamming Pod

A physical pod that is capable of receiving, synthesising and effecting within the RF portion of the EM spectrum.

C.11 Contingency Views

This section contains IVs relating to the contingency management capability of a system and how it is used.

C.11.1 Recognition of the need for a Contingency Response

The purpose of this IV is to illustrate how an abnormal rate of fuel depletion is identified and assessed as requiring a contingency response.

The following considerations are excluded:

- The full range of system inputs that may be used to identify the anomaly and determine its root cause.
- Failure handling (i.e. attempting to rectify the problem).
- Communication with the authorised operator.
- The determination of the contingency task.

C.11.1.1 Pre-Conditions

- The aircraft is airborne.
- Fuel is being depleted at a greater than expected rate.

C.11.1.2 View

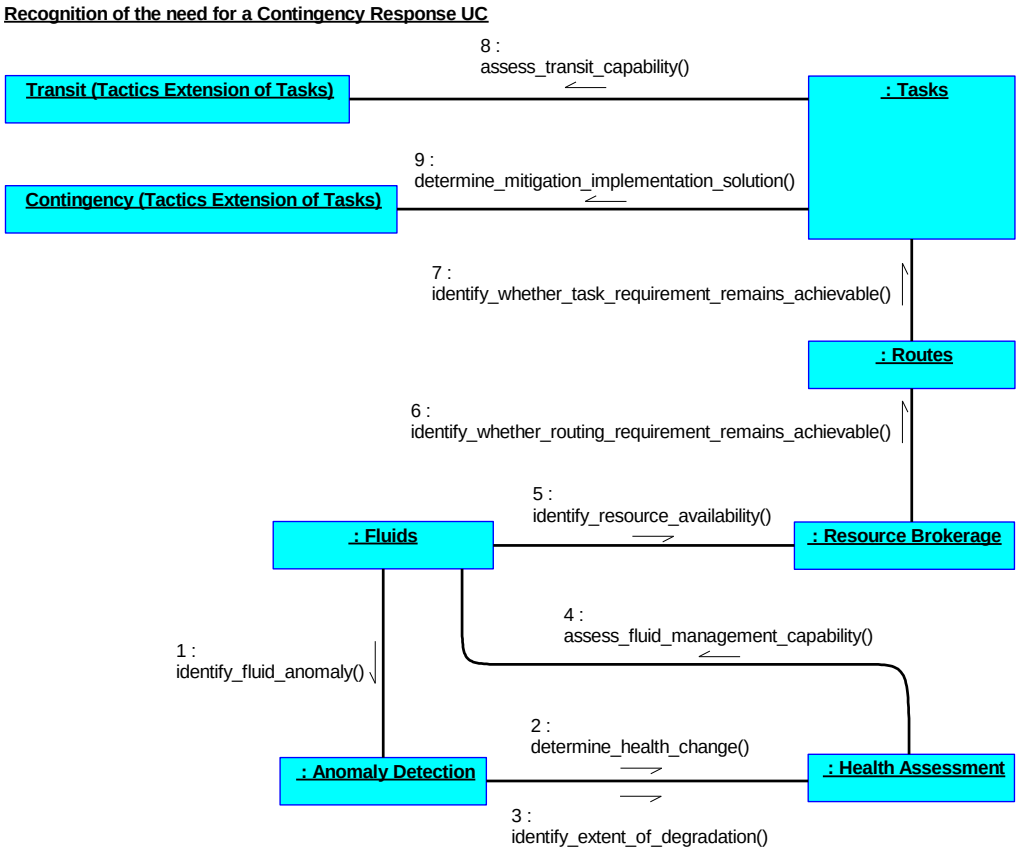

Figure 1370: Recognition of the need for a Contingency Response IV

Anomaly Detection identifies that an anomaly is present as a result of an abnormal rate of fuel depletion. Health Assessment identifies the cause of the anomaly as a damaged fuel tank and determines the extent of the damage. Fluids reassesses its capability to store fuel and reports the impact on fuel availability to Resource Brokerage, which reports an inability to meet its resource allocation to Routes. Routes determines that it can no longer meet the current positioning requirements and reports a change in routing capability to Tasks. Tasks determines, via Transit, that it can no longer meet the requirements of the transit task, and determines the change in transit capability. Tasks, via Contingency, identifies a new contingency task to mitigate the reduction in transit capability.

C.11.1.3 Post-Conditions

- A new contingency task has been identified.

C.11.2 Jettison Management

This IV illustrates the determination and subsequent enactment of a jettison requirement as a method to achieve a reduction in the air vehicle's weight. In this instance, multiple independent elements require jettison, necessitating the need for Jettison to coordinate the overall jettison implementation.

While this IV is only scoped to the reduction of weight being the triggering requirement for jettison, other requirements that require jettison as a mitigation strategy also exist, such as the need to make the air vehicle more aerodynamically efficient.

This IV also shows the routing of the air vehicle to an approved jettison area to allow the jettison to take place.

It is assumed that:

- Jettison is limited only to stores fitted to the air vehicle (e.g. weapons, expendable pods and fuel tanks) and the dumping of fuel directly.
- There is at least 1 store with sensitive data within the jettison package that requires sanitisation prior to jettison.
- The release occurs successfully after jettison is initiated.

The following considerations are excluded:

- Manual triggering of jettison.
- Details of flying the vehicle to a jettison location.
- Details of the physical release of store(s) and fuel from the air vehicle.
- Priming of any fitted weapons with jettison specific data.

C.11.2.1 Pre-Conditions

- The air vehicle is fitted with jettisonable stores and fuel.
- Some stores are behind doors on the air vehicle.
- Authorisation to perform jettison has already been given.
- The Jettison component has knowledge of allowable jettison locations.

C.11.2.2 View

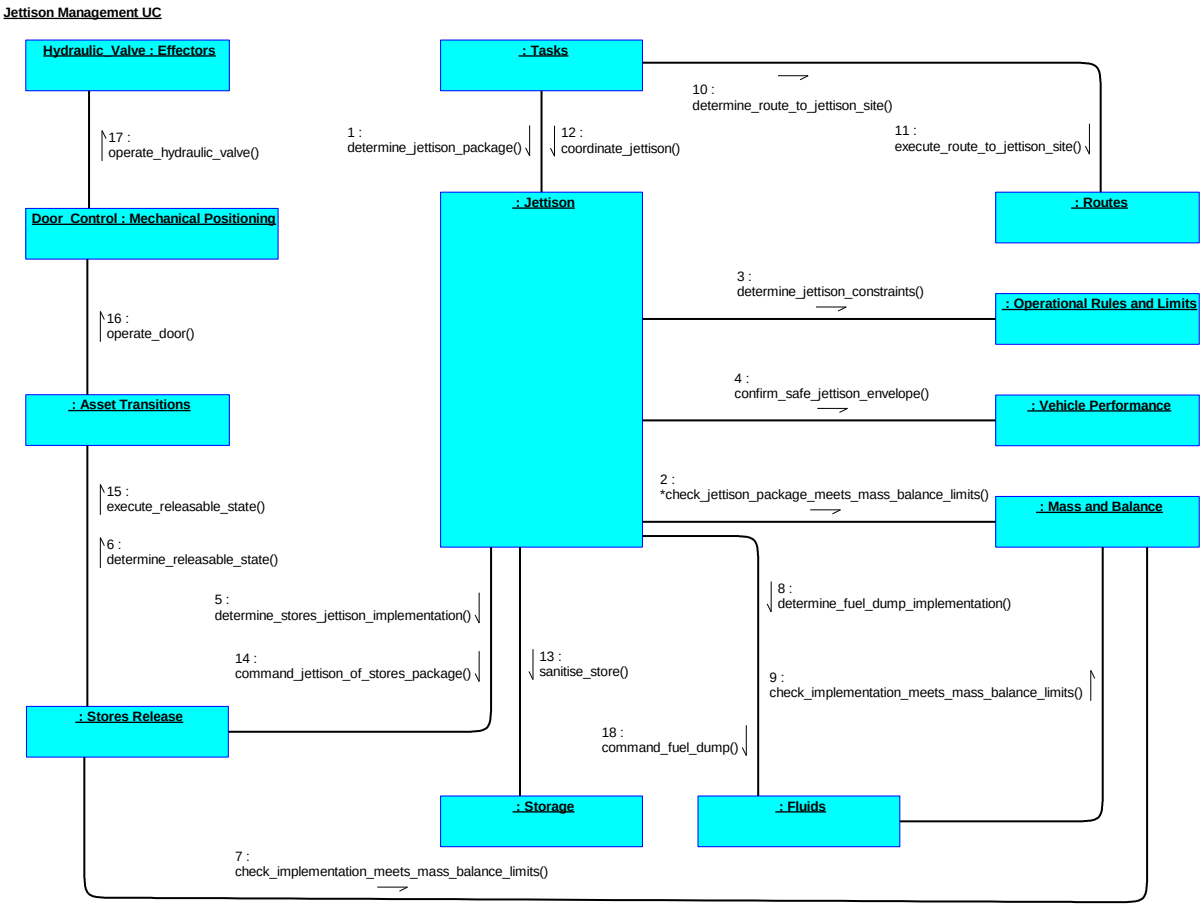

Figure 1371: Jettison Management IV

Tasks determines there is a requirement to make the vehicle lighter to achieve a task, this request is sent to Jettison. Jettison to determines a jettison package that consists of stores and fuel that will achieve the desired reduction of weight. As part of the selection, Jettison ensures that the package conforms to limits set by Mass and Balance, as well as the feasibility of performing the jettison given the limits provided by Operational Rules and Limits and Vehicle Performance.

Jettison requests that Stores Release determines a stores jettison solution. Asset Transitions is instructed by Stores Release to determine an infrastructure state that will make the jettison package releasable (i.e. opening doors before release). Stores Release checks with Mass and Balance that its solution is acceptable. Jettison requests that Fluids determines a fuel dump solution and Fluids checks with Mass and Balance that its solution is acceptable.

Tasks determines that the jettison is feasible, but with the pre-condition that the air vehicle must first be flown to an approved jettison location. Tasks asks Routes to determine the route to the jettison location, and then to fly to the desired location.

Once the desired location has been reached, the jettison pre-condition is satisfied and Tasks commands the execution of the planned jettison solution. Jettison requests Storage to sanitise any stores in the jettison package that contain sensitive data. Jettison then commands Stores Release to execute jettison of the stores. Asset Transitions is instructed by Stores Release to execute an infrastructure state change that will make the jettison package releasable. This is executed via Door_Control: Mechanical Positioning and Hydraulic Valve: Effectors (as the doors are operated by hydraulic pressure).

After the store has been jettisoned, Jettison commands Fluids to perform the fuel dump.

C.11.2.3 Post-Conditions

- A coordinated jettison of a stores package and a fuel dump has been executed.

C.12 Resource Views

This section contains IVs relating to the physical resources of a system.

C.12.1 Fuel Distribution

This IV describes a specific example of fuel being made available to the engines of the air vehicle. The distribution of fuel within the tanks takes into account centre of gravity constraints provided by Mass and Balance.

It is assumed that:

- Fluids has knowledge of all vehicle fuel.
- Fuel is a liquid.

The following related areas of functionality are excluded:

- Health assessment / anomaly detection.
- Refuelling.
- Management of jettison of fuel / fuel tanks.
- HMI.
- Controlling the rate of fuel burn by the engines, in response to throttle demands.

C.12.1.1 Pre-Conditions

- System is initialised and running with all removable tanks accounted for by Inventory.

C.12.1.2 View

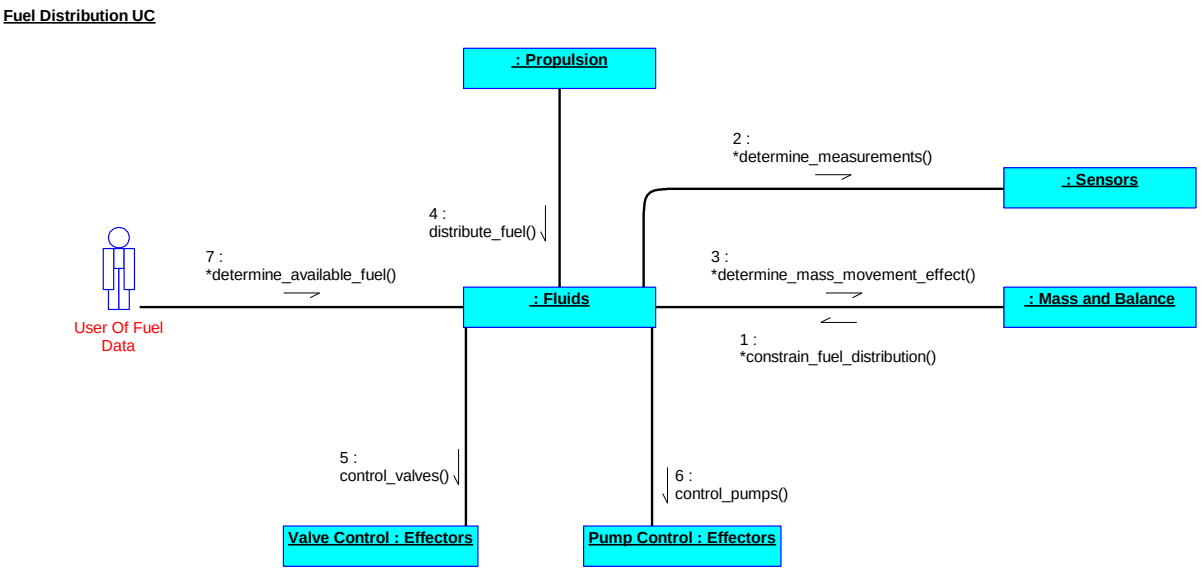

Figure 1372: Fuel Distribution IV

Mass and Balance continuously provides Fluids with constraints regarding the distribution of fuel, to ensure that the centre of gravity remains within limits. Fluids determines from Sensors the raw measurements (e.g. capacity and temperature) so that Fluids can determine the quantity of fuel. Fluids reports to Mass and Balance how the fuel is distributed so that Mass and Balance can determine the effect that the distribution may have on the centre of gravity, mass and inertia. In response to a demand by Propulsion, Fluids makes fuel available to engine(s) using Valve_Control: Effectors and Pump_Control: Effectors (to enable fuel transfer and venting), subject to Mass and Balance constraints. Fluids provides fuel related information, such as the amount of available fuel, to a User Of Fuel Data.

C.12.1.3 Post-Conditions

- A fuel demand from Propulsion is met, and the centre of gravity remains within limits.

C.12.1.4 Actors

User Of Fuel Data

A user of the fuel data provided by Fluids. This may be a crew member observing the data using an HMI or another component within the architecture.

C.12.2 Fuel Management

C.12.2.1 Fuel Management Pre-Mission

This IV describes the high level planning of fuel requirements needed for a mission, prior to mission commencement. Routes determines the fuel costs in order to fulfil the mission requirements placed upon it by Tasks.

It is assumed that:

- Resource Brokerage has received no other unresolved demands for fuel allocation.

The following related areas of functionality are excluded:

- Contingency fuel planning.
- HMI.
- Mission Data Load.

C.12.2.1.1 Pre-Conditions

- A task exists that requires a plan to be generated.

C.12.2.1.2 View

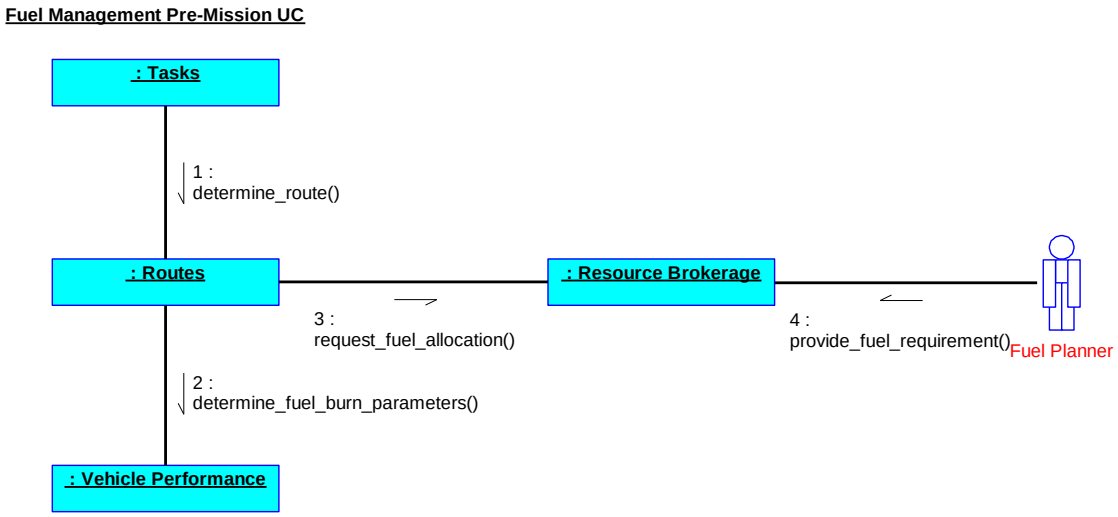

Figure 1373: Fuel Management Pre-Mission IV

Tasks requests a route from Routes based upon requirements. Routes determines a route to the specified location and collates the fuel cost for this route with input from Vehicle Performance. Resource Brokerage then allocates the fuel based upon requests placed upon it from Routes and provides the Fuel Planner with the fuel requirement.

C.12.2.1.3 Post-Conditions

- The Fuel Planner knows the fuel requirement.

C.12.2.1.4 Actors

Fuel Planner

The person responsible for creating a fuelling plan from the fuel requirements.

C.12.2.2 Fuel Management During a Mission

This IV describes how an air vehicle's fuel reserves are allocated and monitored during a mission. The vehicle is following a planned route, but is forced to perform a replan due to a change in routing requirements.

It is assumed that:

- The mission objectives are still achievable after a route replan due to sufficient fuel reserves.

The following related areas of functionality are excluded:

- Further detail of route generation and progression.
- HMI.

C.12.2.2.1 Pre-Conditions

- The vehicle is in flight.

C.12.2.2.2 View

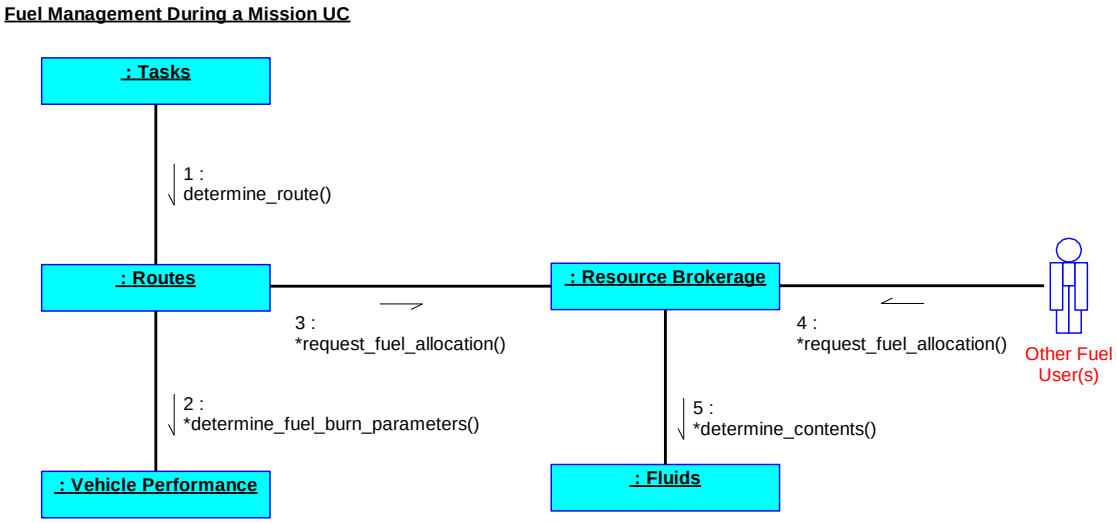

Figure 1374: Fuel Management During a Mission IV

Routes continually updates the planned route fuel cost based on updates from Vehicle Performance. Concurrently, Fluids determines its contents so that Resource Brokerage can allocate the updated route fuel cost alongside updates from Other Fuel User(s), while also ensuring that there is sufficient unallocated fuel. At some point in time Tasks receives a requirement to change route, and therefore requests Routes to determine a suitable new route. The revised route fuel cost is calculated and sent to Resource Brokerage, which determines that in this scenario there is sufficient fuel for the route change. The revised route fuel is then allocated and a request for route authorisation is ready to be sent.

C.12.2.2.3 Post-Conditions

- The route has been successfully updated.

C.12.2.2.4 Actors

Other Fuel User(s)

A planned user of fuel with a corresponding fuel cost previously calculated.

C.12.3 Power Management

This IV illustrates how electrical power management can be implemented for an Exploiting Platform, in this case an air vehicle. It is anticipated that, under normal operating conditions, the demand for electrical power will be adequately met by the supplies available, however there may be occasions when this is not true and the available supplies are restricted, e.g. following a failure or damage to the electrical systems (generators, busbars, etc.) resulting in a diminished power system capability. This IV describes such a situation when demand has to be managed in order not to exceed the available supply. It includes demands from a high powered sensor (including for cooling), engines in maximum reheat and the pre-launch conditioning requirements of a missile, but other demands could be added or substituted.

It is assumed that:

- The air vehicle is operating on its own electrical supply, with on board generators and battery reserves.
- The main generators are part of the propulsion system (and hence the use of maximum reheat leads to diminished power generation capability).
- Any required power management usage profiles, prioritisation policies and interruptibility strategies are loaded into the system.
- The available short range missile requires a dedicated start-up sequence before it can be used and also requires pre-launch conditioning of the missile sensor.
- Additional power required for flight-critical services (such as cooling the equipment in the avionics bay) is ring-fenced.
- The control of the cooling aspects of the scenario is limited to a simple enabling / disabling of power to the cooling systems. Hence the resource required by the Environmental Conditioning component in order to fulfil its actions is power, and the power demand is made by that component directly. The other cases in the scenario are different in that the resources required by the other action layer components are the associated hardware (i.e. engine, sensor or weapon), with power being a secondary resource enabling the capability of that hardware. Hence the power demands are made from the associated resource user components (Propulsion, Sensors or Asset Transitions) in those cases.

The following related areas of functionality are excluded:

- The specific chain of components which translate the solution created by Vehicle Guidance into specific demands on Propulsion.
- Intermediate steps between Tasks and Vehicle Guidance.
- The weapon selection chain beyond Target Engagement.

C.12.3.1 Pre-Conditions

- Whilst carrying out an air vehicle mission, tasking is received to intercept and engage a target as a priority.

C.12.3.2 View

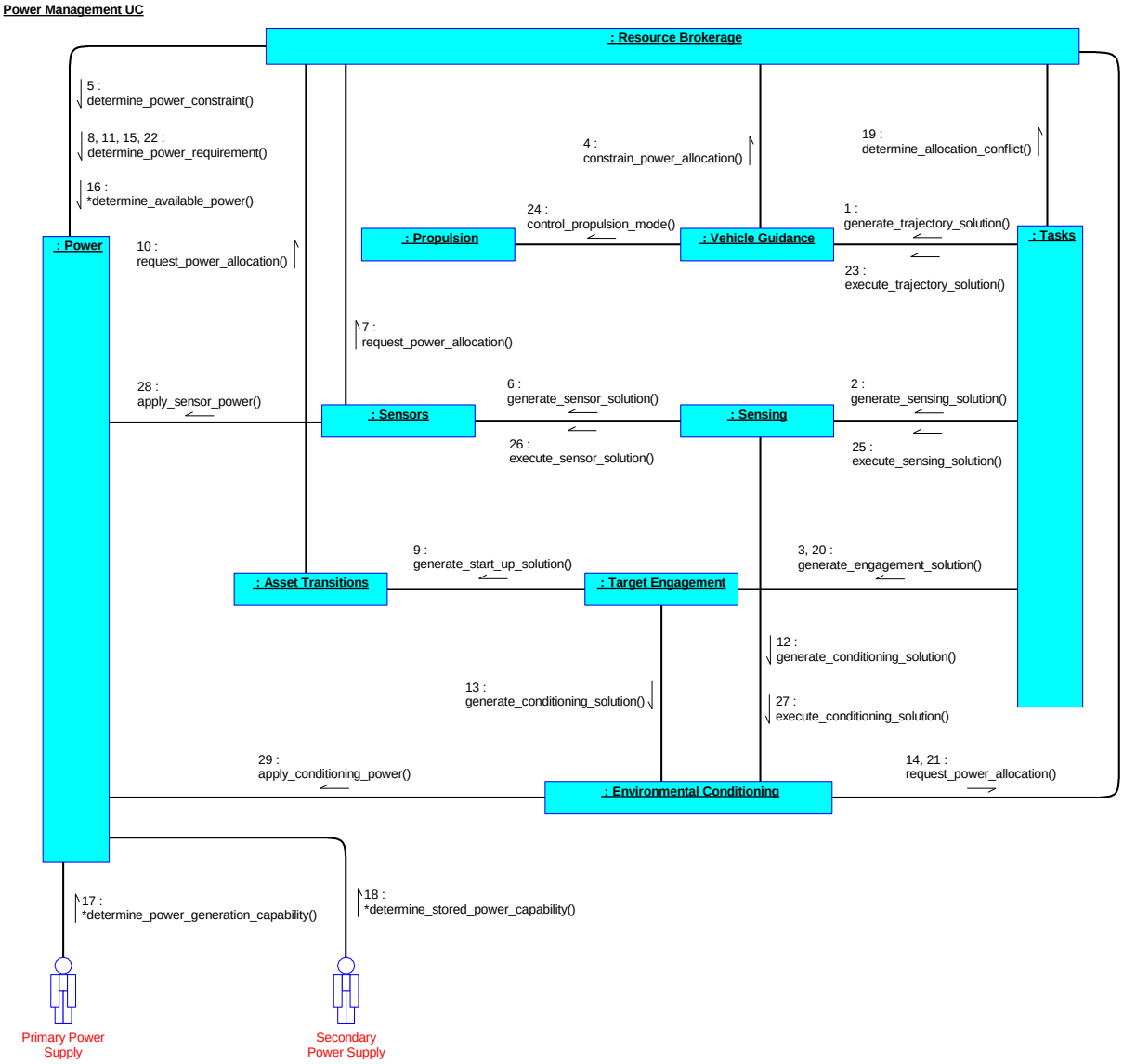

Figure 1375: Power Management IV

Tasks provides initial tasking to:

- Generate a trajectory solution to intercept the target;
- Generate a sensing solution to track the target;
- Generate an engagement solution for engaging the target.

As a result of this:

- Vehicle Guidance determines the need to use the engines' maximum propulsive capability (i.e. max reheat) and places a requirement on Propulsion for maximum propulsive capability. Propulsion informs Power that less generation capacity is available and Power informs Resource Brokerage that there is less power available to be brokered.
- Sensing determines the sensor capability required to execute the sensing solution and requests Sensors to determine how to provide that capability. Sensors determines that power is required and requests an allocation from Resource Brokerage.
- Target Engagement determines the weapon capability required to execute the engagement solution - in this case there is a dedicated start-up sequence required for the missile so Target Engagement requests Asset Transitions to generate the weapon start-up solution. Asset Transitions determines that power is required as part of the start-up sequence and requests an allocation from Resource Brokerage.
- Environmental Conditioning receives the requirements to cool both the sensor and the missile (from Sensing and Target Engagement respectively). It determines that power is required and requests an allocation from Resource Brokerage.

Resource Brokerage requests the levels of available resource from Power in order to determine if conflicts exist. Power gathers information about the available supply of electrical energy from the generators (within the engines) and that stored in battery reserves based on their modes of operation. Based on the power available for allocation, Resource Brokerage determines that the demand for power outstrips the available supply and, in this case, the power allocation requested by Environmental Conditioning cannot be met. The sources of the allocation requests are informed of the outcome.

Environmental Conditioning informs Sensing and Target Engagement that it cannot provide cooling to the sensor and missile concurrently due to a lack of available power. Both components determine that their actions cannot be fulfilled without the cooling resource and inform Tasks of their inability to generate their requested solutions.

Tasks requests details of the conflict in power allocations from Resource Brokerage and updates its task requirements such that the intercept and sensing actions take precedence over the engagement of the target (i.e. the requirement for Target Engagement to generate an engagement solution at this time is removed). The process described above is repeated, but with a reduced power allocation being required by Environmental Conditioning. Resource Brokerage indicates that the reduced amount of power demands can be allocated concurrently and hence Sensing and Vehicle Guidance inform Tasks that solutions can be generated. Tasks commands Vehicle Guidance and Sensing to execute their solutions and Sensing commands Environmental Conditioning to execute the updated conditioning solution (i.e. to cool the sensor only). Demands for the power allocations already agreed are made by Sensors and Environmental Conditioning.

C.12.3.3 Post-Conditions

- Missile conditioning is delayed, while precedence is given to intercept trajectory and tracking.

C.12.3.4 Actors

Primary Power Supply

The primary Power_Source for electrical power (e.g. a generator within the propulsion system)

Secondary Power Supply

Auxiliary or backup electrical Power_Source (e.g. a battery).