This document is no longer extant and has been withdrawn.

MOD Architecture Framework (MODAF)

MODAF is an internationally recognised enterprise architecture framework developed by the MOD to support Defence planning and change management activities. It does this by enabling the capture and presentation of information in a rigorous, coherent and comprehensive way that aids the understanding of complex issues.

Team announcement

This site replacesÂ www.modaf.org.uk. We areÂ keen to ensure that this new site is as user-friendly and useful as possible. Please e-mail any comments or suggestions for improving this site to the contact address shown.

This site has been developed as home for authoritative information relating to all aspects of the use and development of MODAF, the MOD Architecture Framework.

modaf

MOD Architecture Framework

"MODAF is a set of rules on how to organise information about the business".

MODAFÂ Site Contents

The MODAFÂ site is divided into the following main sections:

- MODAF Detailed Guidance
- The MODAF views and viewpoints
- Examples and use of MODAF
- The MODAF Meta Model (M3)
- Frequently asked questions & glossary
- MODAF Configuration Control

Links to these sections may be found in the â€˜Related Pagesâ€™ section.

It is best to start with thw MODAF detailed guidance page, for an overview of the views used in MODAF.

Who should use this site?

This site is set up to support anyone with an interest in MODAF. This audience includes:

- Enterprise Architects, the principal customers for MODAF views, who need to both correctly interpret standard MODAF views provided to them and to specify and control the tasks required to create new views

Information Management

in this section:

- ICAD
- Information Coherence Group
- **MOD Architecture Framework**

Contact us

Contact the MODAF Team

MODAF Version History

Aug 2005 - MODAF V1.0 Released
Apr 2007Â - MODAF V1.1 Released
AprÂ 2008 - MODAF V1.2.003 Released
May 2010 - MODAF V1.2.004 Released

Related pages

MODAF - Guidance
Detailed Guidance on MODAF

Viewpoints and views
Overview of the differentÂ viewsÂ used in MODAF

Examples and use of MODAF
Including an example of a generic architecting process

MODAF Meta Model (M3)
link to the Meta Model

MODAF - FAQ's
Frequently asked questions

Configuration control
Version history

Information Coherence Authority for Defence (ICAD)
Related team

MODAF - Guidance

External links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

- Architectural modellers who need guidance on the creation and interchange of MODAF views (including for example: architecting principles, view coherence rules and tool selection criteria)
- Tool developers and engineers who are implementing architectural data repositories for storing and manipulating MODAF Architecture data elements
- Trainers and educators who require reference material in order to appropriately train and support the previous types of MODAF users
- MODAF users who wish to contribute to the development of MODAF
- Managers who need to understand what views are required to answer their particular questions

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

MODAF â€“ detailed guidance

Documentation supporting the MOD Architecture Framework (MODAF).

Team announcement

This site replacesÂ www.modaf.org.uk. We areÂ keen to ensure that this new site is as user-friendly and useful as possible. Please e-mail any comments or suggestions for improving this site to the contact address shown.

modaf

MOD Architecture Framework

Looking for an overview of MODAF?
Below is an overview of MODAF.

From here you can navigate back to the home page or down for more information.

Guidance on MODAF

MODAF provides a coherent set of rules and templates, known as Views that, when populated, provide a graphical and textual visualisation of the business area being investigated. Each View offers a different perspective on the business to support different stakeholder interests. The Views are divided into seven categories:

- **Strategic Views (StVs)** define the desired business outcome, and what capabilities are required to achieve it
- **Operational Views (OVs)** define (in abstract rather than physical terms) the processes, information and entities needed to fulfil the capability requirements
- **Service Oriented Views (SOVs)** describe the services, (i.e. units of work supplied by providers to consumers), required to support the processes described in the operational Views
- **Systems Views (SVs)** describe the physical implementation of the Operational and Service Orientated Views and, thereby, define the solution
- **Acquisition Views (AcVs)** describe the dependencies and timelines of the projects that will of deliver the solution
- **Technical Views (TVs)** define the standards that are to be applied to the solution
- **All Views (AVs)** provide a description and glossary of the contents of the architecture

To ensure the coherence between the Views, MODAF is underpinned by a model which defines the relationship between all the data in all the Views. This model is called the MODAF Meta Model, also known as the "M3". The M3 also provides a technical standard to enable the exchange of data between architectures developed in different modelling (software) applications.

MODAF supports the application of rigour to requirements capture

The use of MODAF provides a de-facto element of rigour to requirements capture

Information Management

in this section:

- ICAD
- Information Coherence Group
- **MOD Architecture Framework**

Contact us

Contact the MODAF Team

MODAF Version History

Aug 2005 - MODAF V1.0 Released
Apr 2007Â - MODAF V1.1 Released
AprÂ 2008 - MODAF V1.2.003 Released
May 2010 - MODAF V1.2.004 Released

Related pages

Viewpoints and views
Overview of the differentÂ viewsÂ used in MODAF

Examples and use of MODAF
Including an example of a generic architecting process

MODAF Meta Model (M3)
link to the Meta Model

MODAF - FAQ's
Frequently asked questions

Configuration control
Version history

Information Coherence Authority for Defence (ICAD)
Related team

MODAF - Home Page

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture

because the population of the Views requires the application of a structured analytical approach that leads the user from desired outcome to solution options.

MODAF supports the modelling of options

There are a number of commercially available tools that support the use of MODAF. As well as allowing the presentation of the Views, these tools also provide a repository in which the architecture can be stored and enable the modelling of different change options to support decision making.

MODAF supports interoperability

The use of MODAF as the standard architecture framework enables the coherent sharing of architectural information which helps identify gaps and overlaps between operating processes and the systems that support them.

MODAF has pedigree

MODAF was developed by MOD from the US Department of Defense Architecture Framework (DoDAF) version 1.0, but has been extended and modified to meet MOD requirements. MODAF is now itself internationally recognised as a best practice for enterprise architecting, and provided the template against which NATO Architecture Framework version 3.0 was developed.

MODAF has been adopted by organisations outside MOD

As well as MOD, MODAF is widely used by its industry partners, such as BAE Systems, Thales, Lockheed Martin, Boeing and Serco. It is also used by other government departments, such as GCHQ, and external bodies, such as the National Air Traffic Services. MODAF was recently adopted for use by the Swedish Armed Forces.

Â

Framework

Access online conversion tools for Adobe PDF documents

Viewpoints and views

This section provides an overview of the different viewpoints used in the MOD Architecture Framework (MODAF).

MODAF architectures are developed as coherent, contiguous models that when viewed as a whole present a complete picture of the enterprise. MODAF defines a rich selection of relationships which can be used to integrate the various architectural elements.

Producing an enterprise architecture is rarely the work of one person and it is sometimes useful to be able to logically divide an architecture into domains, each concerned with one aspect of how the enterprise works. This also proves useful when publishing an architecture to different stakeholders. For this reason, MODAF defines a set of standard viewpoints.

Each Viewpoint takes a different perspective upon the architectural model; for instance, the Operational Viewpoint considers the operaional nodes (logical "actors" that may be realised by one or more resources) that interact in certain ways in order to achieve a desired outcome.

The MODAF views

Each Viewpoint consists of several views, which highlight slightly different details within the particular Viewpoint. For instance within the Operational Viewpoint, OV-1 provides a high-level conceptual graphic, whilst OV-2 considers the interactions between operational nodes and OV-3 details the information flows.

Whilst the data within each view adds more richness to the overall description of an architecture, it is not necessary for all of the MODAF views to be completed at any particular point in time during the MOD's acquisition life-cycle. Indeed, each group of users within the MOD will have different needs and will only populate and exploit those MODAF Views that are of relevance to them. This means that most of the MOD's Communities of Interest (COIs) will only be dealing with the population and exploitation of a subset of MODAF Views, and few will need to understand and deal with all of the available MODAF Views.

Links to more detailed descriptions of each viewpoint and their constituent views can be found listed in the "Related Pages" section. A link to a high-level summary of views may also be found there.

Interactions between views and interactions between architectures

It is expected that the Strategic Views (StVs) cover more than one operational architecture - i.e. the capabilities defined in the StVs are re-used across a number of architectures. It may also be the case that the architect wishes to conduct an architectural trade study - i.e. there may be multiple possible solutions for a given

Information Management

in this section:

- ICAD
- Information Coherence Group
- **MOD Architecture Framework**

Contact us

Email the MODAF team

Related pages/documents

- **Views summary documents**
- **All-views viewpoint**
- **Strategic view viewpoint**
- **Operational view viewpoint**
- **System view viewpoint**
- **Technical standards view viewpoint**
- **Acquisition view viewpoint**
- **Service-oriented view viewpoint**

MODAF - detailed guidance

External links

- **International Defence Enterprise Architecture Specification (IDEAS) Group**
- **NATO Architcture Framework (NAF)**
- **Department of Defence Architecture Framework**

Access online conversion tools for Adobe PDF documents

requirement in the Operational Viewpoints (OVs).

These relationships are covered in more detail in the guidance for each viewpoint and in the document "MODAF Layers and Viewpoint Linkages" - which can be found on the Views Summary Documents page. See **Related Pages >>>**

These relationships are covered in more detail in the guidance for each viewpoint and in the document â€œMODAF Layers and Viewpoint Linkagesâ€• - which can be found on the MODAF Views Summary Documents page, together with a downloadable version of this page.

This document is no longer extant and has been withdrawn.

Defence |

about defence

Views summary documents

Documents summarising the MOD Architecture Framework (MODAF) Views and Viewpoints.

View Summary is intended to help choose the most appropriate MODAF views. It provides a list of all views and details what each view is used for and what data elements they contain.

Layers and Viewpoint Linkages provides a top-level view of how MODAF views relate to each other.

There is also a downloadable version of the Viewpoints and Views homepage.

- **View summary.pdf PDF [179.4 KB]**
- **MODAF Layers and Viewpoint Linkages.pdf PDF [340.3 KB]**
- **StV2 OV2 OV5 SV1 SV4U views picture.png PNG [144.0 KB]**
- **ViewsHomeDownloadable.pdf PDF [129.8 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

ViewpointsÂ andÂ views
Parent page

All-views viewpoint

Strategic view viewpoint

Operational view viewpoint

System view viewpoint

Technical standards view viewpoint

Acquisition view viewpoint

Service-oriented view viewpoint

MODAF - detailed guidance

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page rated 1 times

This document is no longer extant and has been withdrawn.

A SUMMARY OF MODAF VIEWS BY THEIR USE AND DATA TYPES

This document provides a summary of the MODAF viewpoints; for each viewpoint, it lists the uses for that viewpoint and the data objects which each viewpoint contains.

VIEWPOINT	CATEGORY													
	Tabular		Structural		Behavioural		Mapping		Ontology		Pictorial		Timeline	
All Views	AV-1	Link							AV-2	Link				
Strategic	StV-1	Link	StV-4	Link			StV-3	Link	StV-2	Link				
							StV-5	Link						
							StV-6	Link						
Operational	OV-1b	Link	OV-2	Link	OV-5	Link					OV-1a	Link		
	OV-1c	Link	OV-4	Link	OV-6a	Link								
	OV-3	Link	OV-7	Link	OV-6b	Link								
					OV-6c	Link								
System	SV-6	Link	SV-1	Link	SV-4	Link	SV-3	Link					SV-8	Link
	SV-7	Link	SV-2a	Link	SV-10a	Link	SV-5	Link						
	SV-9	Link	SV-2b	Link	SV-10b	Link	SV-12	Link						
			SV-2c	Link	SV-10c	Link								
			SV-11	Link										
Technical	TV-1	Link												
	TV-2	Link												
Acquisition			AcV-1	Link									AcV-2	Link
Service Oriented	SOV-2	Link			SOV-4a	Link	SOV-3	Link	SOV-1	Link				
					SOV-4b	Link								
					SOV-4c	Link								
					SOV-5	Link								

View Categories

Tabular: Views which are essentially tabular, which includes structured text as a special case

Structural: This category comprises diagrams describing the structural aspects of an Architecture.

Behavioural: This category comprises diagrams describing the behavioural aspects of an Architecture.

Mapping: These views provide matrix (or similar) mappings between two different types of information.

Ontology: Views which extend the MODAF ontology for a particular Architecture.

Pictorial: This category comprises just one view, namely OV-1a, which is essentially a free-form picture.

Timeline: This category comprises diagrams describing the programmatic aspects of an Architecture.

Clicking on the “link” takes you to the summary of the view.

This document is no longer extant and has been withdrawn.

All Views Viewpoint (AV)

An overarching description of the architecture, its scope, ownership, timeframe and all of the other meta data that is required in order to effectively search and query architectural models.

AV View	Used for	Data objects	
AV-1 Overview & Summary Information	• Scoping the project. • Providing context to the project. • Definition of an architecture-based task. • Summarising the findings from an architecture-based task. • Assisting search within an architecture repository.	• Scope • Purpose • Listing of views used	Back to table
AV-2 Integrated Dictionary	• AV-2 presents all the Elements used in an architecture as a stand alone structure. An AV-2 presents all the Elements as a specialisation hierarchy, provides a text definition for each one and references the source of the element (e.g. MODAF Ontology, IDEAS Model, local, etc.). • An AV-2 shows elements from the MODAF Ontology that have been used in the architecture and new elements (i.e. not in the MODAF Ontology) that have been introduced by the architecture.	• Ontology References • Specialisation Relationships (Subtyping) • Type-Instance Relationships	Back to table

Strategic Viewpoint (SV)

These views support to the process of analysing and optimising the delivery of military capability in line with the MOD's strategic intent.

StV View	Used for	Data objects	
StV-1 Enterprise Vision	• Communication of strategic vision regarding capability evolution	• Enterprise Vision • Enterprise Phase • Enterprise Goals • Capability • Enduring Task	Back to table

This document is no longer extant and has been withdrawn.

StV View	Used for	Data objects	
StV-2 Capability Taxonomy	• Identification of capability requirements • Capability planning (capability taxonomy) • Codifying required capability elements • Capability audit • Capability gap analysis • Source for the derivation of cohesive sets of KUR • Providing reference capabilities for architectures	• Capability • Capability Specialisation (relationship between capabilities) • Enterprise Phase	Back to table
StV-3 Capability Phasing	• Capability planning (capability phasing) • Capability integration planning • Capability gap analysis	• Capability • Capability Configuration • Capability Increment (Project Milestone) • Out of Service (Project Milestone) • Enterprise Phase	Back to table
StV-4 Capability Dependencies	• Identification of capability dependencies • Capability management (impact analysis for options, disposal etc)	• Capability • Capability Dependency (relationship) • Capability Composition (relationship)	Back to table
StV-5 Capability to Organisation Deployment Mapping	• Fielding planning • Capability integration planning • Capability options analysis • Capability redundancy/overlap/gap analysis • Identification of deployment level shortfalls	• Capability • Capability Configuration • Resource Interaction (between Capability Configurations or their components) • Actual Organisational Resource (Actual Post, Actual Organisation) • Capability Delivery (Project Milestone) • Capability No Longer Used (Project Milestone)	Back to table
StV-6 Operational Activity to Capability Mapping	• Tracing capability requirements to enduring tasks • Capability audit	• Capability • Standard Operational Activity	Back to table

Operational Viewpoint (OV)

These views describe a requirement for a to-be architecture in logical terms, or as a simplified description of the key behavioural and information aspects of an as-is architecture.

OV View	Used for	Data objects	
OV-1a High-Level Operational Concept Graphic	• Puts an operational situation or scenario into context • Provides a tool for discussion and presentation; for example, aids industry engagement in acquisition • Can provide a common way in to more detailed information in published architectures	• Operational Nodes i.e. Headquarters • Systems i.e. aircraft • Organisations • Information Flows • Environmental context objects i.e. rivers, hills	Back to table
OV-1b Operational Concept Description	• Concept of Operations • Input to URD	OV-1b is a textual description of the OV-1a graphic so does not usually have specific data objects associated with it.	Back to table
OV-1c Operational Performance Attributes	• Definition of performance characteristics. • Measures of Effectiveness (input to URD).	• Metrics associated with performance associated with specific concepts within the scenario specified within the OV-1a.	Back to table
OV-2 Operational Node Relationship Description	• Definition of operational concepts. • Elaboration of capability requirements. • Definition of collaboration needs. • 'Localising' capability. • Problem space definition. • Operational planning. • Supply chain analysis.	• Nodes ("Operational Nodes"). • Needlines (bundles of information exchanges). • Logical Flows (of materiel, people or energy). • Operational Activities. • Locations.	Back to table
OV-3 Operational Information Exchange Matrix	• Definition of interoperability requirements	• Information Exchanges (each associated with a Needline) • Information Elements (each carried by one or more Information Exchange)	

OV View	Used for	Data objects	
OV-4 Organisational Relationships Chart	A **typical** OV-4 may be used for: • Organisational analysis • Definition of human roles • Operational analysis An **actual** OV-4 may be used to: • Identify architecture stakeholders • Identify process owners • Illustrate current or future organisation structures	• Organisation Types • Resource Composition relationships • Resource Interaction relationships • Post Types • Role Types • Actual Posts and Organisations • Competences	Back to table
OV-5 Operational Activity Model	• Description of business processes and workflows. • Requirements capture (input to URD). • Definition of roles and responsibilities. • Support task analysis to determine training needs. • Problem space definition. • Operational planning. • Logistic support analysis. • Information flow analysis.	• Operational activities. • Standard operational activities (originating in StV-6). • Operational Activity Flow Objects • Swimlanes (each associated with a node).	Back to table
OV-6a Operational Rules Model	• Definition of doctrinally correct operational procedures • Definition of business rules • Identification of operational constraints	• Operational constraints	Back to table
OV-6b Operational State Transition Description	• Analysis of business events. • Behavioural analysis. • Identification of constraints (input to SRD).	• States (each associated with a mission, node or operational activity.) • State transitions (each associated with an event).	Back to table
OV-6c Operational Event-Trace Description	• Analysis of operational events. • Behavioural analysis. • Identification of non-functional user requirements (input to URD). • Operational test scenarios.	• Lifelines (each associated with a Node).	Back to table
OV-7 Information Model	• Information architecture. • Information product hierarchy.	• Operational Information Entity.	Back to table

System Views

Describe the resources that realise capability.

SV View	Used for	Data objects	
SV-1 Resource Interaction Specification	• Definition of system concepts • Definition of system options • Interface requirements capture • Capability integration planning • System integration management • Operational planning (capability configuration definition)	• Artefact • Organisation Type • Post Type • Role Type • Software • Capability Configuration • Resource Composition • Resource Interaction	Back to table
SV-2a System Port Specification	• Interface specification • Identification of applicable protocols • Description of system communication paths	• System • System Port • Protocol	Back to table
SV-2b System Port Connectivity Description	• Interface specification	• System • System port • Port connection • Protocol	Back to table
SV-2c System Connectivity Clusters	• Interface specification. • Bandwidth and capacity analysis.	• Physical asset. • Organisational resource (post type or organisation type). • System. • System port. • System port connection.	Back to table
SV-3 Resource Interaction Matrix	• Summarising resource interactions. • Interface (ICD) management. • Comparing interoperability characteristics of solution options.	• Resource types. • Resource interactions.	Back to table
SV-4 Functionality Description	• Description of task workflow. • Identification of functional system requirements. • Functional decomposition of systems. • Relate human and system functions.	• Function • Resource • Data Element	Back to table

SV View	Used for	Data objects	
SV-5 Function to Operational Activity / Service Function Traceability Matrix	• Tracing functional system requirements to user requirements. • Tracing solution options to requirements. • Identification of overlaps.	• Function. • Resource. • Operational activity. • Service function.	Back to table
SV-6 Systems Data Exchange Matrix	• Detailed definition of data flows.	• System. • Resource interaction. • System port connector. • Data element. • Information exchange (OV-2).	Back to table
SV-7 Resource Performance Parameters Matrix	• Definition of performance characteristics. • Identification of non-functional requirements (input to SRD).	• Resource (system, role, or capability configuration). • Measurable property. • Qualitative property.	Back to table
SV-8 Capability Configuration Management	• Development of incremental acquisition strategy. • Planning technology insertion.	• Capability configurations. • Resources that are parts of capability configurations. • Project milestone (reflecting capability delivery).	Back to table
SV-9 Technology & Skills Forecast	• Forecasting technology readiness against time. • HR trends analysis. • Recruitment panning. • Planning technology insertion. • Input to options analysis.	• Resources. • Competences. • Standards. • Forecasts (for the any of the above).	Back to table
SV-10a Resource Constraints Specification	• Definition of implementation logic. • Identification of resource constraints.	• Resource constraint.	Back to table
SV-10b Resource State Transition Description	• Definition of states, events and state transitions (behavioural modelling). • Identification of constraints (input to System Requirements Document).	• Resources. • States (associated with a resource or function). • State transitions (each associated with an event).	Back to table

SV View	Used for	Data objects	
SV-10c Resource Event-Trace Description	• Analysis of resource events impacting operation. • Behavioural analysis. • Identification of non-functional system requirements (input to System Requirement Document).	• Lifelines (each associated with a functional resource or a system port).	Back to table
SV-11 Physical Schema	• Specifying the system data elements exchanged between systems, thus reducing the risk of interoperability errors. • Definition of physical data structure (input to system design).	• System data entity.	Back to table
SV-12 Service Provision	• Service implementation. • Resource audit. • Tracing business processes to the resources that support them.	• Service. • Resource type	Back to table

Technical Standards Viewpoint (TV)

Standards, rules, policy and guidance that are applicable to aspects of the architecture.

TV View	Used for	Data objects	
TV-1 Standards Profile	• Application of standards (informing project strategy) • Standards compliance	• Standard • Protocol	Back to table
TV-2 Standards Forecast	• Forecasting future changes in standards (informing project strategy)	• Standard (evolution over time)	Back to table

Acquisition Viewpoint (AV)

Describe programmatic details, including dependencies between projects and capability integration across the Defence Lines of Development (DLODs).

AcV View	Used for	Data objects	
AcV-1 Acquisition Clusters	• Programme management (specified acquisition programme structure) • Project organisation	• Project • Project Owner • Enterprise Phase	Back to table
AcV-2 Programme Timelines	• Project management and control (including delivery timescales) • Project dependency risk identification • Management of dependencies within a System of Systems (including all Lines of Development) • Portfolio management (for System of Systems acquisition) • Through Life Management Planning (TLMP)	• Projects • Project Milestones • Threads (e.g. DLOD) • Project Dependencies • Capability Configurations	Back to table

Service Oriented Views

Specify Services that are to be used in a Service-Orientated Architecture (SOA).

View	Used for	Data objects	
SOV-1 Service Taxonomy	• SOA Governance • Identification of Services • Service Planning • Service Audit • Service gap analysis • Providing reference services for architectures • Tailoring generic services for specific applications	• Service • Service Generalisation (the specialisation relationship) • Service Attribute • Service Policy (optional, also shown in SOV-3)	Back to table

This document is no longer extant and has been withdrawn.

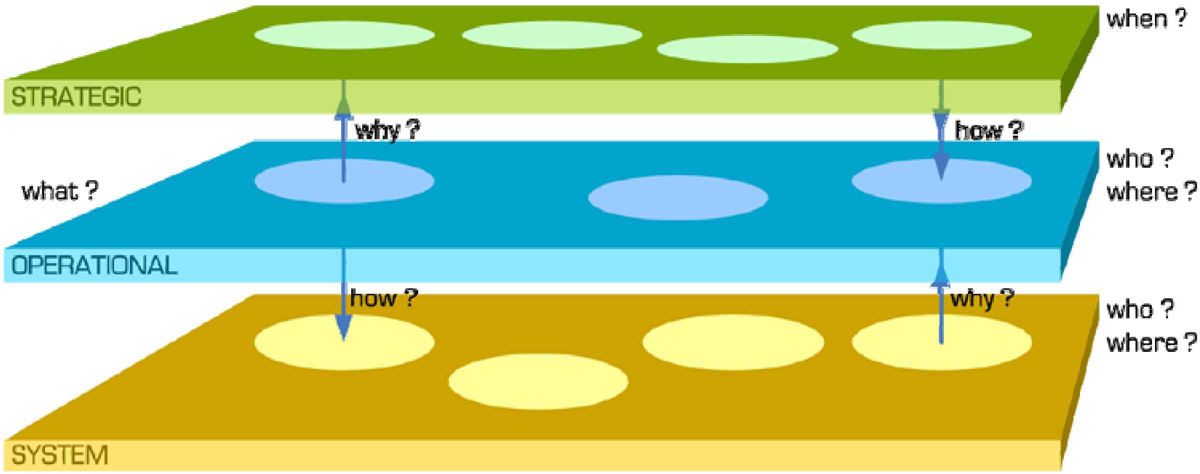
View	Used for	Data objects	
SOV-2 Service Interface Specification	• SOA Governance • Detailed Service Specification • Service Interoperability	• Service (Operational, Information and Application Service) • Service Interface • Service Interface Operation • Service Interface Parameter	Back to table
SOV-3 Capability to Service Mapping	• Service specification & planning • Governance	• Service (Operational, Information and Application Service) • Capability • Service Aims to Achieve (relationship from Service to Capability)	Back to table
SOV-4a Service Constraints	• Service Specification • Service Governance	• Service (Operational, Information and Application Service) • Service Policy	Back to table
SOV-4b Service State Model	• Service Specification	• Service (Operational, Information and Application Service) • Service State Machine	Back to table
SOV-4c Service Interaction Specification	• Service Specification	• Service(Operational, Information and Application Service) • Service Interface • Service Lifeline • Service Consumer	Back to table
SOV-5 Service Functionality	• Service Specification • Functional Requirements Definition	• Service(Operational, Information and Application Service) • Service Function	Back to table

Viewpoint linkages

The relationship between MODAF Viewpoints is best understood using the principle of layering, as explained in the article on MODAF Layers[1]. This article explores the exploitation of these relationships, in terms of the use of MODAF.

The diagram below illustrates the viewpoint linkages from the point of view of the author of a User Requirement Document (URD). Such a user of MODAF will have a focus on the operational viewpoint because models within this viewpoint describe **'what'** the bounded operational capability is that is the subject of the URD. From this perspective, any strategic views that are relevant to the requirements definition activity will provide context (both capability and business context) – the strategic views then answer the **'why'** question.

Similarly models in the system viewpoint offer alternative means of realising the operational capability of interest – these alternatives each provide one way of addressing the capability need, i.e. the **'how'**.



It is important to recognise that the **what-why-how** relationships depend upon the perspective of the MODAF user and the task they are engaged in. For example, a capability planner would regard models in the Strategic Viewpoint as describing **'what'** the capability needs are; then models in the Operational Viewpoint would represent (possibly alternative) representations of **'how'** those needs might be realised.

Similarly an author of a System Requirements Document (SRD) may be focused on the System Viewpoint as described **'what'** the equipment capability is that is needed. Models in the Operational Viewpoint provide operational capability context for that requirements definition task, i.e. **'why'** the equipment is needed (as well as which military tasks it would be used for).

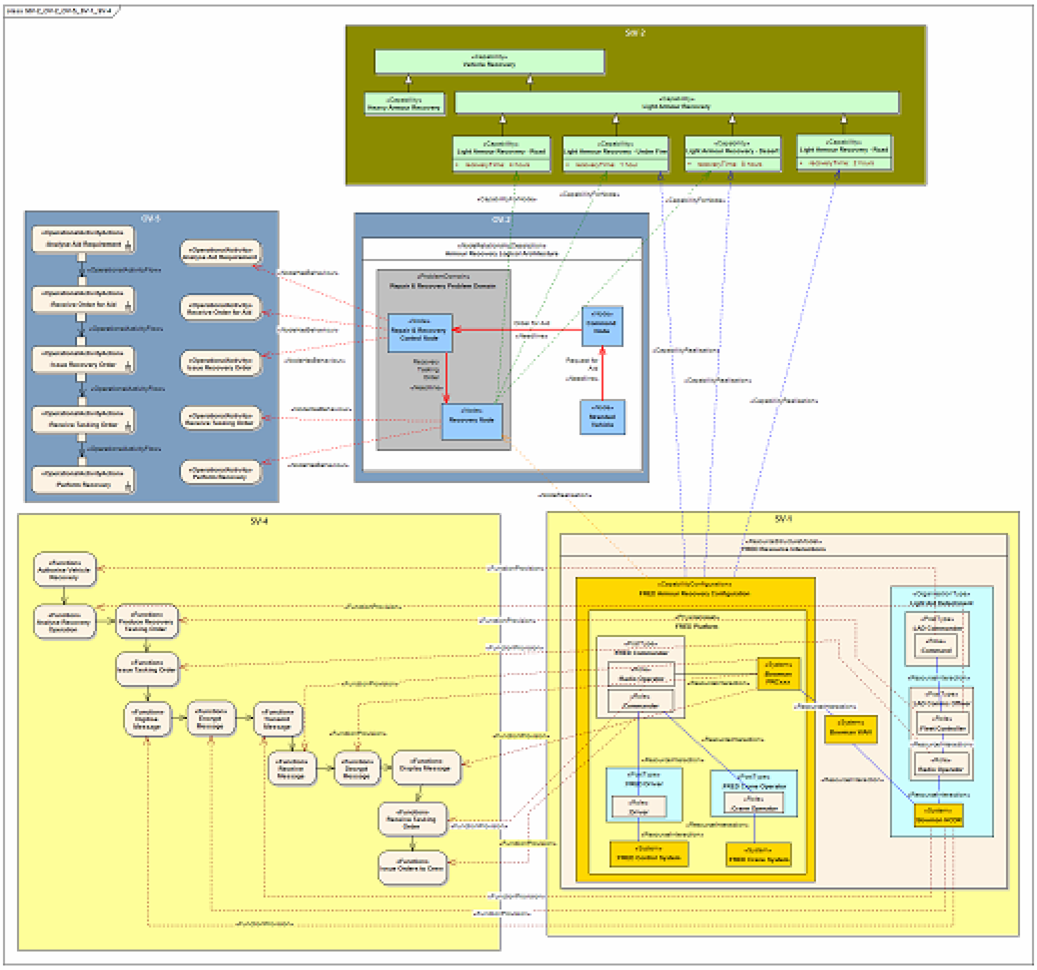
Use of the familiar **what-who-why-how-when-where** paradigm provides an opportunity to obtain a useful characterisation of the MODAF viewpoints (recognising that the **what-why-how** aspects tend to be subjective). This has been found to be particularly useful in differentiating the Strategic Viewpoint from the other MODAF viewpoints.

---

[1] A link to the document "How do MODAF layers interrelate?" is to be found on the "MODAF Concepts" page.

The Strategic Viewpoint addresses issues of timing (specifically capability phasing) but not, for example, personnel (e.g. transformation owners are not currently represented). The Operational Viewpoint addresses issues of personnel (specifically organisation and post types) as well as location (the location of nodes). The System Viewpoint also addresses personnel and location (but now in terms of roles and the deployment to physical assets). While not shown on the diagram, the Acquisition Viewpoint addresses personnel (project ownership) and timing (project timelines) but not, for example, where projects are based.

The example below shows how the core views – StV-2, OV-2, OV-5, SV-1 and SV-4 are linked:



Note: A link to a larger version of the example above is available on the “Views summary documents” page.

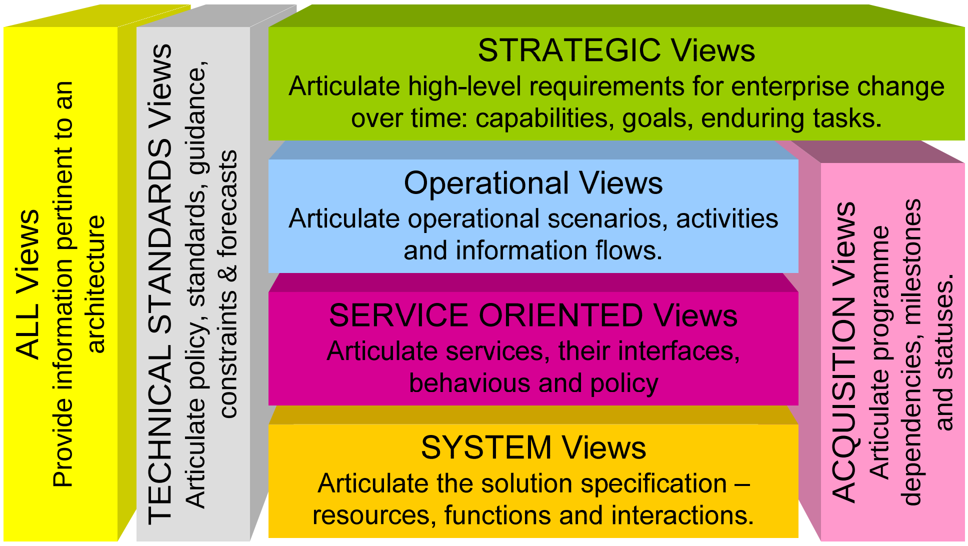
In this example, capabilities are traced to a node to specify what level of capability is required by whatever resource realises it. The links from the node the SV-1 show a capability configuration that can meet that required capabilities. Note also that the capability configuration itself is traced back to capabilities (via <<CapabilityRealisation>>>> dependencies), and in fact exceeds the capability specified for the operational node.

AN INTRODUCTION TO MODAF VIEWS

MODAF architectures are developed as coherent, contiguous models that when viewed as a whole present a complete picture of the enterprise. MODAF defines a rich selection of relationships which can be used to integrate the various architectural elements.

MODAF Viewpoints

Producing an enterprise architecture is rarely the work of one person and it is sometimes useful to be able to logically divide an architecture into domains, each concerned with one aspect of how the enterprise works. This also proves useful when publishing an architecture to different stakeholders. For this reason, MODAF defines a set of standard viewpoints:



How the MODAF Viewpoints relate to each other

This diagram illustrates the relationship between the six MODAF Viewpoints, in particular the way that:

- The Strategic, Operational, and System Viewpoints have a layered relationship.
- The Acquisition Viewpoint sits beneath the Strategic Viewpoint, and has a supporting role across the Operational and System layers.
- The All Views and Technical Standards Viewpoints sit alongside the others in their role of providing a description and ontology for an architecture, plus information on supporting standards.

Each Viewpoint takes a different perspective upon the architectural model; for instance, the Operational Viewpoint considers the operational nodes (logical "actors" that may be realised by one or more resources) that interact in certain ways in order to achieve a desired outcome.

MODAF Views

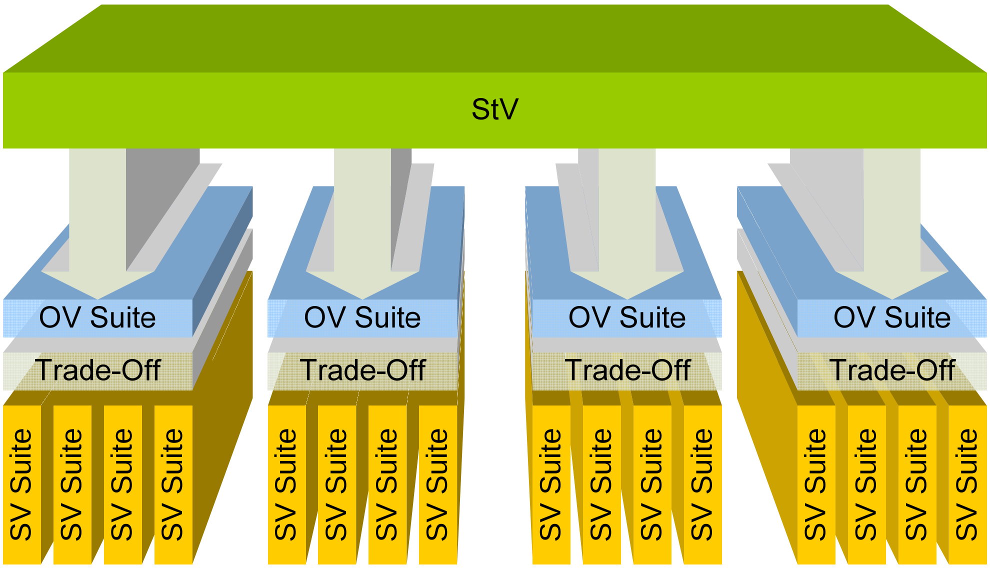
Each viewpoints consists of several views, which highlighting slightly different details within the particular viewpoint. For instance within the Operational Viewpoint, OV-1 provides a high level conceptual graphic, whilst OV-2 considers the interactions between operational nodes and OV-3 details the information flows.

Whilst the data within each view adds more richness to the overall description of an architecture, it is not necessary for all of the MODAF views to be completed at any particular point in time during the MOD's acquisition lifecycle. Indeed, each group of users within the MOD will have different needs and will only populate and exploit those MODAF Views that are of relevance to them. This means that most of the MOD's Communities of Interest (COIs) will only be dealing with the population and exploitation of a subset of MODAF Views, and few will need to understand and deal with all of the available MODAF Views.

Links to more detailed descriptions of each viewpoint and their constituent views can be found listed in the "Related Pages" section. A link to a high level summary of views may also be found there.

Interactions between Views and Interactions between Architectures

It is expected that the Strategic Views (SVs) cover more than one operational architecture – ie the capabilities defined in the StVs are re-used across a number of architectures. It may also be the case that the architect wishes to conduct an architectural trade study – ie there may be multiple possible solutions for a given requirement specified in the OVs:



The relationship between StVs, OVs and SVs

These relationships are covered in more detail in the guidance for each viewpoint and in the document "MODAF Layers and Viewpoint Linkages"[1].

[1] A link to this document can be found on the related links section of the 'view summaries' page.

All views (AV) viewpoint

A description of and guidance for the use of the MOD Architecture Framework (MODAF)Â All ViewsÂ viewpoint.

The All Views (AVs) provide an overarching description of the architecture - its scope, ownership, timeframe and all of the other metadata that is required in order to effectively search and query architectural models. They also provide a place to record any findings arising from the architecturing process. The AVs include a dictionary of the terms used in the construction of the architecture - which helps others fully understand its meaning at a later date.

- **AV viewpoint PDF [186.3 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

ViewpointsÂ andÂ views
Parent page

Views summary documents

Strategic view viewpoint

Operational view viewpoint

System view viewpoint

Technical standards view viewpoint

Acquisition view viewpoint

Service oriented view viewpoint

MODAF - detailed guidance

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page rated 1 times

The MODAF All Views Viewpoint

Viewpoint Summary

The All Views (AVs) Viewpoint provides the essential reference information about the architecture, including: an overarching description of the architecture; its scope; ownership; timeframe; and the metadata necessary to define the terminology used in the architecture, and to enable the architecture to be searched and queried. It also provides a place to record any findings arising from the architecting process.

Since the AVs provide critical information for the future access and exploitation of the architecture, it is essential that they are fully populated whenever a MODAF architecture is created or modified.

All View products provide information pertinent to the entire architecture. They present supporting information rather than architectural models.

Views

There are two views that make up the All Views Viewpoint:

AV-1 - Overview & Summary Information

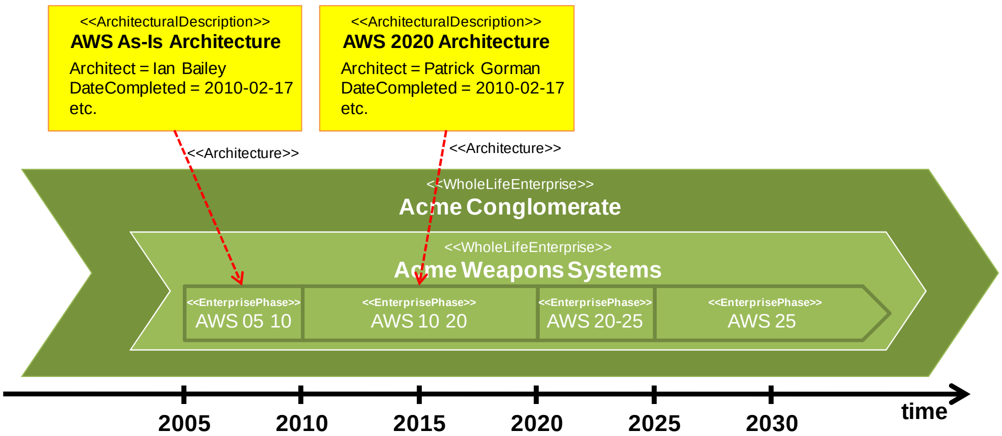
View Summary

The AV-1 provides executive-level summary information about the architecture in a consistent form that allows quick reference and comparison between architectural descriptions. The AV-1 includes assumptions and constraints that may affect high-level decisions relating to an architecture-based work programme. The AV-1 also puts the architecture in context of the Enterprise it describes and the period of time the architecture covers. This context is intended to enable registration and discovery of architectures.

Background

The development of the architecture must be framed in the context of the AV-1 which defines the scope of the enterprise and the phases of that enterprise that the architecture covers.

Enterprises (M3 object = WholeLifeEnterprise) can themselves be decomposed into sub-enterprises and enterprise phases, as in the example below. (This mechanism is an enabler for federated architectures and serves as a front-end for users trying to find a given architecture in a repository). The AV-1, therefore, needs to explicitly define the enterprise phase it is addressing.



Enterprise phases and their architectural descriptions

The AV-1 also serves two additional purposes:

- In the initial stages of architecture development, it serves as a planning guide.
- When the architecture is built, it provides summary information concerning the “who, what, when, why, and how” of the plan, as well as a navigation aid to the views that have been created.

Ultimately, the AV-1 should contain sufficient information to enable an analyst to identify relevant architectures that could be re-used to support other business change activities.

The AV-1 will change as the architecture develops and it is, therefore, important to maintain it throughout the life of the architectural activity that it documents.

Usage

The AV-1 is used for:

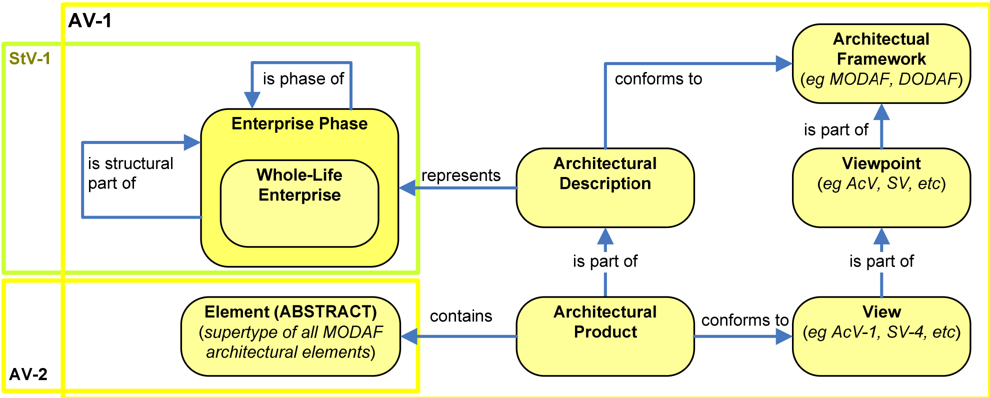
- Scoping the project.
- Providing context to the project.
- Defining an architecture-based task.

- Summarising the findings from an architecture-based task.
- Assisting search within an architecture repository.

Data objects

The data in an AV-1 can include:

- Scope, purpose.
- Whole-Life Enterprise & Enterprise Phase.
- Listing of views used.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Structured Text.
- Enterprise phases may also be shown graphically.

Detailed Product Description

AV-1 is usually a structured text product, and architects may create a template for the AV-1 that can then be used to create a consistent set of information across different architecture-based projects that they are responsible for.

The AV-1 documents the following information about the architecture:

- Architecture Project Identification: the architecture project name, the architect, and the organisation developing the architecture. It also includes assumptions and constraints, identifies the approving authority and the completion date, and records the level of effort required to develop the architecture.
- Scope: the Views and Products that have been developed and the temporal nature of the architecture, such as the time frame covered, whether by specific years or by designations such as current, target, transitional, etc. Scope also identifies the Enterprises and Enterprise Phases that fall within the scope of the architecture.
- Purpose and Perspective[1]: the need for the architecture, what it will demonstrate, the types of analyses that will be applied to it, who is expected to perform the analyses, what decisions

[1] "Perspective" could refer to one or more MODAF viewpoints, the MODAF Community of Interest, a focus for the work (e.g. integration or security), or a combination of these.

are expected to be made on the basis the analysis, who is expected to make those decisions, and what actions are expected to result.

- Context: a description of the setting in which an architecture exists. Context includes such things as mission, doctrine, concepts of operation, threats, environmental conditions, and geographical areas addressed. Context also identifies the rules, criteria, and conventions that are used in the architecture. Any linkages to parallel architecture efforts should be identified.
- Status: a description of the status of the architecture at the time of publication or development of the AV-1 (which might precede the architectural development itself). Status refers to creation, validation and assurance activities.
- Tools and File Formats Used: the tool suite used to develop the architecture and file names and formats for the Architectural Products if appropriate.
- Assumptions and Constraints.
- Date Completed.

An illustrative example is provided below.

AV-1 Overview and Summary Information

- **Architecture Project Identification**
  - Name: **ITT for Service Management System (Network), SMS(N)**
  - Architect: **DCSA DCTO Architecture 4**
  - Organisation Developing the Architecture: **DCSA / DCTO**
  - Assumptions and Constraints: **None**
  - Approval Authority: **Hugh Turbett, Project Manager**
  - Date Completed:
- **Scope: Architecture Views & Products Identification**
  - Views and Products Developed: **AV1, AV2, StV6, OV1 and OV2**
  - Time Frames Addressed: **Present**
  - Organisations Involved: **DCTO and SMS(N) Project Team**

Example AV-1

Normally architecture is used to support analysis; therefore the AV-1 can be extended to include:

- Findings: the findings and recommendations resulting from the architectural effort. These may include capability gaps identified and recommendations on how, for example, processes or systems could be changed to address the gap.
- Costs: the costs that have been incurred in developing the architecture in order to support a cost-benefit analysis of the architectural effort against changes implemented as a result of the architecture.

The AV-1 can be particularly useful as a means of communicating the methods and rationale that have been applied to create the other (MODAF) views in the architecture and the modelling assumptions that have shaped those views. To support this, the AV-1 should list each individual view product and provide a brief commentary against each.

On completion of the architectural activity, a final version of the AV-1 should be produced to summarise the findings for high level decision makers. This version of the AV-1 together with an OV-1, High-Level Operational Concept Graphic, can serve as an executive summary of the architecture.

AV-2 - Integrated Dictionary

The AV-2 catalogues and describes all the Elements used in an architecture, and the Relationships between them. It presents all the Elements as a specialisation hierarchy, provides a text definition for each one, and references the source of each element.

Background

The purpose of the AV-2 is to explain the terms and abbreviations used in building the architecture.

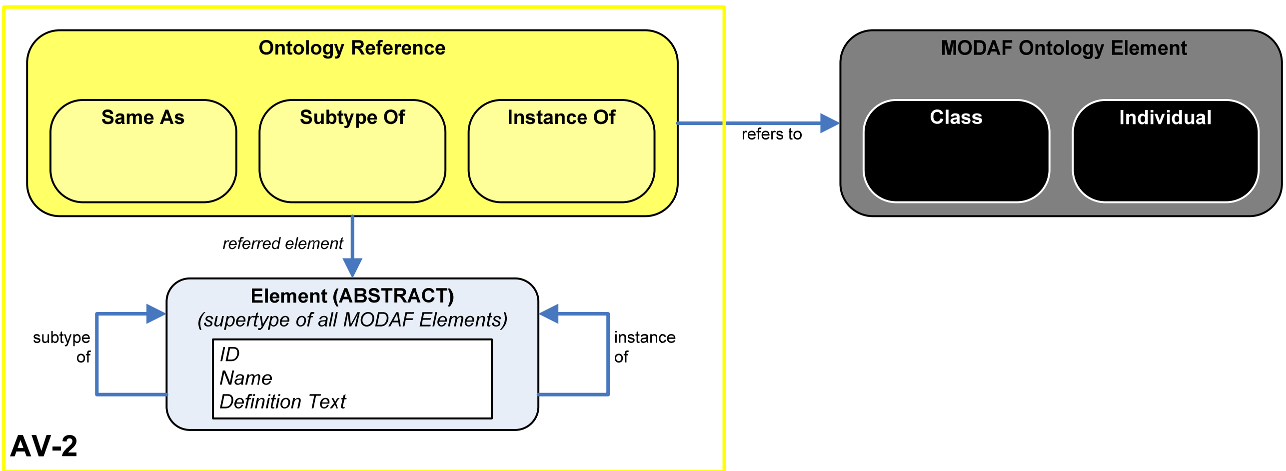
It is, however, essential that organisations within Defence use the same terms to refer to an object. It is MOD policy that architectures developed using MODAF should use terminology that is aligned with the Defence Terminology which is maintained by the Information Coherence Authority for Defence (ICAD)[2].

Where new terms are required, usually when the architecture is covering new technology or business processes, they should be unambiguous and be supported by a description and provenance information.

Data objects

The data in an AV-2 can include:

- References to IDEAS[3] Ontologies.
- Specialisation Relationships (Subtyping).
- Type-Instance Relationships.



Relationships between Key Data Objects (Simplified from M3)

[2] See Joint Services Publication 329, Chapter 5 for further guidance.

[3] Five-nation standards group "International Defence Enterprise Architecture Specification."

Detailed Product Description

Each entry in an Integrated Dictionary should display the following properties:

- The name used for this element in the architecture.
- Alternative names for this element.
- A brief description of the element.
- A list of the views in which the element is used.

An AV-2 is structured using two types of hierarchical relationship between elements: sub-supertype and type-instance. A sub-supertype relationship is a relationship between two classes with the second being a pure specialisation of the first. A type-instance relationship is a relationship between a class and an instance that is a member (instance) of that class. Note that classes may be members of other classes (eg the class, “Colonel” is a member of the class, “Rank”).

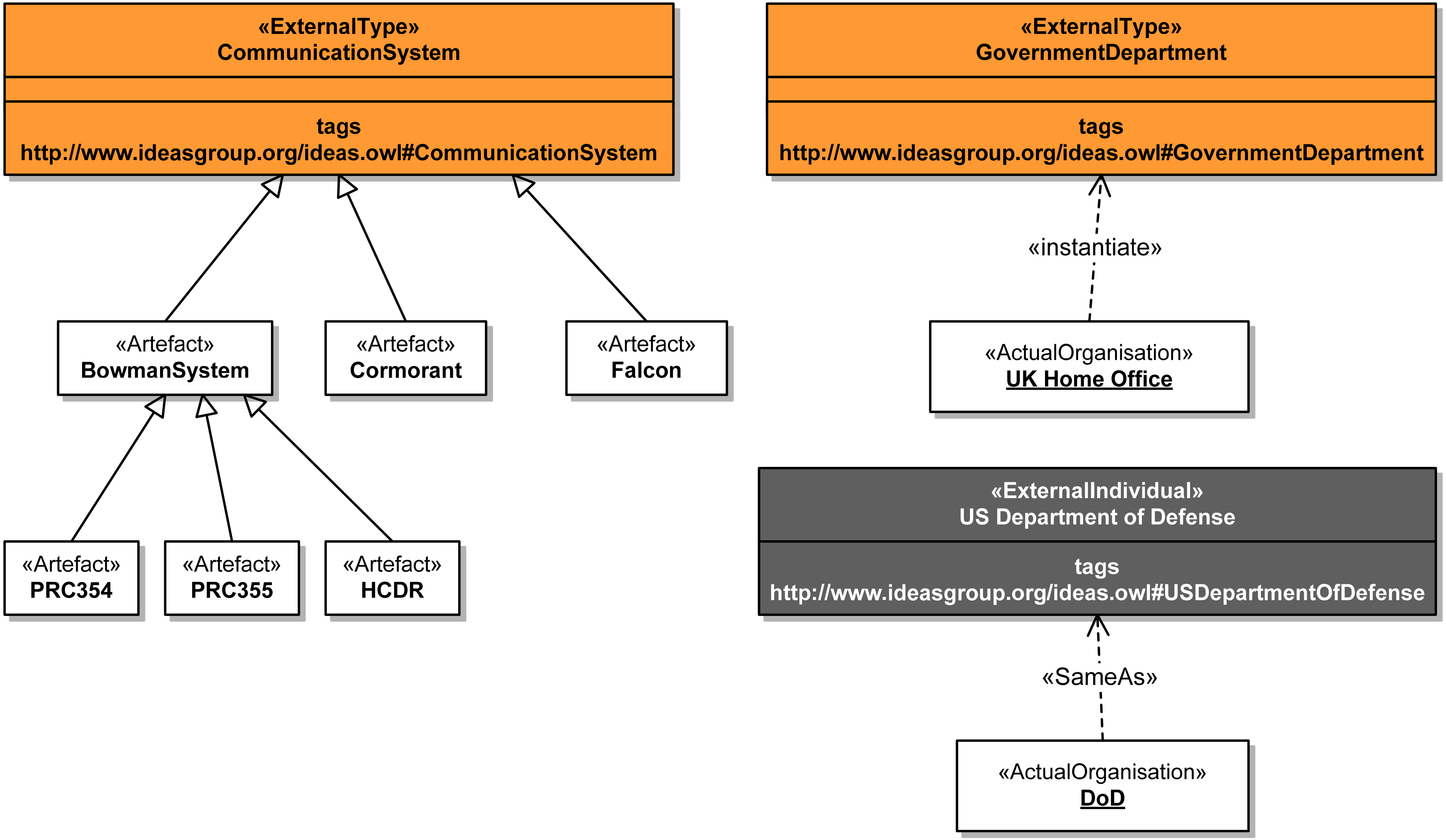
Care should be taken when using these relationships to structure the AV-2 dictionary.

The figure below and the one on the following page show actual examples.

Category	Unique ID	Local Name	Description	Sul
Class	http://www.ideasgroup.org/ideas.owl #CommunicationSystem	Comms System	A system which enables spatially separated parties to communicate	
Class	http://www.ideasgroup.org/ideas.owl #Government Department	Govt Dept	An organization that is an executive body of a national government	
Individual	http://www.ideasgroup.org/ideas.owl #USDepartmentOfDefense	DoD	The federal department responsible for defence of the United States of America	
Class	BowmanSystem	Bowman System	A communications system that is part of the UK Bowman family of land radio	

Example of a Tabular AV-2

UNCONTROLLED COPY WHEN PRINTED



Example of a Graphical AV-2

Strategic view (StV) viewpoint

A description of and guidance for the use of the MOD Architecture Framework (MODAF)Â Strategic View Viewpoint.

Strategic Views (StVs) support the process of analysing and optimising the delivery of military capability in line with the MOD's strategic intent. The StVs achieve this by capturing the capability policy/concepts, decomposing this into a capability taxonomy supported by appropriate measures of effectiveness that can be used for capability audit and gap/overlap analysis.

- **StV viewpoint PDF [600.9 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

- **ViewpointsÂ andÂ views**
  Parent page
- **Views summary documents**
- **All views viewpoint**
- **Operational view viewpoint**
- **System view viewpoint**
- **Technical standard views viewpoint**
- **Acquisition view viewpoint**
- **Service oriented view viewpoint**

MODAF - detailed guidance

Related links

- **International Defence Enterprise Architecture Specification (IDEAS) Group**
- **NATO Architcture Framework (NAF)**
- **Department of Defence Architecture Framework**

Access online conversion tools for Adobe PDF documents

Page not yet rated

The MODAF Strategic Viewpoint

Viewpoint Summary

The Strategic Viewpoint defines the desired business outcome and the capabilities that are required to achieve it; i.e. it provides a means to align an enterprise's strategy with the capabilities required to deliver that strategy, identifying any capability gaps that may exist.

It provides a set of Strategic Views (StVs) which capture the enterprise vision, goals, policies and concepts related to the capability requirements. It enables decomposition of capabilities into a capability taxonomy which, supported by appropriate measures of effectiveness, can be used for capability audit and gap and overlap analysis.

The StVs further detail the dependencies between capabilities, thus enabling capability options to be built to support effective trade-off.

Views

There are 6 StVs that make up the Strategic Viewpoint:

StV-1 - Enterprise Vision

The StV-1 provides the high-level scope for the architecture and a strategic context for the capabilities the architecture contains. In essence it describes the end-state for any business transformation activity.

Background

The purpose of an StV-1 is to provide a strategic context for the capabilities described in the architecture. It also provides a high-level scope for the architecture which is more general than the scenario-based scope defined in an OV-1, High Level Operational Context Graphic.

The Views are high-level and describe the vision, goals, enduring tasks and capabilities using terminology that is easily understood by non-technical readers, which may include the use of terminology and acronyms routinely used by the business, (which will need to be clearly defined in the AV-2, Integrated Dictionary).

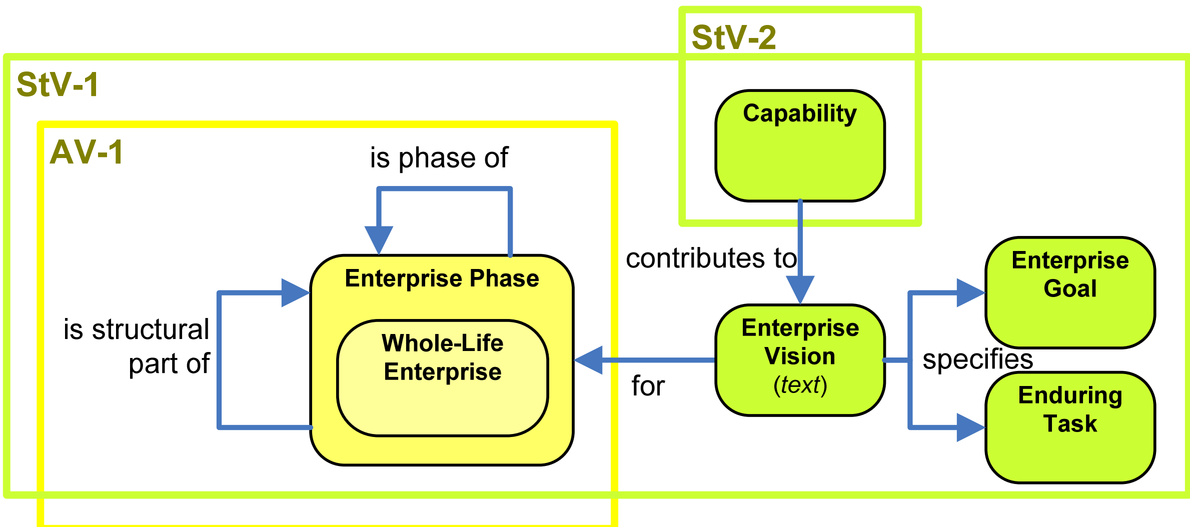
Usage

- Capture and communication of the strategic vision related to capability evolution.
- Identify the capabilities required to meet the vision and goals.
- Identify the required timescales for the capabilities (cf StV-3, Capability Phasing, which provides a summary of when projects are estimated to deliver capability).
- Identify any enduring tasks the enterprise performs.

Data objects

The data in an StV-1 can include:

- Enterprise Vision.
- Enterprise Phase.
- Enterprise Goals.
- Capability.
- Enduring Task.



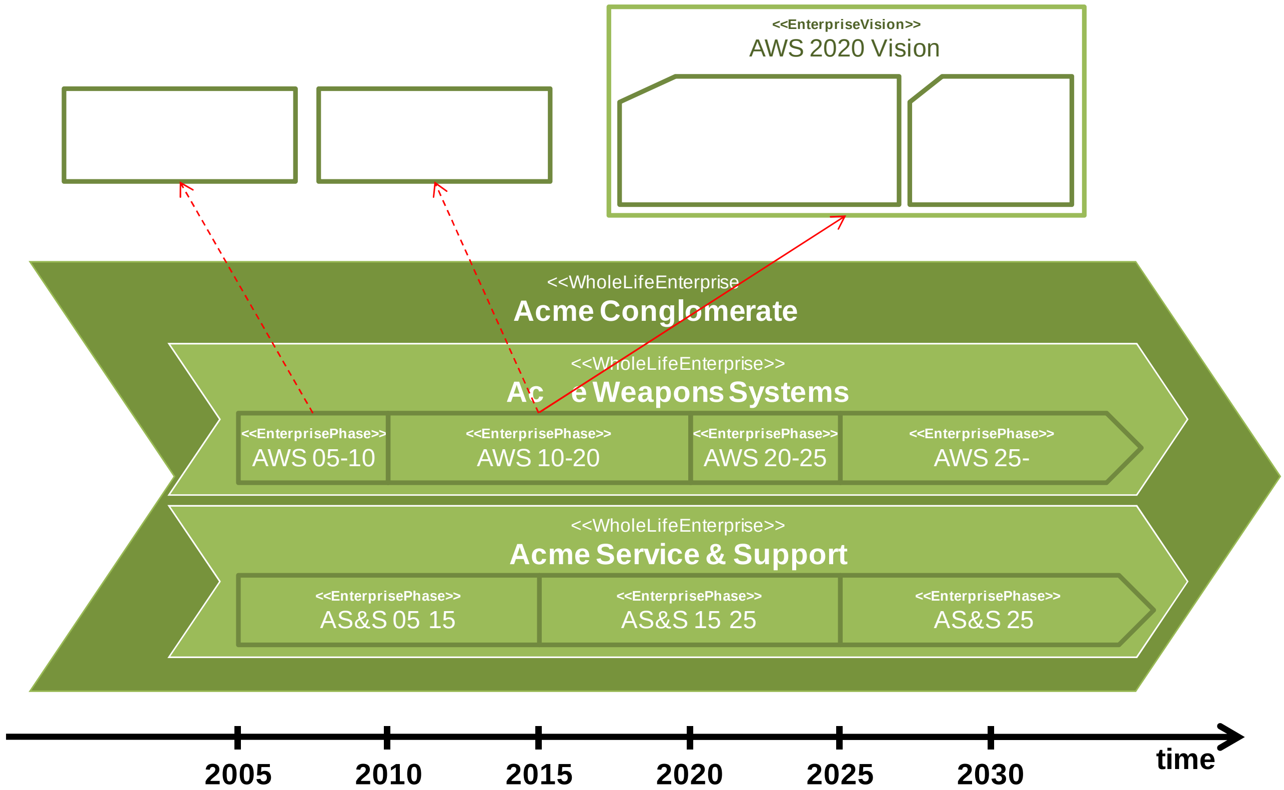
Relationships between Key Data Objects (Simplified from M3)

Representation

- Structured Text.
- UML Composite Structure Diagram.
- SysML Structural Diagrams.

Detailed Product Description

The StV-1 defines the strategic context for a group of capabilities described in the architecture by outlining the vision for an enterprise over a bounded period of time. It describes the high level goals and strategy for the enterprise, and the level of capability the enterprise is expected to achieve over time.



StV-1 Enterprise Structure & Goals Example

An StV-1 can provide the blueprint for a transformational initiative, by showing the expected capabilities that phases of an enterprise will exhibit.

StV-1s may also be textual descriptions of the overarching objectives of the transformation or change programme that the Enterprise is engaged in. Of key importance is the identification of Enterprise Goals, together with the desired outcomes and measurable benefits associated with these.

The StV-1 only shows the capabilities exhibited by enterprises; it does not show how the enterprise is structured in order to deliver those capabilities. OV-2, Operational Node Relationship Description, is used to define the logical structure of the enterprise, and the individual logical nodes in the enterprise that deliver the capability. SV-1, Resource Interaction Specification, and OV-4, Organisational Relationships Chart, are used to show the physical and organisational structure of the enterprise.

UNCLASSIFIED

THE UK JOINT HIGH LEVEL OPERATIONAL CONCEPT

CAPPING PAPER

101. Fighting power comprises conceptual, moral and physical components. The conceptual component of joint fighting power was articulated in UK Joint Vision, where the importance of the enduring nature of the Principles of War was endorsed. The Vision provided broad guidance for future capabilities in the form of a joint High Level Operational Concept (HLOC), an effects based framework for operations and a description of capability as seven discrete but closely interlocking components. However, UK Joint Vision did not develop the conceptual components in detail. Using the Defence Capability Framework, this Analytical Concept describes the components of capability in sufficient detail to inform Joint Operational Concept Committee stakeholders, particularly the single Services, who are developing their own high level operational concepts in parallel. The three components of capability, Command, Inform and Operate, form the capability backbone of the HLOC around which considerations for the remaining four components — Prepare, Project, Protect and Sustain —have been woven to form the complete concept. The concept addresses the 2020 timeframe, assessed as the best compromise between the need to break free from the dominance of current systems without venturing into the purely speculative. It has also been harmonised with US joint concepts, noting the clear guidance from COS that we must be able to operate *with* but not necessarily *as* our close allies.

OPERATE CORE CONCEPT

An agile task-oriented joint force with freedom of action to synchronise effects throughout the Battlespace and with maximum potential to exploit fleeting opportunities.

StV-1 Text-Based Example

StV-2 - Capability Taxonomy

The StV-2 presents capabilities and the hierarchical relationships between them.

Background

The StV-2 presents a hierarchy of capabilities. StV-2 specifies all the capabilities that are referenced throughout one or more architectures – i.e. one StV-2 may provide the definitive list of capabilities for a number of logical and physical architectures. In addition it can be used as a source document for the development of high level use cases and Key User Requirements (KURs).

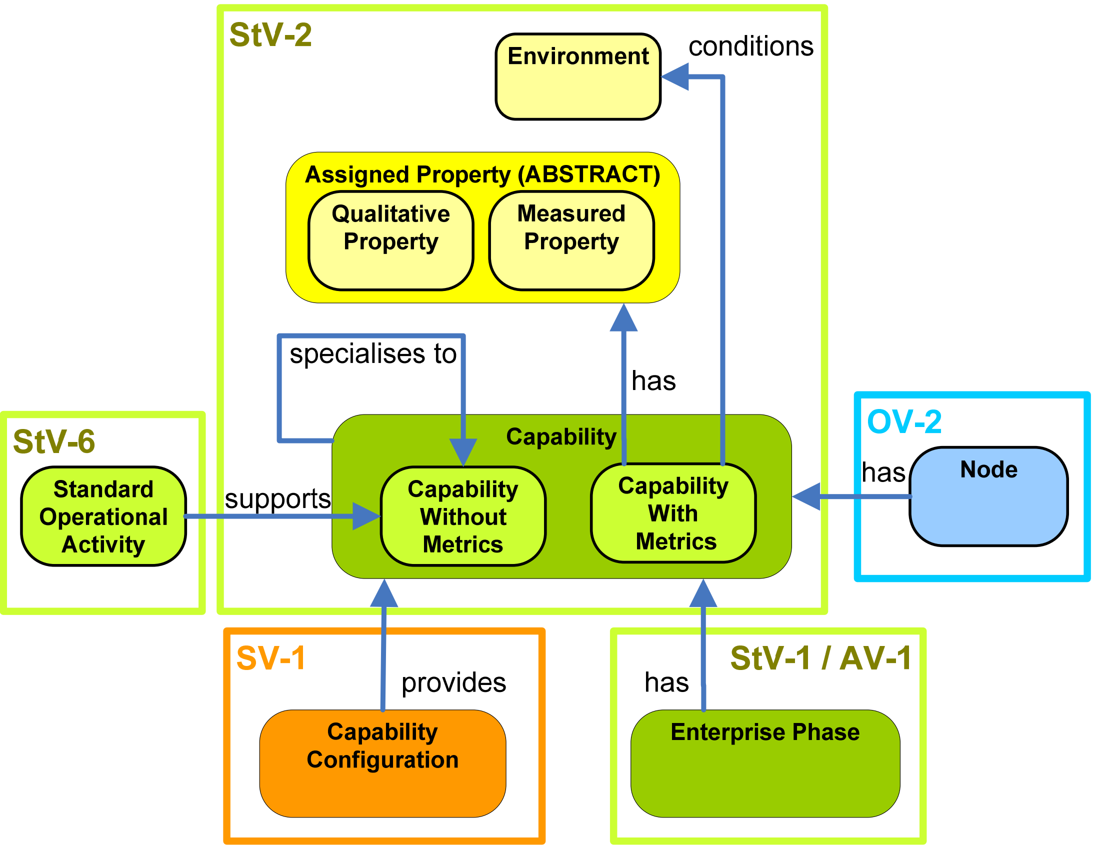
Usage

- Identification of existing and required capabilities.
- Codifying required capability elements.
- Source for the derivation of cohesive sets of KURs.
- Providing reference capabilities for multiple architectures.

Data objects

The data in an StV-2 can include:

- Capability.
- Capability Specialisation (super-subtype relationship between capabilities).
- Capability Composition (whole-part relationship between capabilities).



Relationships between Key Data Objects (Simplified from M3)

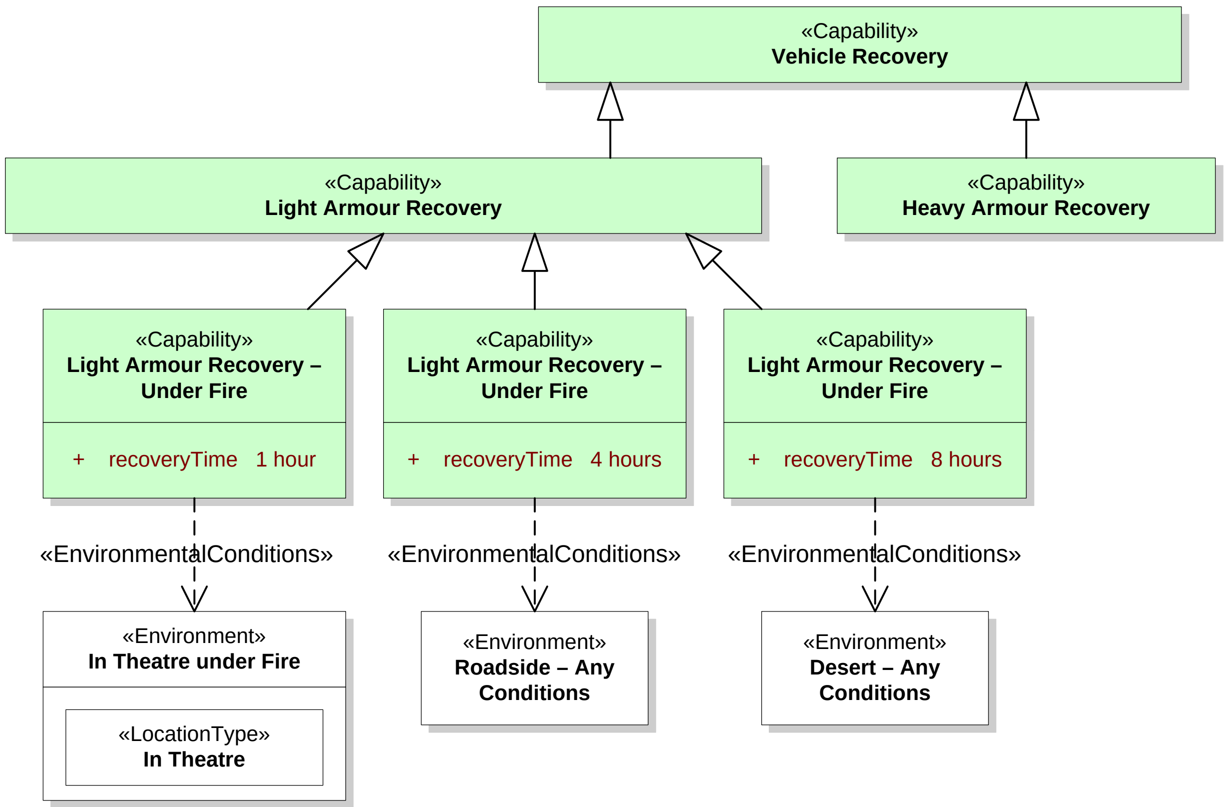
Representation:

- Tabulation.
- Hierarchical (Connected Shapes).
- UML Class Diagram (with generalization and aggregation relationships).

Detailed Product Description

The StV-2 specifies a hierarchy of capabilities. A capability taxonomy persists over time (an architect may wish to specify historical, current or future capability) and may be referenced by multiple architectures. The capabilities specified in an StV-2 are extensively re-used in operational, service-oriented, system and acquisition views; in this way the concept of capability is integral to any MODAF architecture.

In MODAF, a capability is a description of an ability to do something. In StV-2, the capabilities are only described in the abstract – i.e. StV-2 does not specify how a capability is to be implemented. An StV-2 is most commonly structured as a specialisation hierarchy of capabilities, with the most general at the root and most specific at the leaves. At the leaf level, capabilities may have a metric specified, along with an environmental qualifier for the metric:



StV-2 Example with Leaf Metrics

Note that capabilities with a metric specified may not be further specialised. When capabilities are referenced in operational or systems architectures, it may be that a particular Node (from OV-2, Operational Node Relationship Description) or Capability Configuration (from SV-1, Resource Interaction Specification) meets more than one level of capability.

StV-2 is a capability taxonomy view. The view that describes dependencies between the capabilities shown in the StV-2 taxonomy is the StV-4, Capability Dependencies.

The capabilities in an StV-2 are structure using two types of relationships; specialisation and composition. A capability specialisation relationship asserts that one capability is a special case of another (e.g. *ground-based ISTAR* is a specialisation of *ISTAR*). A capability composition relationship asserts that one capability is a necessary component of another (e.g. *intelligence* is a component of *intelligence-led-policing*).

In MODAF, Capabilities are 'enduring', that is they are intended to provide an enduring framework across the lifetime of one or more enterprises. This means that it is feasible to develop a capability taxonomy that will apply to all Enterprise Phases listed in an StV-1, Enterprise Vision, and an StV-2 may even cover more than one architecture.

<u>Command Battlespace Management</u>

A. Decision Support

1. Operational Planning
2. Operational Analysis
3. Mission Rehearsal
4. Situational Awareness
5. Intelligence

B. Information Management and Acquisition

1. Information Management
   a. Analysis
   b. Fusion
   c. Quality Assurance
   d. Dissemination
2. STAR

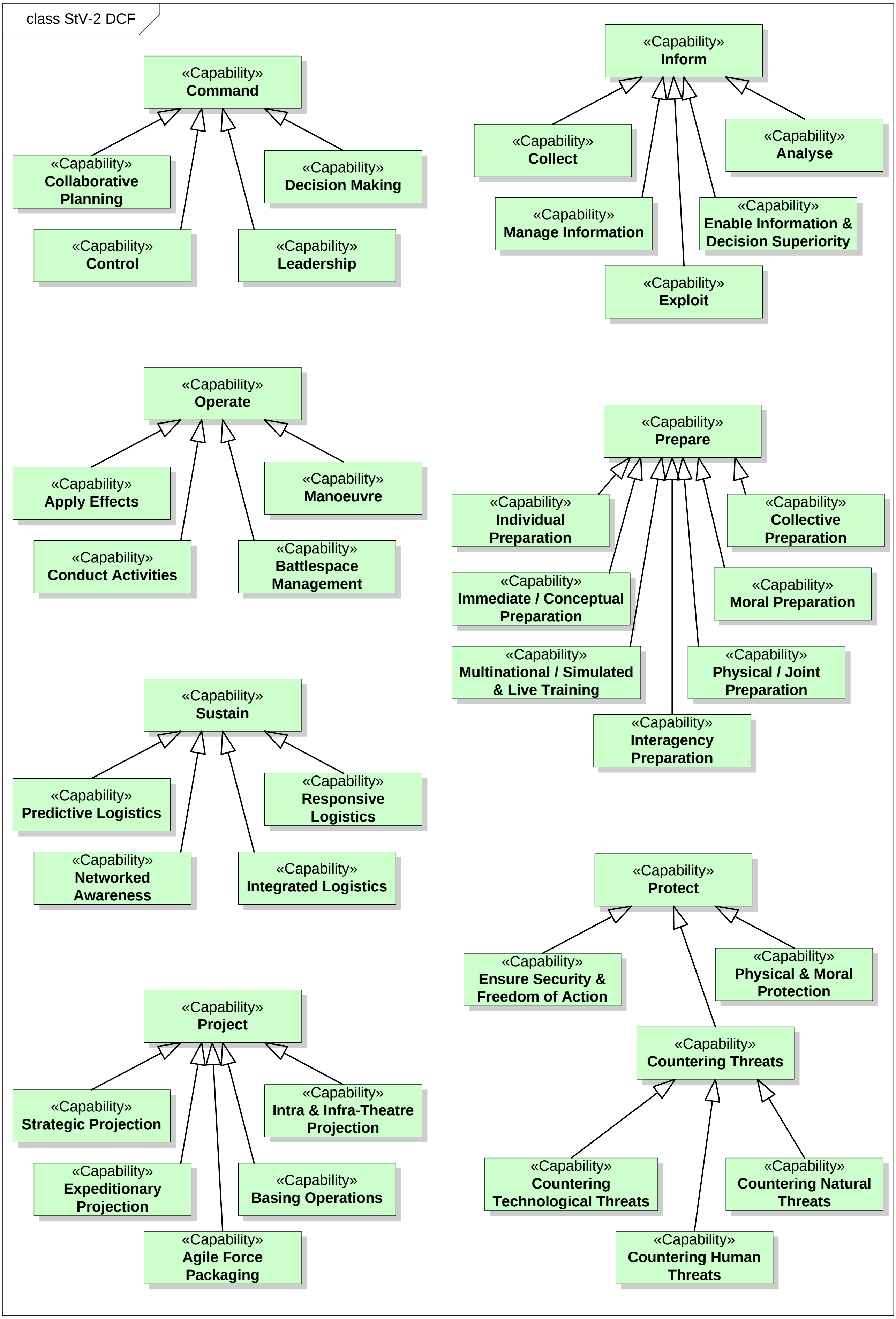
C. Effects

1. Targeting
2. Plan Engagement
   a. Effects Selection
   b. Resource Allocation
   c. Synchronisation
3. Conduct Engagement

Example StV-2

The StV-2 View has no mandated structure although the format selected must be able to support the representation of a structured/hierarchal list and clearly differentiate between specialisation and composition relationships. This structure may be delivered using textual, tabular or graphical methods. The associated attributes and metrics for each capability can either be included on the main StV-2 or in tabular format as an appendix if the inclusion of the attributes and metrics would over complicate the presentation of the view.

It should be noted that UML can be used to develop capability taxonomies; its object-oriented approach naturally includes the concept of specialisation (generalization in UML) and composition (aggregate relationships).


class StV-2 DCF
«Capability» Command
«Capability» Collaborative Planning
«Capability» Decision Making
«Capability» Control
«Capability» Leadership
«Capability» Inform
«Capability» Collect
«Capability» Analyse
«Capability» Manage Information
«Capability» Enable Information & Decision Superiority
«Capability» Exploit
«Capability» Operate
«Capability» Apply Effects
«Capability» Manoeuvre
«Capability» Conduct Activities
«Capability» Battlespace Management
«Capability» Prepare
«Capability» Individual Preparation
«Capability» Collective Preparation
«Capability» Immediate / Conceptual Preparation
«Capability» Moral Preparation
«Capability» Multinational / Simulated & Live Training
«Capability» Physical / Joint Preparation
«Capability» Interagency Preparation
«Capability» Sustain
«Capability» Predictive Logistics
«Capability» Responsive Logistics
«Capability» Networked Awareness
«Capability» Integrated Logistics
«Capability» Protect
«Capability» Ensure Security & Freedom of Action
«Capability» Physical & Moral Protection
«Capability» Countering Threats
«Capability» Countering Technological Threats
«Capability» Countering Natural Threats
«Capability» Countering Human Threats
«Capability» Project
«Capability» Strategic Projection
«Capability» Intra & Infra-Theatre Projection
«Capability» Expeditionary Projection
«Capability» Basing Operations
«Capability» Agile Force Packaging

StV-3 - Capability Phasing

The StV-3 provides a representation of the actual or estimated availability of capabilities over a period of time (derived from capability delivery milestones in acquisition projects).

Background

StV-3 Views support the Capability Audit process and similar processes used across different communities of interest by providing a method to identify gaps or duplication in capability provision.

The view indicates capability increments, which are derived from delivery milestones within acquisition projects.

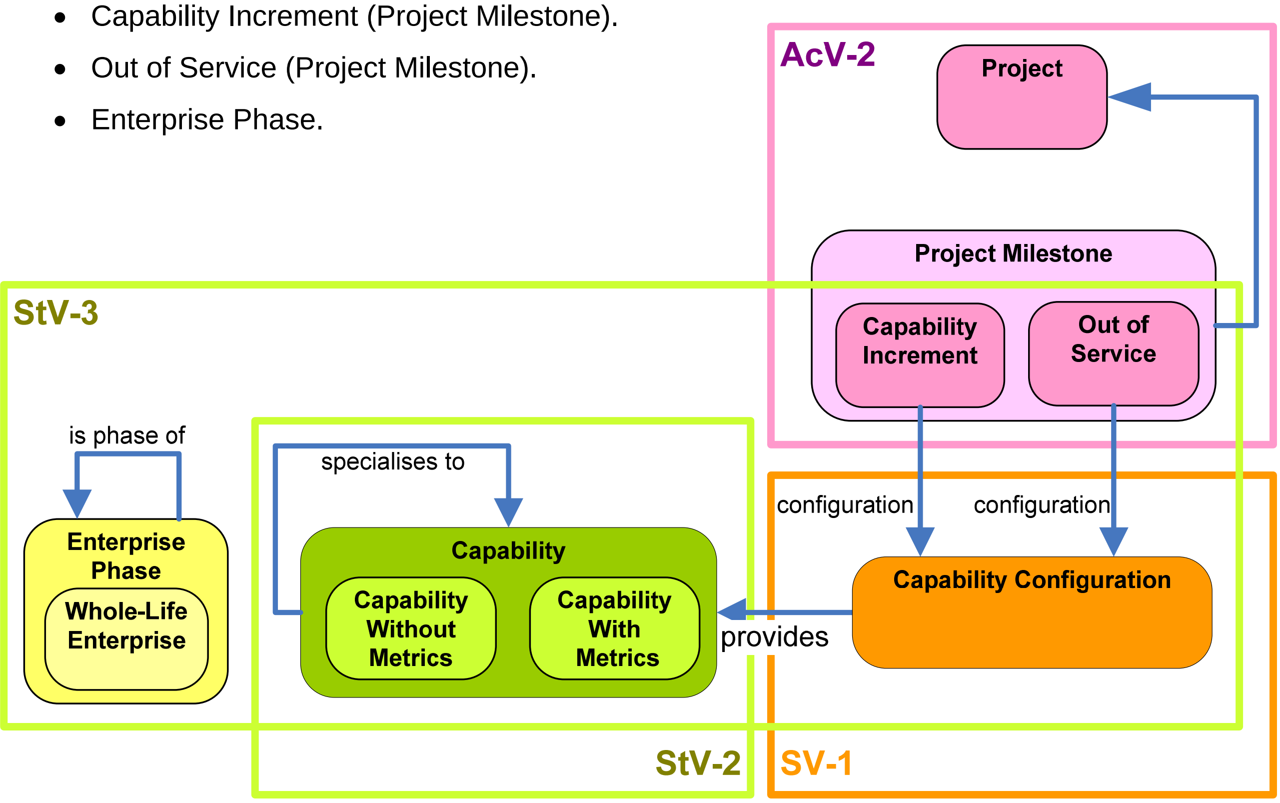
Usage

- Capability planning (capability phasing).
- Capability integration planning.
- Capability gap analysis.
- High-level dashboard for acquisition management.

Data objects

The data in an StV-3 can include:

- Capability.
- Capability Configuration.
- Capability Increment (Project Milestone).
- Out of Service (Project Milestone).
- Enterprise Phase.



Relationships between Key Data Objects (Simplified from M3)

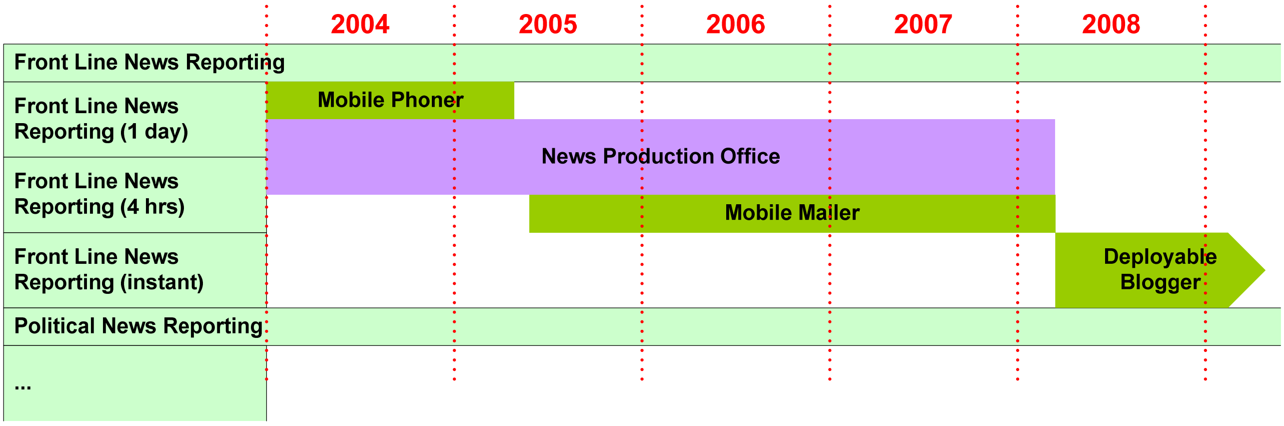
Representation

A time based chart in the style of a Gantt chart.

Detailed Product Description

The StV-3 provides a representation of the available capability levels at different points in time or during specific periods of time (associated with the Enterprise Phases – see StV-1, Enterprise Vision).

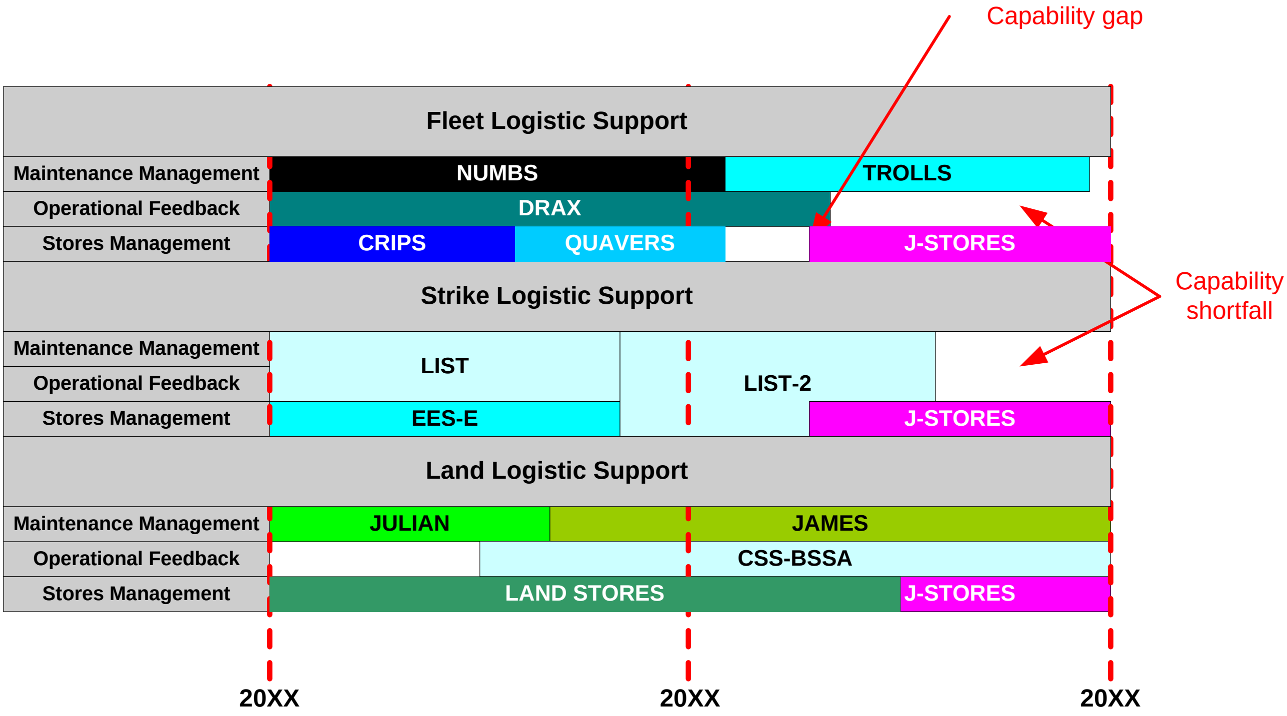
The StV-3 is presented as a timing chart showing capabilities on the vertical axis and time on the horizontal axis. Active capability configurations are shown as bars against the capabilities they provide, with the start and end of the bars corresponding to the capability configuration coming into and going out of service. Where nothing meets a particular capability at a particular time, whitespace is shown so as to highlight capability gaps.



Example StV-3

The view is created by analysing project data to determine when projects providing elements of military capability are to be delivered, upgraded and/or withdrawn (this data may be provided in part by a AcV-2, Programme Timelines, and the mappings from capability to resources that is provided in SV-1, Resource Interaction Specification, and SV-8, Capability Configuration Management).

The output from this iterative approach is a chart that represents the required capability phasing. Normally, an StV-3 will not include the project information, however, there is a variant view in which the table is overlaid with the names of the projects that will deliver the capability increments. An example of this can be found on the following page.



Example StV-3

The essence of this variant view is the relationship between projects, capabilities and time. The view may be used to determine the need for interventions in projects (to fulfil a capability gap) or to represent current plans (the availability of capability according to their delivery timescales).

An StV-3 can be used to assist in the identification of capability gaps/shortfalls (i.e. no fielded capability to fulfil a particular capability function) or capability duplication/overlap (i.e. multiple fielded capabilities for a single capability function).

Note that StV-1 specifies the requirement for capability at each Enterprise Phase, whereas StV-3 reflects the output of capability programmes (e.g. acquisition, training, etc.) over time. The requirement specified in StV-1 may or may not be met by those programmes. StV-3 is strictly-speaking an Acquisition View (AcV) as it is presenting programmatic information rather than strategic intent.

StV-4 - Capability Dependencies

The StV-4 describes the dependencies between capabilities.

Background

The StV-4 is intended to provide a means of analysing the dependencies between capabilities, including those within capability compositions (sometimes called "capability clusters"), in order to guide capability management.

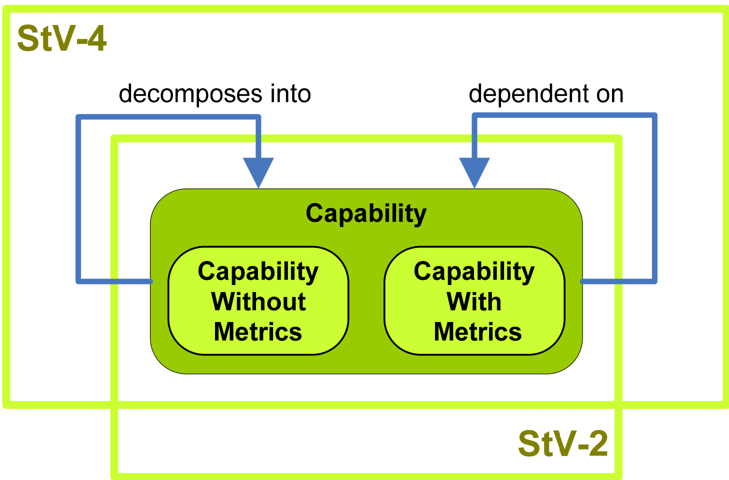
Usage

- Identification of capability dependencies.
- Capability management (impact analysis for options, disposal etc).

Data objects

The data in an StV-4 can include:

- Capability.
- Capability Dependency (relationship).
- Capability Composition (relationship).



Relationships between Key Data Objects (Simplified from M3)

Representation

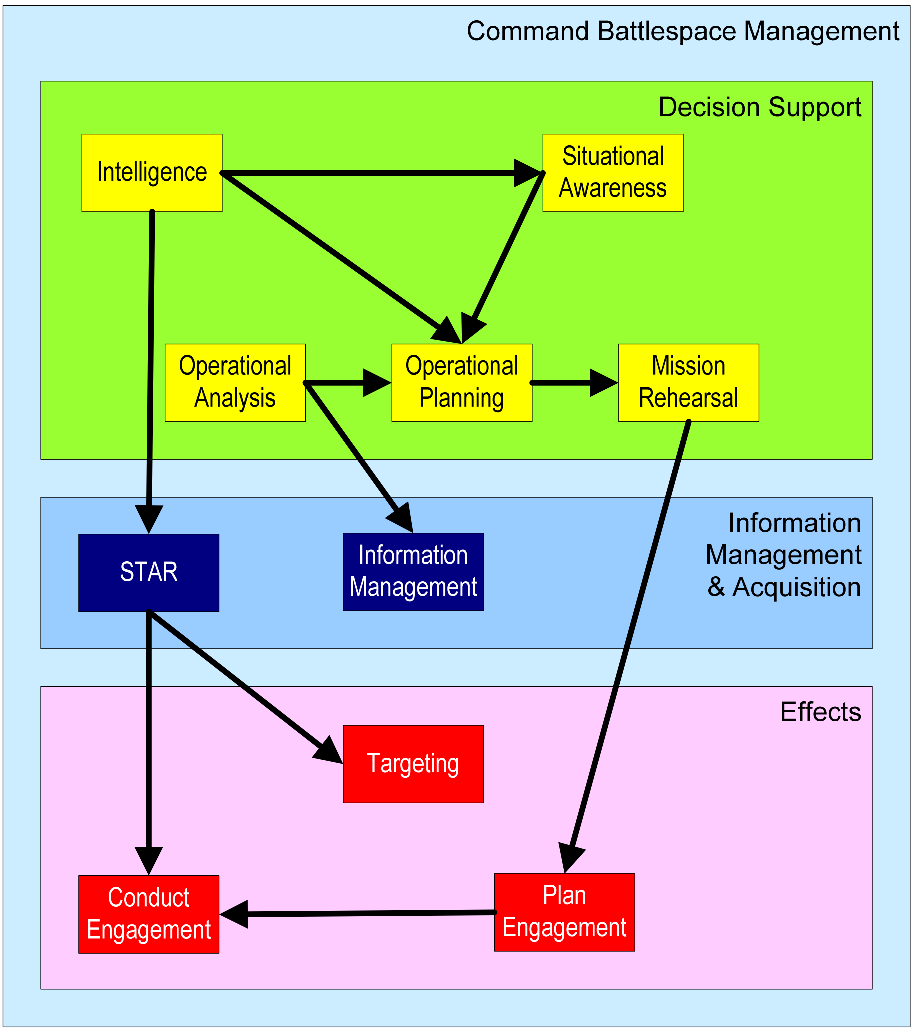
- 'Nested box' Diagram.
- UML Class Diagram.
- UML Composite Structure Diagram.
- SysML Structural Diagrams.

Detailed Product Description

The StV-4 describes the relationships between capabilities. This contrasts with StV-2, Capability Taxonomy, which also deals with relationships between capabilities, but StV-2 only addresses specialisation and composition relationships.

The StV-4 is intended to provide a means of analysing the dependencies between capabilities and between capability clusters. In particular, it will highlight potential integration requirements and the interactions needed between acquisition projects in order to achieve the overall capability.

The recommended notation for StV-4 is a functional dependency diagram which shows how functions are clustered together and the relationships between the individual functions or clusters of functions. It may also be useful to supplement the functional dependency diagram with a functional n-squared diagram. Examples of the StV-4 follow.



Example StV-4 (Graphical Format)

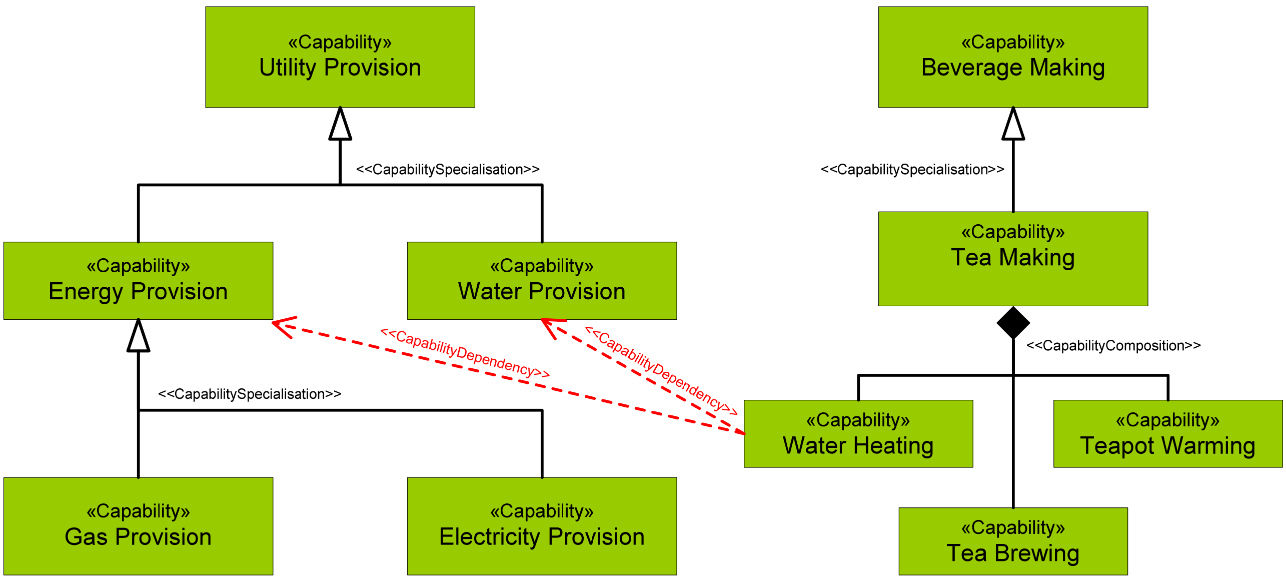
	A	B	C	D	E	F	G	H	I	J
A										
B										
C	X	X		X						
D	X									
E			X							
F	X									
G		X								
H						X				
I					X					
J						X			X	

A = Intelligence
B = Operational Analysis
C = Operational Planning
D = Operational Awareness
E = Mission Rehearsal
F = STAR
G = Information Management
H = Targeting
I = Plan Engagement
J = Conduct Engagement

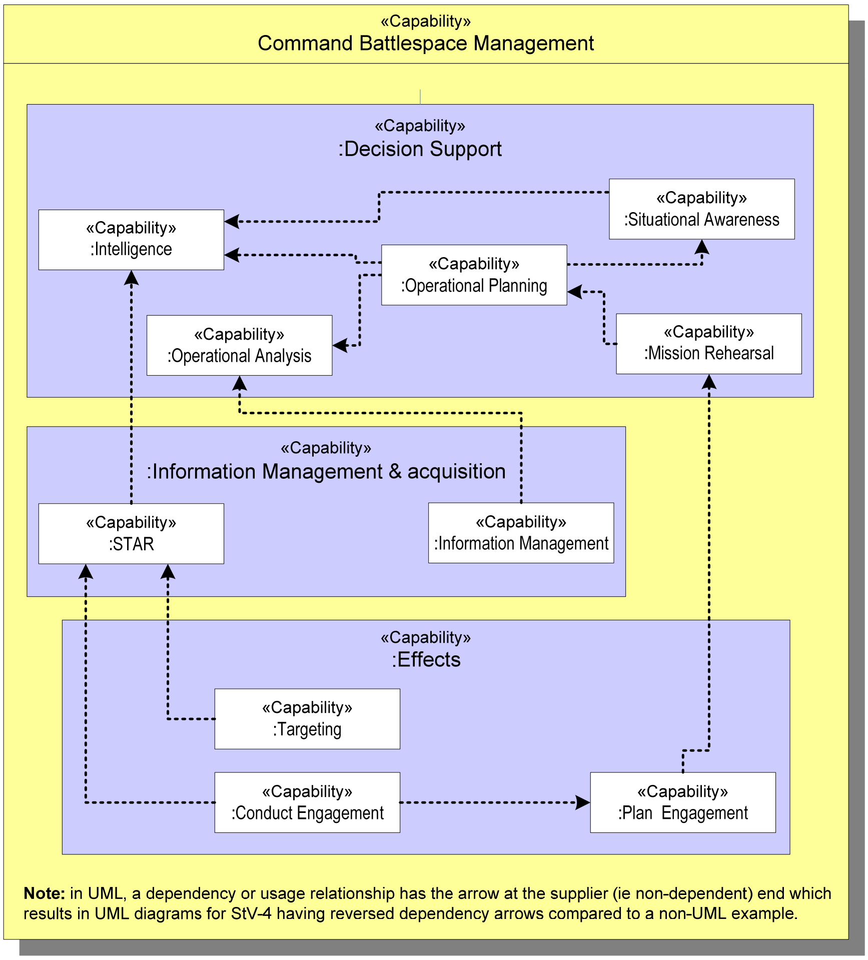
Example StV-4 (N-squared Format)

In some cases it may be important to distinguish between different types of dependency in the StV-4. Graphically this can be achieved by colour-coding the connecting lines, or by using dashed lines.

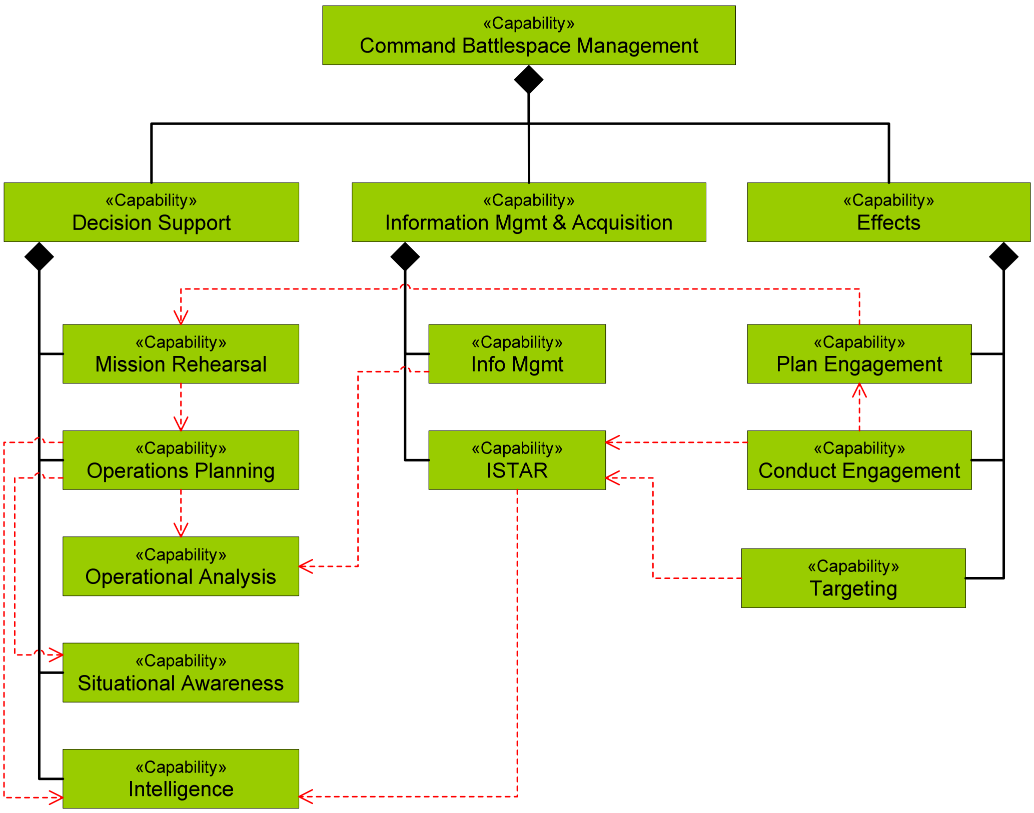
UML can be used to define capability dependency view products as illustrated below.



Example StV-4 (UML Aggregation Format – allows simpler combination of composition and specialisation)



Example StV-4 (UML Composite Structure Form)



Example StV-4 (UML Aggregation Form)

StV-5 - Capability to Organisation Deployment Mapping

The StV-5 shows the deployment of capability to specific organisations over time.

Background

This view shows the planned capability deployment related to the organisations fielding it for a particular period of time (Enterprise Phase). Multiple StV-5s are used to show how the deployment of new capability propagates through organisations over time.

The StV-5 should be seen as a summary of the delivery schedules for capabilities and can, therefore be used to support the capability management process and, in particular, assist the planning of fielding.

To prevent constraining the solution space, StV-5 should not be produced at the time of developing capability / user requirements. Like StV-3, Capability Phasing, it is more of an informative programmatic view – i.e. a "dashboard".

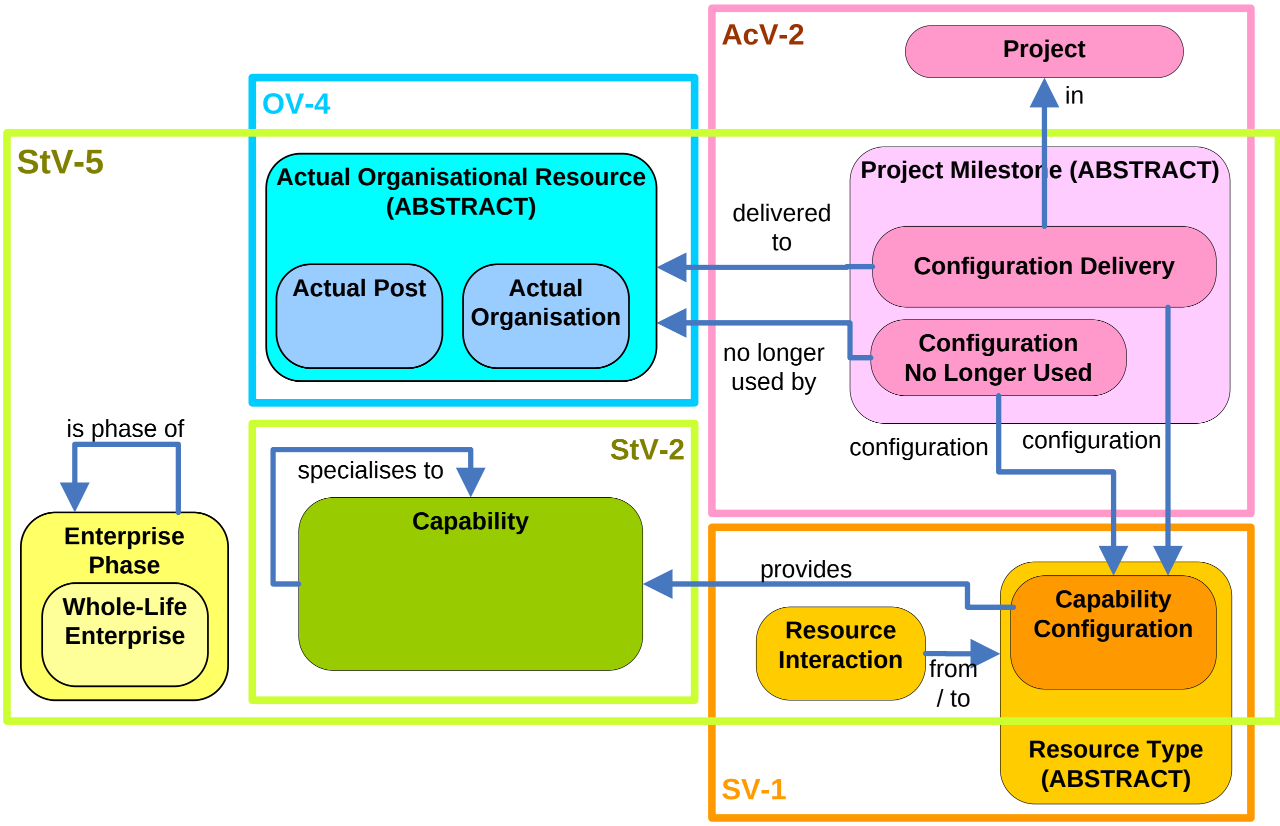
Usage

- Fielding planning.
- FE@R and Operations planning.
- Capability integration planning.
- Capability options analysis.
- Capability redundancy/overlap/gap analysis.
- Identification of deployment level shortfalls.

Data objects

The data in an StV-5 can include:

- Capability.
- Capability Configuration.
- Resource Interaction (between Capability Configurations or their components).
- Actual Organisational Resource (Actual Post, Actual Organisation).
- Capability Delivery (Project Milestone).
- Capability No Longer Used (Project Milestone).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabulation.
- Structured Timeline View.

Detailed Product Description:

The StV-5 shows deployment of capability configurations to specific organisations during a specific Enterprise Phase. The information used to create the StV-5 is drawn from other MODAF views:

- AcV-2, Programme Timelines;
- StV-2, Capability Taxonomy;
- OV-4, Organisational Relationships Chart;
- SV-1, Resource Interaction Specification.

The timing is based on project milestones from AcV-2 which indicate when a capability configuration will be delivered to an organisation and, in addition, the point at which the organisation stops using a particular capability configuration.

In order to conduct a comprehensive analysis, multiple StV-5s can be created to represent the different Enterprise Phases. In addition, the StV-5 can be compared with the StV-3, Capability Phasing, and the SV-8, Capability Configuration Management, to provide a better understanding of the temporal aspects of the architecture.

The StV-5 may also show interactions between capability configurations, where these have been previously defined in an SV-1.

The StV-5 is usually presented in a tabular form with the organisational structure represented along one axis, with the capabilities along the other axis. Graphical objects representing capability configurations are placed in the relevant positions relative to these axes.

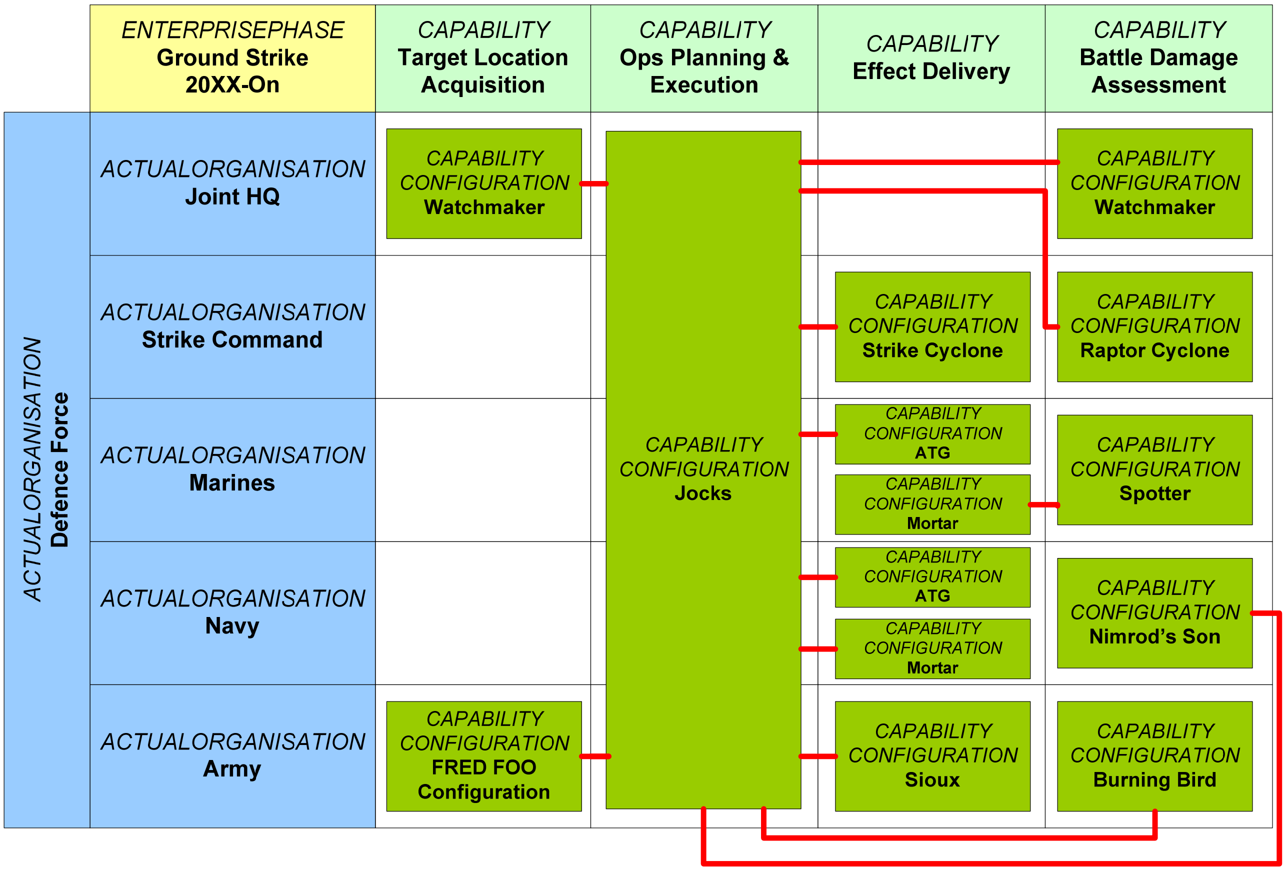
ENTERPRISEPHASE **MOD News & PR** **2005-2007**	*CAPABILITY* **Front Line News Reporting**		
	CAPABILITY **Front Line News Reporting (1 day)**	*CAPABILITY* **Front Line News Reporting (4 days)**	*CAPABILITY* **Front Line News Reporting (instant)**
ACTUALORGANISATION **xx Air Assault Brigade**		*CAPABILITY CONFIGURATION* **Mobile Mailer**	
ACTUALORGANISATION **yy Armoured Brigade**	*CAPABILITY CONFIGURATION* **Mobile Phoner**		
ACTUALORGANISATION **zz Armoured Brigade**	*CAPABILITY CONFIGURATION* **Mobile Phoner**		
ACTUALORGANISATION **xy Mechanised Brigade**	*CAPABILITY CONFIGURATION* **Mobile Phoner**		
ACTUALORGANISATION **xz Light Brigade**		*CAPABILITY CONFIGURATION* **Mobile Mailer**	

ENTERPRISEPHASE **MOD News & PR** **2007-2009**	*CAPABILITY* **Front Line News Reporting**		
	CAPABILITY **Front Line News Reporting (1 day)**	*CAPABILITY* **Front Line News Reporting (4 days)**	*CAPABILITY* **Front Line News Reporting (instant)**
ACTUALORGANISATION **xx Air Assault Brigade**			*CAPABILITY CONFIGURATION* **Deployable Blogger**
ACTUALORGANISATION **yy Armoured Brigade**		*CAPABILITY CONFIGURATION* **Mobile Mailer**	
ACTUALORGANISATION **zz Armoured Brigade**		*CAPABILITY CONFIGURATION* **Mobile Mailer**	
ACTUALORGANISATION **xy Mechanised Brigade**		*CAPABILITY CONFIGURATION* **Mobile Mailer**	
ACTUALORGANISATION **xz Light Brigade**			*CAPABILITY CONFIGURATION* **Deployable Blogger**

ENTERPRISEPHASE **MOD News & PR** **2009-On**	*CAPABILITY* **Front Line News Reporting**		
	CAPABILITY **Front Line News Reporting (1 day)**	*CAPABILITY* **Front Line News Reporting (4 days)**	*CAPABILITY* **Front Line News Reporting (instant)**
ACTUALORGANISATION **xx Air Assault Brigade**			*CAPABILITY CONFIGURATION* **Deployable Blogger**
ACTUALORGANISATION **yy Armoured Brigade**			*CAPABILITY CONFIGURATION* **Deployable Blogger**
ACTUALORGANISATION **zz Armoured Brigade**			*CAPABILITY CONFIGURATION* **Deployable Blogger**
ACTUALORGANISATION **xy Mechanised Brigade**			*CAPABILITY CONFIGURATION* **Deployable Blogger**
ACTUALORGANISATION **xz Light Brigade**			*CAPABILITY CONFIGURATION* **Deployable Blogger**

Three Sequential StV-5s for Different Enterprise Phases, Showing Delivery of News Reporting Capability

Resource Interactions (from an SV-1) can also be shown on an StV-5. In addition, where a Capability Configuration is deployed across a number of Organisations, a parent Organisation should be created for context purposes, and the Capability Configuration stretched across the domain of the parent Organisation. In the example below, the Resource Interactions are the red lines, and “Defence Force” is the parent organisation that provides the domain for “Jocks”:



StV-5 Example Showing Capability Deployed Across Organisations, and Interactions between Capability Configurations

StV-6 – Standard Operational Activities to Capability Mapping

The StV-6 specifies standard (e.g. doctrinal) operational activities, and optionally traces them to the capabilities they support.

(*Note: the name of this MODAF view was changed from 'Operational Activity to Capability Mapping'*).

Background

Some processes are standard across the whole enterprise, or even more than one enterprise. The StV-6 specifies Standard Operational Activities that can be re-used across multiple logical architectures (i.e. the MODAF Operational Viewpoints).

An StV-6 can also show which capabilities the standard operational activities support

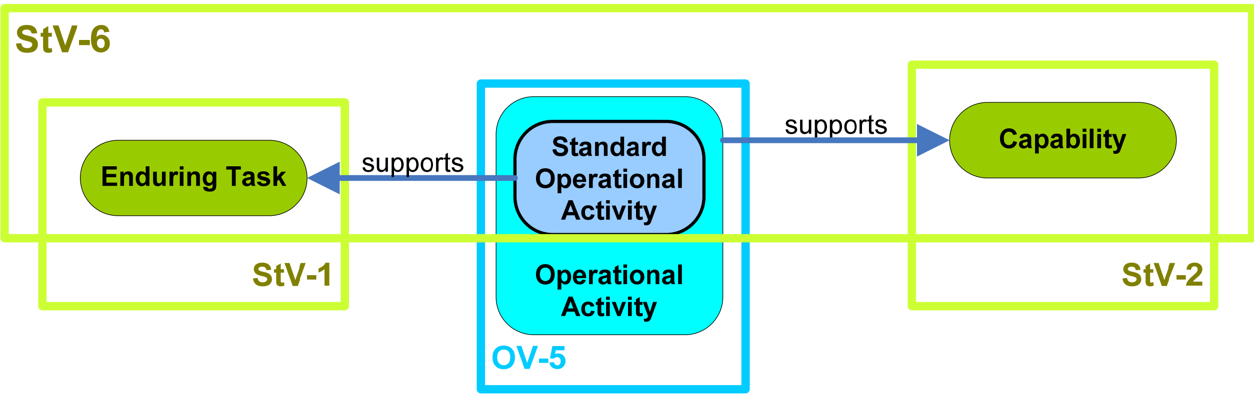
Usage

- Specification of doctrine
- Tracing capabilities to enduring tasks.
- Tracing capabilities to standard operational activities
- Capability audit.

Data objects

The data in an StV-6 can include:

- Capability.
- Standard Operational Activity.
- Enduring Task



Relationships between Key Data Objects (Simplified from M3)

Representation

- Table.
- Tracing Diagram.

Detailed Product Description

The StV-6 view specifies the Standard Operational Activities, and the enduring tasks and capabilities they support.

It is created with the Strategic Architecture (i.e. before the creation of supporting Operational Views), and will consist of a library of pre-defined functions taken from Doctrine[1]. Consequently, the OV-5, Operation Activity Model, should contain activities that are generalisations of the Standard Operational Activities from the StV-6.

An StV-6 is usually shown in the form of a table, optionally listing the supported capabilities and enduring tasks.

Standard Operational Activities	Capabilities Supported	Enduring Tasks Supported
Recce	Information Acquisition	Military Intelligence Conduct Operations
Collate Intelligence	Information Management	Military Intelligence Conduct Operations
Conduct Estimate	Information Management	Conduct Operations
Coordinate Plan	Information Acquisition Information Management Effects	Conduct Operations
Attack	Information Acquisition Information Management Effects	Conduct Operations
Recouperate	Information Management	Conduct Operations

StV-6 Tabular Example

[1] The Joint Essential Task list is an example of a source for such functions.

Operational view (OV) viewpoint

A description of and guidance for the use of the MOD Architecture Framework (MODAF) OperationalÂ View Viewpoint.

The Operational Views (OVs) define the logical aspects of the architecture. A suite of OV products may be used to describe a requirement for a to-be architecture in logical terms, or as a simplified description of the key behavioural and information aspects of an as-is architecture. The OVs re-use the capabilities defined in the StVs and put them in context of an operation or scenario. The OVs can be used at a number of points through the MOD lifecycle including the development of user requirements, capturing future concepts and supporting the operational planning process. A number of stakeholders will develop and exploit OVs during the MOD acquisition lifecycle.

More details on the OVs and Information Exchange Requirements (IERs) are contained in the document IERs in MODAF.

- **OV viewpoint PDF [1.1 MB]**
- **IERs in MODAF PDF [301.0 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

ViewpointsÂ andÂ views
Parent page

Views summary documents

All views viewpoint

Strategic view viewpoint

System view viewpoint

Technical standards view viewpoint

Acquisition view viewpoint

Service oriented view viewpoint

MODAF - detailed guidance

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page rated 2 times

The MODAF Operational Viewpoint

Viewpoint Summary

The Operational Viewpoint provides a logical perspective of the architecture: i.e. it defines (in abstract rather than physical terms) the processes, information and entities needed to fulfil the capability requirements, but does not consider how the solution may manifest itself.

In MODAF the Operational Viewpoint is represented in a series of Operational Views (OVs) that depict organisational entities, processes and information, and the relationships between them, in the context of an Enterprise or Enterprise Phase established in the Strategic Views (StVs).

It should be noted that in MODAF the OVs also include some service elements.

Views

There are 11 OVs (including sub-views) that make up the Operational Viewpoint:

Introduction to OV-1a, OV-1b and OV-1c

The OV-1 provides a high level, scenario-based, description of how a business or military objective might be achieved. It describes a mission[1] or type of mission within the scenario, highlighting the main operational elements and any interesting or unique aspects of the operation.

The OV-1 provides a means of organising the operational architecture models into distinct groups based on scenario context and it communicates the essence of the scenario context in a graphical form supported by textual descriptions, which is ideal for communicating the purpose of the architecture to non-technical stakeholders.

There are three parts to the OV-1:

- OV-1a is the graphic itself.
- OV-1b is a text description providing more detail.
- OV-1c is a tabular representation of key parametric data associated with the scenario (often showing evolution of capability over time). These should reflect the capability requirements.

[1] In the MODAF Meta Model (M3), a Mission is defined in quite general terms as a purpose to which resources may be directed.

OV-1a - High-Level Operational Concept Graphic

The OV-1a provides a graphical view of what the architecture is addressing and an idea of the players and operations involved. Its main use is to aid human communication, and it is intended for presentation to high-level decision makers.

Background

The OV-1a provides a graphical, scenario-based, description of a mission[2] or class of mission could fulfil a business objective. It shows the main nodes (see the definition in OV-2, Operational Node Relationship Description) and interesting or unique aspects of operations. It describes the interactions between the subject architecture and its environment (including external systems), in order to convey the right amount of information to stakeholders. Its main utility is to communicate the purpose of the architecture to non-technical, high-level decision makers.

Unlike an OV-2, an OV-1a may show elements of the solution (physical) architecture – in other words, OV-1a is not strictly speaking a logical view.

Usage

- Puts an operational situation or scenario into context.
- Provides a tool for discussion and presentation; for example, aids industry engagement in acquisition.
- Provides an overview of more detailed information in published architectures.

Data objects:

An OV-1a is typically just a graphic, but MODAF allows each symbol in the graphic to be traced back to elements and relationships in the M3. The data to be included in an OV-1a can be any business objects of interest, including:

- Nodes (e.g. headquarters).
- Systems.
- Organisations.
- Information Flows.
- Environmental context objects (e.g. rivers, hills).

Representation

- Graphic.
- Structured graphic.
- UML class diagram (context diagram).
- UML use case.

Detailed Product Description

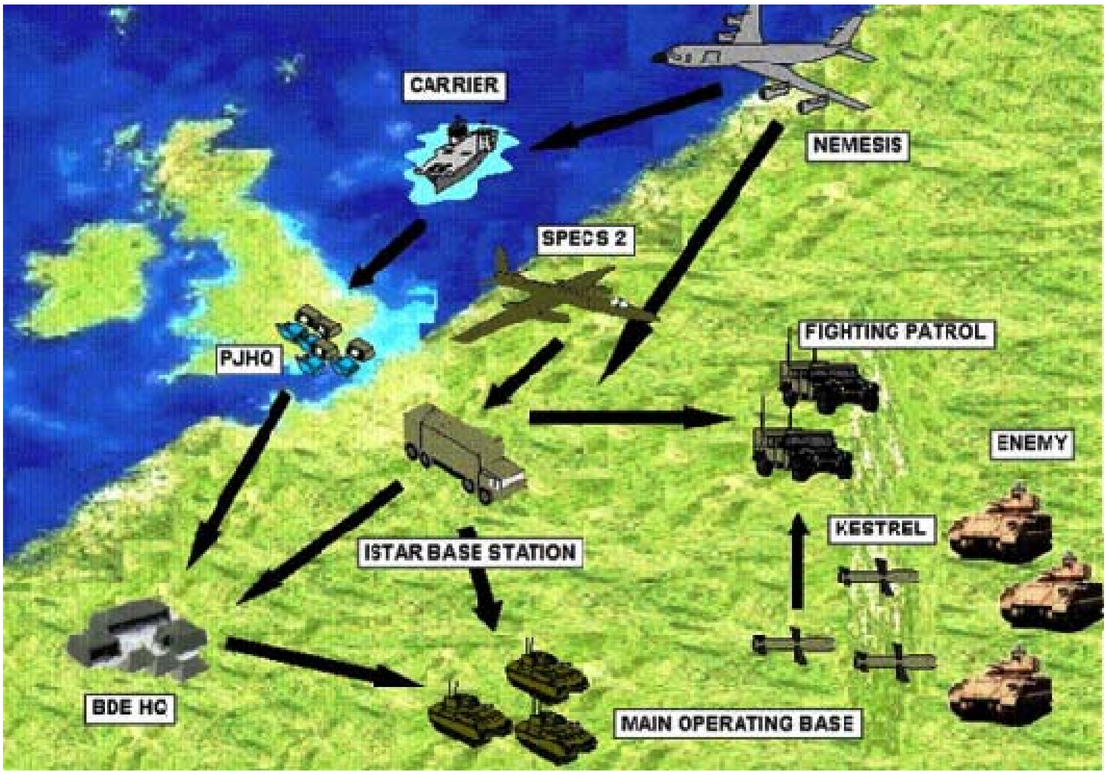
[2] In the MODAF Meta Model (M3), a Mission is defined in quite general terms as a purpose to which resources may be directed.

Each operational view describes one or more Enterprise Phases.

An OV-1a depicts the mission or domain covered by the architecture. In simple terms, an OV-1a will communicate the purpose of architecture is about and provide an idea of the players and operations involved.

The OV-1a provides a graphical executive summary of the architectural endeavour, which describes the interactions between the subject architecture and its environment, and between the architecture and external systems. A textual description accompanying the graphic is essential, with labels on the graphic and a detailed description in the OV-1b. Graphics alone are not sufficient for capturing the necessary architecture data.

The purpose of OV-1a is to provide a quick, high-level description of the business objective that the architecture is addressing, and how that objective might be achieved. An OV-1a can be used to orient and focus detailed discussions. Its main utility is to communicate the purpose of the architecture to non-technical, high-level decision makers.

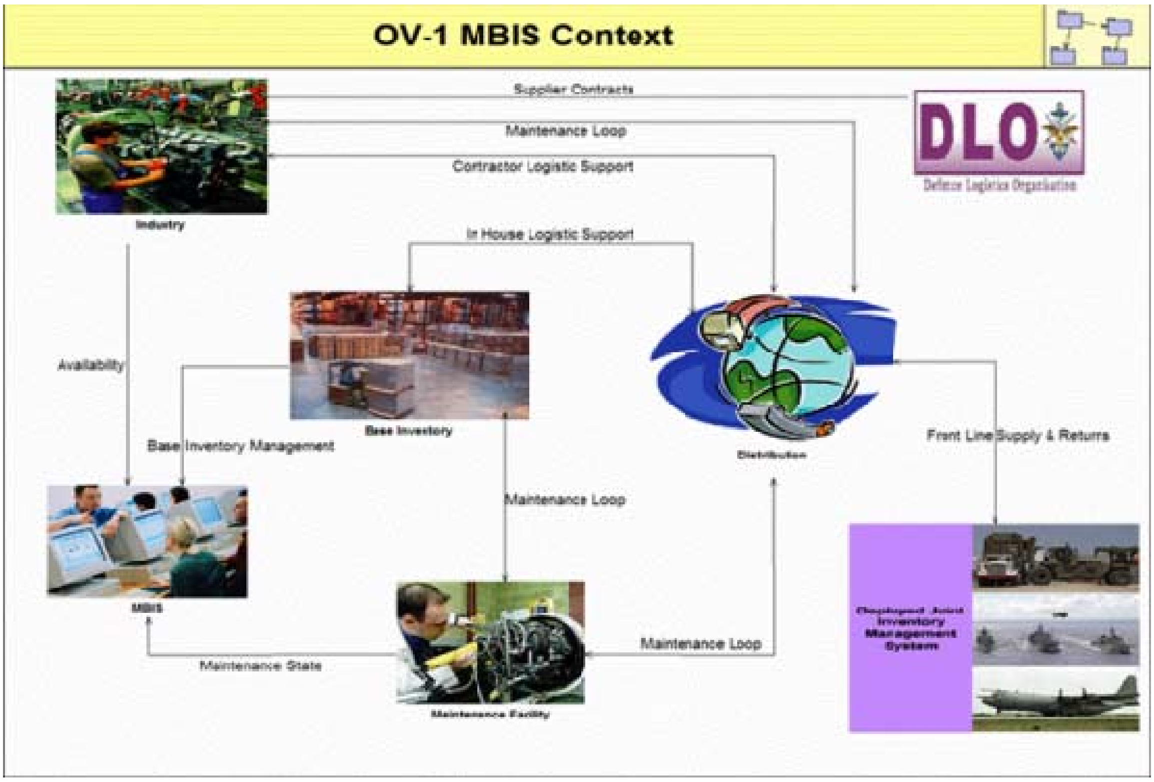


Example OV-1a

The content of an OV-1a depends on the scope and intent of the architecture, but in general it describes the missions, high-level operations, organisations, and geographical distribution of assets. It will provide an overview of the operational concept (what happens, who does what, in what order, to accomplish what goal) and highlight interactions to the environment and other external systems. The content should, however, reflect the executive summary level of the OV-1a, as the other OVs provide the detail of the interactions and sequencing.

In some cases, OV-1a is the last product to be developed, as it conveys summary information about the whole architecture for a given scenario.

OV-1a is the most general of the architectural views and the most flexible in format. Because the format is freeform and variable, no template is shown for this view. An OV-1a product will usually, however, consists of graphics and/or text presented in a form that best communicates the idea of the architecture to the stakeholders.



Example OV-1a

From a modelling perspective, the OV-1a view is useful in establishing the context for a suite of related operational views. This context may be in terms of an Enterprise Phase, a time period, a mission and / or a location. In particular, this provides a container for the spatio-temporally constrained performance parameters depicted in OV-1c.

For example, the operational performance metrics for desert warfare in Phase 1 may be different to those in Phase 2. The metrics for jungle warfare in Phase 2 may be different to those for desert warfare in Phase 2.

OV-1b Operational Concept Description

The OV-1b provides a textual description that explains and details the scenario contained within the associated OV-1a, High Level Operational Concept Graphic. The OV-1b should always be developed alongside the associated OV-1a.

Background

An OV-1b product is used to explain and add further detail to the graphical presentation of the scenario shown in the associated OV-1a. It will be developed alongside the OV-1a, and used together they will provide a comprehensive summary of the scenario or use case described within the operational views of the architecture.

Usage

- Concept of operations.
- Input to User Requirements Document (URD).

Data objects

OV-1b is a textual description of the OV-1a graphic so does not usually have specific data objects associated with it.

Representation

- Text.

Detailed Product Description

The Operational Concept Description (OV-1b) View provides a supplementary textual description that explains and details the scenario contained within the associated High Level Operational Concept Graphic (OV-1a) View.

The nature and type of description in an OV-1b product will be very dependant upon the level of detail and maturity in the operational scenario or architecture being described.

Regardless of the method of representation, it is imperative that the information in the view is consistent with the other OVs, and when the OV-1b is updated or modified these changes are cascaded throughout the architecture.

> ISTAR information is currently provided by the SPECS system, the LOOKER UAV system and the NEMESIS system. SPECS is an operational level asset and communicates via a data link to its dedicated base station. LOOKER is a tactical UAV system that can transmit real-time video footage directly to either Fighting Patrols or the Brigade HQ. NEMESIS is a strategic asset that has considerable on-board processing capability, enabling the data to be exploited during flight. The resultant information can be communicated either by satellite communications or directly to a receiver based on board a naval vessel.

Example OV-1b

OV-1c - Operational Performance Attributes

The OV-1c provides detail of the operational performance attributes associated with the scenario / use case represented in the OV-1a, High Level Operating Concept Graphic, and how these might evolve over time.

Background

An OV-1c Product is used to specify quantifiable attributes and values within the scenario / use case represented in the OV-1a. The values expressed define the performance of specific or multiple capability elements, and can be represented as either single values or a range of values across a defined timescale. The data may indicate changes in particular performance parameters from one Enterprise Phase to the next.

Usage

- Definition of performance characteristics.
- Measures of effectiveness (input to URD).

Data objects

The data in an OV-1c can include:

- Metrics associated with performance associated with specific concepts within the scenario specified within the OV-1a.

Representation

- Tabulation.
- SysML parametric diagram.

Detailed Product Description

The performance of an operational scenario or use case can be measured in a variety of different ways depending upon the scenario context, the capabilities needed to satisfy the requirement and the systems deployed to provide the required capabilities. Possible attributes may include operational tempo, accuracy of targeting, fratricide rate, etc. Furthermore, it may be possible to link the attributes to a specific system or it may only possible to consider the attributes as an emergent property, for example they are dependent upon all of the elements that are interacting within the scenario, rather than an attribute of the individual elements.

The attributes and values that are specified may be as single values (eg the target engagement process is to be concluded in a maximum time of 25 minutes) or may be used to represent trends or targets that are expected to be achieved. This type of attribute would be represented by a number of values for various points in time or periods of time.

Attribute	Measure	Value			
		As-Is	Period of Time 1	Period of Time 2	Target
Operational Tempo	Rate of advance for an armoured brigade against light resistance	20 km/day	40 km/day	60 km/day	80 km/day
Synchronisation of Effects	Simultaneous rounds on impact delivered by and artillery battery	30 rounds	40 rounds	60 rounds	100 rounds
Sortie rate	Period to refuel and rearm aircraft	4 hours	3 hours	2 hours	1 hour

Example OV-1c

The measurable values that appear in an OV-1c may be performance parameters as shown in the previous example. However, other measurable parameters can be shown as in the following example which refers to sustainability parameters.

Attribute	Measure	2025	2026	2027	2028	2029	2030	Target
SPECS 2 Availability	Number of days down time	45	30	20	18	15	14	10
SPECS 2 Maintainability	Support personnel required to maintain SPECS 2	50	50	45	40	35	34	30
SPECS 2 Reliability	Number of days unplanned down time	10	8	7	6	5	5	5

Example OV-1c

OV-2 - Operational Node Relationship Description

The OV-2 shows, at a high level, the interactions between logical[3] nodes and depicts the capabilities that those nodes bring to the architecture.

Background

The primary purpose of the OV-2 is to specify nodes (elements of capability) in context with each other. The context is usually expressed in terms of the information that flows between the nodes (e.g. the information flow requirements between capabilities in a given scenario). However, the context may also be flows of materiel, human resource or energy.

With MODAF V1.2, the OV-2 has been developed to:

- Adopt a more formal definition of logical flows to represent node connections.
- Support the introduction of Service Oriented Architecture (SOA).
- Accommodate the use of known resources (as defined in SV-1).

Usage

- Definition of operational concepts.
- Elaboration of capability requirements.
- Definition of collaboration needs.
- Associating capability with a location.
- Problem space definition.
- Operational planning.
- Supply chain analysis.
- Security models – e.g. domain-based security and entity trust models.

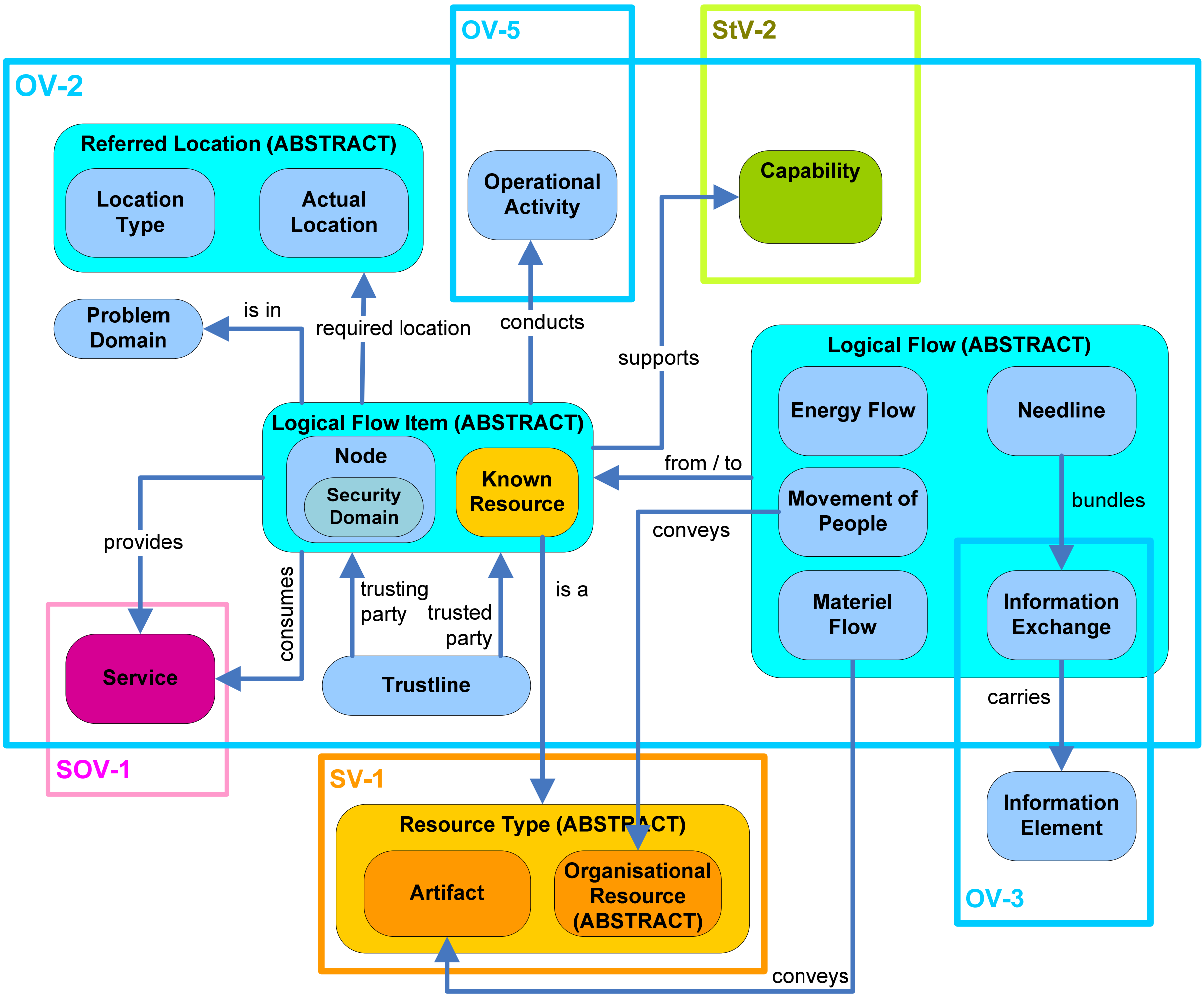
Data objects

The data in an OV-2 can include:

- Nodes.
- Needlines (bundles of information exchanges).
- Flows of materiel, people or energy.
- Operational Activities.
- Security Domains.
- Trust Lines (for entity trust models).
- Locations ('real' or logical).
- Services.

---

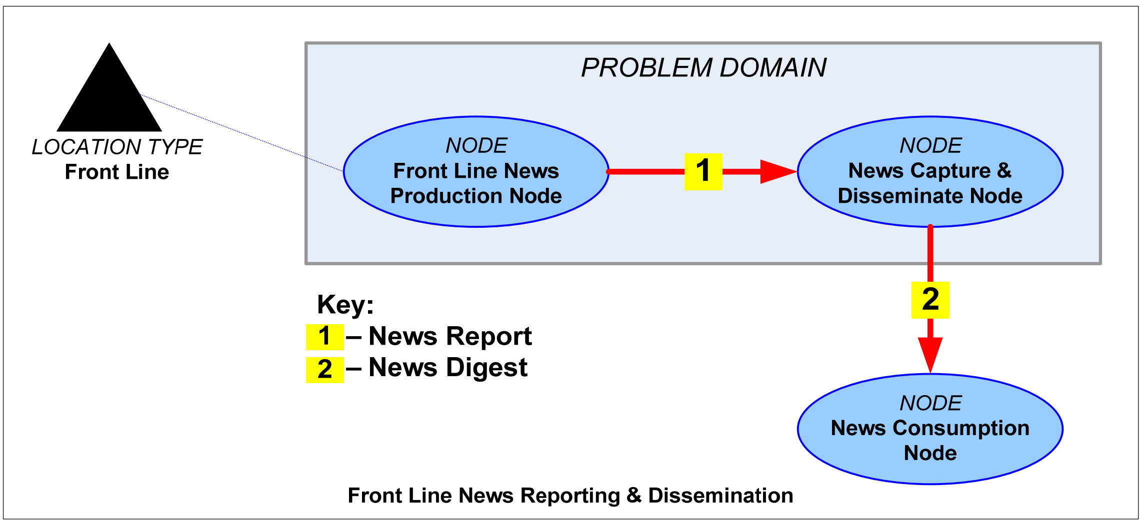
[3] i.e. in abstract rather than physical terms, so as to be solution independent.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Topological (connected shapes).
- UML composite structure diagram.



Non-UML OV-2 Example

Structured text may also be used to provide a fuller description of the needlines.

Detailed Product Description

The OV-2 depicts nodes and the needlines between them, primarily to indicate a need to exchange or share information. The OV-2 may, however, also show the location (or type of location or environment) of nodes, and may optionally be annotated to show flows of people, materiel or energy between nodes.

The primary purpose of an OV-2 is to define the logical structure of architecture. Building on the strategic intent identified in StV-1, Enterprise Vision, OV-2 takes the required capabilities and expresses them as nodes which interact by exchanging information or producing / consuming services.

The nodes in the logical architecture do not represent specific organisations, systems or locations. This enables Information Exchange Requirements (IERs) to be established without prescribing the way that the information exchange is handled. OV-2 does not, therefore, depict the physical connectivity between the nodes. An OV-2 can be a powerful way of expressing the differences between an as-is architecture and a proposed architecture to non-technical stakeholders, as it can be used to emphasise how information flows (or does not flow) without becoming over-complex.

Nodes

A node is a logical element of capability that may produce, consume, or process information, energy, materiel or people. What constitutes a node can vary among architectures. Here are some examples:

- A logical or functional grouping (e.g. Logistics Node, Intelligence Node).
- The headquarters for an organisation (e.g. Command HQ) or an organisation type (e.g. Joint Task Force HQ).
- A capability or other facility of importance to the business expressed in context of a requirement to interoperate with other capabilities.

Nodes conduct operational activities and, therefore, an OV-2 indicates the key players and the interactions necessary to conduct the corresponding operational activities of an OV-5, Operational Activity Model

Known Resources

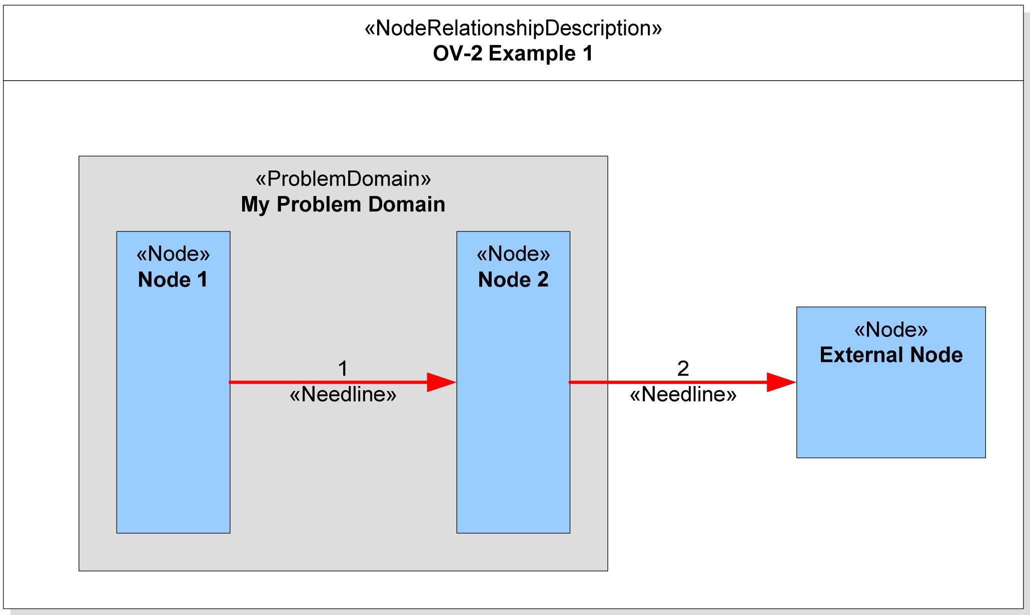
In addition to logical nodes, MODAF allows known resources to be depicted on OV-2s. These resources are defined in SV-1, and are to be used where a constraint on the logical solution exists due to existing resources, such as in a maritime operation where an aircraft carrier is always part of the solution. Known resources shall not be used in a problem domain.

Needlines

Needlines document the required or actual exchange of information between nodes; they are conduits for one or more information exchanges i.e. they represent a logical bundle of information flows.

A needline does not indicate how the information transfer is implemented. For example, if information produced at Node A is simply routed through Node B and is used at Node C, then Node B would not be shown on the OV-2 diagram – the Needline would go from Node A to Node C. OV-2 is not a communications link or communications network diagram but a high-level definition of the logical requirement for information exchange between elements of capability (nodes).

Needlines are represented by arrows that indicate the direction of flow and are annotated with a diagram-unique identifier and a phrase that is descriptive of the principal type of exchange. It may be convenient to present these phrases in a key to the diagram to prevent cluttering. It is important to note that the arrows (with identifiers) on the diagram represent needlines only. This means that each arrow indicates only that there is a need for some kind of information transfer between the two connected nodes.



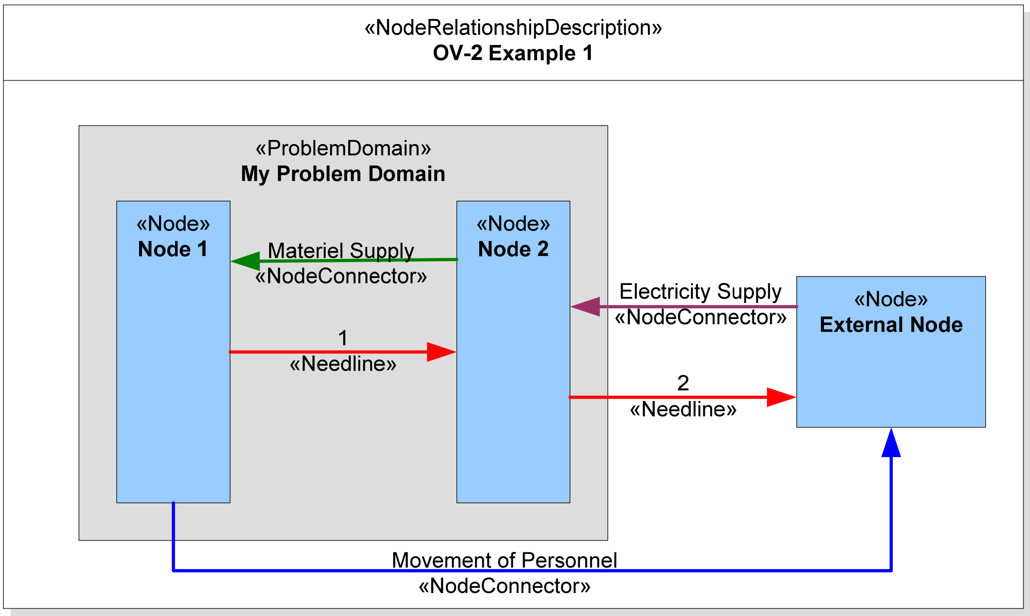
Generic OV-2 showing Needlines

The diagram may include the needline identifiers as numerical labels (as in the example above). Alternatively short phrases may be used.

Because needline identifiers are often needed to provide a trace reference for information exchange requirements (see OV-3), a combined approach with numerical and text labels can be used.

In most cases there will be only one needline between any two nodes (which may carry multiple information exchanges). This is not mandatory, however, and the architect may choose to group the exchanges into more than one needline.

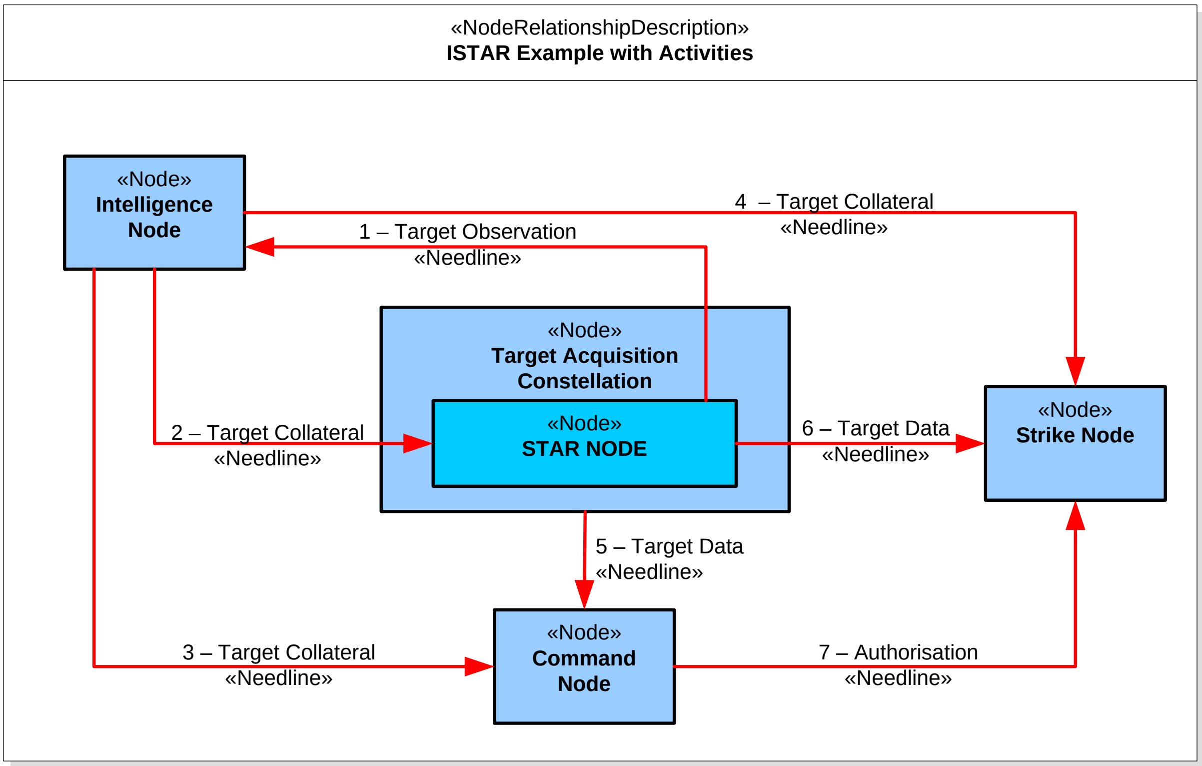
There is a one-to-many relationship from needlines to information exchanges (e.g. a single needline in OV-2 represents multiple individual information exchanges). The mapping of the exchanges to the needlines of OV-2 occurs in the OV-3, Operational Information Exchange Matrix. For example, an OV-2 may have a needline labelled “Situation Report” which represents a number of information exchanges, consisting of various types of reports (information elements), and their attributes (such as periodicity and timeliness). The identity of the individual elements and their attributes are documented in OV-3, along with the producing and consuming activities from OV-5, Operational Activity Model.



Generic OV-2 showing Needlines and Flows

Nesting of Nodes

It is often convenient to model nodes as being nested, in other words one node is part of another. A simple example is shown below.



Example OV-2 with nested Nodes

This technique may be used to include the same node more than once on the diagram. This works because each occurrence of the node has a different usage context. Care should be taken when using nested nodes, particularly when OV-2 is being used to express a user requirement. Nesting nodes would imply a structure on the solution architecture and so could close off some avenues of innovation.

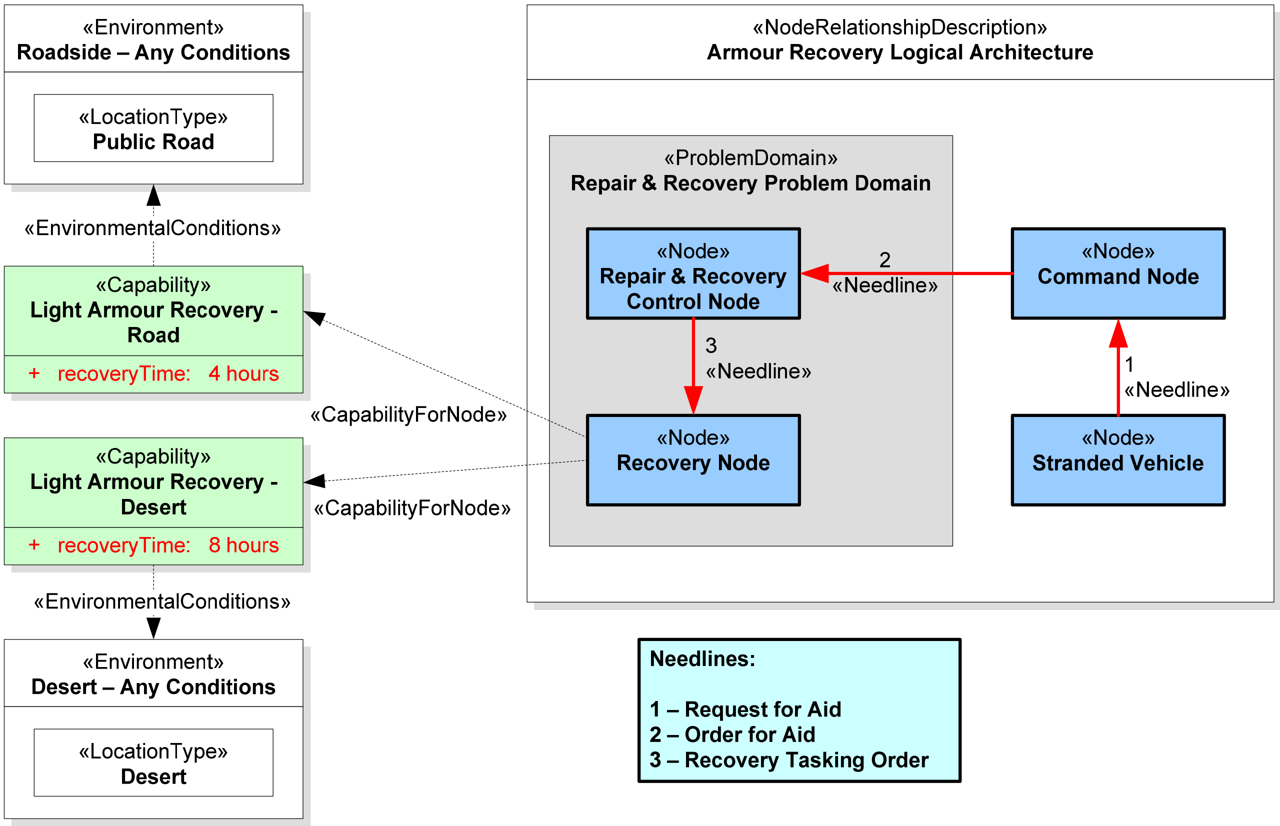
Nesting is also sometimes used to show 'roles' associated with each node (often together with the activities those roles perform). It is important that the OV-2 logical view maintains focus on the operational requirements and avoids 'solutioneering'. The 'roles' that may feature on an OV-2 are used as a convenient means to compartmentalise the logical architecture. A legitimate example of this is the use of an OV-2 to depict a generic set of functional cells within a generic headquarters (such as a Land Battlegroup HQ). The capability required of each functional cell may be delivered by people alone or through a combination of systems and people (see capability configurations within SV-1).

Trade-Space and Requirements

The OV-2 may also represent operational concepts that are of critical importance to requirements definition. In OV-2 this is achieved by mapping capabilities onto nodes to represent the required level of capability in the architecture. The requirements specified in this way may then be realised by more than one suite of SVs; i.e. there may be multiple potential specifications that can be traded off against each other.

The OV-2 can also describe the trade-space by using an M3 concept called Problem Domain. Those nodes that are within the problem domain are those expected to be delivered in the solution. Those outside the problem domain are not part of the solution but represent external elements the solution will be expected to interact with. This is important for several reasons:

- A User Requirements Document is intended to define a bounded operational capability and it is therefore helpful to reflect this in any operational architecture models that provide context for those requirements.
- It is essential to model capabilities outside the boundary to be able identify dependencies so that interoperability requirements can be modelled (in terms of collaboration across the boundary), and external constraints can be highlighted. The definition of boundary provides focus for several other views (e.g. OV-3, Operational Information Exchange Matrix, and SV-2, Systems Communications Description).



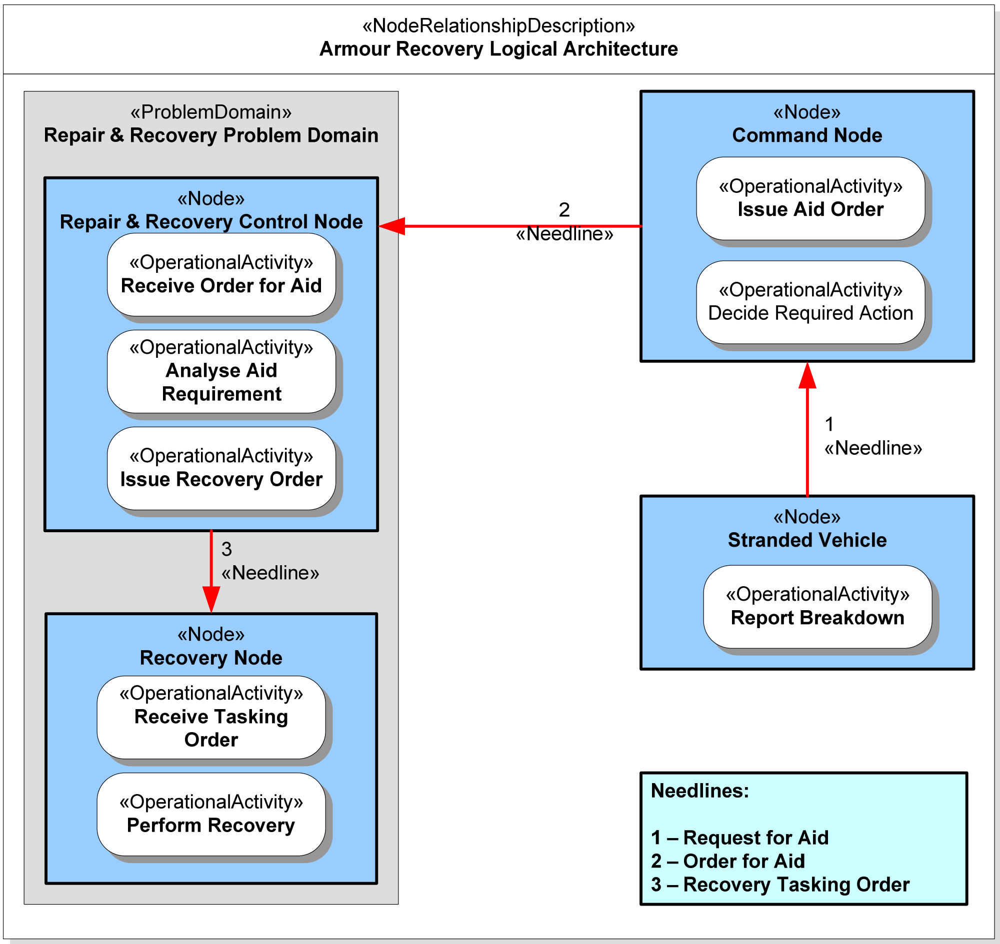
OV-2 showing Problem Domain

A node can be realised by a resource or combination of resources (specified in SV-1).

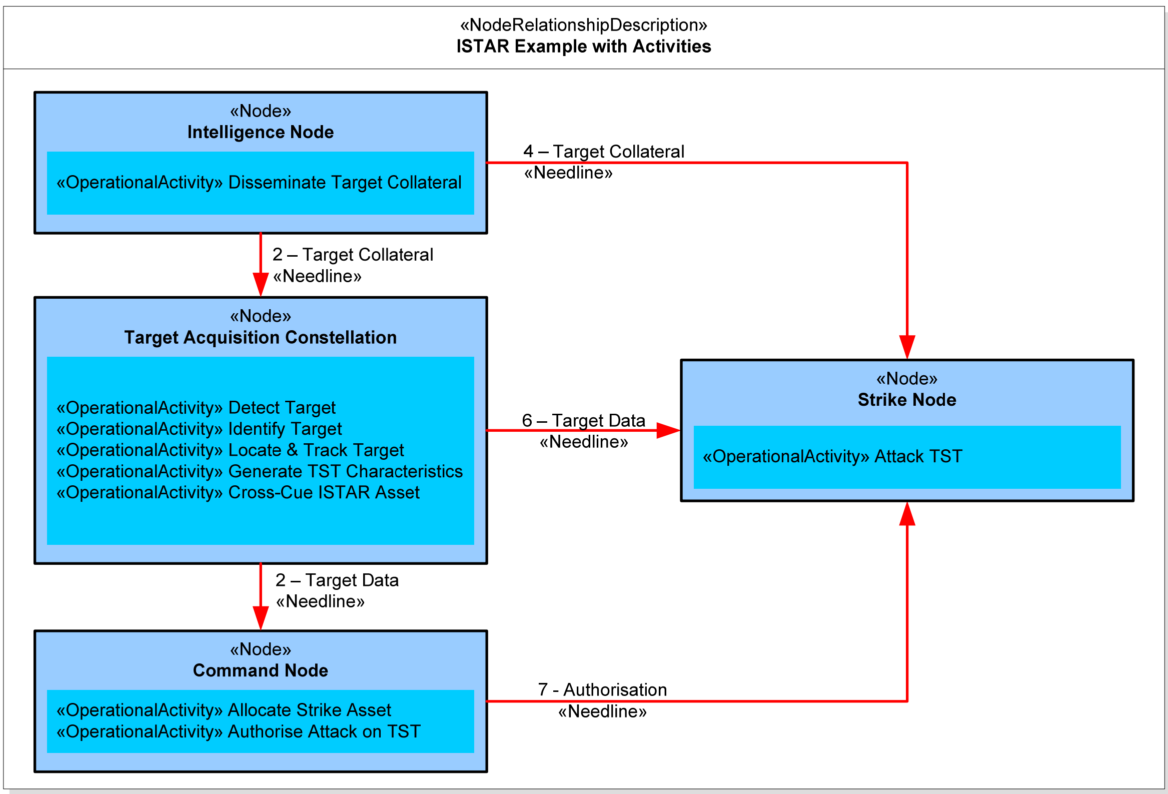
Operational Activities

The operational activities (from the OV-5, Operational Activity Model) performed by a given node may be listed on the graphic, if space permits. OV-2 and OV-5 are complementary descriptions. OV-2 focuses on the nodes, with the activities being a secondary adornment. OV-5, on the other hand, places first-order attention on operational activities and only second-order attention on nodes, which can be shown as annotations or swim-lanes on the activities. In developing a logical architecture, OV-2 and OV-5 are often the starting points and these may be developed iteratively.

Examples of how this can be depicted are illustrated in following diagrams on the following pages.



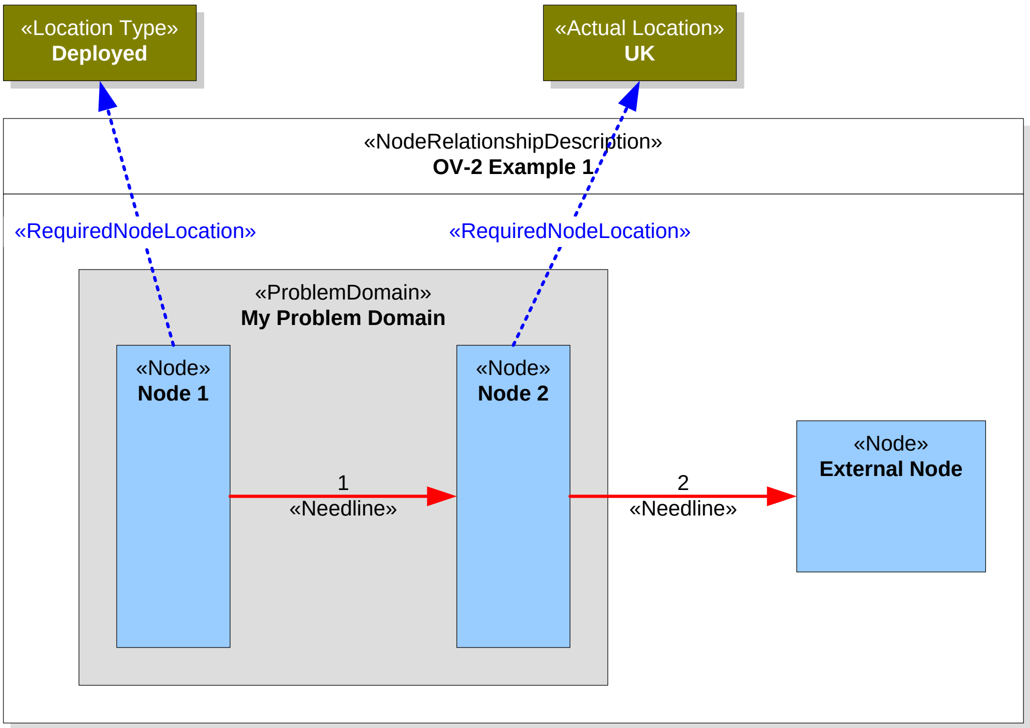
OV-2 showing Nodes having Operational Activities



OV-2 showing Nodes having Operational Activities

Locations

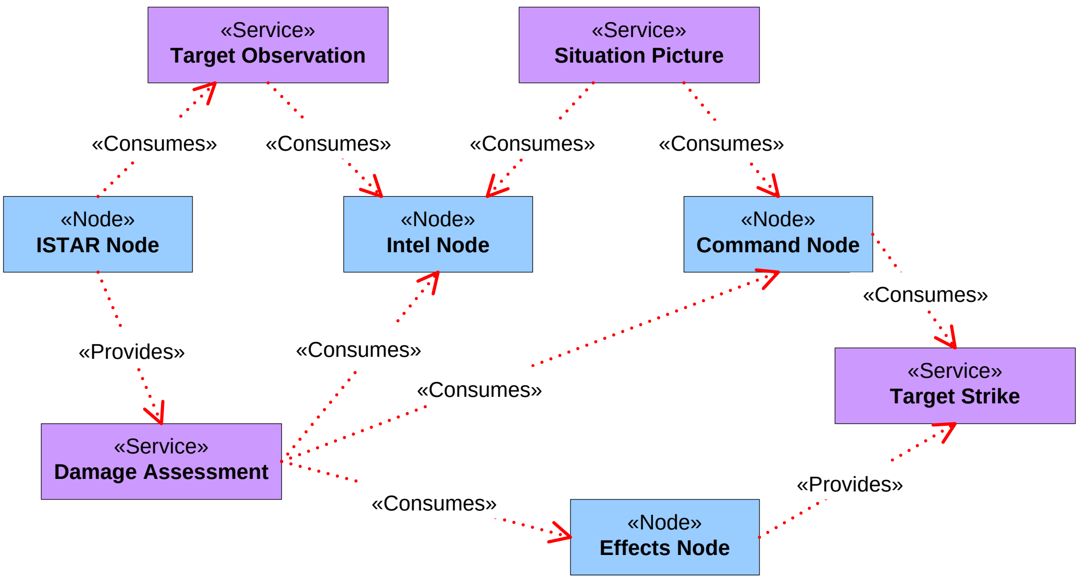
An OV-2 can also show the location of each node, if the location is known or knowable. The location may be specified geographically, and this in turn may be a specific geographic location (eg “RAF Wyton”) or a type of location or environment (eg “behind enemy lines”).



Generic OV-2 with locations

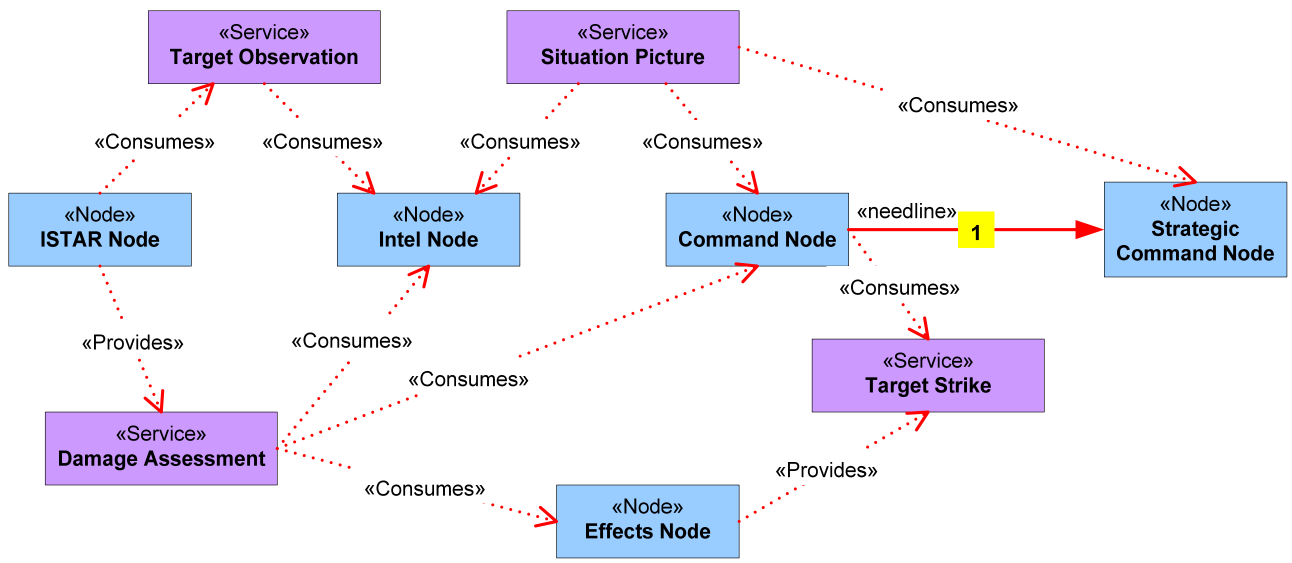
Service Oriented Architectures

If the architect is developing a Service Oriented Architecture (SOA), an OV-2 may be used to show which logical agents (nodes) produce and consume services. The concept of producing and consuming services replaces the idea of fixed needlines – loose coupling is a tenet of SOA.



OV-2 showing Service Elements

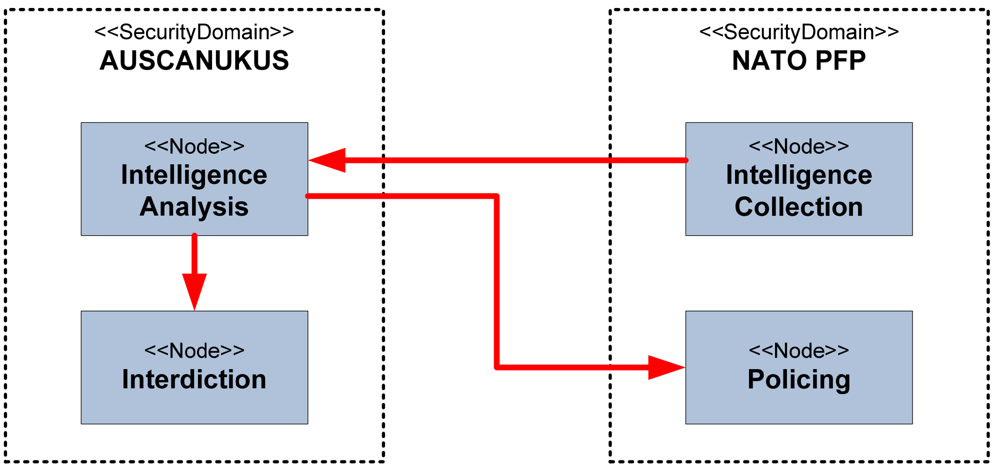
As with a non-SOA OV-2, the capabilities of the nodes may also be shown. Most architectures are likely to consist of point-to-point connections as well as service interactions, so it is possible to have OV-2 products which combine the needline and service approach:



OV-2 showing Service Elements with traditional needlines

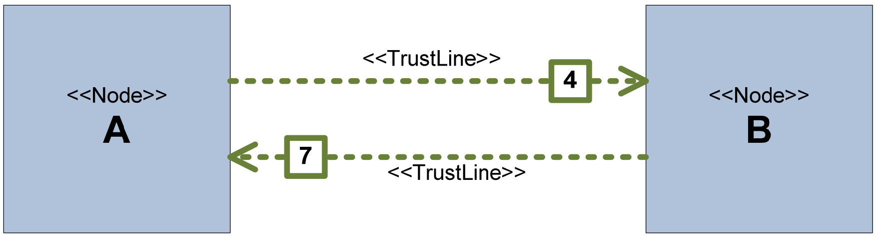
Security Models in OV-2

An OV-2 may also be used to model certain aspects of security. In particular, security domains may be shown, which can be used to assert a common security policy over a number of nodes or known resources:



OV-2 showing needlines between nodes belonging to different security domains

As well as showing security domains, OV-2 may also be used to specify entity trust relationships between nodes and known resources. The trust is shown as a line between the nodes or resources, specifying a numeric level of trust. The arrow of the trust line points to the trusted party, and the number indicates how much that party is trusted by the party at the non-arrow end. MODAF does not specify a scale of trust, and the numeric values used may be different for each architecture, and policy should be outlined as to what nature of information (e.g. protective marking) may be shared along trust lines assigned those values. Note that this mechanism allows for cases where mutual trust differs – e.g. party A trusts party B more than B trusts A.



OV-2 excerpt showing trust lines between nodes

Trust lines may also be specified between security domains – meaning that every element inside one domain trusts every element in the other domain to a given level.

OV-3 - Operational Information Exchange Matrix

The OV-3 details the operational information exchanges between nodes, as defined in the OV-2, Operational Node Relationship Diagram.

Background

Information exchanges help define the interoperability requirements associated with the operational capability of interest. Although the primary purpose of this view is to specify information exchanges, an OV-3 may also list flows of materiel, energy and human resources.

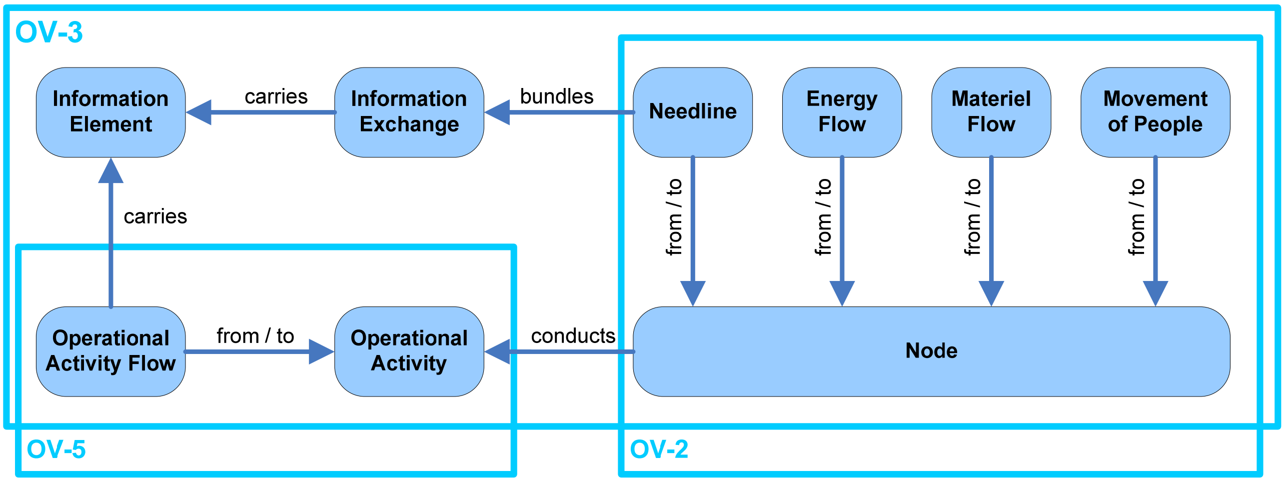
Usage

- Definition of interoperability requirements.

Data objects

The data in an OV-3 can include:

- Information Exchanges (each associated with a needline).
- Information Elements (each carried by one or more information exchange).
- Operational Activities (that produce and consume the information elements).
- Nodes (between which the information exchanges take place).



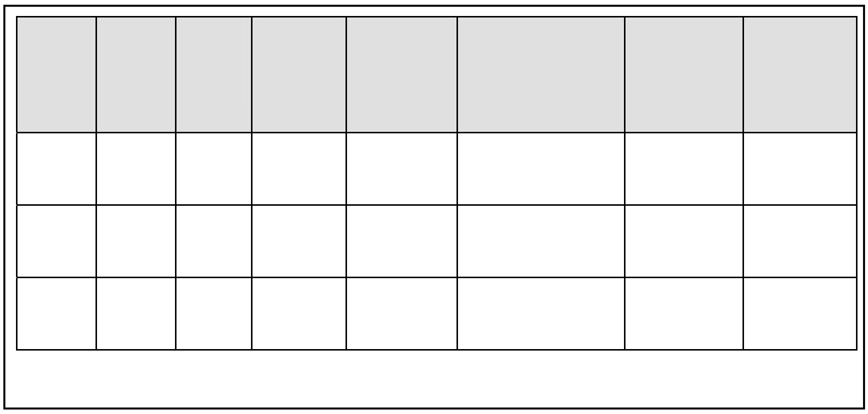
Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabulation.

Detailed Product Description

The OV-3, Operational Information Exchange Matrix, identifies the information transfers that are necessary to enable the nodes to achieve a business objective. This view is initially constructed from the information contained in OV-2, Operational Node Relationship Description, and OV-5, Operational Activity Model; however, OV-3 provides a more detailed definition of the information flows between nodes.



The Operational Information Exchange Matrix details information exchanges by identifying which nodes exchange what information, with whom, why the information is necessary, and the key attributes of the associated information products. Information exchanges express the relationship across the three main M3 elements for the Operational Viewpoint (operational activities, nodes, and information flows) with a focus on the specific aspects of the information flow and the information content. OV-3 is one of a suite of operational views that address the information content of the operational architecture (the others being OV-2, Operational Node Relationship Diagram, OV-5, Operational Activity Model, and OV-7, Information Model).

The OV-3 maps information elements to the producing and consuming nodes, the needlines between them, and the activities that they support.

An information element is a piece of information that is subject to an operational process. The structure of the information element may be defined by a logical data model (see OV-7, Information Model). Information elements are carried on operation activity information flows (in OV-5) and information exchanges (in OV-2). The same information element may be used in one or more information exchanges.

An architect may specify attributes for the Information exchanges in OV-3. Typical attributes would be “timeliness”, “availability”, “protective marking”, “non-repudiation”, etc.

Multiple information exchanges may be bundled into one needline. In OV-3, this information is captured in tabular form, usually with the needline identifier being shown in one of the columns.

The column headings in an OV-3 matrix are not prescribed by MODAF, this allows the architect to select the most appropriate headings for a given architecture. Most OV-3 tables will at least have columns for:

- Information Exchange ID.
- Producing Node.
- Consuming Node.
- Needline ID.
- Producing Activity.
- Consuming Activity.

A more complex example of an OV-3 is shown on the next page.

The emphasis in this view is on the logical and operational characteristics of the information being exchanged. It is important to note that OV-3 is not intended to be an exhaustive listing of all the details contained in every information exchange of every node associated with the architecture in question. Rather, this product is intended to capture the most important aspects of selected information exchanges.

Needline Identifier	Information Exchange Identifier	Information Element Description					Producer		Consumer		Nature of Transaction				
		Information Element Name and Identifier	Content	Scope	Accuracy	Language	Sending Operational Node Name and Identifier	Sending Operational Activity Name and Identifier	Receiving Operational Node Name and Identifier	Receiving Operational Activity name and Identifier	Mission / Scenario UJTL or METL	Transaction Type	Triggering Event	Interoperability Level Required	Criticality
1	WOC-JFAC C1	BDA Report	Report on Battle Damage	Theatre	1 Day	English	WOC	Conduct Battle Damage Assessment	JFACC	Conduct Munitions Effects Assess-ments	Combat Assessment	Collab-orative	AirStrike 072200, 0615	2A	High
1	WOC JFAC C2	Target Nominations	Report on Possible Targets	Theatre	2 Hours	English	WOC	Recommend Airstrike	JFACC	Request Target Materials	Combat Assessment	Direct	AitTO XX, 072300	1B	High
2															
...															
11	MAW JFAC C1	BDA Report	Report on Battle Damage	Theatre	1 Day	English	MAW	Conduct Battle Damage Assessment	JFACC	Conduct Munitions Effects Assess-ments	Combat Assessment	Collab-orative	AirStrike 072200, 0615	2A	High

More Complex OV-3 Example

OV-4 - Organisational Relationships Chart

The OV-4 shows organisational structures and interactions. OV-4 exists in two forms: typical (e.g. a generic brigade command structure) and actual (e.g. an organisation chart for a department or agency).

Background

A **typical** OV-4 shows the possible relationships between organisational resources (organisations and posts); the key relationship being the composition; i.e. how organisational resources are structured within a parent organisation. It may also show the posts in an organisation and the roles[4] associated with each post. Interactions may be specified between organisational resources (organisations, posts and roles), which may be command relationships. Interactions typically illustrate the fundamental roles and management responsibilities. A **typical** OV-4 can be considered as a special type of SV-1, Resource Interaction Specification, where the resources shown are purely organisational.

An **actual** OV-4 shows the structure of a real organisation at a particular point in time, and is used to provide context to other parts of the architecture such as AV-1, (Architecture) Overview and Summary and the StVs.

Usage

A **typical** OV-4 may be used for:

- Organisational analysis.
- Definition of human roles.
- Operational analysis.

An **actual** OV-4 may be used to:

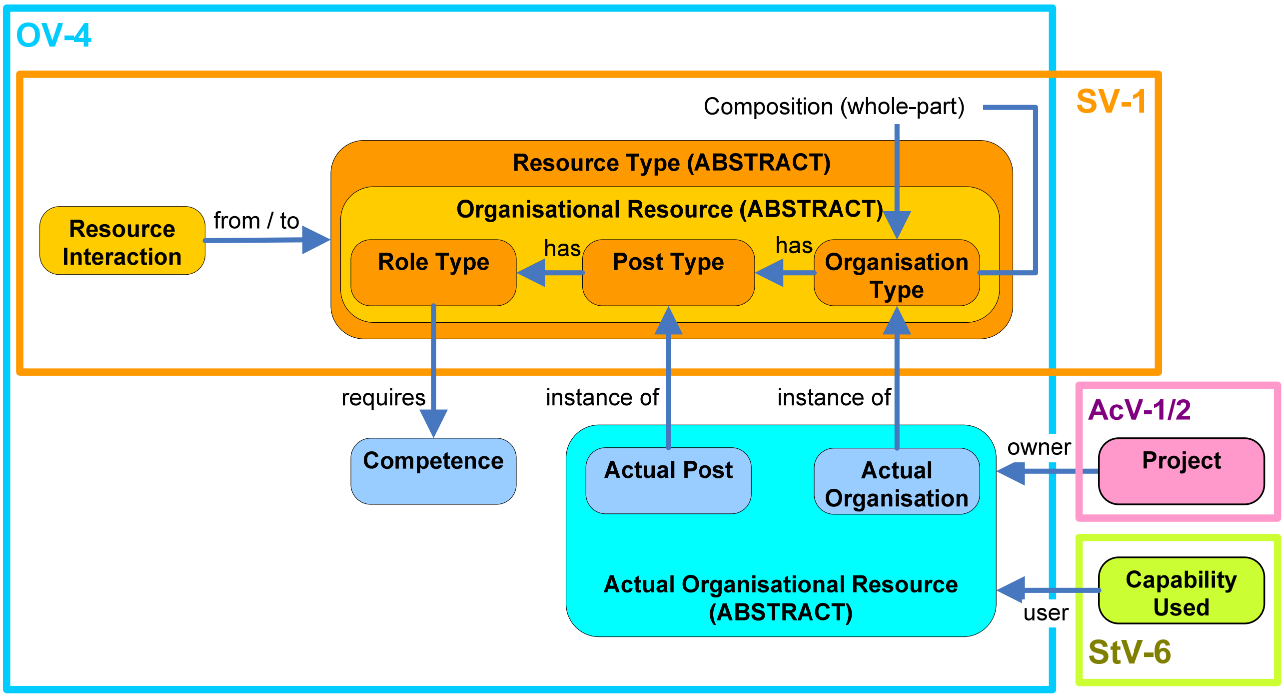
- Identify process owners.
- Illustrate current or future organisation structures.

Data objects

The data in an OV-4 can include:

- Organisation types.
- Resource composition relationships.
- Resource interaction relationships.
- Post types.
- Role types.
- Actual posts and organisations.
- Competences.

[4] The roles represent the functional aspects of organisational resources.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Graphical.
- UML composite structure diagram (**typical**).
- UML instances (**actual**).

Detailed OV-4 Product Description

The OV-4, Organisational Relationships Chart, addresses the organisational aspects of an architecture.

A **typical** OV-4 illustrates the command structure or relationships (as opposed to relationships within a business process flow) among human roles, organisations, or organisation types that are the key players in the business represented by the architecture.

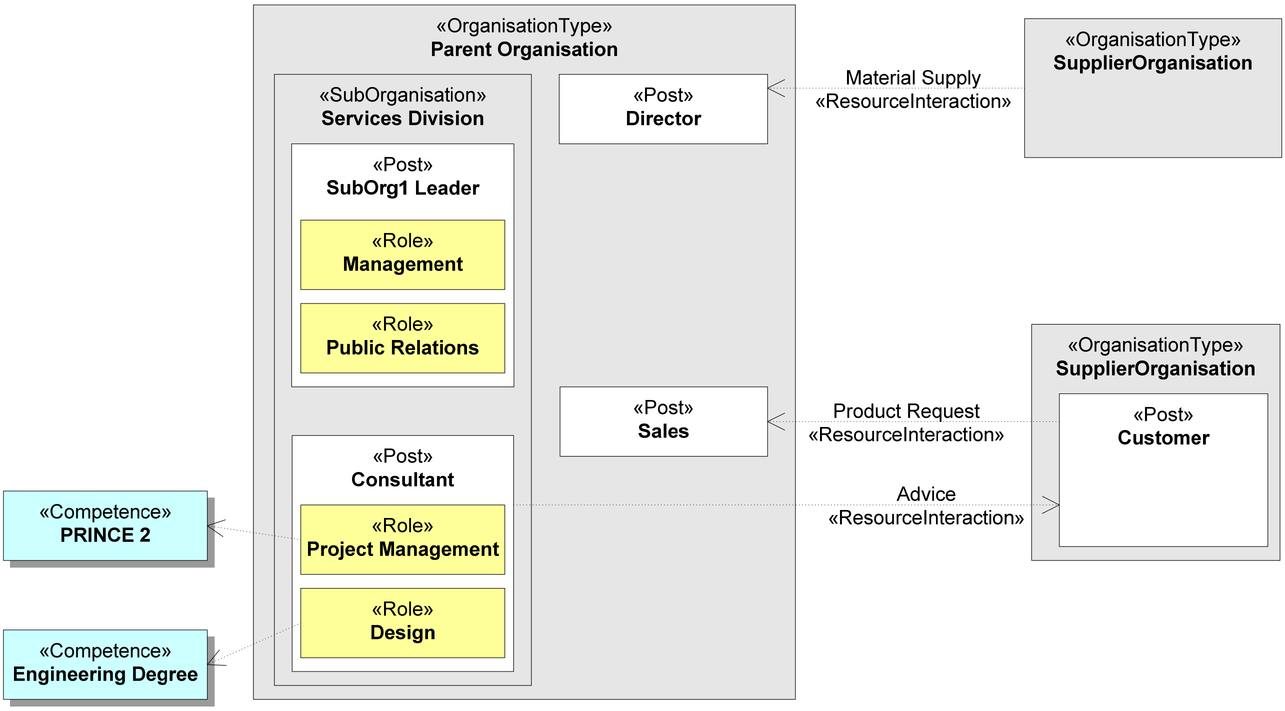
An **actual** OV-4 shows real organisations and posts and the relationships between them.

MODAF only defines two fundamental relationships between Organisational Resources: structure (whole-part) and interaction (which includes the command relationship). When there is a need for other types of organisational relationships, these should be recorded and defined in the AV-2, Integrated Dictionary.

An OV-4 clarifies the various relationships that can exist between organisations and sub-organisations within the Architecture and between internal and external organisations.

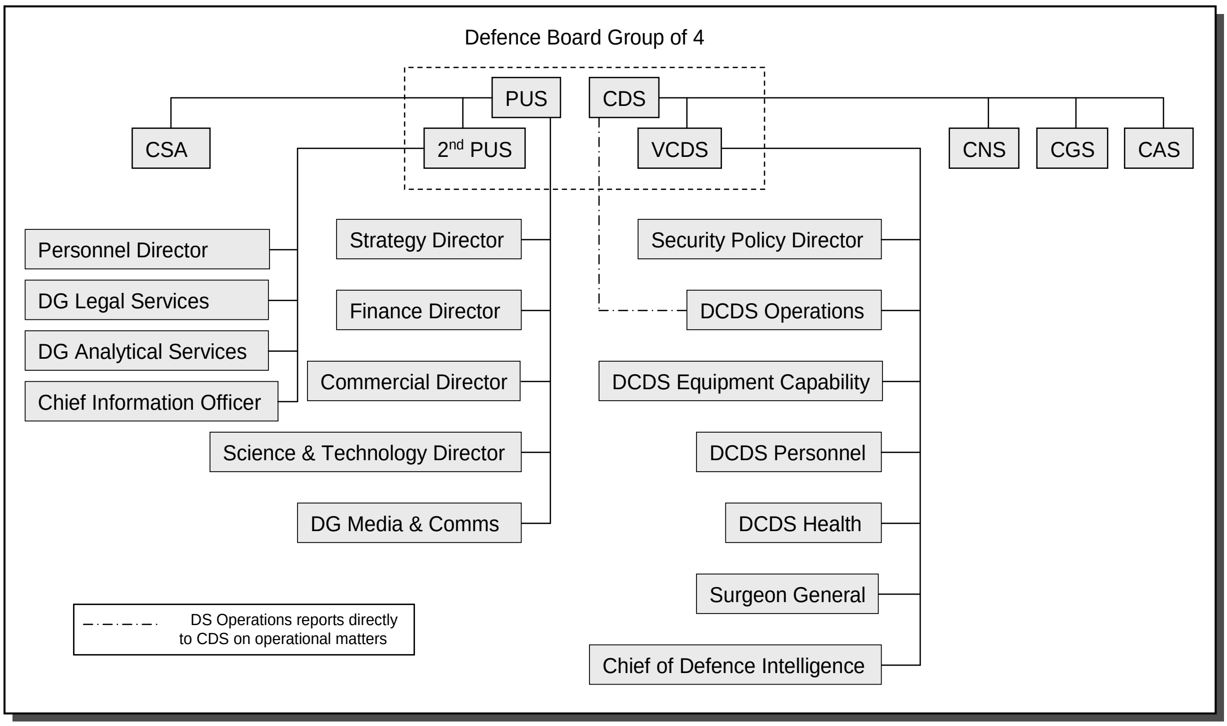
Note that individual people are not modelled in MODAF, but specific posts may be detailed in an **actual** OV-4.

A **typical** OV-4 product may show types of organisations and the typical structure of those organizations:



OV-4 Example

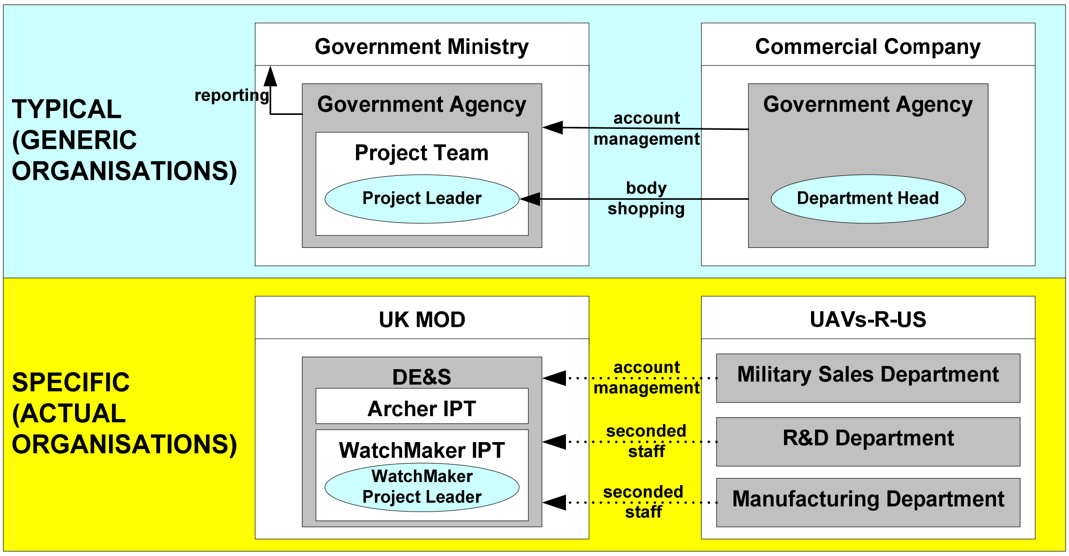
OV-4 products may alternatively show actual, specific organisations (eg "The UK Ministry of Defence Head Office") at some point in time:



OV-4 Example

(Source: MOD Management Framework)

Alternatively, an OV-4 may be a hybrid diagram showing **typical** and **actual** organisation structures:



OV-4 Example

In both the **typical** and **actual** cases, it is possible to overlay resource interaction relationships which denote relationships between organisational elements that are not strictly hierarchical (e.g. a customer-supplier relationship).

In an SV-1, Resource Interaction Specification, the organisational resources defined in a **typical** OV-4 may be part of a capability configuration. Also, **actual** organisations may form elements of a fielded capability which realises the requirements of a node at the system level (again, this may be depicted on an SV-1).

The organisational resources depicted in an OV-4 (typical) may perform functions (SV-4). An OV-4 (actual) may depict operational activities (OV-5) and enduring tasks (StV-1), but only to show ownership of processes.

Roles may require certain competences and this should be modelled where applicable.

The organisations and types of organisation that are modelled using OV-4 in the Operational Viewpoint may also appear in other views, for example SV-1 (organisational constituents of a capability configuration) and AcV-1 (actual organisations that own projects).

OV-5 - Operational Activity Model

The OV-5 describes the activities that are conducted in the course of achieving a mission or a business goal. It describes operational activities, the flows (inputs and outputs) between activities and may optionally show the nodes that conduct the activities.

Background

OV-5 describes the operational activities (or business processes) that are being conducted within the mission or scenario.

OV-5 activity models describe the business processes associated with the architecture, as well as the:

- Relationships or dependencies among the business processes.
- Information exchanged between business processes.
- External interchanges (from/to business processes that are outside the scope of the model).

An operational activity is a logical process, specified independently of how it is carried out. To maintain this independence from implementation, logical nodes in OV-2, Operational Node Relationship Description, are used to represent the structure which carries out the operational activities. Operational activities are realised as functions in the SV-4, Functionality Description; i.e. the OV-5 describes that "what" which is mapped to the SV-4 that defines the "how".

In the Operational Viewpoint, the OV-5 complements the OV-2. OV-2 focuses on the nodes, and the OV-5 focuses on the operational activities undertaken by those nodes. Consequently, the OV-2 and OV5 are usually developed together in an iterative fashion.

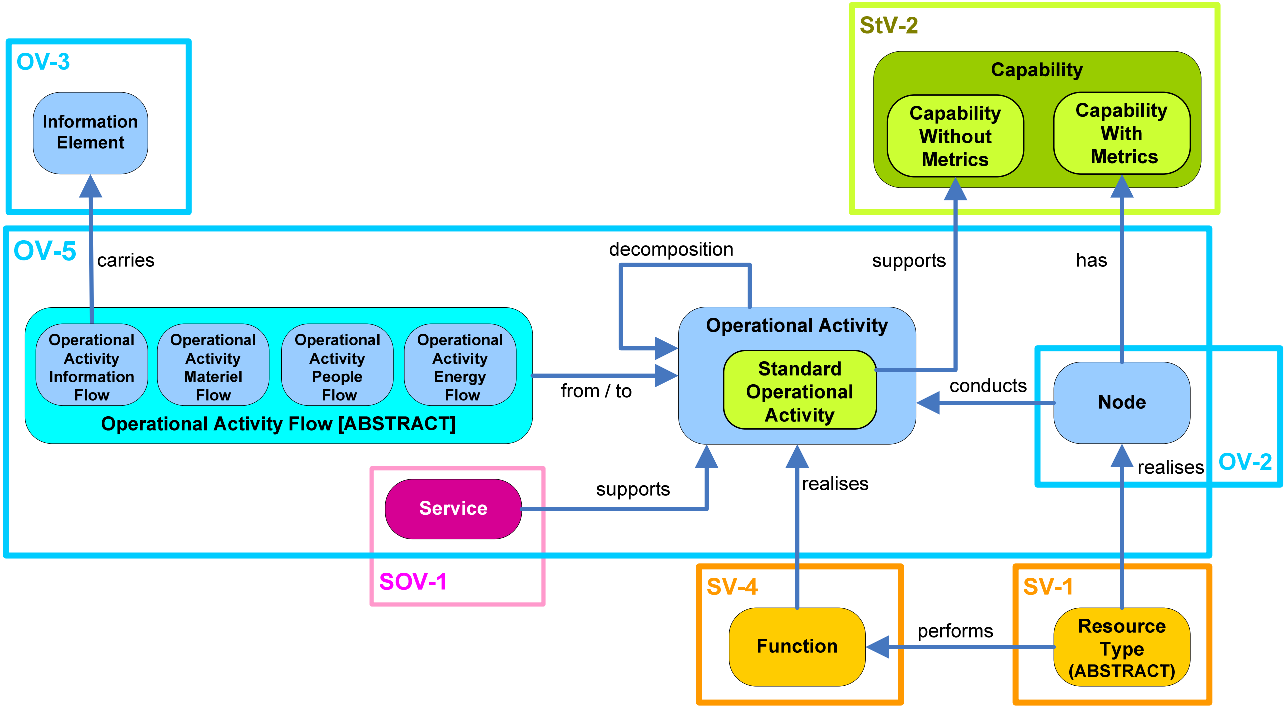
Usage

- Requirements capture (input to URD).
- Description of business processes and workflows.
- Operational planning.
- Logistic support analysis.
- Information flow analysis.
- Support task analysis to determine training needs.

Data objects

The data in an OV-5 can include:

- Operational activities.
- Standard operational activities (originating in StV-6).
- Operational Activity Flow Objects
- 'Swimlanes' (each associated with a node).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Hierarchy chart.
- IDEF0 activity model.
- BPMN diagram.
- UML activity diagram.
- UML activity diagram (with swimlanes).

Detailed Product Description

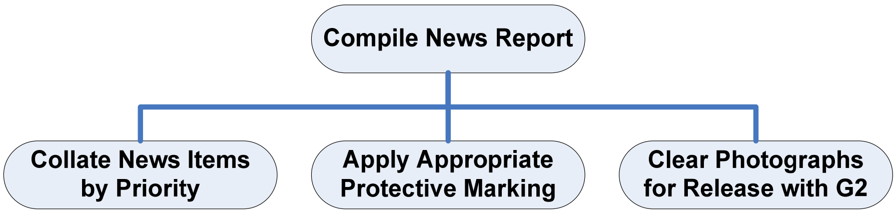
The OV-5 describes the activities that are normally conducted in the course of achieving a mission or a business goal. It describes operational activities (or business processes) and the input and output flows between those activities.

The activities described in an OV-5 may be standard operational activities which are defined in StV-6, Standard Operational Activities to Capability Mapping. Standard operational activities are those defined in doctrine, but which are not tailored to a specific requirement, i.e. they may be used across multiple logical architectures.

There are two basic ways to depict activity models:

- The activity hierarchy shows activities depicted in a tree structure and is typically used to provide a navigation aid.
- The activity flow diagram shows activities connected by information flow arrows.

The OV-5 activity hierarchy chart helps provide an overall picture of the activities involved and a quick reference for navigating the OV-5 input/output flow model.



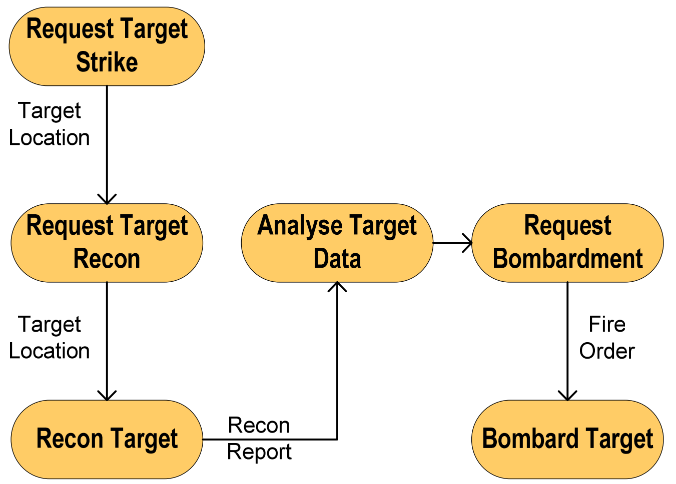
Example OV-5 Activity Hierarchy

The OV-5 activity flow diagram shows activities related by flows. Input / outputs of operational activities relate to information elements of OV-3, Operational Information Exchange Matrix, and are further characterised by the information exchange attributes described in OV-3. The information elements may be further described using OV-7, Information Model.

The operational activities in an OV-5 are undertaken by nodes from the corresponding OV-2. Consequently, the level of detail and decomposition in the OV-5 will be aligned with the complexity of the relationship of the nodes in the OV-2.

Operational activities may consume or produce information. When these cross node boundaries they are carried by information exchanges shown in the OV-2. In this way an OV-5 can contribute to IER analysis.

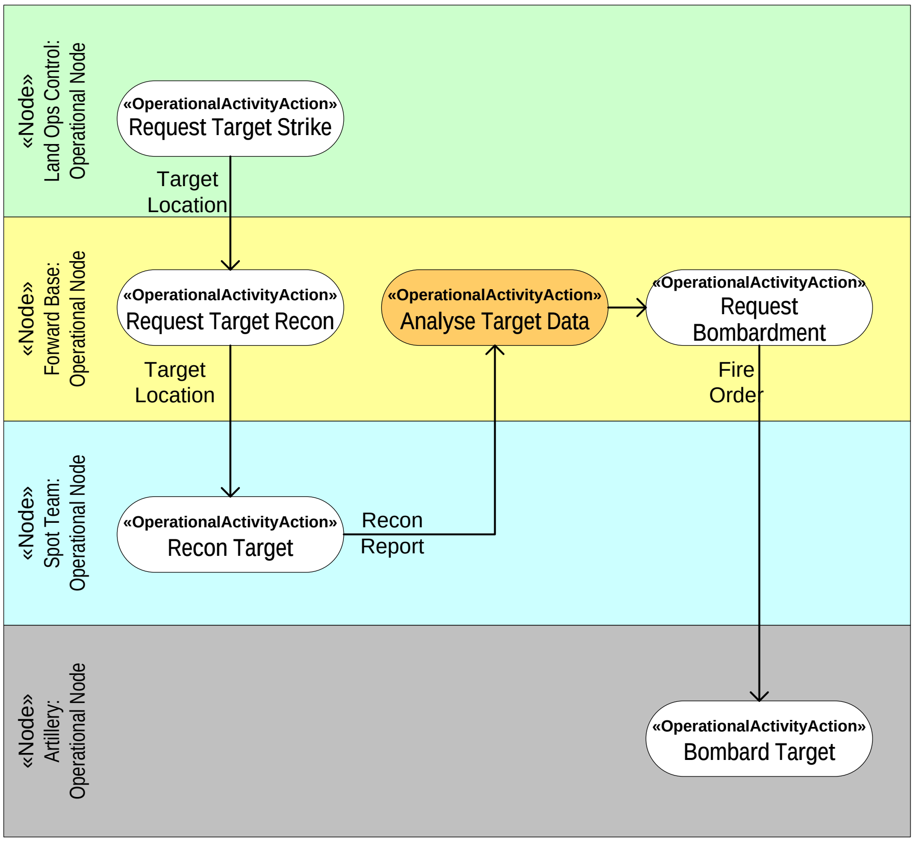
Annotations to the activities may also identify the costs (actual or estimated) associated with performing each activity.



Example OV-5 Flow Diagram

The business rules that govern the performance of the activities can be keyed to each activity - the business rules may be described in OV-6a, Operational Rules Model.

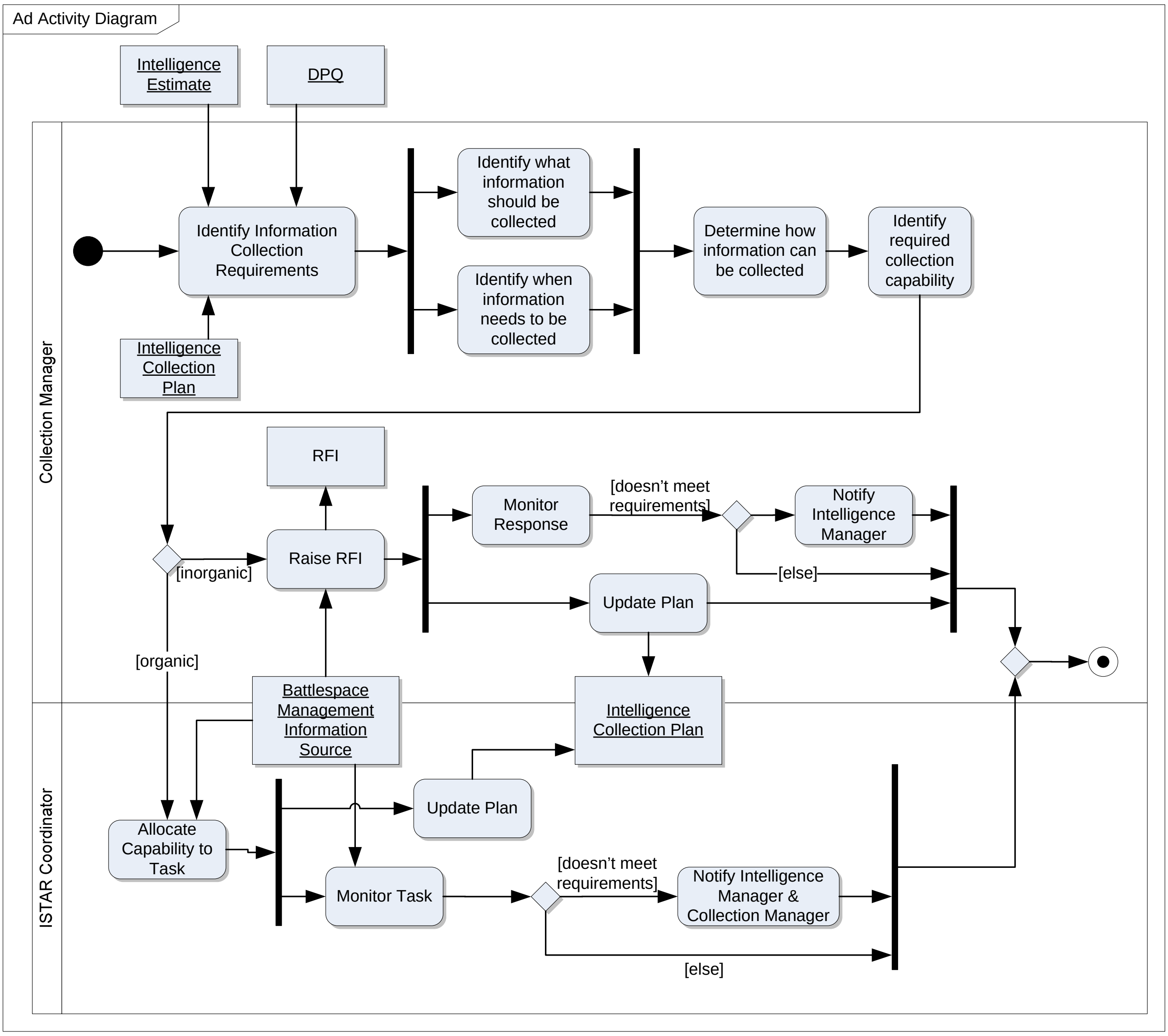
In addition, a process flow model may be annotated with the names of the nodes responsible for conducting those activities, in swimlanes.



Example OV-5 Flow Diagram with Swimlanes

Alternatively, operational activities can be annotated (eg via the mechanism arrow in an IDEF0 diagram) with the corresponding node from OV-2.

If the Unified Modelling Language (UML) method is used, then the activity models can contain decision points and branching.



Complex OV-5 Flow Diagram with Swimlanes (UML)

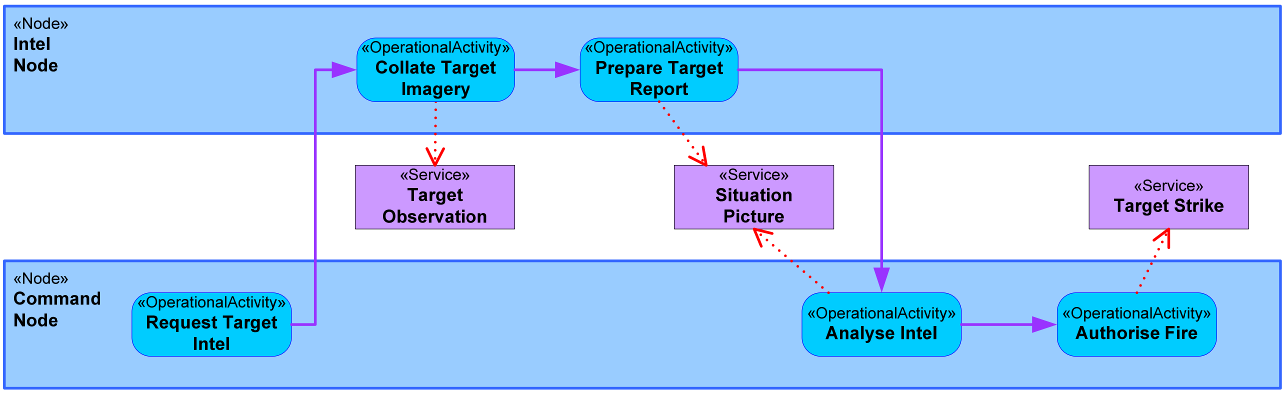
If the Integration Definition for Function Modelling (IDEF0) method is used, the activities also show controls (factors that affect the way that the activity is performed) and may show mechanisms (the resources, including nodes, that perform the activity). While some may illustrate corresponding systems as mechanisms in this model, the reader is cautioned that the introduction of system data early in the development of the OV may result in limiting system design and implementation decisions.

The OV-5 may be used in conjunction with OV-6c, Operational Event Trace Diagram, to specify the sequence in which information exchanges take place. From a modelling perspective, operational activities can be designated as 'acting upon' particular information entities. This relationship between activities and information entities is different to the input / output flow relationship described above. This is intended to address information management types of activities where the information entity is the subject of some management action but is not necessarily part of an input-output activity flow.

As with OV-2 and OV-3, flows on an OV-5 may also carry materiel, human resources or energy.

Service Oriented Architectures

If the architect is developing a Service Oriented Architecture (SOA), an OV-5 may be used to show which services are required to support the conduct of operational activities. This type of view is commonly termed a "service orchestration diagram", because it helps define what services are needed to support an operation and when they are needed.



OV-5 Diagram with Services

Introduction to OV-6a, OV-6b and OV-6c

Many of the critical characteristics of architecture are only discovered when the behaviour of the architectural elements is modelled. OV-5 provides a functional description of this behaviour. OV-6 augments this functional description with rules, states and sequences.

OV-6 consists of three views. The first (OV-6a) is not strictly a behavioural view – it specifies operational rules, which may be behavioural, or may simply be non-functional constraints. OV-6b describes the typical states a node may have and the possible transitions between those states. OV-6c augments the OV-3, Information Exchange Matrix, by outlining the sequence in which information exchanges take place between nodes.

The OV-6 views describe logical rules, states and sequences – i.e. they are specified independently of any given solution. SV-10 provides the solution-specific equivalent to OV-6, detailing the rules, states and sequences that derive from the OV-6 for a specific physical architecture.

OV-6a - Operational Rules Model

An OV-6a specifies operational or business rules that are constraints on the way that business is done in the Enterprise.

Background

At a top level, rules will at least embody the concepts of operations defined in OV-1a, High Level Operational Concept Graphic. These will also provide guidelines for the definition of more detailed rules and behavioural definitions that will be captured as the architecture is developed. (Rules can also be shown as constraints on other diagrams)

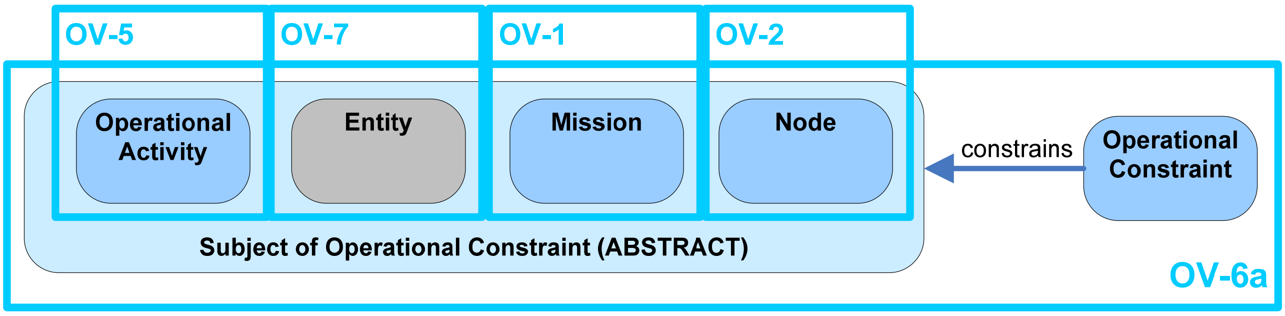
Usage

- Definition of doctrinally correct operational procedures.
- Definition of business rules.
- Identification of operational constraints.

Data objects

The data in an OV-6a can include:

- Operational constraints.
- Nodes.
- Operational Activities.
- Missions.
- Entities (from OV-7, Information Model).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Structured Text.
- UML diagram with associated UML constraints.

Detailed Product Description

The OV-6a specifies operational or business rules that are constraints on the way business is done in the enterprise. Whilst the other OVs describe the structure and operation of a business, for the most part they do not describe the constraints and rules under which it operates.

At the mission level, OV-6a may be based on business rules, such as those contained in doctrine, guidance, rules of engagement, etc. At lower levels, OV-6a describes the rules under which the architecture or its nodes behave under specified conditions. Such rules can be expressed in a textual form, for example:

> "*If (these conditions) exist, and (this event) occurs, then (perform these actions).*"

These rules are contrasted with the business or doctrinal standards themselves, which provide authoritative references and provenance for the rules (see TV-1, Technical Standards View).

The rules captured in OV-6a are logical[5] whereas constraints that are specific to resources are defined in SV-10a, Resource Constraints Specification. OV-6a rules can include such guidance as the conditions under which operational control passes from one entity to another or the conditions under which a human role is authorised to proceed with a specific activity.

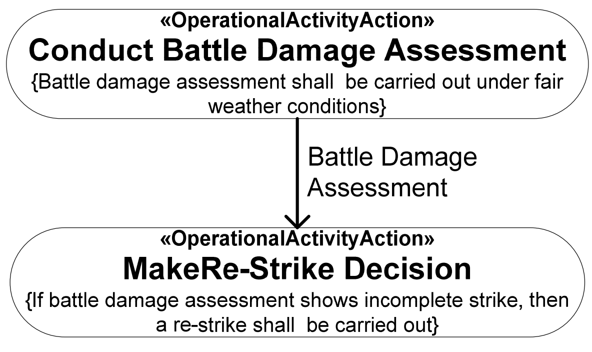
Rule ID	Applies to	Rule Specification
R1	All	All communications shall be encrypted to TS level according to CESG guidelines
R2	Conduct BDA (Operational Activity)	Battle Damage Assessment shall be carried out under fair weather conditions
R3	Make Re-Strike Decision (Operational Activity	If Battle Damage Assessment shows incomplete strike then a re-strike shall be carried out

Operational Rules Example (Structured Text)

From a modelling perspective, operational constraints may act upon nodes, operational activities, missions and entities (OV-7). Consequently, OV-6a rules may be associated with activities in OV-5 and it is often useful to overlay the rules on an OV-5, Operational Activity Model (See diagram – right).

In this example, a rule "battle damage assessment shall be carried out under fair weather conditions" is shown linked to the "Conduct BDA" activity in the OV-5.

OV-6a can also be used to extend the capture of business requirements by constraining the structure and validity of OV-7, Information Model, elements.



Overlaid Operational Rules Example (OV-5)

Detailed rules can become quite complex, and the structuring of the rules themselves can often be challenging. MODAF does not specify how OV-6a rules will be specified, other than being written in English.

[5] In abstract rather than physical terms, so as to be solution independent.

OV-6b - Operational State Transition Description

The OV-6b is a graphical method of describing how a node changes in response events that affect it.

Background

The OV-6b specifies the states a node can have, and the possible transitions (i.e. changes of state) between them. Triggers for state changes may also be defined.

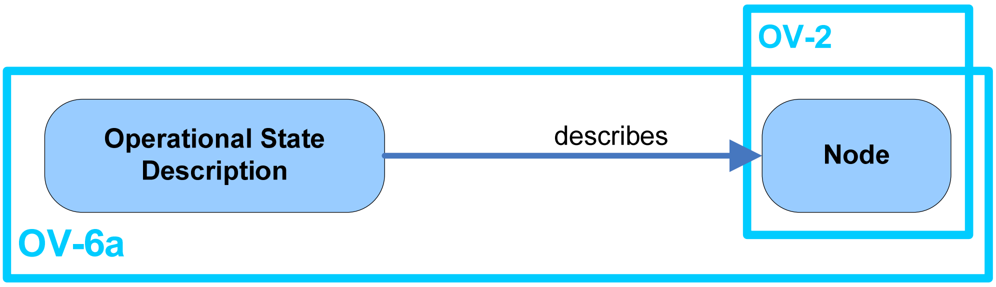
Usage

- Analysis of business events.
- Behavioural analysis.
- Identification of constraints (input to SRD).

Data objects

The data in an OV-6b can include:

- States (each associated with a node).
- State transitions (each associated with an event).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Topological (Connected Shapes).
- UML state diagram.

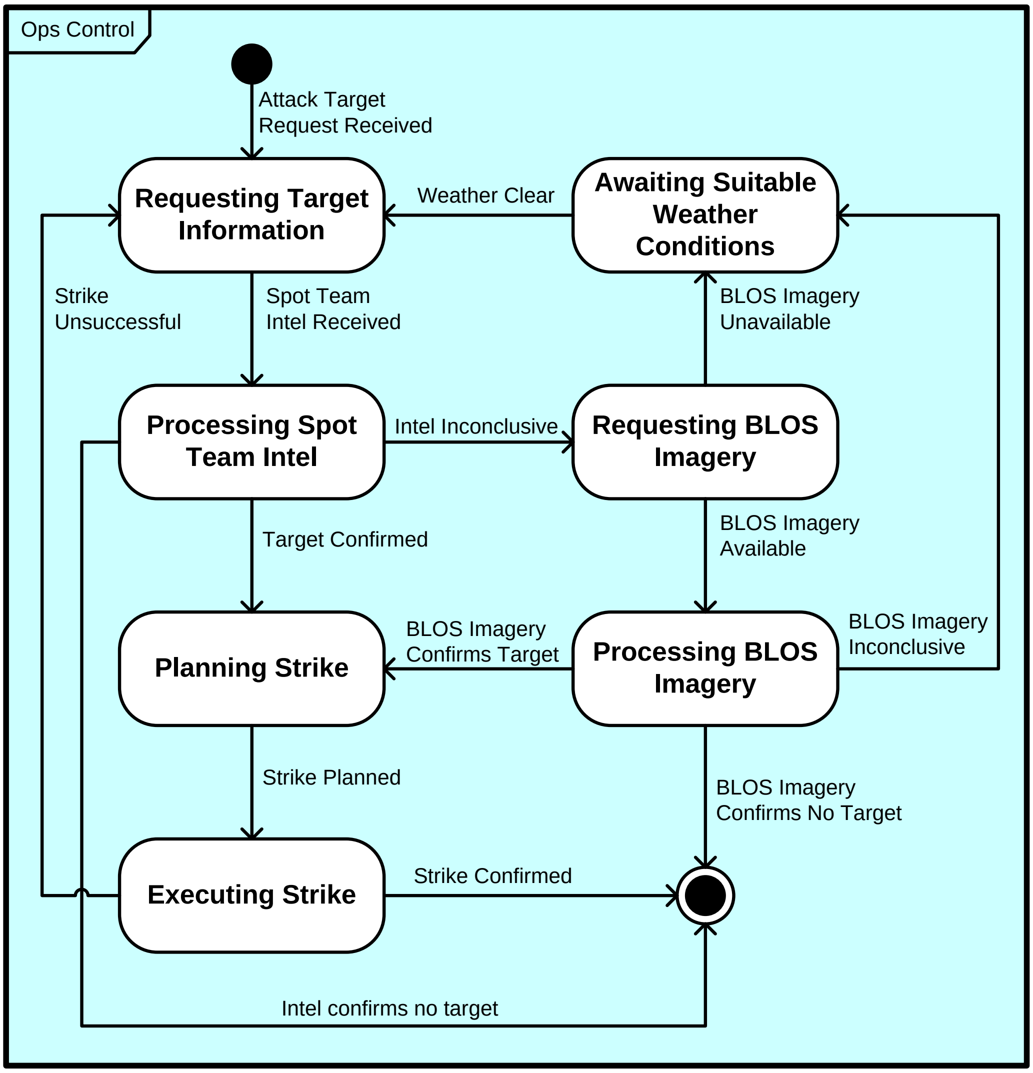
Detailed Product Description

An OV-6a depicts states and state transitions for a node.

The Figure below, based on a State chart Diagram, provides a template for a simple OV-6b. The black dot and incoming arrow point to initial states (usually one per diagram), while terminal states are identified by an outgoing arrow pointing to a black dot with a circle around it. States are indicated by rounded corner box icons and labelled by name or number and, optionally, any actions associated with that state. Transitions between states are indicated by one-way arrows labelled with an event/action notation that indicates an event-action pair, and which semantically translates to: when an event occurs, the corresponding action is executed.

State transitions

The following figure provides an example view product.



Example OV-6b

States in an OV-6b may be nested. This enables quite complex models to be created to represent operational behaviour.

OV-6c - Operational Event-Trace Description

The OV-6c provides a time-ordered examination of the exchanges between participating nodes in a particular scenario. There may be multiple OV-6c products to represent different scenarios, and each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

Background

Operational Event-Trace Descriptions, sometimes called sequence diagrams, event scenarios or timing diagrams, allow the tracing of interactions between nodes in a scenario or critical sequence of events. The node interactions usually correspond to flows of information, but may optional describe flows of energy, materiel or people. The OV-6c, along with OV-6b, Operational State Transition Description, and OV-5, Operational Activity Model, specify the behaviour of nodes.

OV-6c is valuable for increasing the level of detail from the initial operational concepts, and can help define node interactions and operational threads. It can also help ensure that each participating node has the necessary information it needs at the right time in order to perform its assigned operational activity.

Usage

- Analysis of operational events.
- Sequences of interactions between nodes.
- Behavioural analysis.
- Identification of non-functional user requirements (input to URD).
- Operational test scenarios.

Data objects

The data in an OV-6c can include:

- Lifelines (each associated with a node).
- Messages
- Information Elements

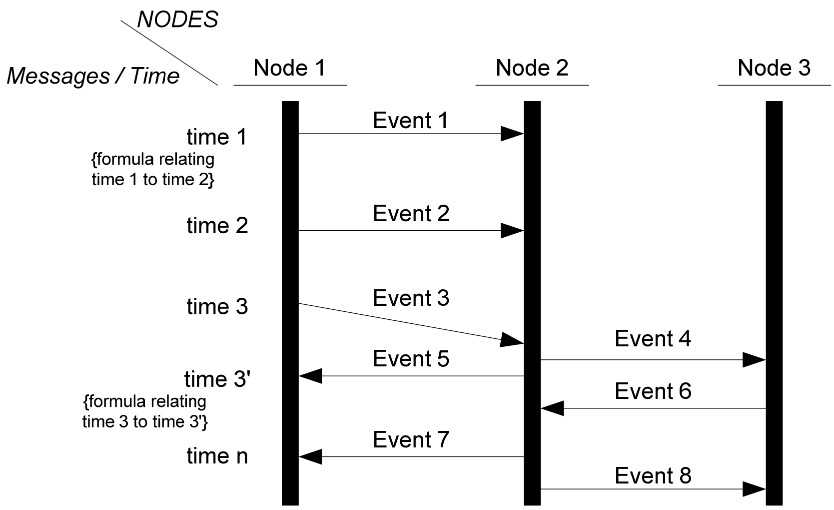
Representation

- UML sequence diagram.

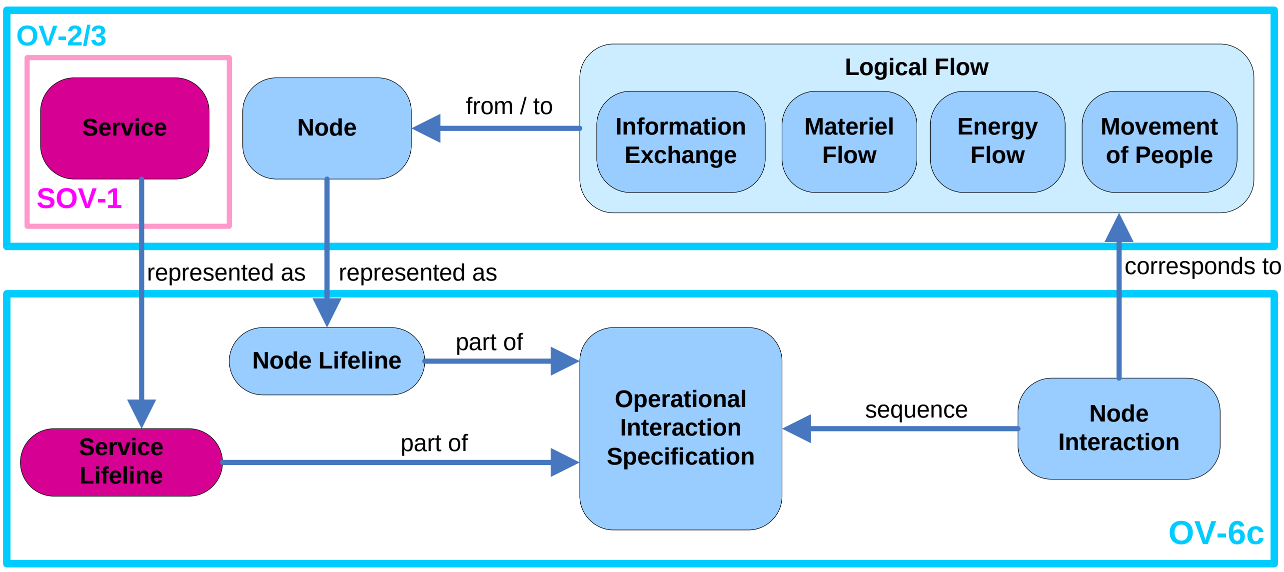
Detailed Product Description

OV-6c allows the tracing of interactions in a scenario or critical sequence of events and can be used by itself or in conjunction with OV-5, Operational Activity Model, and/or an OV-6b to describe the dynamic behaviour of nodes in a mission or operational thread.

The diagram below shows the components of an OV-6c. The items across the top of the diagram are nodes. Each node has a vertical timeline associated with it. Specific points in time can be labelled running down the left-hand side of the diagram. Directed lines between the node time lines represent interactions (e.g. information exchanges) between nodes, and the points at which they intersect the timelines represent the times at which the nodes become aware of the events.

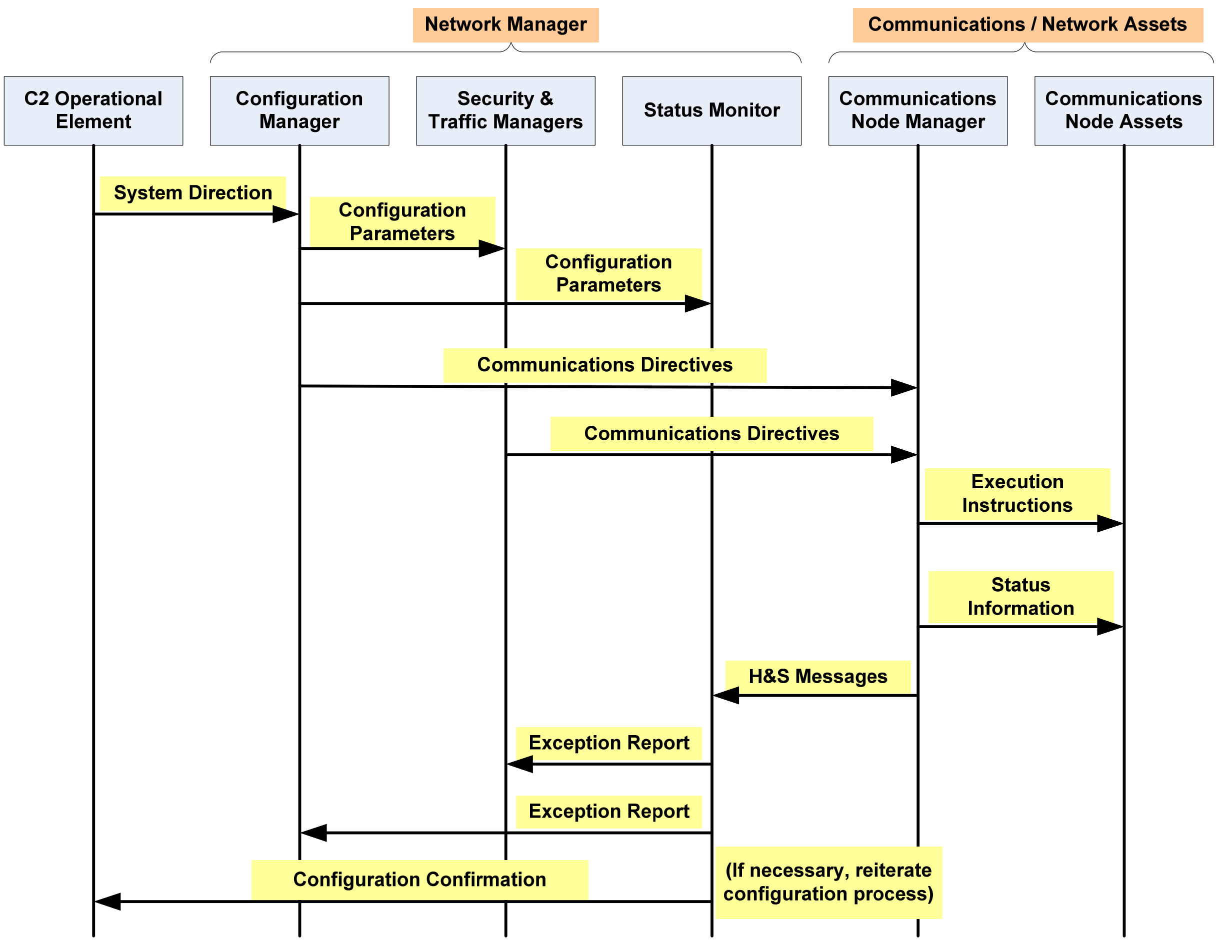


OV-6c Schematic



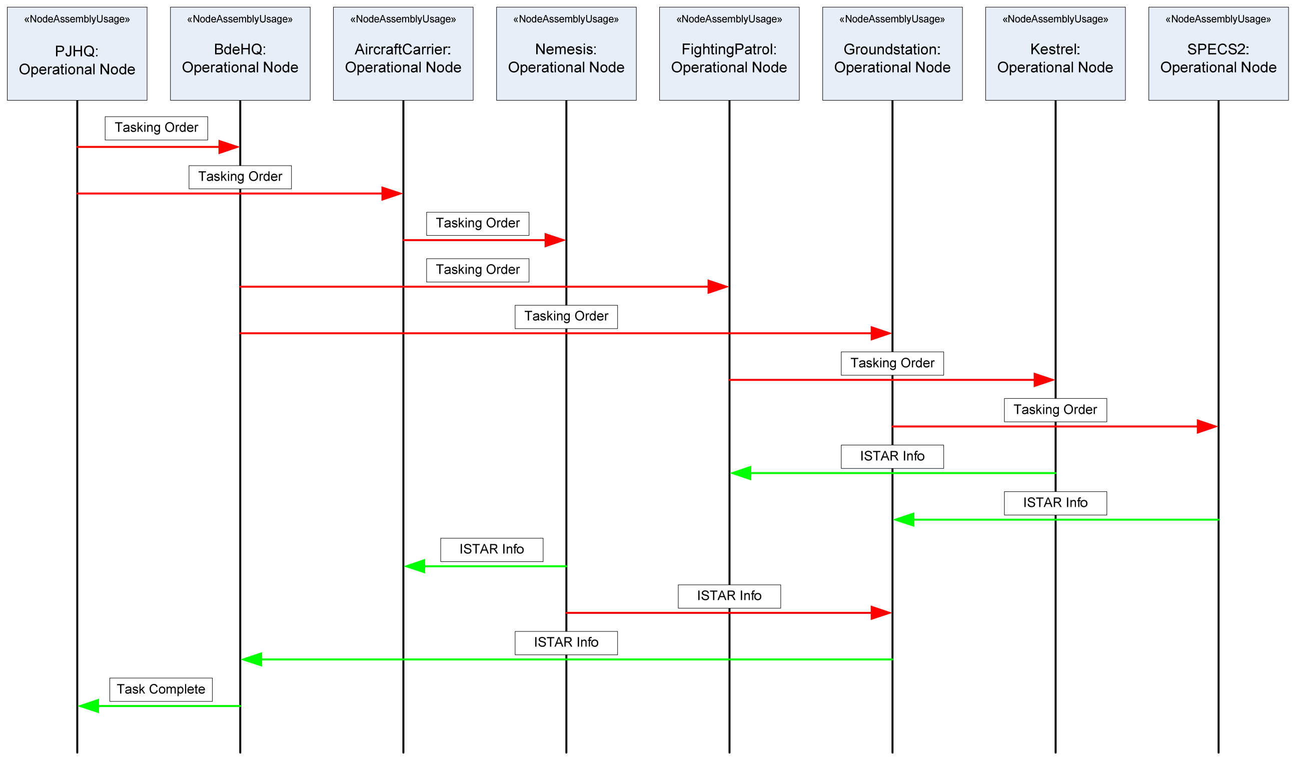
Relationships between Key Data Objects (Simplified from M3)

MODAF does not generally endorse a specific modelling methodology, however in the case of OV-6c, UML sequence diagrams seem the most appropriate. If UML cannot be used, an OV-6c may be developed using any modelling notation that supports the layout of timing and sequence of activities along with the information exchanges that occur between Nodes for a given scenario. Different scenarios will be depicted by separate diagrams.



OV-6c Example)

The figure on the following page shows an example OV-6c Product using a UML sequence diagram.

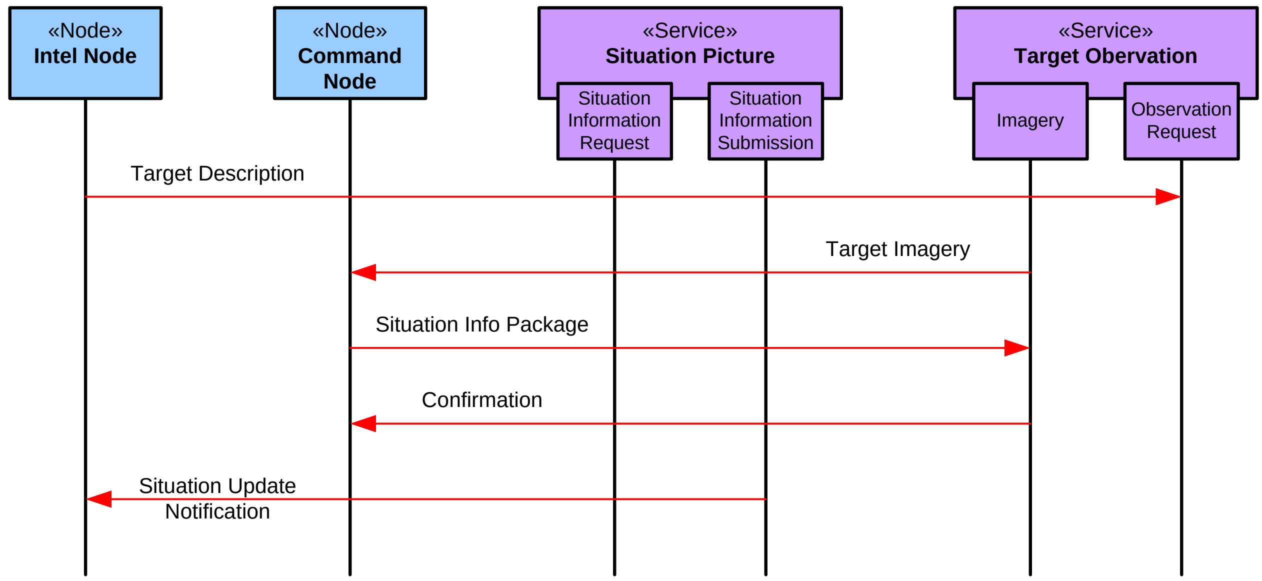


OV-6c Example

The information content of 'messages' that connect life-lines in an OV-6c view product may be related, in modelling terms, with the information flows, from OV-3 and OV-5, and information entities, from OV-7, Information Model.

Service Oriented Architectures

If the architect is developing a Service Oriented Architecture (SOA), an OV-6c product may be used to show the sequence of interactions required to support operational activities. An "SOA OV-5" shows which services support which operational activities. However it may be useful to show how those services are required to interact in order to support operations. An "SOA OV-6c" shows lifelines for services, and the sequence of interactions between those services:



Services Example of OV-6c

OV-7 - Information Model

The OV-7 addresses the information modelling perspective of an operational architecture.

Background

The OV-7 is used to document the business information requirements of the enterprise. It describes the information that is associated with the information exchanges specified in OV-3, Information Exchange Matrix. An OV-7 defines a logical data model consisting of entities, attributes and relationships. The entities in the model define the structure of information elements that are exchanged between nodes.

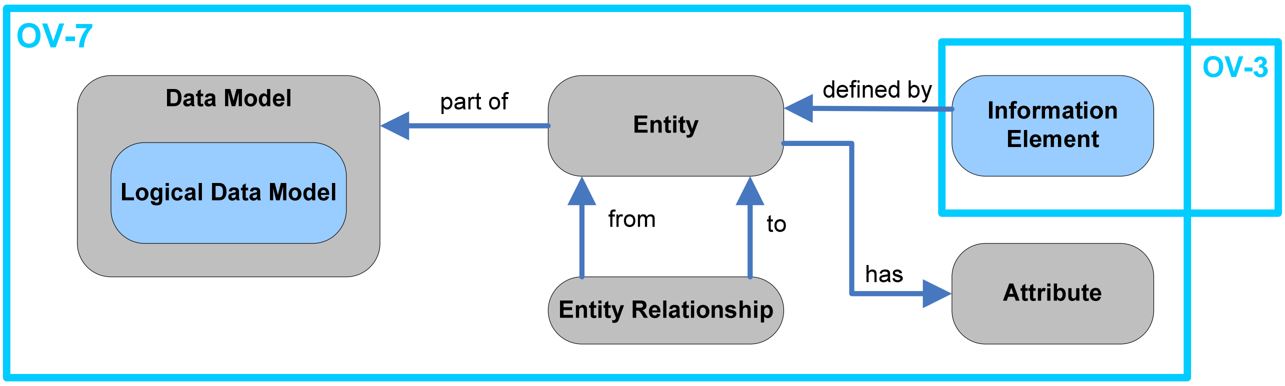
Usage

- Information architecture.
- Logical data modelling.

Data objects

The data in an OV-7 can include:

- Logical Data Model.
- Entity.
- Attribute.
- Entity Relationship.



Relationships between Key Data Objects (Simplified from M3)

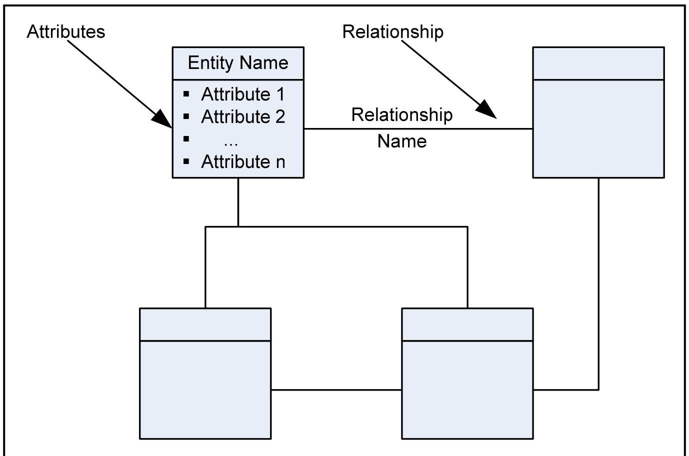
Note that an Entity within an OV-7 may define the structure of an Information Element in an OV-3, Operational Information Exchange Matrix.

Representation

- Entity-Relationship diagram (e.g. IDEF1X)
- UML class diagram.

Detailed Product Description

The OV-7 describes the logical data model for the architecture. It provides a definition of information types (entities), their attributes or characteristics, and their interrelationships.

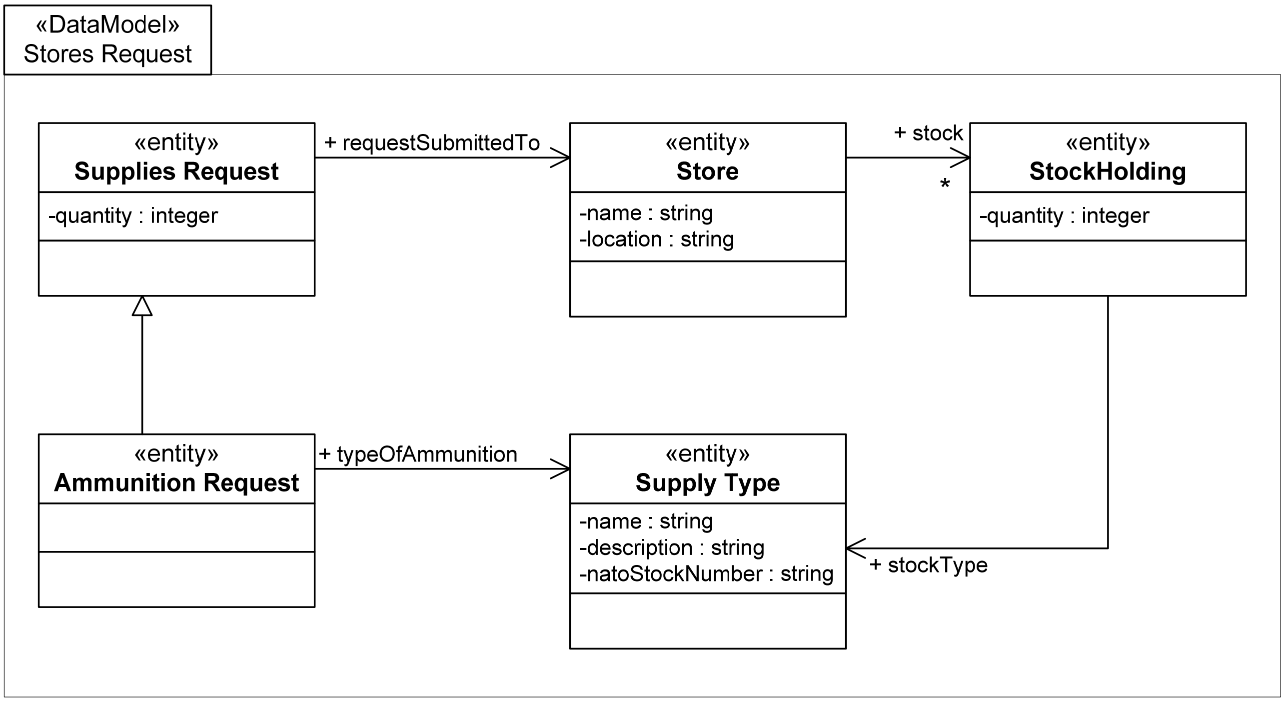


Generic example of OV-7 Information Model

Note that MODAF refers to 'information' in the Operational Viewpoint and 'data' in the System Viewpoint. The intention of this is that OV-7 describes information of importance to the business (e.g. information products that might be referred to in doctrine, SOPs etc) whereas SV-11 describes data relevant at the system level.

OV-7 defines each kind of information class associated with the architecture domain, mission, or business as its own entity, with its associated attributes and relationships. These entity definitions specify the structure of OV-3 information.

Usually, an entity-relationship notation will be used for OV-7, but it is also possible to use UML (with appropriate M3 stereotypes)for OV-7. An example UML usage is shown below.



Example OV-7 (UML)

UNCLASSIFIED

Representing Information Exchange Requirements using the MOD Architecture Framework (MODAF)

Version 1.1

12 November 2007

Ian Bailey

This document outlines the preferred approach to developing operational architectures in MODAF. It stresses a logical approach (which has been greatly clarified in v1.1 of MODAF) and identifies a simple iterative process for establishing information exchange requirements (IERs)

It is aimed at anyone wishing to use MODAF when developing IERs, or anyone who may need to read or review a MODAF architecture. The document was produced to clarify the preferred approach, which will help to improve consistency across MOD's architectural products.

Bringing service to life serco

This document was produced by Model Futures Limited under sub-contract to Serco Consulting for the UK MOD.

Introduction

There has always been some confusion about how the OV-2 and OV-5 views in MODAF should be used. These views form part of the Operational Viewpoint in MODAF. OV-2 represents the logical structure of the enterprise[i], and OV-5 presents a very simple activity-based representation of the behaviour of the enterprise. Together, these views provide a useful tool for specifying and analysing information exchanges.

MODAF version 1.1 has sought to tighten-up their usage, and their relationship to other views, however, consistent Enterprise Architecture is not guaranteed by the existence of a framework alone. There is also a need for a consistent approach in the use of the framework. The need for a methodology for MODAF has been debated in the past, and is still being debated. What is being suggested here is not a methodology as such, but just a clarification about how some of the key MODAF views are related.

Background

There were some significant changes to MODAF with the release of version 1.1. At first sight they seem minor, but they were intended to clean up some of the inconsistencies in the framework and to ensure that architects were clear about how certain views were to be used.

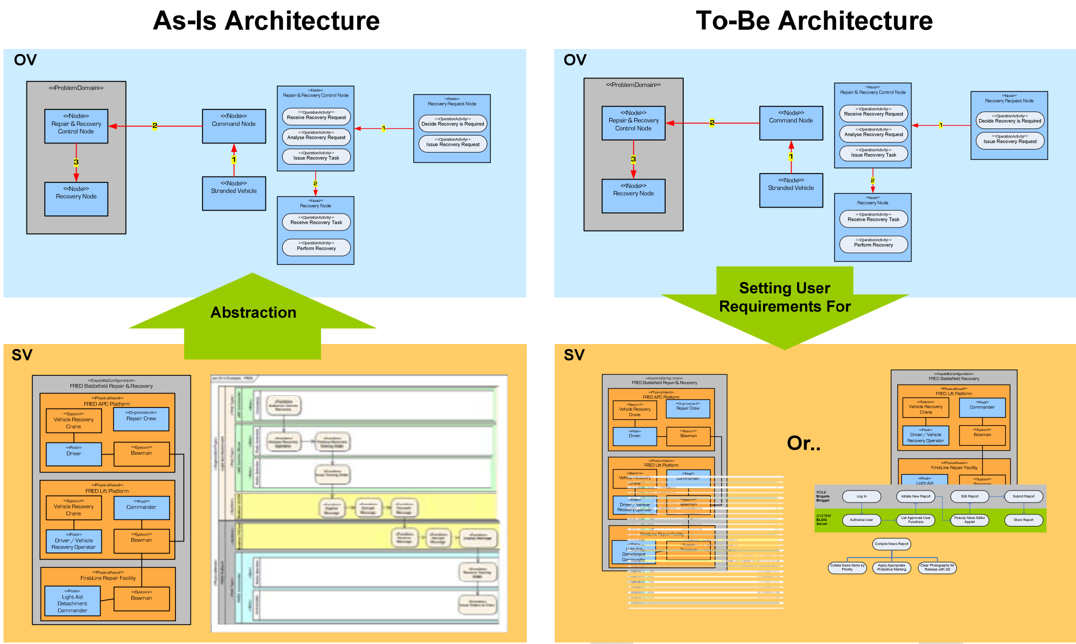
One of the reasons they seem minor is that the views retain the same codes (OV-2, OV-5, SV-1, SV-4, etc.). There was probably a strong case for renaming the Operational Views (OVs) and the System Views (SVs). The OVs aren't necessarily operational in the military sense of the word, but are logical representations of required or existing capabilities in context of each other. Similarly, the SVs describe systems in the broadest sense – i.e. they include humans. The original names were kept in order to maintain consistency with MODAF v1.0 and DoDAF, but it is quite useful to think of the OVs as "Logical" and the SVs as "Physical".[ii]

Previous to these changes in MODAF v1.1 some architects were interpreting the operational views to be business models and the systems views to be technical models. This was never the intent, nor was it the original intent of DoDAF. This tighter adherence to logical and physical distinctions means that architects need to think clearly before deciding if their process model is an OV-5 or an SV-4 (even if it is a business process model – remember the SVs now model human factors). Similarly, it is not acceptable to just treat the OV-2 nodes as platforms or organisations.

Logical Architecture

It is important to stress that the intent of a MODAF operational architecture is to present an abstracted, logical view of the enterprise. OV-2 describes *nodes* and the relationships between them (needlines). *Nodes* are simple agents that perform *operational activities* which, in turn, are the subject of OV-5. The OV-2 & OV-5 serve two purposes depending on the type of architecture[iii]:

- In an as-is architecture, they are used to abstract away physical complexity to show the key logical features of an enterprise. The actual implementation of this functionality is described elsewhere (in MODAF System View products). An as-is logical architecture allows a consistent view of enterprise functionality to be maintained, so that various what-if SV designs can be trialled in change programmes – i.e. it provides a logical baseline against which the impact of change can be measured.
- In a to-be architecture, they provide a logical description of what the enterprise shall do, and they key actors needed in order to do it. For this particular logical requirement, there may be many possible solutions, and these may be traded-off against each other (and may be described using MODAF System View products).



This then begs the question; what is a logical architecture? A broad rule of thumb is to consider the *what* rather than the *how*. A logical architecture specifies what the enterprise is, does, or is required to do, and what the logical actors in that process are. The actors (Nodes) may correspond directly to individual organisations, people or systems, but this does not have to be the case. For example, in a to-be ISTAR[iv] architecture, we may specify only one Operational Node that conducts the observation activities, but in reality this is implemented by a variety of different sensors and human observers.

How far the architecture is logically abstracted is really a matter of fitness for purpose, and MODAF allows for multiple levels of decomposition in OV-2 and OV-5 (i.e. the architect can produce different, but related OV products for different stakeholders). However, in general the overall level of detail should be kept as simple as possible – so as to abstract away complexity in an as-is architecture, and to allow the necessary freedom of design expression in a to-be architecture.

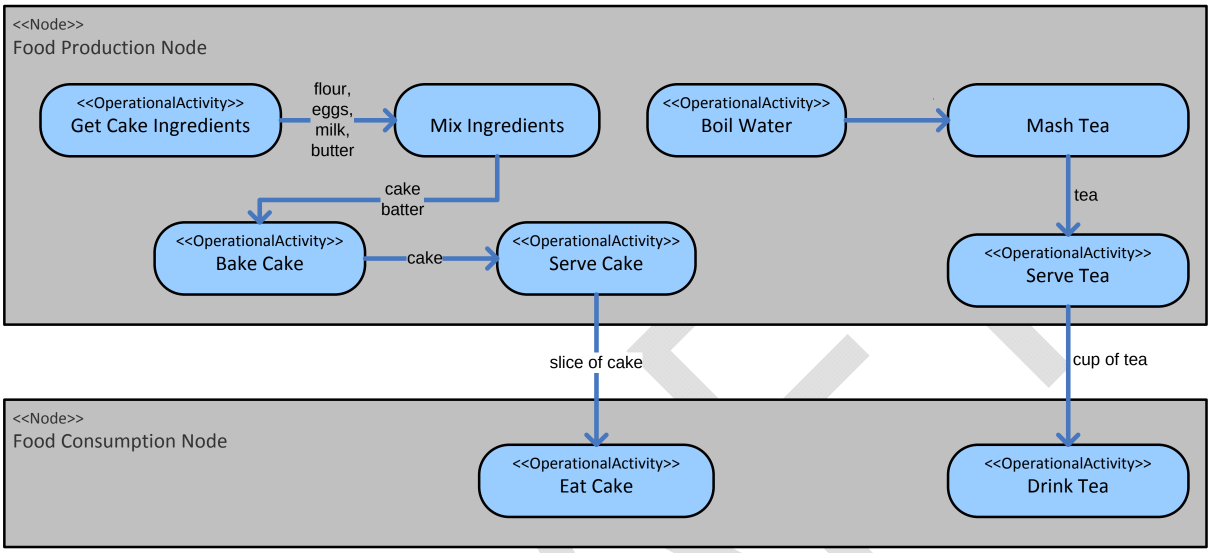
Information Exchange Analysis

Note: *The term "Information Exchange" is used here rather than Information Exchange Requirement (IER). An IER tends to suggest a requirement for some future capability rather than something that exists. MODAF architectures are either to-be or as-is. Therefore an information Exchange in a to-be architecture will indicate a requirement, whereas in an as-is it illustrates and existing capability.*

A logical architecture is commonly used to identify key information exchanges in an enterprise – these may be actual exchanges (in an as-is architecture) to requirements for exchange (to-be). The logical approach is particularly useful in information exchange analysis, as it allows the architect and other stakeholders to clearly see the information exchanges without the complexity of implementation. A real implementation of an information exchange may be very convoluted, requiring relays, junctions, redundant connections, etc. and although this is useful information to a technical architect, it doesn't always help in getting the message over to managers and users.

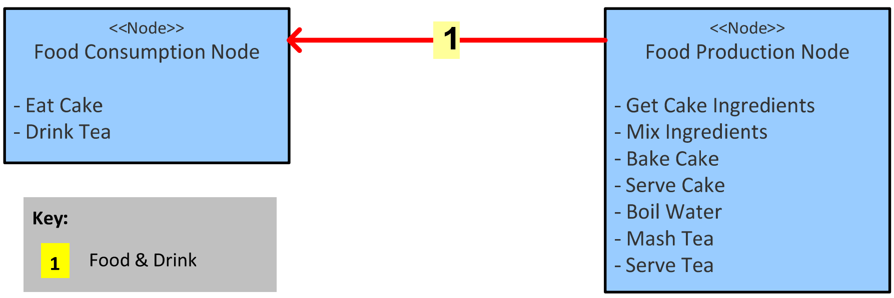
In developing a logical architecture in MODAF, there is always something of a chicken-and-egg aspect to OV-2 and OV-5 – do we specify our processes first, or our nodes[v]? Most experienced DoDAF and MODAF architects will say that actually the OV-2 & OV-5 are both developed together, and there are usually a few iterations as new Nodes are introduced, processes established, etc. Any information exchange analysis, therefore, will also be an iterative one. The basic approach in MODAF (and in DoDAF) is to model the flows of information between Operational Activities. This can be quite a difficult task, but the best way to think about it is to consider what information an actor (Operational Node) would require in order to carry out a task (Operational Activity). It should then be reasonably clear which other tasks produce that information and so a flow can be established. Again, this is an iterative process – the architect may realise that there is no activity which produces the information required, and so this must be modelled.

Once the information flows between activities are in place, and it is clear which nodes conduct the activities, the flows across nodal boundaries become apparent – in the example below, the flows are not information (they're *food* and *drink*) but the principle is the same:



We've used tea and cake here rather than information for two main reasons 1) it is universally understood; 2) the cake baking example is usually used in process modelling classes to teach the principles. In the example, there are clearly two flows which cross the nodal boundaries, hence there are two information (or in this case food) exchange requirements. Note that the model says nothing about how any of the activities take place (e.g. do we use a kettle or a pan to boil the water ?), which keeps everything suitably abstract for analysis purpose. The description of how the water is to be boiled will be specified in an SV-4 functional model, with the accompanying SV-1 resource interaction model. By hiding the complexity of how it is actually done (or could be done), the key information exchanges are exposed in sharp relief, and the need to highlight the exchanges should be a guiding principle in developing OV-2 and OV-5 products.

An information flow between activities (official MODAF name: Operational Activity Flow) has greater significance when it crosses nodal boundaries. When this occurs, it signifies some need to exchange information between people, systems, sites, etc. An OV-2 highlights these key exchanges by simplifying the OV-5 view down to nodes and "needlines":



Clearly, the OV-2 provides similar information to the OV-5, but hides the activity flows. In this case, the cup of tea and slice of cake are bundled into one "Food & Drink" needline. How many information flows are bundled into a needline is again a design decision for the architect – OV-2 is all about presenting a complex architecture in a simple way. The usual practice, however is to bundle those flows that carry information that is likely to be related. *Note that it is optional to overlay nodes on an OV-5, and also optional to show activities on an OV-2.*

In most cases, information flows / exchanges will be finer grain than needlines – i.e. needlines will tend to be collections of information exchanges. In some cases though, a needline may carry only one information exchange. MODAF's meta-model relates needlines and information exchanges by introducing the idea of an information element. These elements are carried by information flows between activities. Selected elements can then be identified as part of the information exchange, and needlines can be used to group the exchanges. The OV-3 itself is a tabular view, showing the needlines, information exchanges and information elements. The tabular approach also allows the architect to define properties of the information exchanges such as information assurance attributes, bandwidth requirements, etc.

Needline Nº	IE Nº	From Act	To Act	Bandwidth Needed	Security Classification	etc.	etc.
1	1	Serve Cake	Eat Cake		Unclassified		
1	2	Serve Tea	Drink Tea		Top Secret		

Conclusion

The key to developing OV-2 and OV-5 products is to keep the modelling as abstract as possible. In so doing, this creates a structure against which information exchanges can be modelled without the need to show the complexity of the underlying physical architecture (or, in the case of a to-be architecture, to prevent "solutioneering" in the user requirement).

The combination of nodes and behaviour (operational activities) shouldn't be any great surprise to a systems engineer, though it may be unfamiliar to requirements managers who tend to work with textual specifications for user requirements. Not all systems engineers will be accustomed to working at this level of abstraction though, and it requires a certain level of discipline to prevent the operational architecture becoming physical. It is very easy to represent systems, people, organisations, etc. as nodes, but this doesn't really add anything above and beyond the physical architecture (which is handled by the Systems Views in MODAF).

Information flows are specified between operational activities. Those activities are performed by nodes. When a flow exists between activities conducted by different nodes, there will be an information exchange. In a to-be architecture, this represents a potential[vi] information exchange requirement. In an as-is architecture, the information exchanges and needlines are used to simplify the presentation of the key exchanges that take place in the enterprise.

Following these simple guidelines will help result in consistent MODAF architectures, where the operational views do more than simply re-present the physical architecture.

[i] For the purposes of this document (and for Enterprise Architecture in general), an enterprise may be of any size from a small project to a large multinational company or Govt department. It is a common mistake to assume enterprise architecture is about modelling the whole business. EA is about bringing together the different business and technical strands to provide an "enterprise view" – i.e. not one that is purely business or IT.

[ii] It is useful to put the MODAF OVs in context of the Strategic Views (StV) and SVs. The MODAF Capability Taxonomy (StV-2) describes capabilities in general. It does not specify the implementation of the capabilities, nor how they are deployed. A capability in MODAF is simply a statement of some ability to deliver an effect. The architect may specify metrics for the level to which the effects are achieved (e.g. a maximum rate of advance for a ground manoeuvre capability). StV-6 specifies standard (i.e. doctrinal) processes, and as with StV-2, these are specified independently of implementation or deployment (so think JETL more than METL). To be of any use, the capabilities and processes must be shown logically in context of a particular scenario, and this is the job of the OV. The nodes in an OV-2 may be instances of one or more capability, and the activities in an OV-5 may well be defined originally in an StV-6 . Finally, the logical nodes specified in the OV-2 are detailed as resources (systems, people, platforms, etc.) in the SV-1, and the logical activities presented in OV-5 are put in context of the resources that deliver them in an SV-4 functional model.

[iii] Enterprise architectures usually either describe the existing state of an enterprise (as-is) or some future, required state of the enterprise (to-be)

[iv] Intelligence, Surveillance, Target Acquisition, and Reconnaissance

[v] Nodes in MODAF are logical agents (or actors) that conduct operational activities – they may be realised by people or systems, or combinations of the two.

[vi] Potential, because in developing the physical architecture, it may be more economic to co-locate the information provider and consumer than implement the necessary communications interface.

System view (SV) viewpoint

A description of and guidance for the use of the MOD Architecture Framework (MODAF)Â System View Viewpoint.

The System Views (SVs) are a set of views that describe resources that realise capability. The Systems Views describe resource functions and interactions between resources and can also provide detailed system interface models. Note that these views address the involvement of humans in both the operation of systems and in carrying out functions in their own right. The SVs can be used to specify solutions to requirements specified in the Operational Viewpoints (OVs), or simply to provide more detail to the logical OV architecture.

- **SV viewpoint PDF [1.5 MB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

ViewpointsÂ andÂ views
Parent page

Views summary documents

All views viewpoint

Strategic view viewpoint

Operational view viewpoint

Technical standards view viewpoint

Acquisition view viewpoint

Service oriented view viewpoint

MODAF -Â detailed guidance

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page not yet rated

The MODAF System Viewpoint

Viewpoint Summary

The System Viewpoint provides a perspective that describes the resources (see section below) that realise capability and/or implement services. It can also be thought of as a Solution or Specification Viewpoint as it specifies a requirement for a system or presents a solution without delving too deep into the design elements of the system.

The System Viewpoint consists of 17 System Views (SVs) that describe resource functions, interactions between resources, and system interfaces. In addition, SVs depict the involvement of humans in the operation of systems, and as resources that carry out functions.

One of the primary uses of the SVs is in the development of system requirements that satisfy user requirements; i.e. the SVs can be used to specify solutions to requirements identified in the Operational Views (OVs). They can alternatively be used to provide more detail to the logical architecture depicted by the OVs.

Introduction to Resources in MODAF

Since version 1.1 of MODAF, the SVs have included aspects of human factors – i.e. the views are not limited to just depicting technical systems. In this sense, the SVs depict "Systems" in the broadest sense of that term.

The SVs are primarily concerned with the MODAF Meta Model (M3) concept of *Resource* and *Functions* that resource performs. Resources may be one of:

- Artefact – a physical resource that is man-made or manufactured.
- Software – executable computer code or a fragment thereof.
- Organisational Resource – a human resource, which may be one of:
  - Organisation Type – a type of organisation, where organisation is defined to be any group of people brought together for a purpose.
  - Post Type – a type of responsible office within an organisation which may be occupied by a person or another organisation.
  - Role Type – a type of role a human resource may have in an organisation or function.
- Physical Architecture – a composite of any of the above resources that forms a re-usable architecture. This may also be:
  - Capability Configuration – a composite of resources that, with the appropriate doctrine, can deliver a capability.
  - Service Implementation – a composite of resources that can deliver a Service.

These resources can be assembled, and re-used from architecture to architecture. When one resource is part of another, MODAF requires that the architect specifies the context in which it is a part:

- Artefacts may be *parts* or *systems* in another Artefact.
- Artefacts may be *platforms*, *systems,* or simply *physical assets* (i.e. serving no function) in a Physical Architecture.
- Software may be *hosted software* on an Artefact, or a *software component* in other Software.
- Physical Architectures may be *used configurations* in other Physical Architectures.
- Organisational Resources may be *human resources* in a Physical Architecture.
- Organisation Types may be *sub-organisations* in other Organisation Types.
- Post Types may be *posts* in Organisation Types.

- Role Types may be *roles* in Post Types.

Views

There are seventeen views, including sub-views, which make up the System Viewpoint:

SV-1 – Resource Interaction Specification

An SV-1 specifies the composition and interaction of resources[1].

Background

The SV-1 links together the operational and systems architecture views by depicting how resources are structured and how they interact in order to realise the logical architecture specified in an OV-2, Operational Node Relationship Description. An SV-1 may represent the realisation of a requirement specified in an OV-2 (i.e. in a to-be architecture) and, consequently, there may be many alternative SV suites that provide candidate solutions that realise the operational requirement.

The SV-1 depicts interactions between resources. A resource interaction is a simplified representation of a pathway or network, usually depicted graphically as a connector (i.e. a line that can be labelled with supporting information). Note that interactions between systems (Artefacts used as systems) may be further specified in detail in the SV-2, Systems Communications Description series, and SV-6, Systems Data Exchange Matrix.

Resources may be decomposed in SV-1 to any level (i.e. depth) that the architect sees fit. When one resource is part of another, the architect must specify the context in which the part is used.

Usage

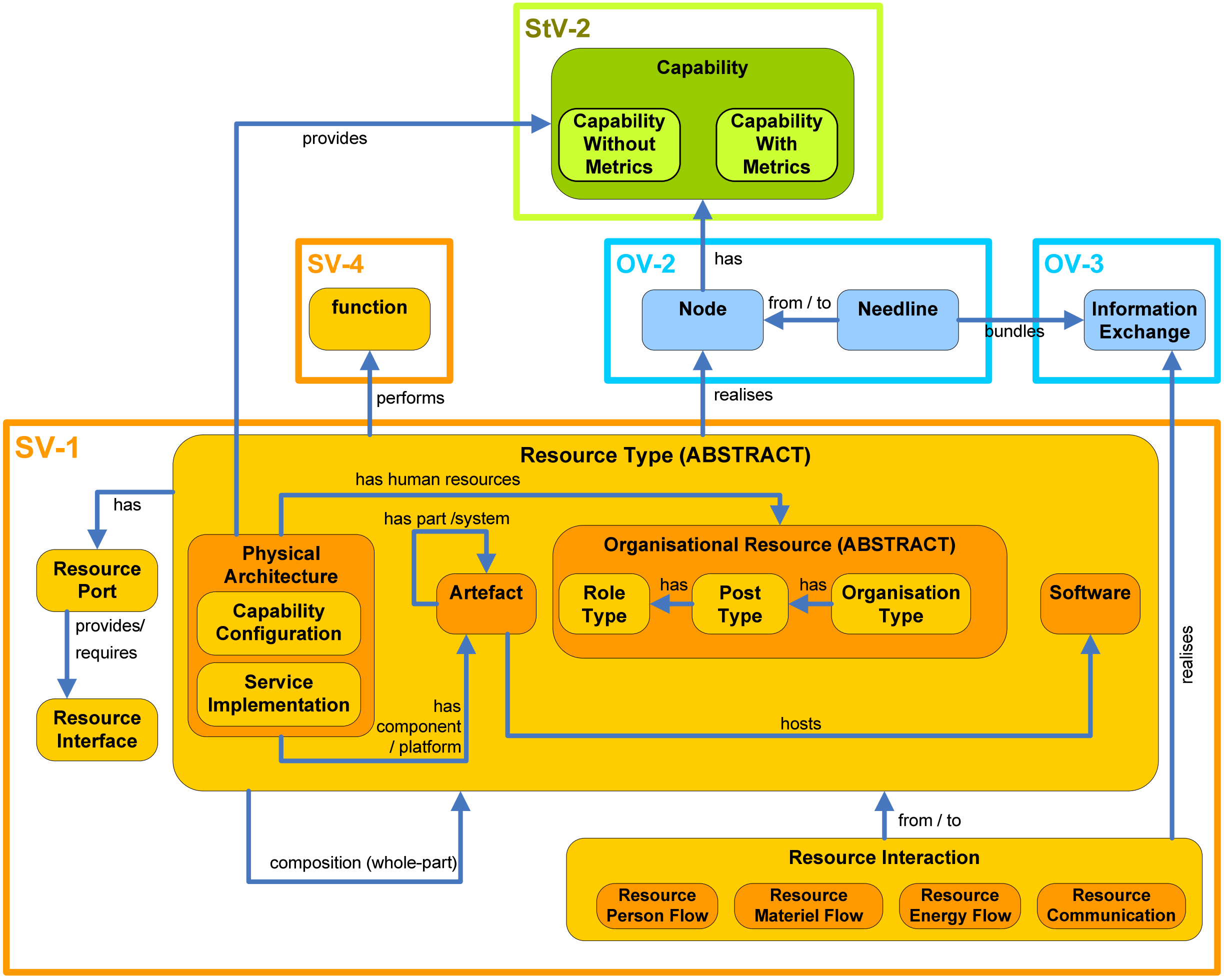
- Solution specification.
- Definition of solution options.
- System Requirements specification.
- Interface requirements capture.
- Capability integration planning.
- System integration management.
- Operational planning (capability configuration definition).

Data objects

The data in an SV-1 can include:

- Artefact.
- Organisational Resource (Organisation Type, Post Type, Role Type).
- Software.
- Physical Architecture (Capability Configuration, System Implementation).
- Resource Composition.
- Resource Interaction (flows of data, materiel, human resources or energy).
- Traceability to Nodes (OV-2) and Capabilities (StV-2)

[1] See Introduction to Resources in MODAF on page 1



Relationships between Key Data Objects (Simplified from M3)

Representation

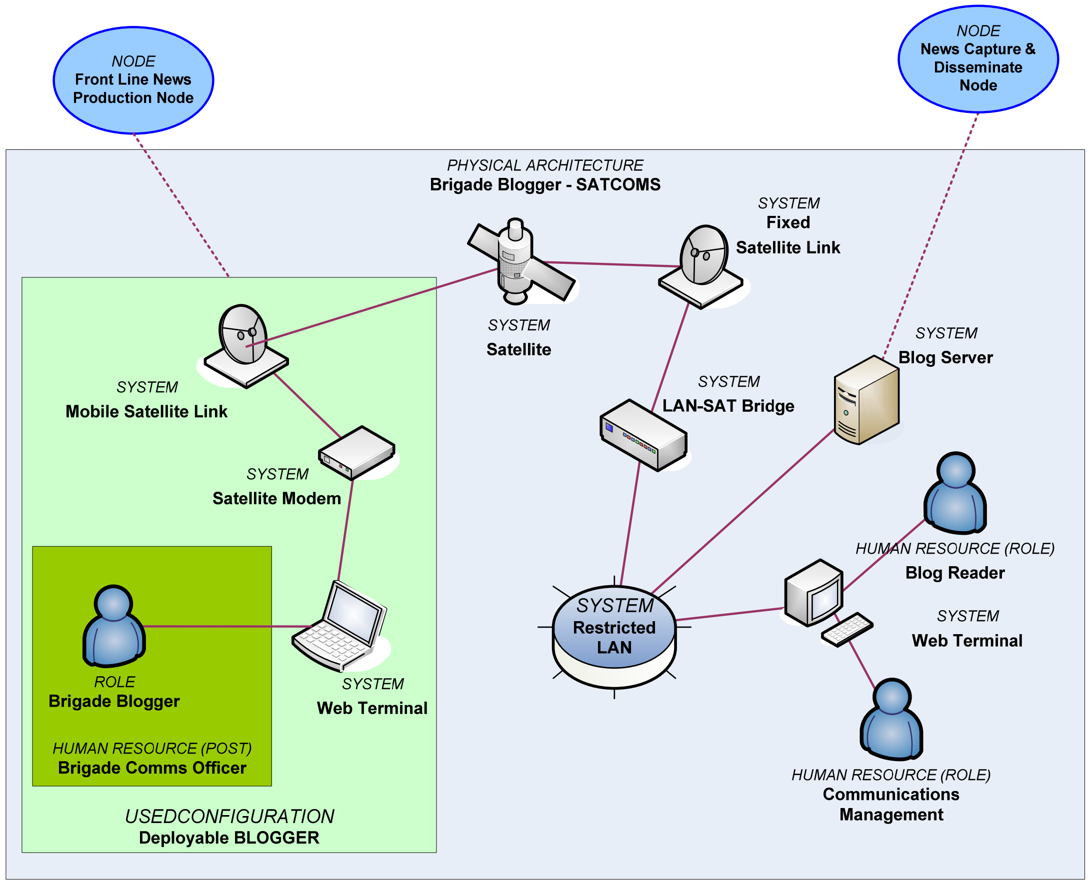
- Topological (connected shapes).
- UML composite structure diagram.
- SysML blocks diagram.

Detailed Product Description

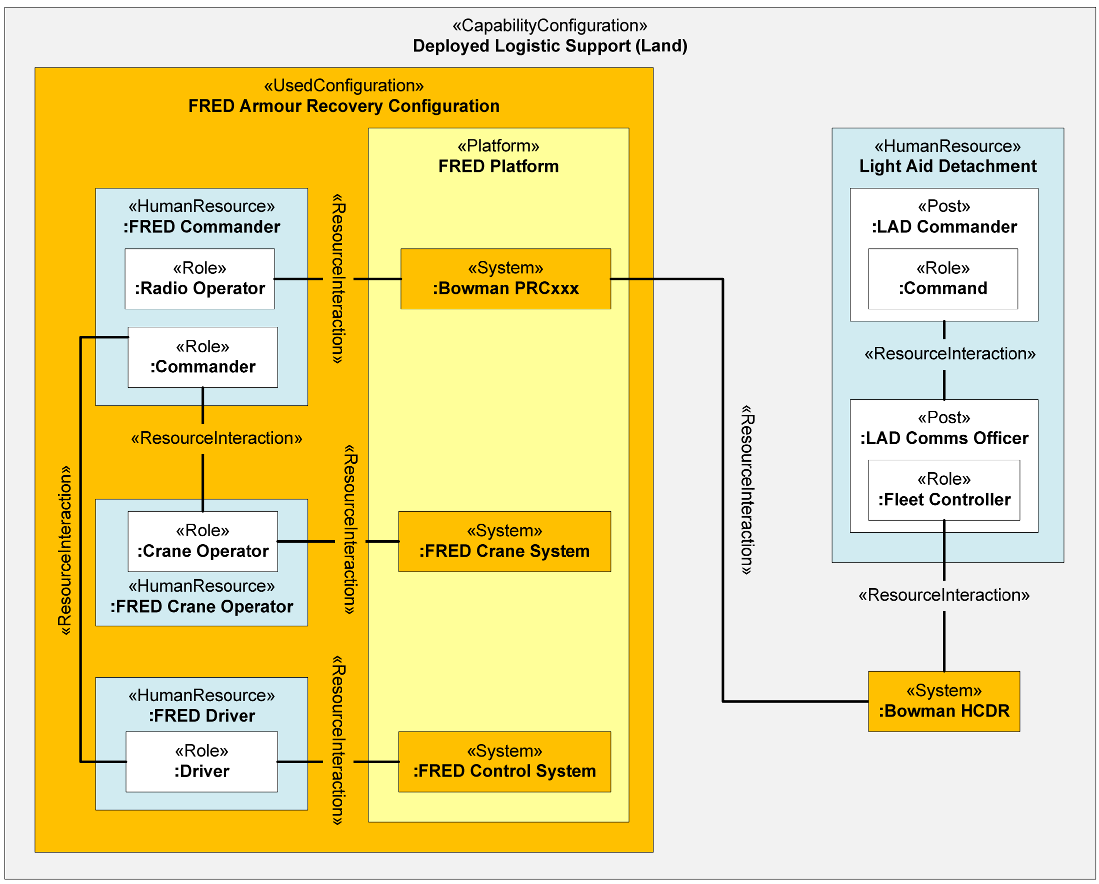
The primary purpose of an SV-1 is to show resource structure; i.e. to identify the primary sub-systems, posts and roles and their interactions. SV-1 contributes to user understanding of the structural characteristics of the capability. Resource structures may be identified in SV-1 to any level (i.e. depth) of decomposition the architect sees fit. An SV-1 may be adorned with nodes originally specified in OV-2. In this way, traceability can be established from the logical OV structure to the physical SV structure.

In its simplest form, an SV-1 can be used to depict systems and sub-systems, and identify the interfaces between them; however, this rarely adds more to that which can be shown in an SV-2, product. The real benefit of an SV-1 is its ability to show the human aspects of an individual architecture, and how these interact with systems. In addition, MODAF has the concept of a ‘capability configuration’ which is used to gather together systems, assets and people into a configuration which can meet a specific capability.

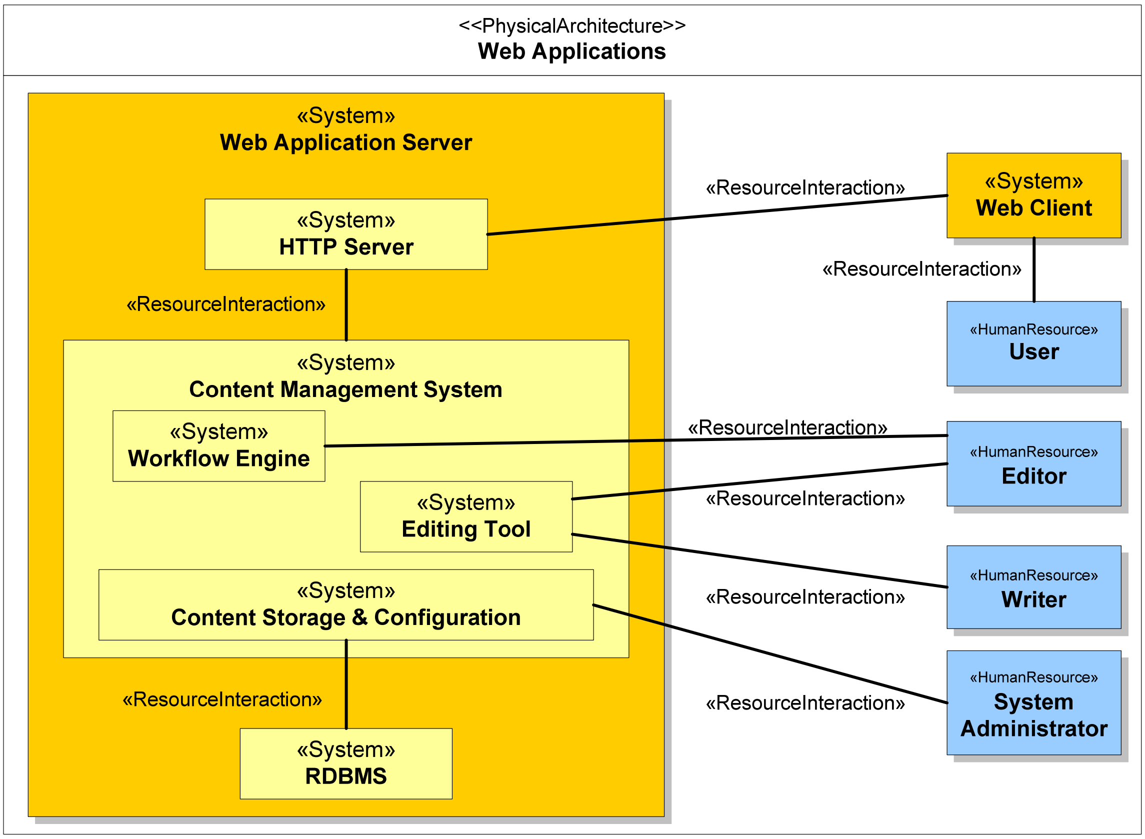
If possible, an SV-1 will show resources and their interactions for the entire architecture on the same diagram. If a single SV-1 is not possible, the structure should be decomposed into multiple SV-1s.



SV-1 Example with elements traced back to logical nodes



SV-1 Example Showing Capability Configuration



SV-1 Example (UML Composite Structure Model)

Functions

MODAF adopts a simplified model where any resource may perform functions. SV-1 and SV-4, Functionality Description, provide complementary representations (structure and function). The functions from an SV-4 can optionally be overlaid on an SV-1.

Interactions in SV-1

In addition to depicting resources and their structure, SV-1 addresses interaction relationships between resources. An interaction, as depicted in SV-1, is an indicator that data passes between one resource and another. In the case of systems, this can be described in further detail in an SV-2b, System to System Port Connectivity Description. Interactions provide a specification for how the exchanges specified in OV-2 needlines are realised. A single needline shown in the OV-2 may translate into multiple interactions.

The actual implementation of an interaction may take more than one form (e.g. multiple physical links). Details of the physical links and communications networks that implement the interfaces are documented in SV-2. Resource Interactions are summarised in an SV-3, Resource Interaction Matrix. If SV-1 is developed as a composite structure model (e.g. in SysML, UML), Resource Ports may be used to convey how interactions are dealt with internal to the resource when the resource has parts. Resource Ports may also specify the interfaces they require or provide. Note that when connecting resources via interfaces and ports, the architecture is tight-coupled. For loose-coupled architectures, a service-oriented approach should be taken (see Service Oriented Views and SV-12, Service Provision and Service Composition).

Interactions between resources need not be restricted to communication of data. An SV-1 may also show interactions where materiel, human resources or energy flow from one resource to another.

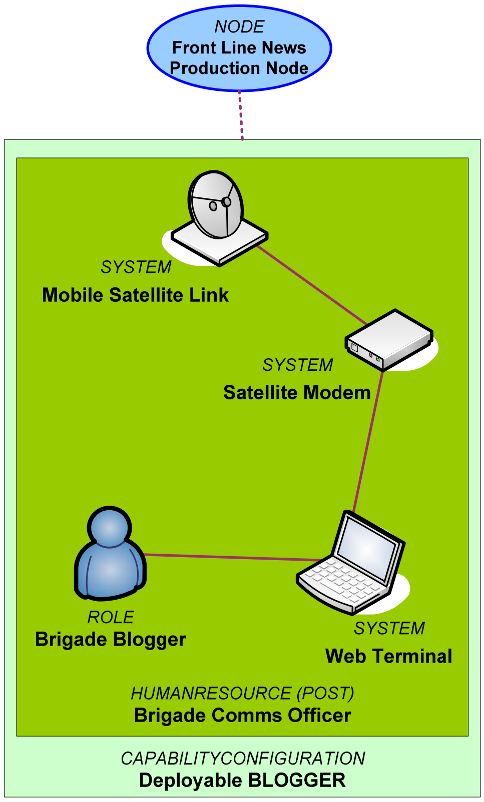
SV-1 in Solution Architecture

A key feature in MODAF is the ability to represent configurations of resources that, when put together with the appropriate functions (e.g. doctrine), provide one or more capability. These are known as 'capability configurations'. A capability configuration is a combination of organisational resources (with their competencies) and equipment (artefacts and software) that combine to provide a capability.

Capability configurations fulfil the operational capability needs and are usually defined to fulfil the requirements associated with nodes (see OV-2). [For more detail, please refer to "MODAF Support to Systems Requirement Definition"[2]].

The example to the right illustrates the relationship between capability configuration and nodes.

Use of capability configurations allows architects to include all of the Defence Lines of Development (DLOD) rather than just systems and platforms. [For more information on this, please refer to the document "MODAF Support to Analysis of Capability Integration in the Context of the Defence Lines of Development" [3].]



Capability Configuration Tracing to OV-2 Node

Fielded Capability

A 'fielded capability' is a particular instance of a capability configuration. For example, a capability configuration may be a Type 45 destroyer configured for an anti-air role, of which HMS Daring will be a fielded capability. Fielded capabilities should be used only when a specific instance of a Capability Configuration is required.

---

[2] http://www.mod.uk/DefenceInternet/AboutDefence/CorporatePublications/InformationManagement/MODAF/UseAndExamplesOfModaf.htm

[3] http://www.mod.uk/DefenceInternet/AboutDefence/CorporatePublications/InformationManagement/MODAF/UseAndExamplesOfModaf.htm

An Introduction to SV-2a, SV-2b and SV-2c v1.2

The SV-2 Systems Communications Description is comprised of a set of 3 views that can provide representation of the communications networks and pathways that link communications systems[4], and provides details regarding their configuration.

The networks and pathways documented through these views represent the physical implementation of the SV-1, Resource Interaction Specification, and the information needlines identified in an OV-2, Operational Node Relationship Description.

The SV-2 view comprises of three views which define the communications links between systems:

- SV-2a System Port Specification – defines the ports on each system, and the communication protocol / hardware stack that is specified or implemented for each of those ports.
- SV-2b System to System Port Connectivity – defines the connections between individual ports and shows the communication protocols and hardware spec used for each connection.
- SV-2c System Connectivity Clusters – defines the bundles of system to system connections that go to make up a connection between the Artefacts that host the connected systems (see SV-1).

The purpose of these views is to provide a comprehensive specification of how systems are connected, what interfaces each system exposes (ports), the hardware interface used and the protocols transmitted across the interface. Key elements are repeated from view to view and are also common to the SV-1. These key elements are:

- Artefacts (used as Systems and Platforms).
- Ports.
- Protocols.
- System Port connections.

SV-2 differs from SV-1 in that it only features Physical Architectures, Software and Artefacts (as systems) – i.e. SV-2 does not feature any organisational resources. SV-2 also provides a great deal more technical detail than SV-1, specifying the protocols implemented by systems and used by the connections between those systems.

It is important to understand the differences between SV-1 and SV-2 to ensure that the correct detail is captured in each view. In essence, the SV-2 expands on the SV-1 by providing more detail of the physical characteristics of interactions between systems. For example, the SV-1 interaction perspective shows a single-line representation of interfaces between nodes, whereas the SV-2 would show a more detailed representation of the communications infrastructure that provides the connections.

[4] Formally, in M3, these are Artefacts that are being used as systems in the context of a given Physical Architecture

SV-2a System Port Specification v1.2

An SV-2a specifies the ports provided by a system, and the protocols used by those ports when communicating with other systems.

Background

An SV-2a provides a specification for each system port that is modelled in the architecture.

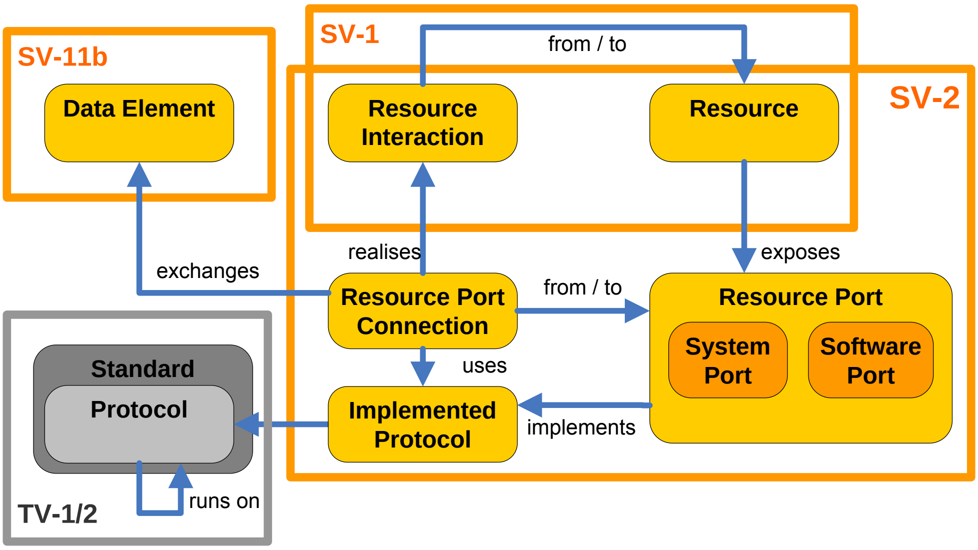
Usage

- Interface specification.
- Identification of applicable protocols.
- Description of system communication paths.

Data objects

The data in an SV-2a can include:

- System.
- System port.
- Protocol.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Topological (connected shapes).
- UML composite structure diagram.
- SysML structural diagram.

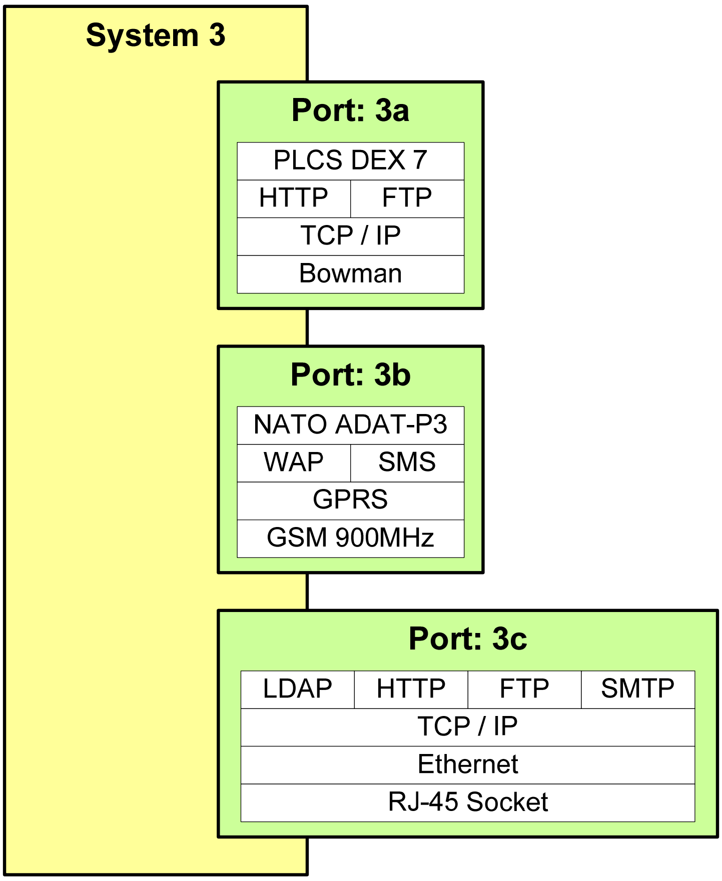
Detailed Product Description

An SV-2a is used to describe the interface protocols and hardware specifications of each port on a system. The view comprises of one diagram for each system in the architecture. Each port on the system is specified in terms of:

- Its name.
- The interface protocols used (e.g. OSI Stack).
- The physical port specification (e.g. the physical element of the stack).

In many cases, a physical port may support more than one protocol in parallel (e.g. a TCP/IP network supporting http, ftp, telnet, etc.). All supported protocols relevant to the architecture shall be shown through the SV-2a for the various systems.

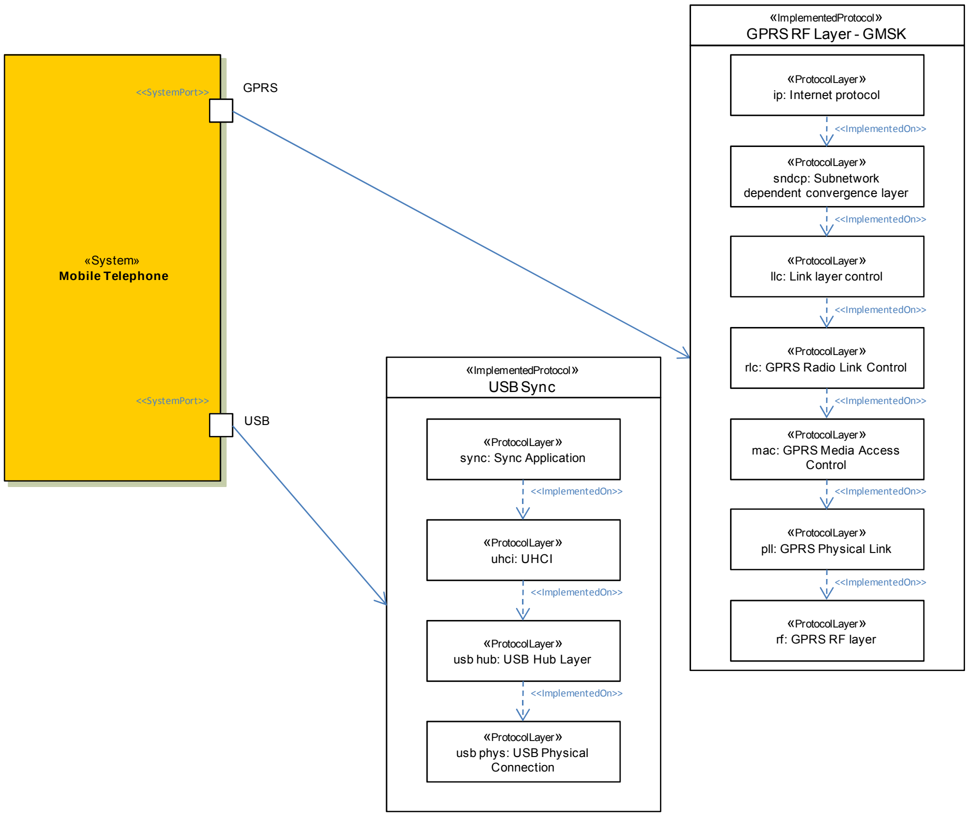
The figure below shows an example port specification port 3a uses a physical port to support HTTP and FTP over TCP/IP.



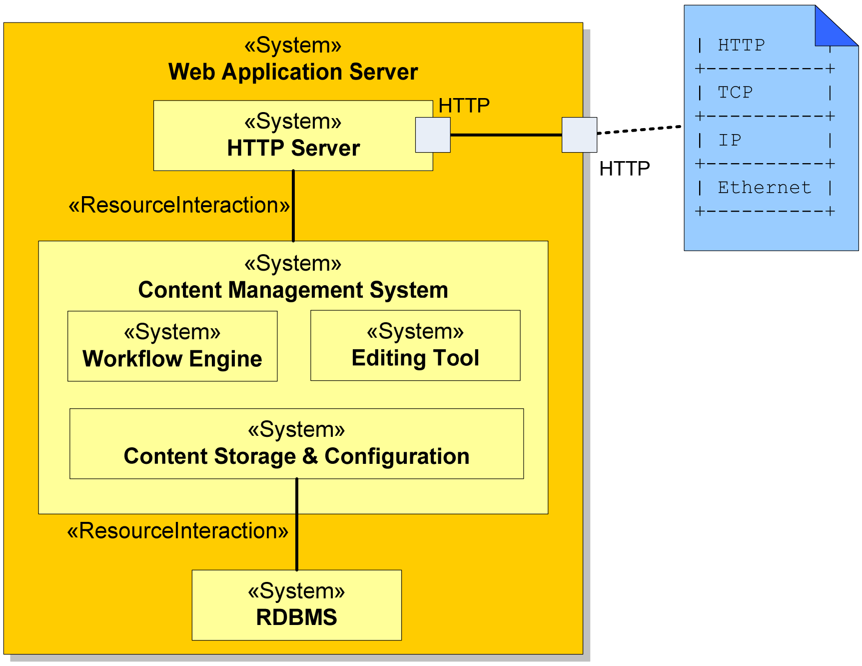
Non-UML Example Showing Alternative Protocols at Different Levels in the Stack

If a port supports a particular data protocol that is in it's self supported by a physical data model (from SV-11, Physical Schema), then this will also be specified. In the above figure, Port 3a supports the PLCS DEX 7 XML Schema definition for in-service feedback.

Any protocol referred to in an SV-2a diagram must be listed and defined in the TV-1 Technical Standards View.



Example SV-2a in UML – Mobile Phone



Example SV-2a in UML Using Simplified Stack Notation

SV-2b - System Port Connectivity Description

An SV-2b specifies the communications links between systems and may also list the protocol stacks used in connections.

Background

An SV-2b is used to give a precise specification of a connection between systems.

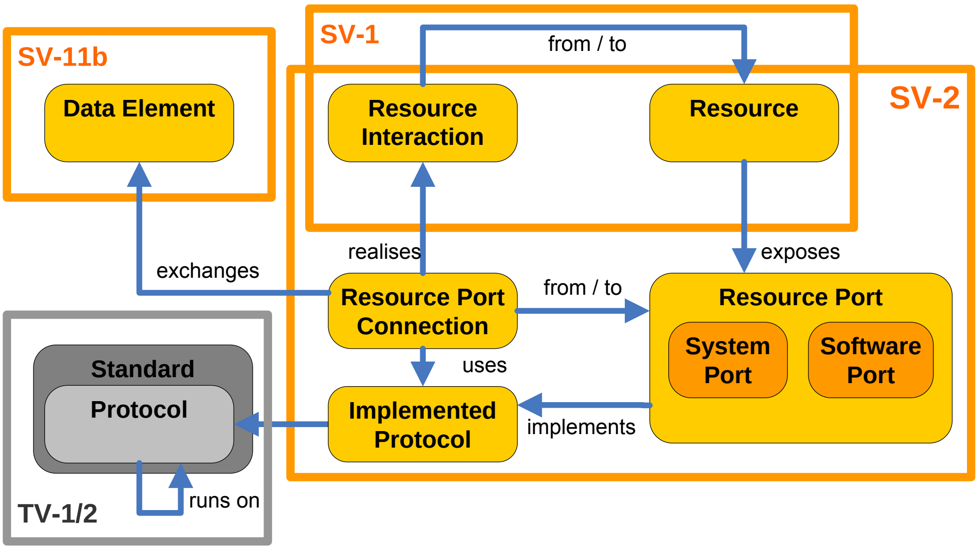
Usage

- Interface specification.

Data objects

The data in an SV-2b can include:

- System.
- System and software ports.
- Port connection.
- Protocol.



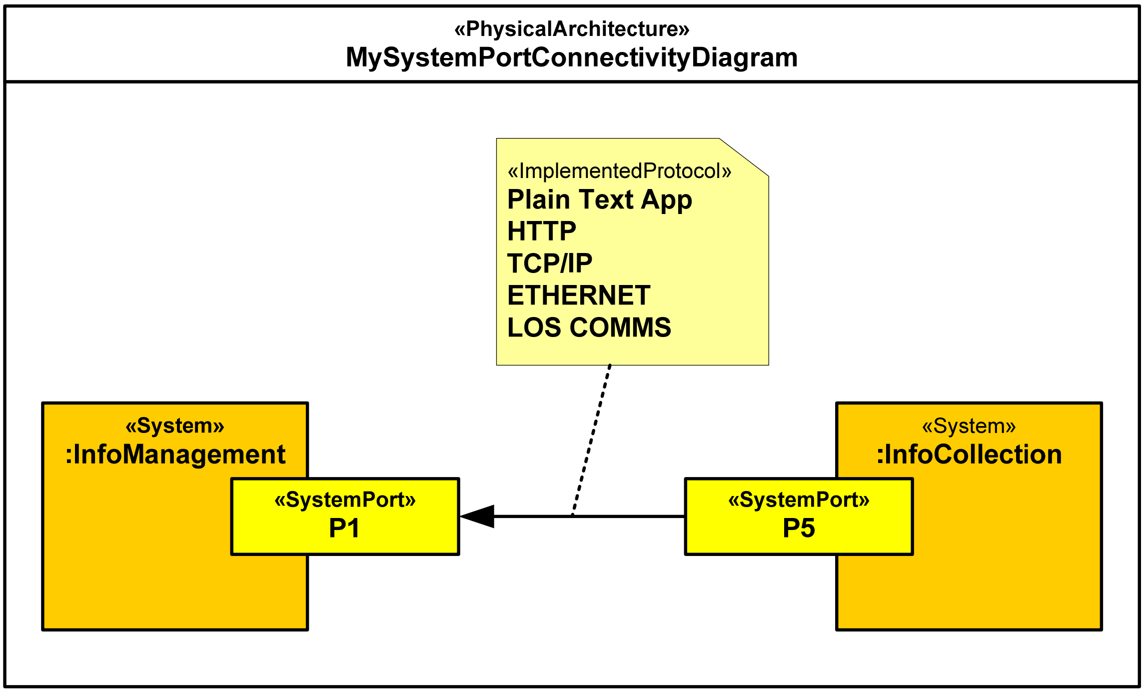
Relationships between Key Data Objects (Simplified from M3)

Representation

- Topological (connected shapes).
- UML composite structure diagram.
- SysML block diagram.

Detailed Product Description

An SV-2b is comprised of systems, their communications ports and the connections between those ports. The architect may choose to create a diagram for each pair of connected systems in the architecture or to show all the connections on one diagram if this is possible.

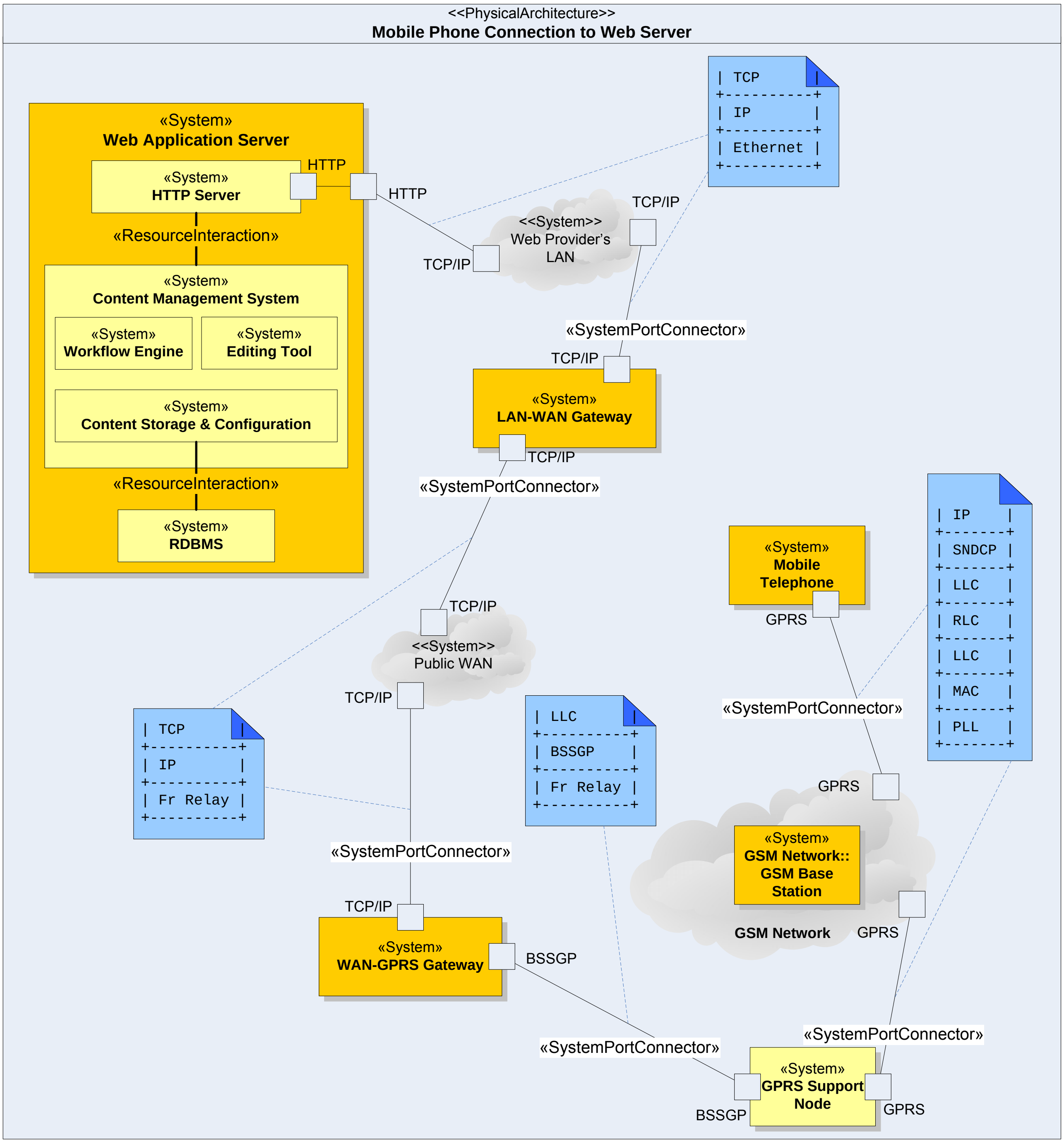


Simple SV-2b Example Showing Pair of Systems

Each diagram shall show:

- Which ports are connected.
- The systems to which the ports belong.
- The definition of the connection in terms of the physical connectivity and any protocols that are used in that connection.

The SV-2b view is closely related to the SV-2a System Port Specification that specifies the available protocols on each port. Any connection specified in an SV-2b view shall conform to the protocols specified on the corresponding port definitions in the SV-2a view.



SV-2b Example based on Mobile Data Communications

Note that networks are represented as systems. The architect may choose to show other systems being components of the network if they are part of the network infrastructure.

Any protocol referred to in an SV-2b must be defined in the TV-1 Technical Standards View.

SV-2c - System Connectivity Clusters

An SV-2c defines how individual connections between system ports are grouped when the systems share common parent resources.

Background

An SV-2c defines the connectivity requirements between resources which host one or more systems. Typically the hosting resource will be a physical asset, although it could also be an organisational resource.

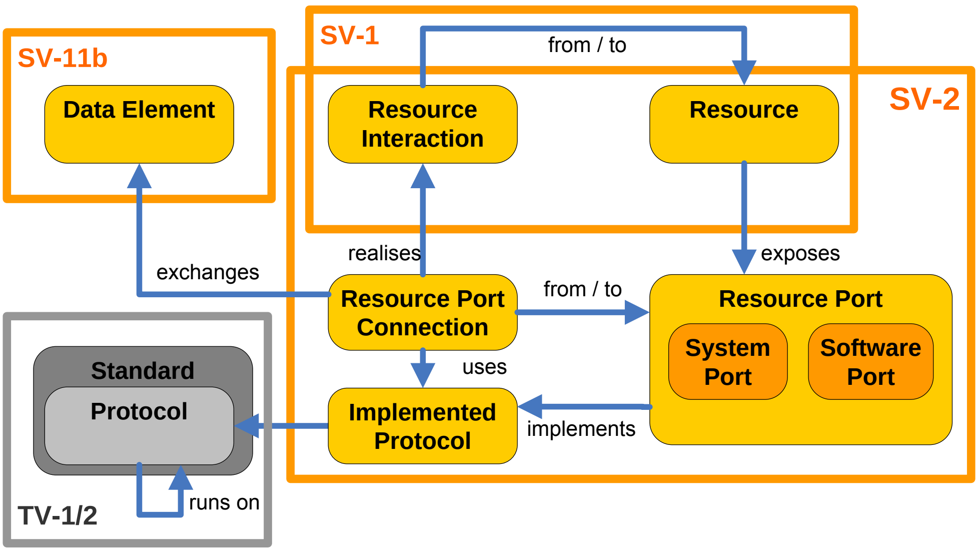
Usage

- Interface specification.
- Bandwidth and capacity analysis.

Data objects

The data in an SV-2c can include:

- Physical asset.
- Organisational resource (post type or organisation type).
- System.
- System port.
- System port connection.



Relationships Between Key Data Objects (Simplified from M3)

Representation

- UML composite structure diagram.
- Topological (connected shapes).

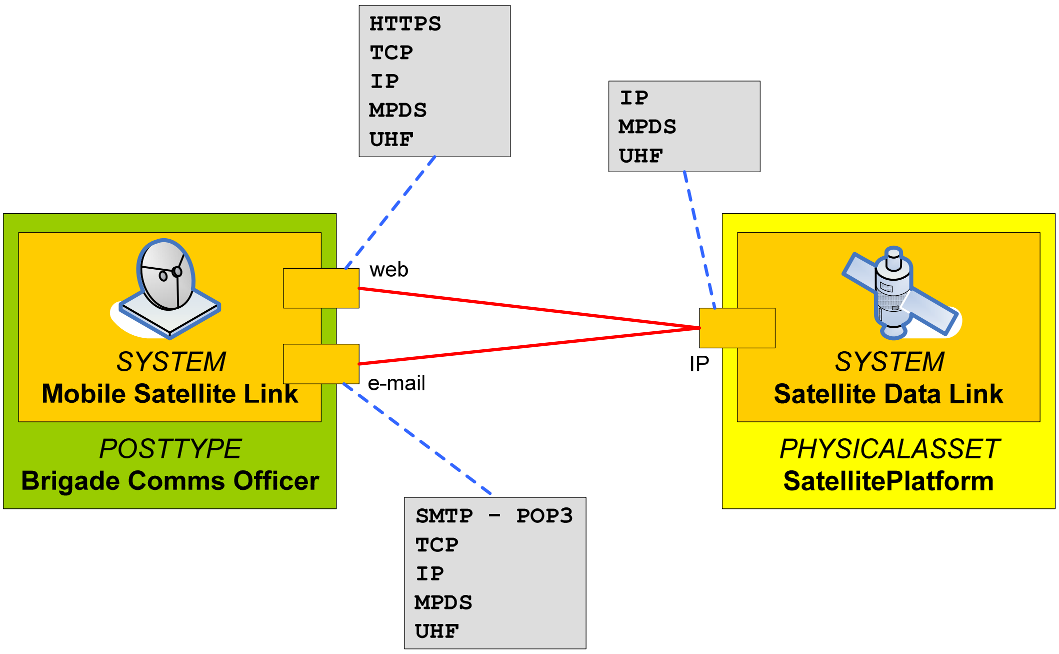
Detailed Product Description

An SV-2c defines the connectivity requirements between resources and may be used for estimating requirements for physical routing and bandwidth, as well as defining the physical architecture within a system or system of systems. An SV-2c view is particularly useful when planning physical connections and routings between physical assets.

The SV-2c is intended to aid analysis of the connectivity between systems that are hosted separately. In particular it is a useful way of highlighting redundancy issues that is, showing when too many or too few connections are used. This could indicate opportunities for cost savings from using a common network, or that there may be a need for redundancy to increase reliability.

An SV-2c consists of a diagram for each connection between assets and shows:

- The hosting resources and their systems (this should be a simple 2-level decomposition).
- The system-to-system connections that run between.
- Which ports are used in which system-to-system connections.



Example SV-2c

SV-3 - Resource Interaction Matrix

The SV-3 provides a summary of the resource interactions specified in the SV-1, Resource Interaction Specification.

Background

An SV-3 allows a quick overview of all the resource interactions specified in one or more SV-1 diagrams. The matrix format supports a rapid assessment of potential commonalities and redundancies or, if fault-tolerance is desired, the lack of redundancies.

The SV-3 can be organised in a number of ways to emphasise the association of groups of system pairs in context with the architecture's purpose.

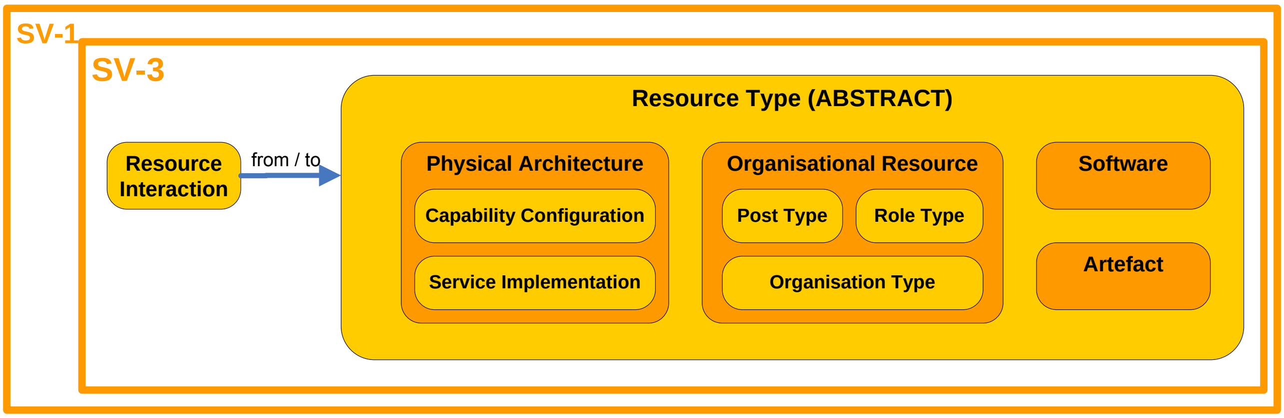
Usage

- Summarising resource interactions.
- Interface management.

Data objects

The data in an SV-3 can include:

- Resource types.
- Resource interactions.



Relationships Between Key Data Objects (Simplified from M3)

Representation

- Tabulation.
- N-squared table.

Detailed Product Description

The SV-3 summarises the resource interactions depicted in the SV-1.

Depending upon the purpose of the architecture, there could be several SV-3 products. The suite of SV-3 products can be organised in a number of ways (e.g. by domain, by operational mission phase or by solution option) to emphasise the association of groups of resource pairs in context with the architecture's purpose.

SV-3 is similar to an N2-type matrix, where the resources are listed in the rows and columns of the matrix and each cell indicates an interaction between resources if one exists.

	Blog Server	Web Terminal	Restricted LAN	LAN-SAT Bridge	Fixed Satellite Link	Satellite	Mobile Satellite Link	Satellite Modem	Web Terminal	Brigade Blogger	BLOG Reader	Comms Management
Blog Server			X									
Web Terminal			X									
Restricted LAN	X	X		X								
LAN-SAT Bridge			X		X							
Fixed Satellite Link				X		X						
Satellite					X		X					
Mobile Satellite Link						X		X				
Satellite Modem							X		X			
Web Terminal								X		X	X	X
Brigade Blogger									X			
BLOG Reader		X										
Comms Management		X										

Example SV-3

SV-4 - Functionality Description

An SV-4 specifies the functions carried out by all types of Resource, including organisational resources.

Background

The primary purposes of the SV-4 are to:

- Specify the behaviour of resources in the architecture.
- Develop a clear description of the necessary data flows that are input (consumed) by and output (produced) by each resource.
- Ensure that the functional connectivity is complete (i.e. that a resource's required inputs are all satisfied).
- Provide implementation-specific realisations of the operational activities specified in OV-5, Operational Activity Model.

The Functionality Description provides detailed information regarding the:

- Allocation of functions to resources.
- Flow of data between functions.

The SV-4 is the systems view counterpart to the OV-5.

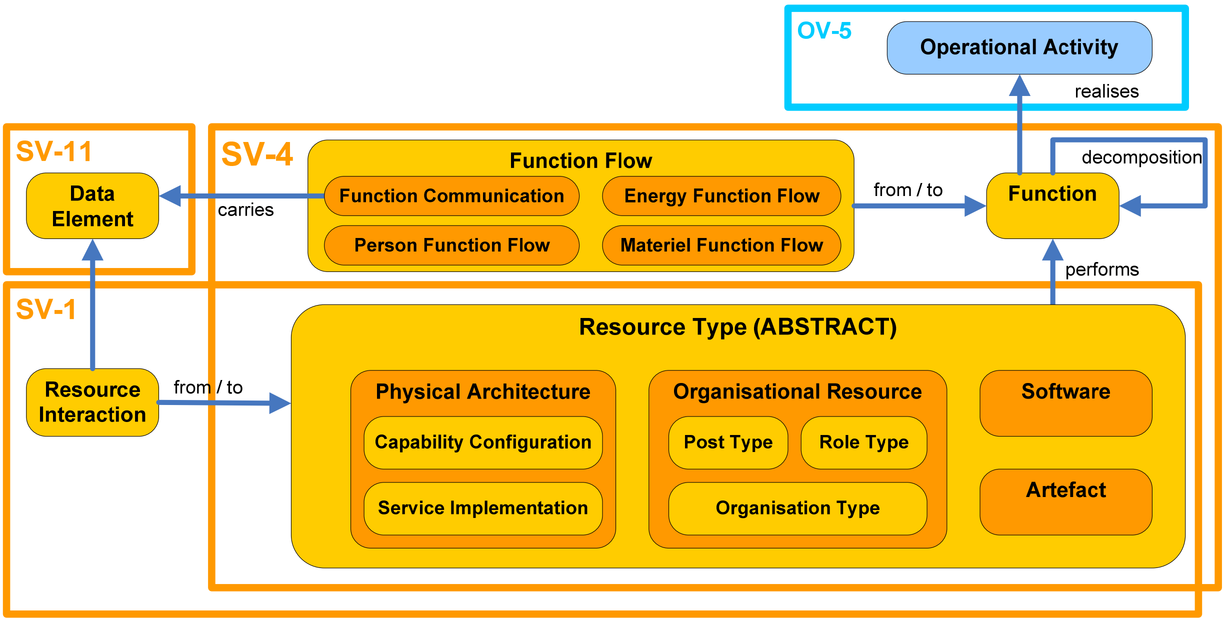
Usage

- Description of task workflow.
- Identification of functional system requirements.
- Functional decomposition of systems.
- Relating human and system functions to provide detail about interactions.

Data objects

The data in an SV-4 can include:

- Function.
- Resource.
- Data Element.



Relationships Between Key Data Objects (Simplified from M3)

Representation

- Topological (connected shapes).
- UML activity diagram.
- UML activity diagram (with swimlanes to represent resources).
- Functional Breakdown (decomposition).
- SysML activity diagram.

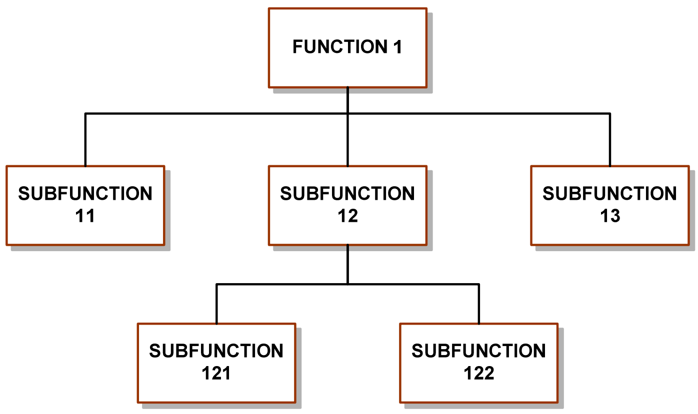
Detailed Product Description

The SV-4 is used to specify the functionality of resources in the architecture. SV-4 is the functional counterpart to the structures specified in SV-1, Resource Interaction Specification, (in the same way that OV-5 is the functional counterpart to OV-2, Operational Node Connectivity Description).

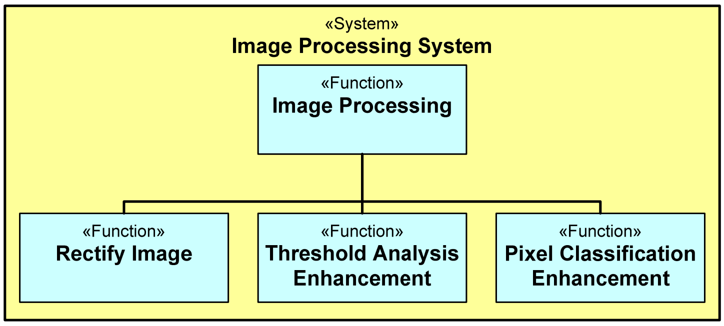
The scope of this view may be capability wide, without regard to which resources perform which functions, or it may be resource-specific (usually with the resources depicted as swimlanes). There are two basic ways to depict SV-4:

- The functional hierarchy shows a decomposition of functions depicted in a tree structure and is typically used where tasks are concurrent but dependent, for example, on a production line.
- The functional flow diagram that shows functions connected by data and control flow arrows.

The functional hierarchy approach may be particularly useful in capability-based acquisition where it is necessary to model the functions that are associated with particular capability configurations depicted in the SV-1.

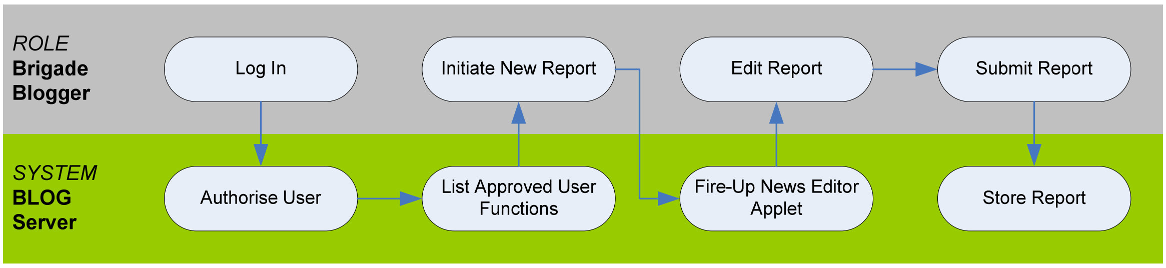


SV-4 Hierarchy Schematic

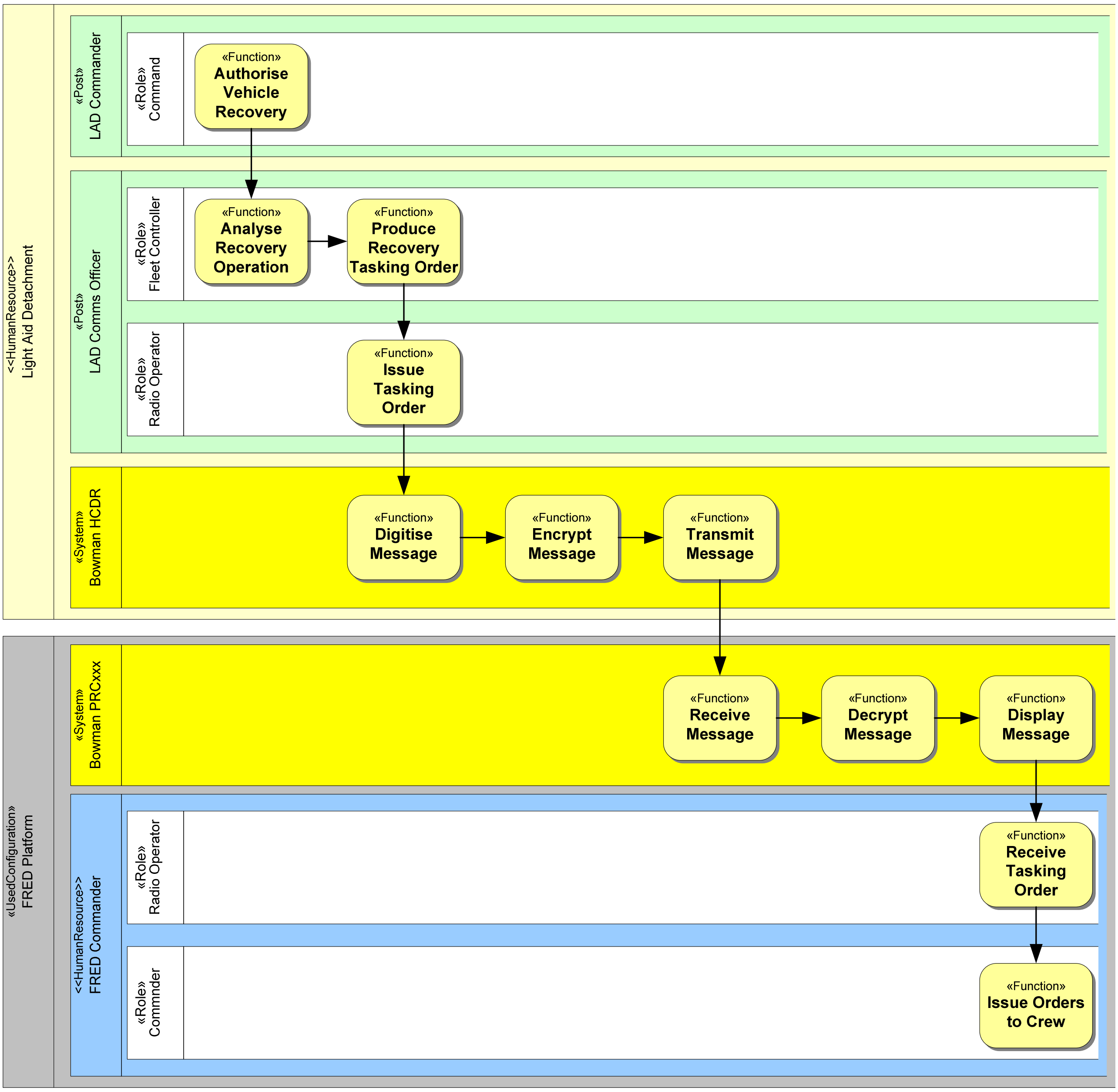


SV-4 Hierarchy Schematic with System Context

Within a system architecture, SV-4 flow diagrams document resource functions and the flows of data between those functions. Any type of resource may be used in an SV-4, and it is often used to depict the functional interactions between people and systems.



Simple Example of SV-4

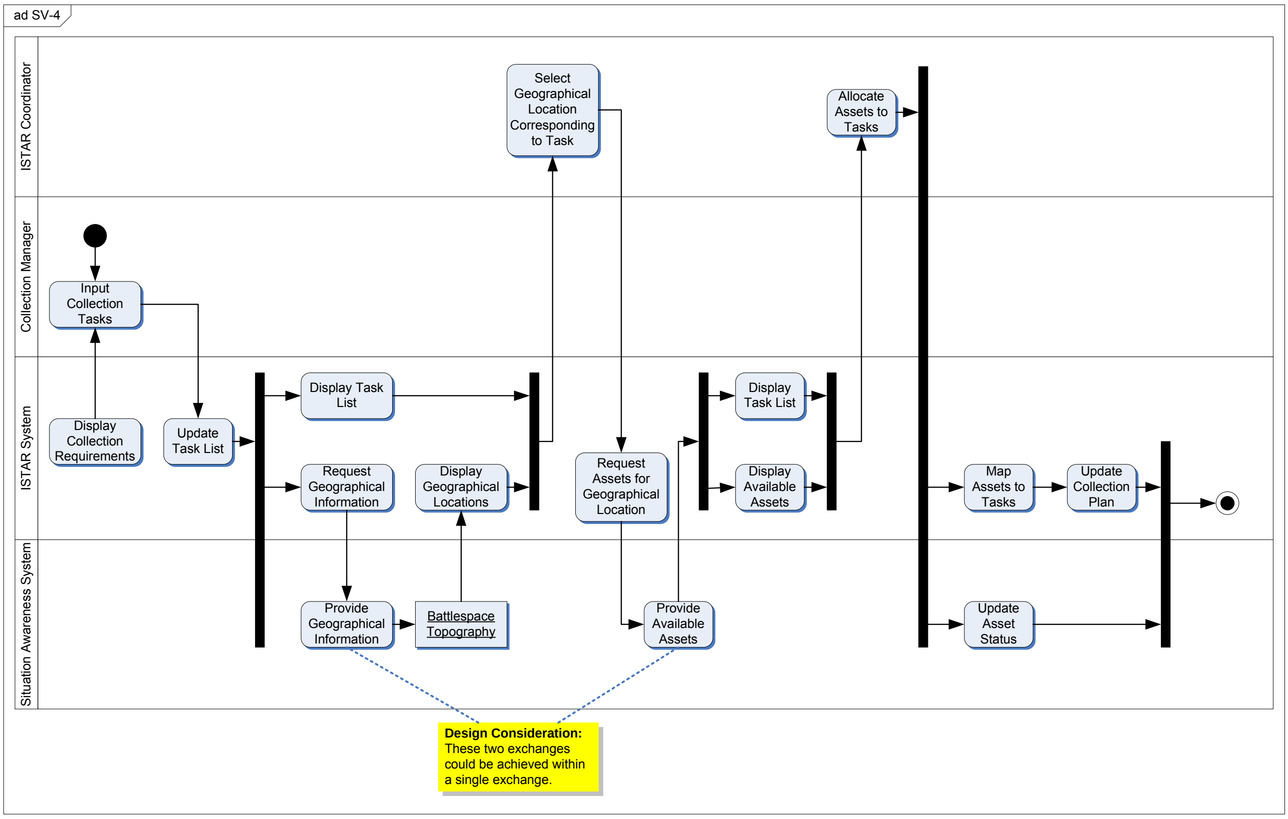


SV-4 Data Flow Schematic

The functions may realise Operational Activities captured in OV-5 and these Operational Activities may optionally be shown in SV-4, traced to the functions that realise them. The full mapping is documented in SV-5, The Function to Operational Activity/Service Function Traceability Matrix.

An SV-4 functional flow view may be used with ‘swimlanes’. A swimlane may be associated with a resource, for example a system, a capability configuration (usually based on a physical asset) or a role.

Swimlanes are presented either vertically or horizontally. A function is placed in the swimlane associated with the resource that performs it. This provides a graphical means of presenting the interactions between systems or capability configurations (shown through resource interactions on SV-1) in functional terms.



Example SV-4 Function Description (Swimlanes) (Source: IPT Deskbook)

MODAF also has the relationship 'functions upon' between functions and data elements, materiel, human resources, or energy. This allows architects to specify that the functions operate on a particular element.

In addition to information flows between functions, an SV-4 may show flows of materiel, energy or human resources.

SV-5 - Function to Operational Activity/Service Function Traceability Matrix

The SV-5 provides two alternate views:

- The mapping between functions described in SV-4, Functionality Description, and the operational activities in OV-5, Operational Activities Model.
- The mapping between functions in SV-4 and the service functions in SOV-5, Service Functionality.

Background

The SV-5 depicts the mapping of functions (and optionally, the resources that perform them) to operational activities or service functions. It therefore identifies the transformation of an operational need into a purposeful action performed by a resource. For service functions, SV-5 provides the link between the services used at the operational level and the specific functions provided by the resources that implement the services.

During requirements definition, SV-5 plays a particularly important role in tracing the architectural elements associated with system requirements to those associated with user requirements.

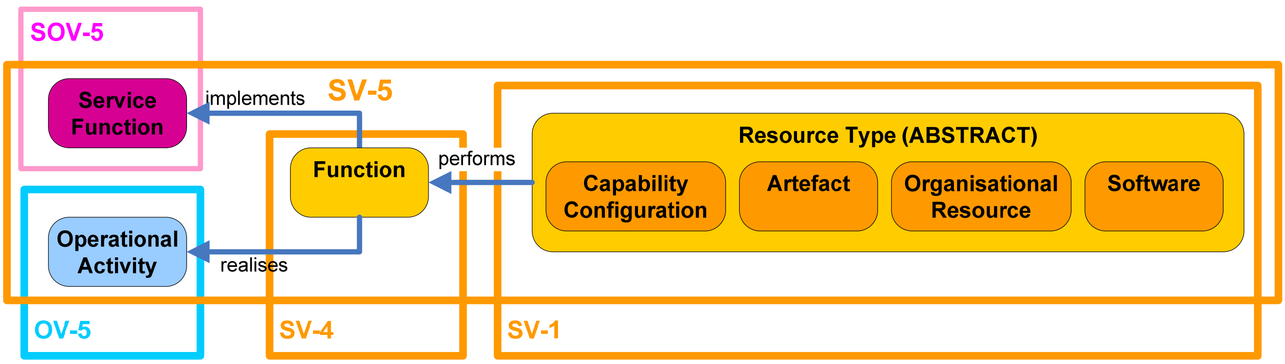
Usage

- Tracing functional system requirements to user requirements.
- Tracing solution options to requirements.
- Identification of overlaps.

Data objects

The data in an SV-5 can include:

- Function.
- Resource.
- Operational activity.
- Service function.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabulation.

Detailed Product Description

The SV-5 is a specification of the mapping between the set of operational activities or service functions to the functions that realise them.

MODAF uses the term ‘operational activity’ in the OVs and the term ‘function’ in the SVs to refer to essentially the same kind of thing, that is, both activities and functions are tasks that are performed, accept inputs, and develop outputs. The distinction between an operational activity and a function is a question of “what” and “how”; an operational activity is a specification of what is to be done, regardless of the mechanism used whereas a function specifies how a resource carries it out. For this reason, the SV-5 is a significant view, as it ties together the logical specification in the OV-5 with the physical specification of the SV-4. This logic can also be applied to services where the service functions are a specification of what functionality is to be delivered, specified independently of implementation.

The relationship between functions and operational activities or service may be many-to-many (i.e. one activity / service function may be supported by multiple functions and one function may support multiple activities / service functions).

The SV-5 is normally a matrix showing the relationship between functions, and operational activities / service functions.

SPECS 2 Functions (↓) / Operational Activity (→)	Recce	Collate Intelligence	Conduct Estimate	Co-ordinate Plan	Attack	Recuperate
Provision of Real-Time Video Imagery	X	X			X	X
Provision of Real-Time IR Imagery	X	X			X	X
Monitoring of Airspace	X	X				X
Timelapse Recording of Designated Areas	X				X	
Communications Relay	X	X	X	X	X	X
Command and Control				X	X	

Example SV-5

SV-5 may be further augmented with the resources (e.g. systems, roles and capability configurations) that conduct the functions. The architect may also wish to hide the functions in an SV-5 so that the table simply shows the mapping from resources to operational activities / service Functions.

SPECS 2 Sub-systems / Operational Activity	Recce	Collate Intelligence	Conduct Estimate	Co-ordinate Plan	Attack	Recuperate
Real-Time Imagery Sub-System	X	X			X	X
Imagery Reference Library		X	X			X
Analyst Exploitation Station		X	X	X		
Communications Sub-System	X	X	X	X	X	X
Mission Support System				X	X	

Variant SV-5 (systems mapped to operational activities)

SV-6 - Systems Data Exchange Matrix

The SV-6 specifies the characteristics of the data exchanged between systems.

Background

SV-6 focuses on the specific aspects of the system data flow and the system data content in a tabular format.

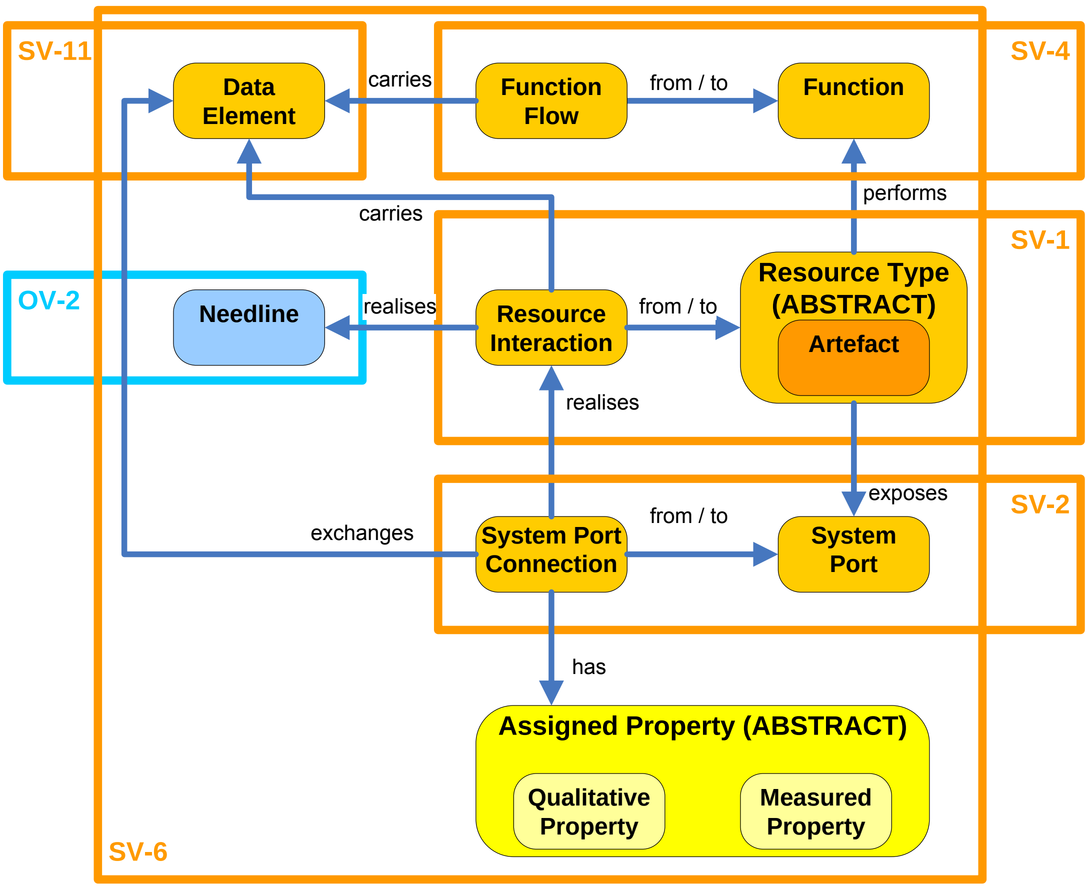
Usage

- Detailed definition of data flows.

Data objects

The data in an SV-6 can include:

- System.
- Resource interaction.
- System port connector.
- Data element.
- Information exchange (from OV-2, Operational Node Relationship Description).



Relationships between Key Data Objects (Simplified from M3)

Note: *The term 'System Data Exchange' does not refer to one M3 element. A data exchange as described in SV-6 is a combination of a system port connector and the data elements that flow through it. Any properties shown in an SV-6 table will be properties of the system port connector.*

Representation

- Tabulation.

Detailed Product Description

The SV-6 specifies the characteristics of data exchanges between systems. SV-6 is the physical equivalent of the logical OV-3, Operational Information Exchange Matrix, and provides detailed information on the system connections that implement the information exchanges specified in an OV-3. Human communications, such as verbal orders, are captured in the SV-1, Resource Interaction Specification, and SV-3, Resource Interaction Matrix, products only.

SV-6 describes, in a tabular format, data exchanged between systems, and the functions that produce and consume that data. SV-6 also specifies properties of these system data exchanges. MODAF does not mandate a set of standard properties, but it is typical to see periodicity, timeliness, throughput, size, information assurance and security characteristics of the exchange. In addition, the data elements, their format and media type, accuracy, units of measurement, and system data standard may also be described in the matrix.

Data Exchange ID	Sending System	Sending Port	Receiving System	Receiving Port	Description	Data Elements / Formats	Data Rate	Contention Ratio	Readio Frequency (MHz)	Operational Info Exchange
1	Web Application Server	HTTP1	Web Provider's LAN	TCP-IP-WPL1	Local Area Connection for Web Server	HTML, Javascript, PNG, JPEG	100 Mb/s	1:1	N/A	1
2	LAN-WAN gateway	TCP-IP-LWG1	Web Provider's LAN	TCP-IP-WPL2	Local Area Connection for LAN-WAN Gateway	Any	100 Mb/s	1:1	N/A	1
3	LAN-WAN gateway	TCP-IP-LWG2	Public WAN	TCP-IP-PW1	LAN-WAN Gateway Connection to Public Wide Area Network	Any	20 Mb/s	20:1	N/A	1
4	WAN _GPRS gateway	TCP-IP-WG1	Public WAN	TCP-IP-PW2	GPRS-WAN Gateway Connection to Public Wide Area Network	HTML, Javascript, PNG, JPEG, POP3, SMTP, IMAP, FTP	T4	5:1	N/A	1
5	WAN _GPRS gateway	TCP-IP-WG1	GPRS Support Node	BSSGP-GSN-1	WAN-GPRS Gateway Connection to GPRS Support Node	HTML, Javascript, PNG, JPEG, POP3, SMTP, IMAP, FTP	1 Gb/s	1:1	N/A	1
6	GPRS Support Node	GPRS-GSN-1	GSM Network	GPRS-GSM-1	GPRS Support Node to GSM Network	HTML, Javascript, PNG, JPEG, POP3, SMTP, IMAP, FTP	128 Kb/s	1:1	900-1800	1
7	Mobile Phone	GPRS-MB-1	GSM Network	GPRS-GSM-2	Mobile Phone to GSM Network connection	HTML, Javascript, PNG, JPEG, POP3, SMTP, IMAP, FTP	128 Kb/s	9:1	900-1800	1

Example SV-6 Based on GPRS SV-2b Example

Where a suite of SVs provides a physical specification for a logical requirement specified in a suite of OVs, the SV-6 properties should cover all the information exchange properties specified in OV-3. Similarly, it is recommended that all data elements carried by the data exchanges are shown.

It should be noted that each data element exchanged may be related to the function (from SV-4) that produces or consumes it. However, there need not be a one-to-one correlation between data elements listed in the SV-6 matrix and the data flows (inputs and outputs) that are produced or consumed in a related SV-4.

Because an SV-6 is about showing flows across system boundaries, data flows between system functions performed by the same systems may not be shown in the SV-6 matrix; there will be no corresponding system port connection.

Note: although flows of materiel, human resources and energy are permitted in SV-1 and SV-4, they are *not* permitted in SV-6.

SV-7 Resource Performance Parameters Matrix

The SV-7 depicts the performance characteristics of a Resource.

Background

The SV-7 expands on the information presented in an SV-1, Resource Interaction Specification, by depicting the characteristics of the resources shown in the SV-1.

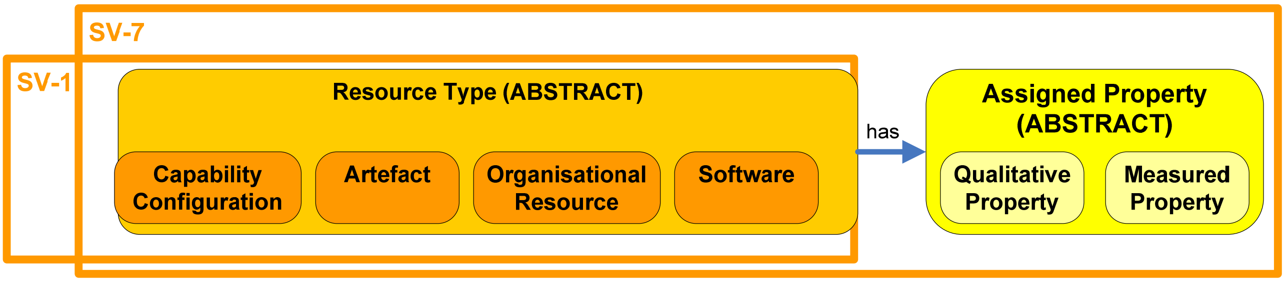
Usage

- Definition of performance characteristics.
- Identification of non-functional requirements (input to System Requirements Document).

Data objects

The data in an SV-7 can include:

- Resource.
- Measurable property.
- Qualitative property.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabulation.

Detailed Product Description

The SV-7 specifies qualitative and quantitative characteristics of resources; i.e. the performance parameters of each resource. The SV-7 is typically a tabular view.

One of the primary purposes of the SV-7 is to communicate which characteristics are considered most crucial for the successful achievement of the mission goals assigned to the resource. These parameters can often be the deciding factor in acquisition and deployment decisions, and will figure strongly in systems analyses and simulations done to support the acquisition decision processes and system design refinement.

The Figure below is a template of SV-7, listing notional user defined performance. It should be noted that these are example metrics – MODAF does not mandate a specific set of resource characteristics.

Parameter ID	System / Element	Performance Requirement	Metric	Measure
Hardware				
H 1.1	SPECS 2 Transmitter	Transmission Rate	2	GB
H 2.1	SPECS 2 Receiver	Gain	60	dB
H 2.2		Signal to Noise ratio	20	dB
H 3.1	SPECS 2 Signal Processor	Comms Channel Bandwidth Support	2	GB
H 4.1	SPECS 2 Video Recorder	Top-end Resolution	1024x768	Pixels
H 4.2		Storage Capacity	20	Hours @ top-end resolution
Software				
S 1.1	Video Analysis	Minimum target location co-ordinate accuracy	10	metres
		Minimum target speed accuracy	5	metres / second
S 1.2	Target Status Alerting	Minimum status Change alert accuracy	500	metres
		Minimum Alert Response time	30	seconds

Example SV-7

It is sometimes useful to analyse resource evolution by comparing performance characteristics for current and future resources. For this reason, repository tools may produce hybrid SV-7 products that span architectures for multiple enterprise phases.

SV-8 - Capability Configuration Management

The SV-8 depicts the whole lifecycle view of a resource, describing how its configuration changes over time.

Background

The SV-8 provides an overview of how a capability configuration changes over time. It shows the structure of several capability configurations mapped against a timeline.

Usage

- Development of incremental acquisition strategy.
- Configuration Management.
- Planning technology insertion.

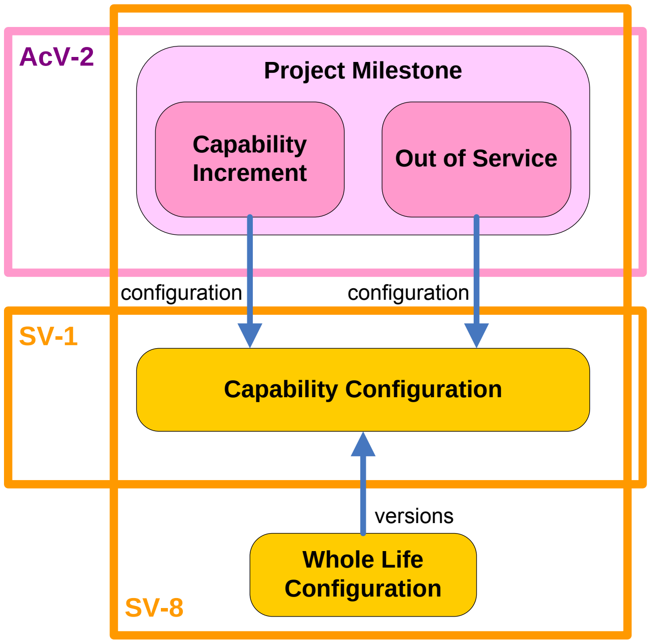
Data objects

The data in an SV-8 can include:

- Capability configurations.
- Resources that are parts of capability configurations.
- Project milestone (that reflect equipment delivery).

Representation

- Timeline view.
- 'Herringbone' diagram.



Relationships Between Key Data Objects (Simplified from M3)

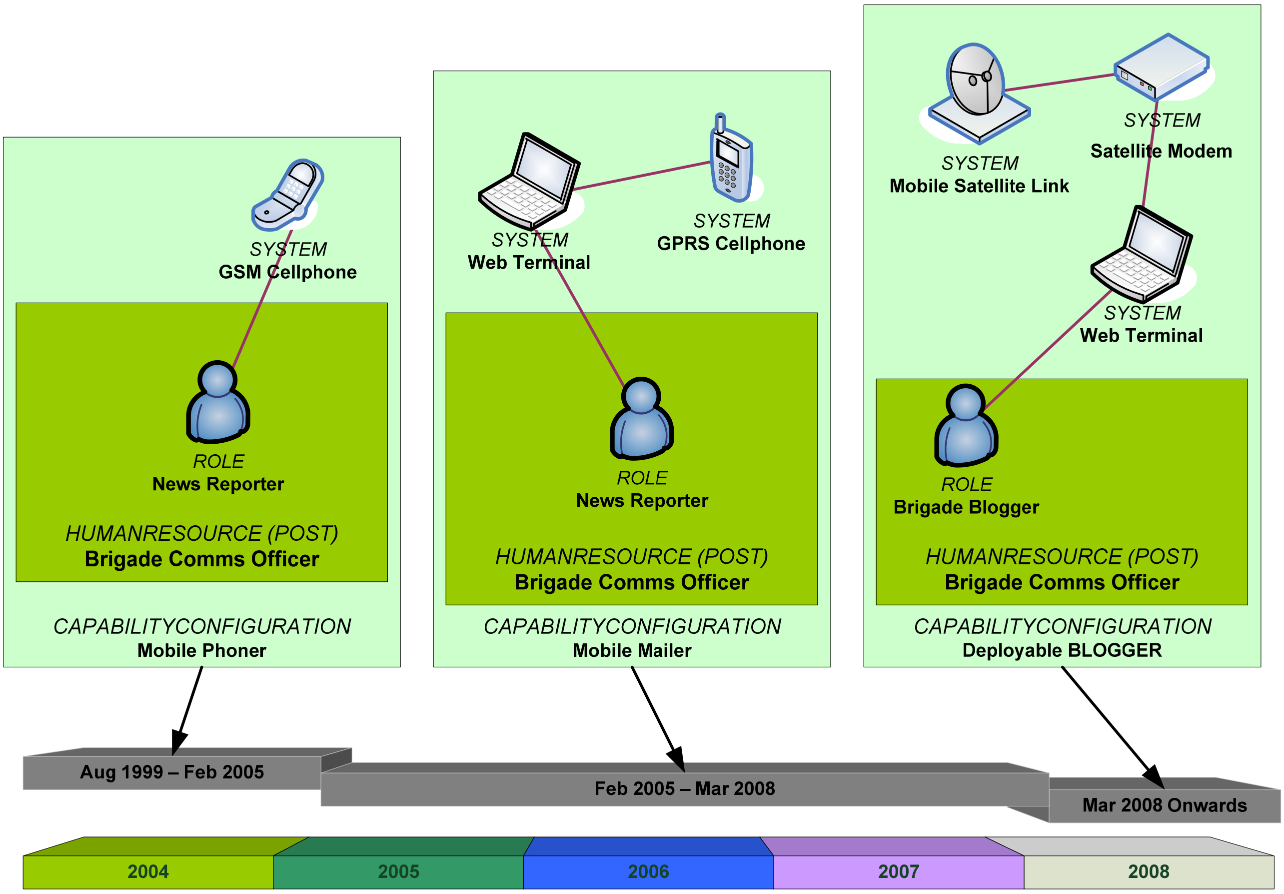
Detailed Product Description

An SV-8 provides a rich definition of how the enterprise and its capabilities are expected to evolve over time, especially when linked together with other evolution views such as AcV-2, Programme Timelines, StV-3, Capability Phasing, and TV-2, Standards Forecast.

In this manner, the view can be used as an architecture evolution project plan or transition plan. In meta-model terms, an SV-8 product is constructed from data specified in AcV-2 and SV-1, Resource Interaction Specification, though there may be several SV-1 products – one for each version of the configuration. It may therefore be possible for MODAF tools to automatically generate SV-8 products from existing data.

An SV-8 can describe legacy, current and future capability configurations against a timeline. Using similar modelling elements as those used in SV-1, (resource composition, resources, etc), the view shows the structure of each capability configuration. Resource interactions which take place *within* the capability configuration boundaries may also be shown.

The changes depicted in the SV-8 View are derived from the project milestones that are also shown in AcV-2.



Example SV-8 Showing Evolution of Front Line News Reporting

SV-9 - Technology & Skills Forecast

The SV-9 identifies the technologies and skills required by the Entererprise over time. These are technologies and skills that can be reasonably forecast against the current state and expected improvements / trends. New technologies and skills will be tied to specific time periods, which can correlate against Enterprise Phases.

Background

SV-9 provides a summary of the current and emerging technologies and skills that impact on the Resources that constitute the Architecture. The SV-9 provides descriptions of relevant:

- Emerging and current technologies.
- Industry trends.
- Predictions (with associated confidence factors) of the availability and readiness of specific hardware and software products.
- Current and possible future skills (modelled using M3 Competence elements).

In addition to providing an inventory of trends, capabilities and products, the SV-9 also includes an assessment of the potential impact of these items on the enterprise (provided as text in the M3 Forecast element). Given the future-oriented nature of this product, forecasts are typically made in short, mid and long-term timeframes, such as six, 12 and 18-month intervals.

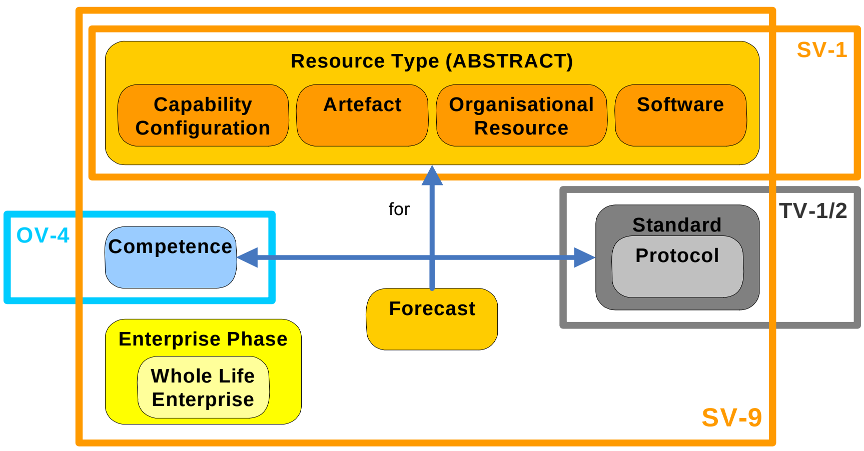
Usage

- Forecasting technology readiness against time.
- HR trends analysis.
- Recruitment planning.
- Planning technology insertion.
- Input to options analysis.

Data objects

The data in an SV-9 can include:

- Resources.
- Competences.
- Standards.
- Forecasts (for the any of the above).



Relationships Between Key Data Objects (Simplified from M3)

Representation

- Timeline view.
- ‘Herringbone’ diagram.

Detailed Product Description

An SV-9 summarises predictions about trends in technology and personnel skills, and is usually presented as a table, with forecasts categorised by periods of time (e.g. 6-month, 12-month, 18-month intervals). Architects will generally produce separate SV-9 products for technology and skills.

The specific time periods selected (and the trends being tracked) will usually correspond to Enterprise Phases. However where technology is fast-moving (e.g. CPU and Storage technology) then shorter periods may be required. The forecast includes a text summary of potential impacts on current architectures and thus influences the development of transition and target architectures.

If standards are an integral part of the technologies important to the evolution of a given architecture, then it may be convenient to combine SV-9 with the TV-2, Technical Standards Forecast.

Where applicable, the SV-9 may relate forecasts to those resources that are impacted by the technology changes – this is stored as plain text in the M3 "Forecast" element.

TECHNOLOGY AREA	TECHNOLOGY FORECASTS		
	SHORT TERM	MEDIUM TERM	LONG TERM
APPLICATION SOFTWARE			
OFFICE APPLICATIONS	MICROSOFT OFFICE 2000	MICROSOFT OFFICE 2005 (DISTRIBUTED)	MICROSOFT DISTRIBUTED OFFICE APPS
BUSINESS APPLICATIONS	INDIVIDUAL APPS – BATES QP24	BISAs	INTEGRATED BISA SUITE
APPLICATION PLATFORM			
DATA MANAGEMENT	ORACLE 9	ORACLE 10	
OPERATING SYSTEM	WINDOWS 2000	NEXT WINDOWS OS	OPEN SOURCE OS
EXTERNAL ENVIRONMENT			
USER INTERFACE		THIN TOUCH SCREEN	BIOMETRIC INTERFACE
STORAGE	HDD	SOLID STATE MEMORY CHIPS	ORGANIC STORAGE
COMMS	BOWMAN	BOWMAN + VOIP	ALL IP COMMS

Example Systems Technology Forecast (SV-9)

Introduction to SV-10a, SV-10b and SV-10c

Behavioural modelling and documentation are key to a successful architecture description. OV-5, Operational Activity Model, and the OV-6 series provide a logical specification of behaviour, which are mirrored in the SV-4, Functionality Description, and SV-10 specifications of resource behaviour.

SV-4 provides a functional specification of the behaviour of resources, however, it is useful to augment this with specifications of the constraints the resources are subject to, the states they can have and the sequence in which interactions between them take place. SV-10 provides this additional specification. SV-10 is in three parts:

- Resource Rules Model (SV-10a).
- Resource State Transition Description (SV-10b).
- Resource Event-Trace Description (SV-10c).

SV-10a - Resource Constraints Specification

The SV-10a specifies functional and non-functional constraints on the implementation aspects of the architecture (i.e. the structural and behavioural elements of the Strategic Viewpoint).

Background

The SV-10a describes constraints on the resources, functions, data and ports that make up the SV physical architecture. The constraints are specified in text and may be functional or structural (i.e. non-functional).

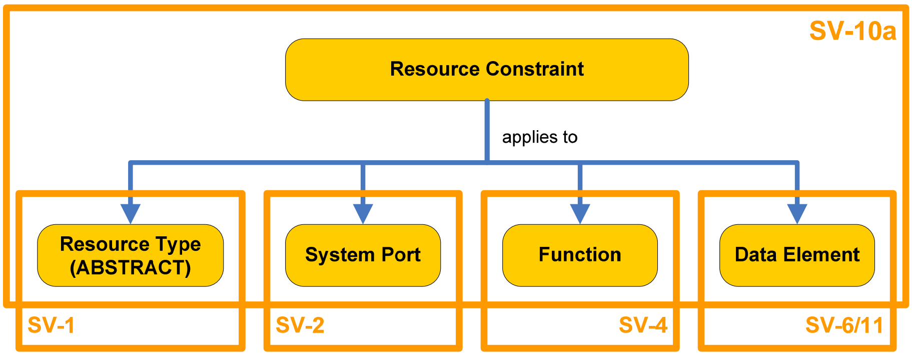
Usage

- Definition of implementation logic.
- Identification of resource constraints.

Data objects

The data in an SV-10a can include:

- Resource constraint.



Relationships Between Key Data Objects (Simplified from M3)

Representation

- Text (preferably specified in a computer-interpretable constraint language such as Object Constraint Language (OCL).
- Tabular.

Detailed Product Description

The SV10-a describes the rules that control, constrain or otherwise guide the implementation aspects of the architecture. Resource constraints are statements that define or constrain some aspect of the technology or business, and may be applied to:

- Resources.
- Functions.
- System ports.
- Data elements.

Constrained Element	Constraint	Description
«PostType» FRED Crane Operator	Age > 18 Yrs Old	All FRED Crane Operators must be older than 18 for insurance purposes.
«System» Bowman PRCxxx	Range > **km	A Bowman PRCxxx shall have a useable transmission range of **km.
«Function» Analyse Recovery Operation	Duration < 1hr	A fleet controller must be able to conduct the op analysis in less than one hour.

An SV-10a Presented in Tabular Form

OV-6a, Operational Rules Model, provides a specification of logical constraints (i.e. rules that will apply in general, regardless of what resources are used). The SV-10a provides a set of resource-specific constraints that are applied in order to satisfy the general constraints from OV-6a.

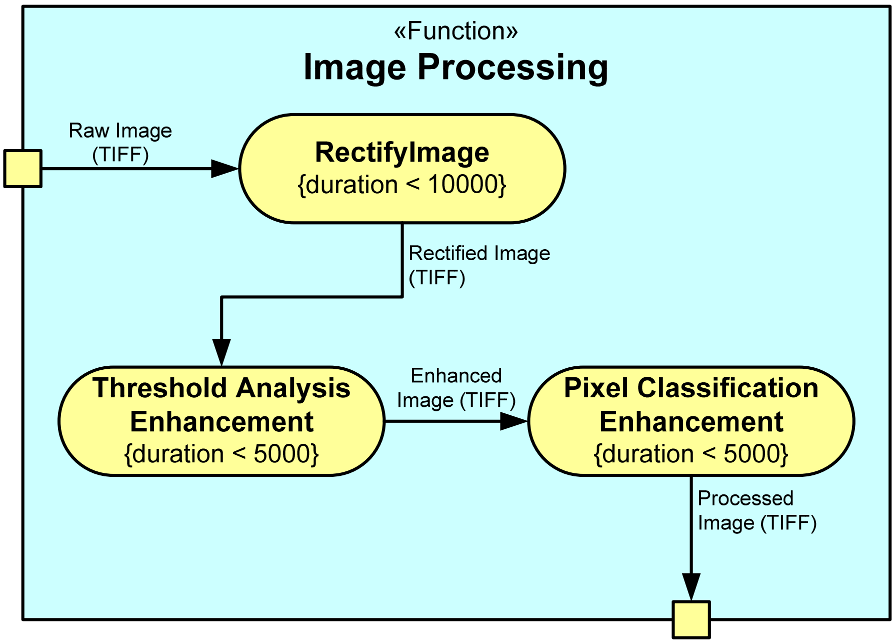
MODAF categorises resource constraints as follows:

- Structural assertions – non-functional constraints governing some physical aspect of the architecture.
- Action assertions – functional constraints governing the behaviour of resources (constraints on functions).
- Derivations – these involve algorithms used to compute facts.

Where a resource constraint is based on some standard, then that standard should be listed in the Standards Profile (TV-1).

Some resource constraints can be added as annotations to other views, in which case SV-10a should provide a listing of the complete set of those rules and any others that are not shown in other views.

With potentially complex resource constraints it may be more useful to express these rules in Object Constraint Language (OCL), as below.



Example SV-4 with SV-10a OCL Constraints Embedded in Functions

SV-10b - Resource State Transition Description

The SV-10b is a graphical method of describing a resource's response to various events in terms of its changes of state. The view specifies the possible states a resource can be in, the possible transitions between those states, and the triggers for those changes.

Background

The functional specification of a resources behaviour presented in SV-4, Functionality Description, can show the flows of control and data between resources, but it cannot reflect the changes of state that occur when control or data is passed from one resource to another. SV-10b provides this additional information.

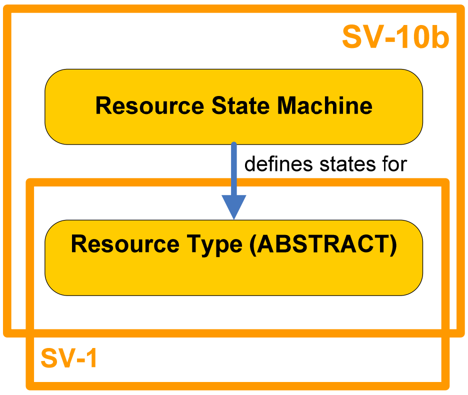
Usage

- Definition of states, events and state transitions (behavioural modelling).
- Identification of constraints on possible states (input to System Requirements Document).

Data objects

The data in an SV-10b can include:

- Resources.
- States (associated with a resource or function).
- State transitions (each associated with an event).



Relationships Between Key Data Objects (Simplified from M3)

NB. The M3 does not provide much detail on SV-10b, the assumption being that the UML meta-model (that underpins M3) is sufficient to cover all the requirements for state transition diagrams.

Representation

- UML state diagram (preferred).

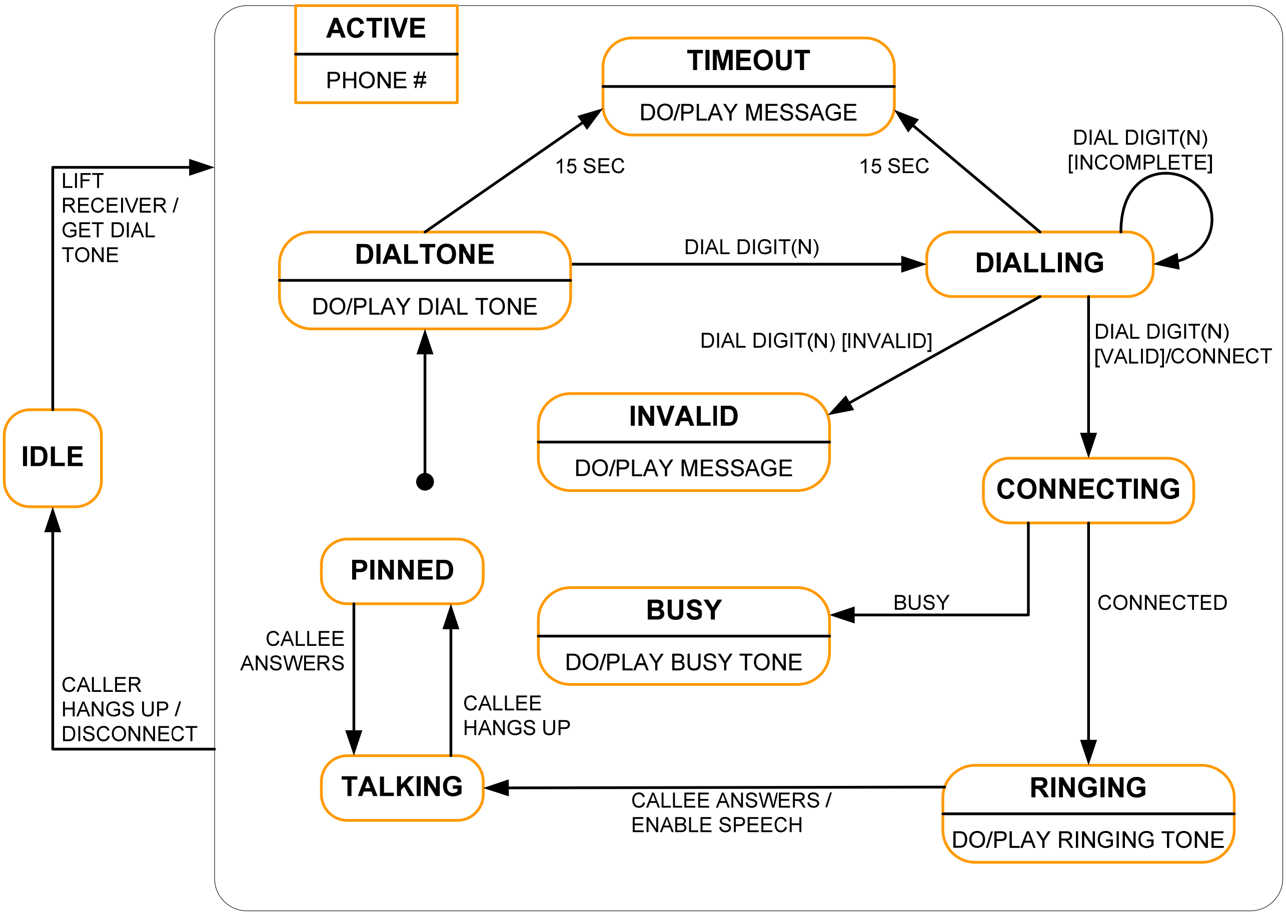
Detailed Product Description

The SV-10b relates events to resource states and describes the transition from one state to another. SV-10b describes state transitions from a resource perspective, with a focus on how the resource responds to stimuli (e.g. triggers and events). As with the OV-6b, Operational State Transition Description, these responses may differ depending upon the rule set or conditions that apply as well as the resource's state at the time the stimuli is received.

The Figure below provides a template for a simple SV-10b. The black dot and incoming arrow point to initial states (usually one per diagram), while terminal states are identified by an outgoing arrow pointing to a black dot with a circle around it. States are indicated by rounded corner box icons and labelled by name or number and, optionally, any actions associated with that state. Transitions between states are indicated by one-way arrows labelled with event or action notation, which indicates an event-action pair and shows, when an event occurs, the corresponding action which is executed. This notation indicates the event that causes the transition and the ensuing action (if any) associated with the transition.

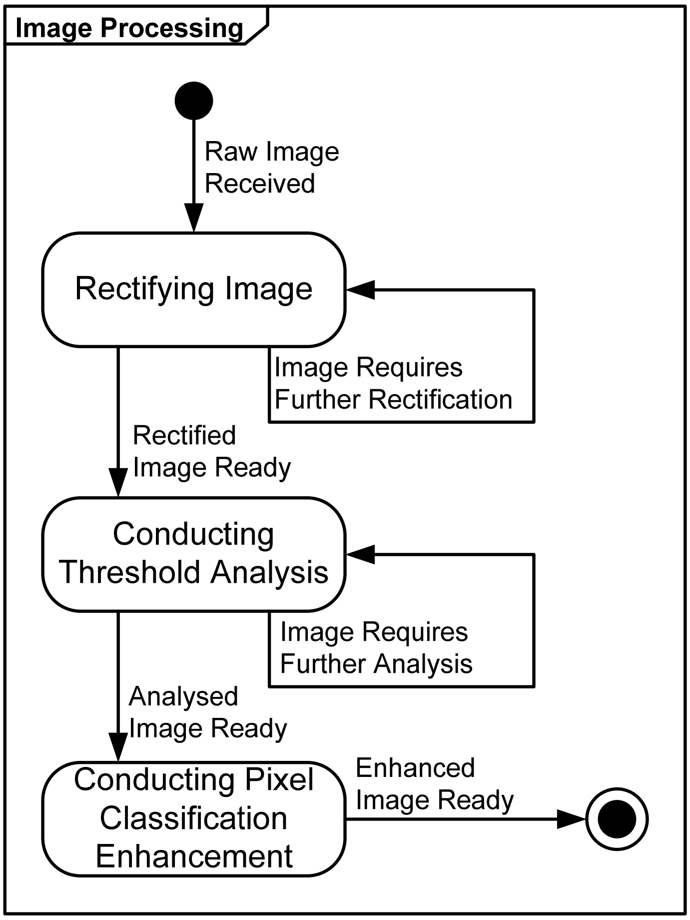
Generic State Transitions Diagram

Composing state transitions provides a model of states known as a state chart.



Example of a State Chart

The SV-10b relates states, events, and actions. A state and its associated action(s) specify the response of a resource or function to events. When an event occurs, the next state may vary depending on the current state (and its associated action), the event and the rule set or guard conditions. A change of state is called a transition. Each transition specifies the response based on a specific event and the current state. Actions may be associated with a given state or with the transition between states.



Example SV-10b

States in SV-10b products may be nested. This enables quite complex models to be created to represent resource behaviour.

SV-10c - Resource Event-Trace Description

The SV-10c provides a time-ordered examination of the interactions between resources. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

Background

The SV-10c provides a valuable mechanism for improving the level of detail from the initial solution design, in order to help define a sequence of interactions, and to ensure that each participating resource or resource port has the necessary information it needs, at the right time, in order to perform its assigned functionality.

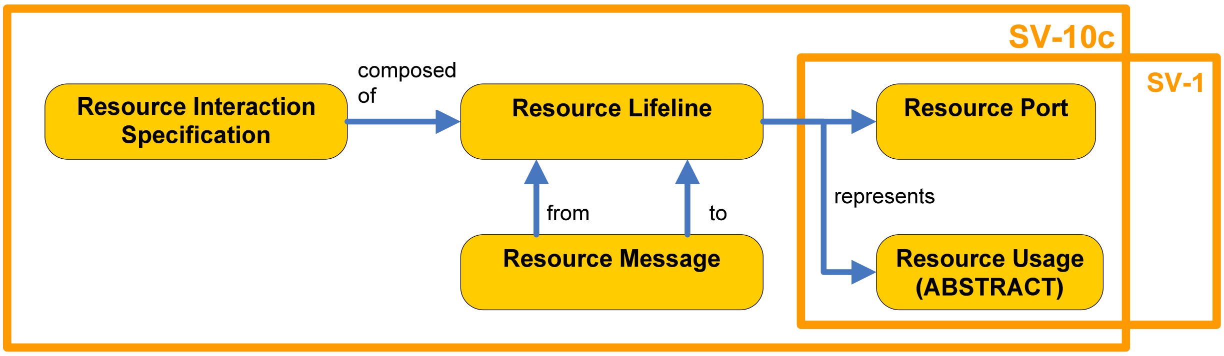
Usage

- Analysis of resource events impacting operation.
- Behavioural analysis.
- Identification of system requirements (input to System Requirement Document).

Data objects

The data in an SV-10c can include:

- Resource Interaction Specification
- Lifelines (each associated with a functional resource or a system port).
- Resource Message



Relationships between Key Data Objects (Simplified from M3)

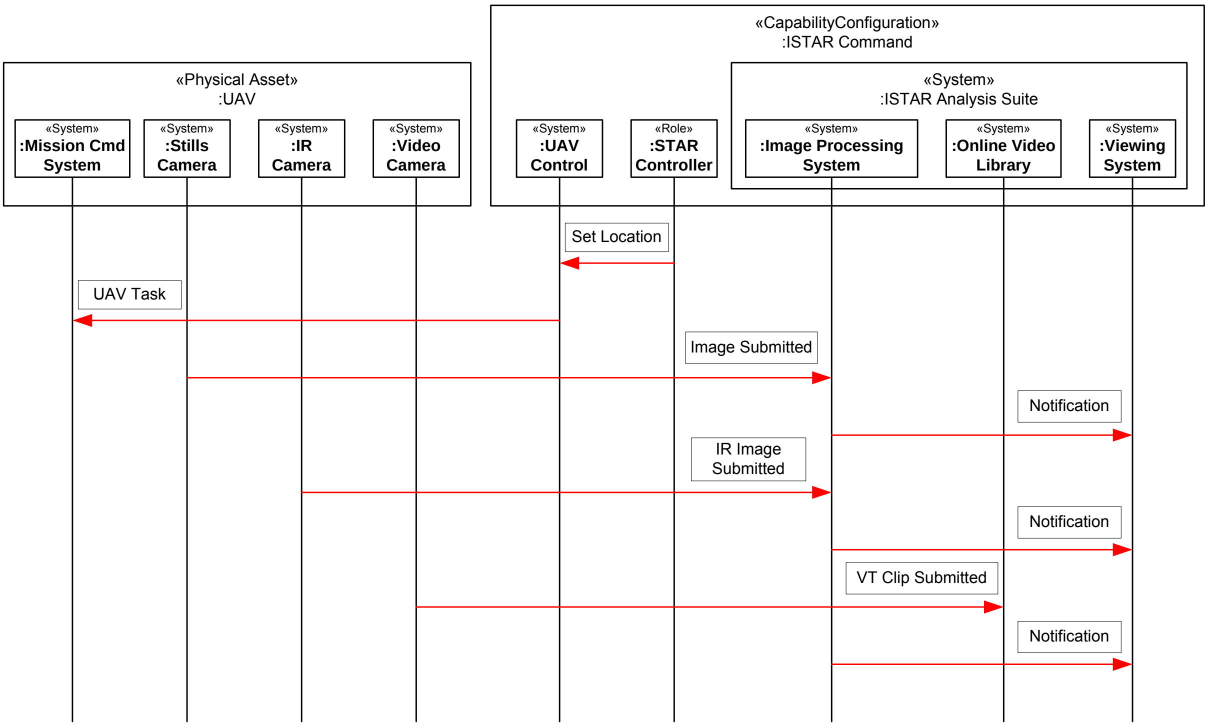
Representation

- Topological (connected shapes).
- UML Sequence Diagram (preferred).

Detailed Product Description

The SV10-c specifies the sequence in which data elements are exchanged in context of a Resource or Resource Port. Resource event / trace descriptions are sometimes called ‘sequence diagrams’, ‘event scenarios’ or ‘timing diagrams’.

The diagram below shows the components of an SV-10c. The items across the top of the diagram are usages of resources or resource ports. The lifelines are depicted as vertical lines descending from the resources and ports.



Systems Event-Trace Description (SV-10c)

Arrows between the lifelines represent exchanges of messages (data), materiel, energy or human resources. The direction of the event lines represents the flow of information from one resource / port to another. The SV-10c provides a time-ordered examination of the data elements exchanged between participating resources or system ports, and the required time interval between exchanges may be shown as a measure between the arrows. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

The content of exchanges that connect lifelines in an SV-10c may be related, in modelling terms, with resource interactions (from, SV-1, Resource Interaction Specification, SV-3, Resource Interaction Matrix), data flows (from SV-4, Functionality Description, and SV-6, Systems Data Exchange Matrix) and data schema entities (from SV-11, Physical Schema) modelled in other views.

The interactions in SV-10c are not just limited to representing information flows. As in SV-1 and SV-4, they may also represent flows of materiel, human resources or energy.

SV-11 - Physical Schema

The SV-11 defines the structure of the various kinds of system data that are utilised by the systems in the architecture.

Background

The SV-11 provides an implementation specific data model that realises the logical data model presented in an OV-7, Information Model.

Implementation constraints may mean that one logical entity in OV-7 is in fact realised by more than one entity in SV-11 and vice versa.

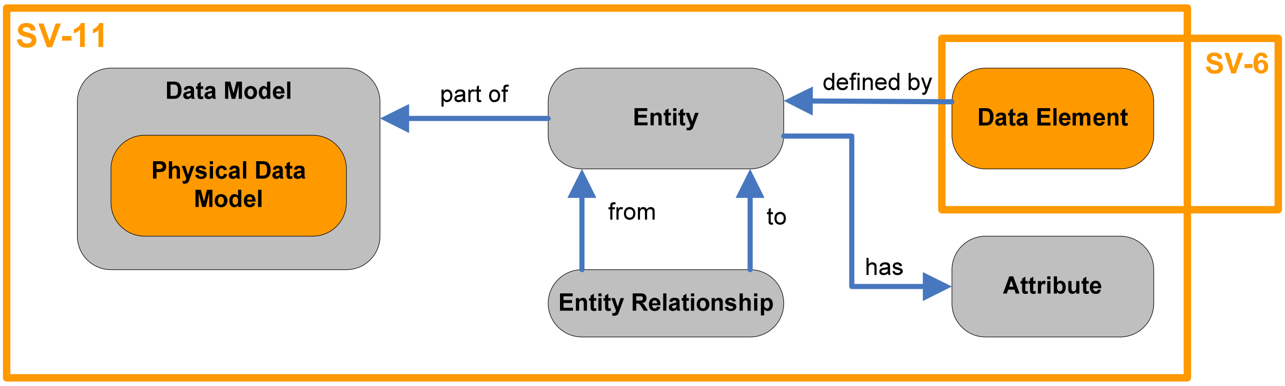
Usage

- Specifying the system data elements exchanged between systems (thus reducing the risk of interoperability errors).
- Definition of physical data structure (input to system design).

Data objects

The data in an SV-11 can include:

- System data entity.



Relationships Between Key Data Objects (Simplified from M3)

Representation

- Formal text data modelling language (e.g. SQL, ISO10303-11, etc.).
- Topological (connected shapes).
- UML class diagram.

Detailed Product Description

SV-11 allows implementation-level detail of data structures to be modelled. The view serves several purposes, including:

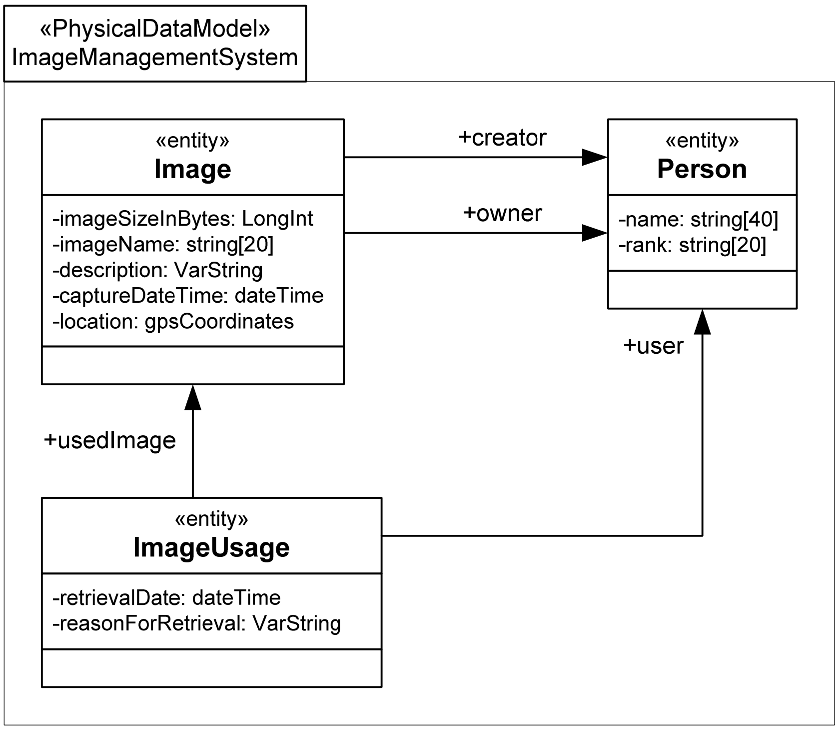
- Providing the detailed information on the system data elements exchanged between systems, thus reducing the risk of interfacing errors.
- Providing system data structures for use in the system design process, if necessary.

SV-11 is an implementation-oriented data model that is used to describe how the information requirements represented in OV-7 are actually implemented for a given solution. The entities specified in SV-11 represent data elements (from the SV-6, Systems Data Exchange Matrix).

There should be a mapping from a given logical data model (OV-7) to the physical data model (SV-11) if both models are used. This mapping is often not trivial (e.g. there may be conditional mappings), in which case a formal mapping language such as ISO10303-14 should be used (as M3 comments) against the data model or individual entities.

Standards associated with entities are also often identified in the development of the SV-11 view product; these should be recorded in the TV-1 Standards Profile.

UML provides a suitable language for developing physical schema (via class diagrams).



Example SV-11 (UML)

Note that an SV-11 also simply be a text schema (e.g. in the case of SQL or ISO10303-11).

Introduction to SV-12a and SV-12b

The SV-12 series of views specify standard configurations that deliver services. These may be configurations of resources (SV-12a) or configurations which utilise other services (SV-12b). In both cases, these views specify how services are implemented. They do not specify the services themselves (this is covered by the Service Oriented Viewpoint), nor do they describe deployed services.

SV-12a - Service Provision

The SV-12a specifies configurations of resources that can deliver a service, and the levels of service that those resources can deliver in different environments.

NAF V3 Equivalency

Note that the MODAF SV-12a differs to the strict NAF definition in NSV-12, Service Provision, which only shows where systems contribute to services. In addition, certain parts of the NAF documentation refer to this view as NSV-13, System Service Provision.

Background

The Service Oriented Views (SOVs) in MODAF provide a specification of what a service is to do and how it presents its functionality to service consumers. The SOVs deliberately avoid specifying how a service is to be implemented, so that maximum creative flexibility is available to service providers. However, when a service is implemented (and its implementation specified in the architecture), it is useful to know what resources are used to implement it. SV-12a provides the mapping from services to the resources that provide those services. An SV-12a may also show the inverse relationship of when a resource uses a service.

Note that SV-12a does not describe actual deployed resources that deliver a service, it specifies typical configurations (i.e. templates) of resources that together can deliver the services.

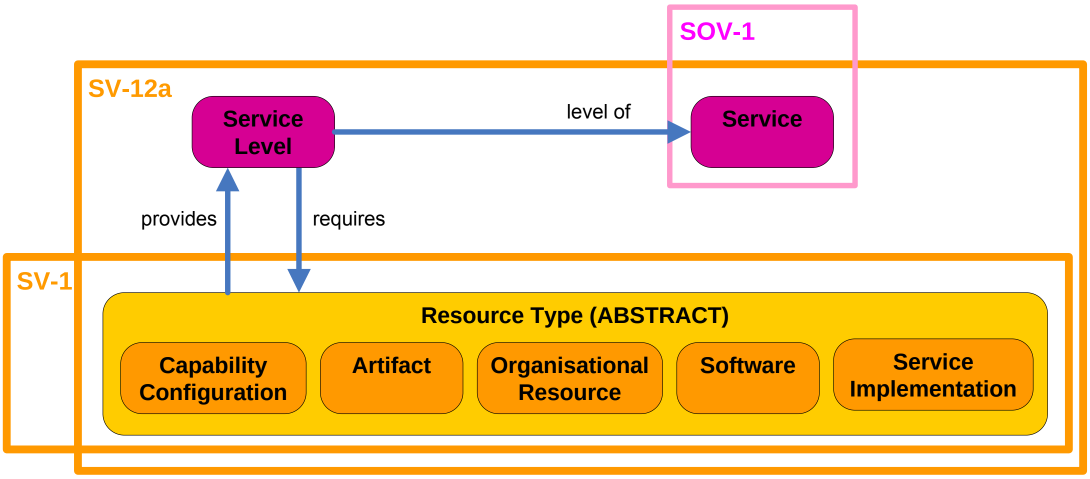
Usage

- Service implementation.
- Resource audit.
- Tracing business processes to the resources that support them.

Data objects

The data in an SV-12a can include:

- Service.
- Service Level
- Resource type.



Relationships Between Key Data Objects (Simplified from M3)

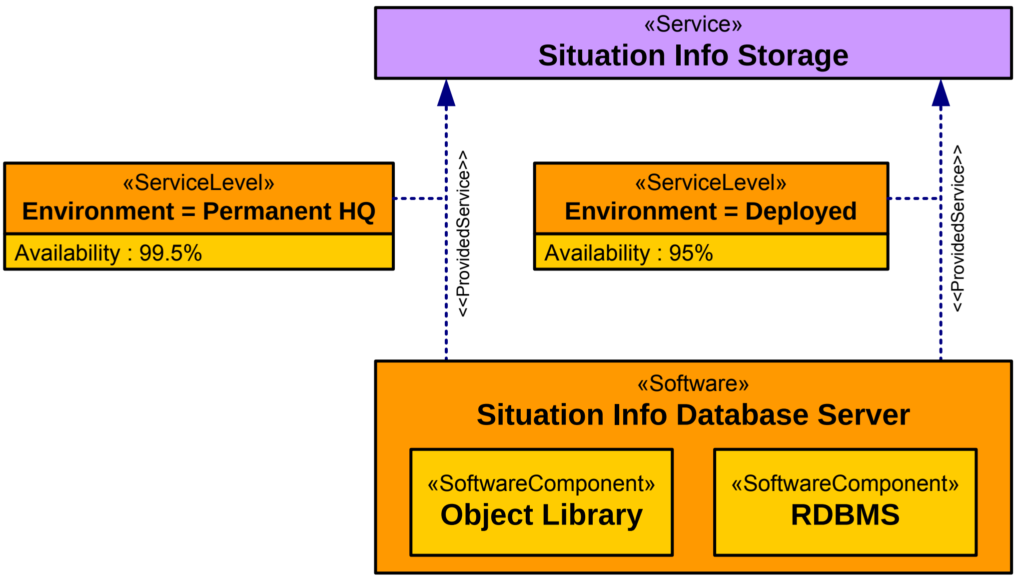
Representation

- Mapping (matrix).
- Topological – connected shapes.
- UML Composite Structure Diagram.
- SysML Blocks Diagram

Detailed Product Description

An SV-12a maps a resource (which may itself be constructed from other resources) to the services it can provide. SV-12a products are usually presented as a structural model (e.g. a UML composite structure), with tracing relationships to services. It is also be possible to present an SV-12a as a table, with services on one axis and resources on the other. Care should be taken with this approach, however, as it tends to hide any underlying structure the resources might have.

A given implementation may provide a different level of service depending on the environment in which is it used. The service attributes defined in SOV-1, Service Taxonomy, can be given values in an SV-12a and related to the environment under which those values are true.



UML Representation of SV-12a

SV-12b - Service Composition

The SV-12b specifies service implementations that use other services.

NAF V3 Equivalency

Note that the MODAF SV-12b differs to the NAF definition in NSV-12, Service Provision. The closest NAF view to SV-12b is NSOV-6, Service Composition (in NAF Chapter 5 version 3.1).

Background

The Service Oriented Views (SOVs) in MODAF provide a specification of what a service is to do and how it presents its functionality to service consumers. The SOVs deliberately avoid specifying how a service is to be implemented – even if that implementation is only based on other services – so as to maintain the principle of service opacity.

SV-12b specifies how a service implementation uses other services to provide its own service(s).

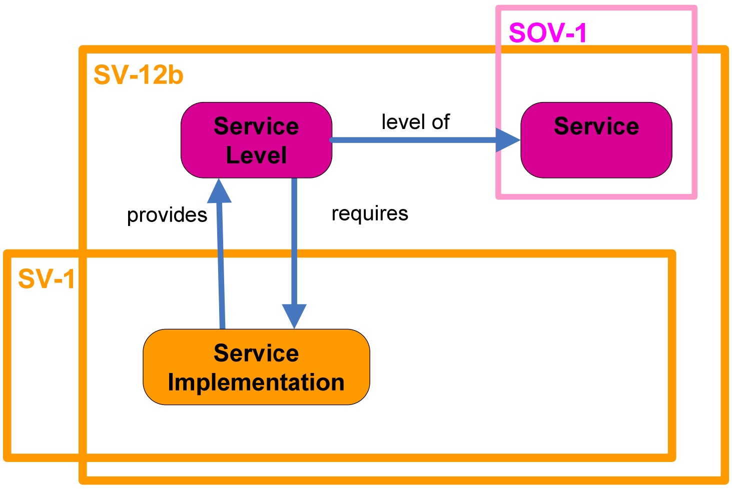
Usage

- Service implementation.
- Tracing business processes to the resources that support them.

Data objects

The data in an SV-12b can include:

- Service.
- Service Implementation.
- Service Level.



Relationships Between Key Data Objects (Simplified from M3)

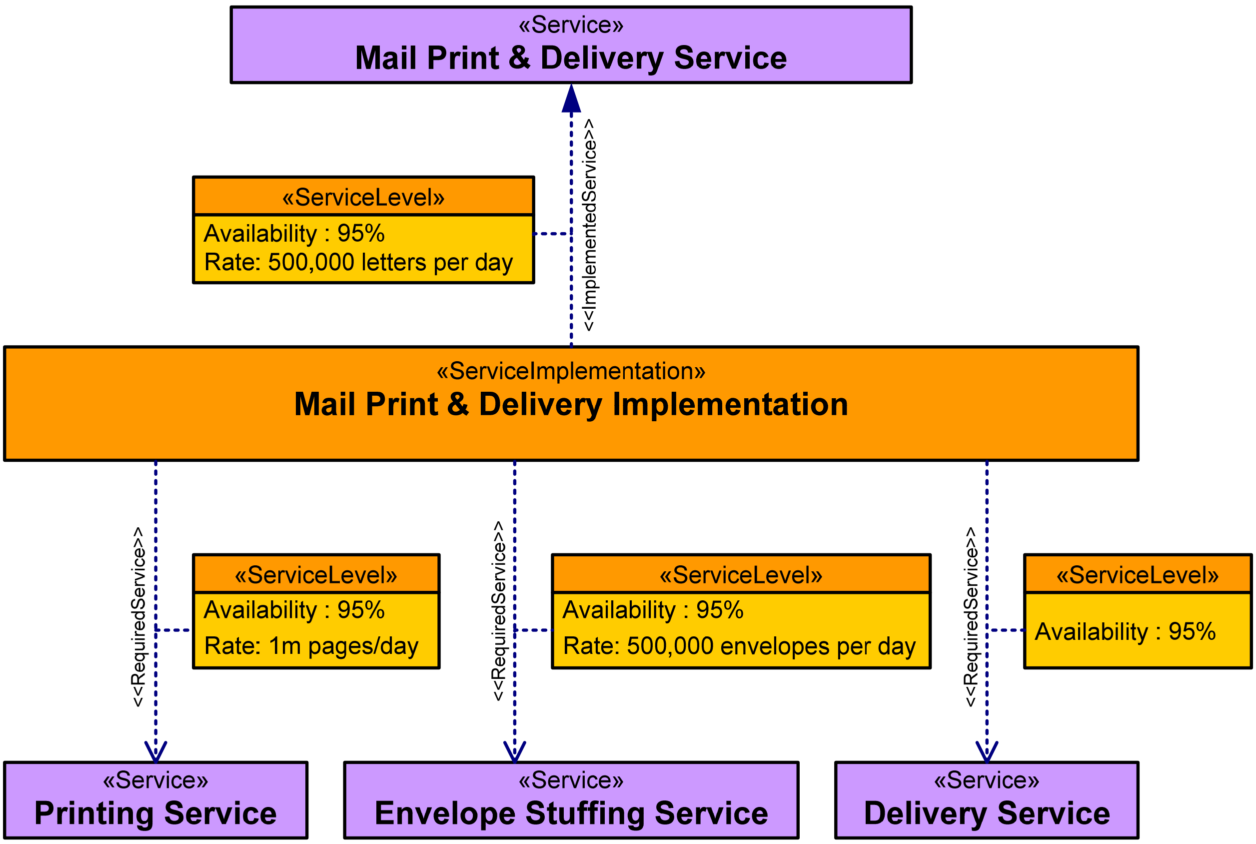
Representation

- Diagram.
- UML.

Detailed Product Description

An SV-12b shows what services are required by a Service Implementation in order to deliver one or more other services. This is effectively a composition of services.

In specifying the services the implementation requires, SV-12b will also specify the level of service required. Similarly, the level of service provided by the implementation will also be specified.



UML Representation of SV-12b

Technical standards view (TV) viewpoint

A description of and guidance for the use of the MOD Architecture Framework (MODAF) Technical Standards View Viewpoint.

The Technical Standards Views (TVs) are tabular views containing standards, rules, policy and guidance applicable to aspects of the architecture. The contents of the TVs do not necessarily need to be of a technical nature and can apply just as much to operational activities (e.g.) doctrine, Standard Operating Procedures and Tactics, Techniques and Procdures) as they do to systems (e.g. standards and protocols). The content of TVs will come from a number of sources including the policy setting organisations in MOD and core interoperability standards from The Sponsor.

- **TV viewpoint PDF [123.8 KB]**

Related pages

ViewpointsÂ andÂ views
Parent page

Views summary documents

All views viewpoint

Strategic view viewpoint

Operational view viewpoint

System view viewpoint

Acquisition view viewpoint

Service oriented view viewpoint

MODAF - detailed guidance

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page not yet rated

The MODAF Technical Standards Viewpoint

Viewpoint Summary

The Technical Standards Viewpoint details the standards, rules, policy and guidance that are applicable to aspects of the architecture.

The Technical Standards Viewpoint provides 2 Technical Standard Views (TVs) that present both technical and non-technical standards - i.e. TVs apply to systems (e.g. standards, and protocols), AND operational activities (e.g. operational doctrine and Standard Operating Procedures (SOPs)). In addition they can be used for other non-technical standards, e.g. those associated with industry process.

The elements contained in the TVs will come from a number of sources including the policy setting organisations in MOD, and interoperability standards from the Sponsor of the architecture activity. The TVs should then be managed and updated throughout the acquisition lifecycle by the standardisation officers within Delivery Teams.

Views

There are 2 TVs that make up the Technical Standards Views Viewpoint:

TV-1 - Standards Profile

TV-1 defines the technical and non-technical standards, guidance and policy applicable to the architecture. The standards specified in TV-1 may be traced to elements elsewhere in the architecture to indicate those elements conform to the standards.

Background

Standards are essential to the coherent running of businesses and to the delivery of reliable, interoperable systems. The TV-1 lists all the currently ratified standards that have been used throughout the architecture, and so acts as a checklist to help the architect ensure conformance.

A TV-2, Standards Forecast, should additionally be used if emerging standards have been identified and used in the architecture.

Usage

- Application of standards (informing project strategy).
- Standards compliance.

Data objects

The data in a TV-1 can include:

- Standard.
- Protocol.
- Ratification Body (the organisation that ratified the standard).
- Spectrum Allocation (standard ranges of RF spectrum, e.g. national frequency tables).

Representation

- Tabulation.

Detailed Product Description

A TV-1 view is typically a table showing the standards used throughout the architecture. Apart from the standard itself, the table may optionally show:

- o The version identifier of the standard (e.g. v1.1).
- o The ratification body responsible for the standard (e.g. ISO, NATO, MOD DEFSTAN, etc.).
- o The ratification date of the standard.
- o The URI of the website where the standard can be found.
- o The publisher of the standard, if different to the ratification body.
- o The elements in the architecture which conform to the standard.
- o Any other supporting information (e.g. service area), which would be captured as a comment on the standard in M3[1].

[1] MODAF Meta Model.

The standards need not be technical, and may be related to business or military doctrine, best practice, or even legislation.

Name	Version	Date	Ratification Body	Website
XMI	2.1	Sept 2005	Object Management Group	www.omg.org
UML	2.1	Apr 2006	Object Management Group	www.omg.org
IEEE1471	2000	Sept 2000	IEEE	www.ieee.org
MODAF	1.2	Apr 2008	UK MOD	www.mod.uk/modaf

Example of a Standards Profile (Technical)

Service Area	Service	Applicable Elements	Standard / Policy
ISTAR Governance	Operations	All fielded capability	SOPs
	Acceptance	All fielded capability	ISTAR Acceptance Procedure
MOD Strategy	Systems Engineering	SPECS 2 & interfaces	MOD Systems Engineering Management Plan (SEMP)
US Interoperability	Communications / Networking	SPECS 2 external US communications network interfaces	US Guidance Notes
Sustainment & Logistics	Ability to sustain capability	SPECS 2 & DLoD support	MOD Sustainment guidelines

Example of a Standards Profile (Operational)

The protocols referred to interface and communications descriptions (see SV-2) are examples of standards and these should also be included in the TV-1 listing, irrespective of which views they appear in, or are referred from.

TV-2 - Standards Forecast

The Standards Forecast is identical to a TV-1 Standards Profile except that the standards listed in TV-2 are not yet ratified (i.e. they are emerging or draft standards).

Background

The time from initial concept to fielded capability may be very long. It is therefore necessary to be able to refer to standards which, although not ratified at the time of producing the architecture, will have an impact on the capability. This could be anything from expected changes in legislation around spectrum management to future environment and safety standards. Being able to refer to emerging standards also enables the architect to mitigate the risk of outmoded specifications - so called "designed obsolescence".

Usage

- Forecasting future changes in standards (informing project strategy).
- Specifying standards that will have an impact on the architecture and the capability it is to deliver.

Data objects

The data in a TV-2 can include:

- Standard.
- Protocol.
- Ratification Body (the organisation that ratified the standard)
- Spectrum Allocation (standard ranges of RF spectrum, e.g. national frequency tables).

Representation

- Tabulation.

Detailed Product Description:

The TV-2 presents a table of draft or emerging standards that are likely to have a bearing on the architecture or the capability being architected. As with TV-1 the standards may be technical or business related, and may include future legislation. All the same information that TV-1 presents may also be shown:

- The version identifier of the standard (e.g. v1.1).
- The ratification body responsible for the standard (e.g. ISO, NATO, MOD DEFSTAN, etc.).
- The ratification date of the standard (in the case of TV-2, this is the forecast ratification date).
- The URI of the website where the standard can be found.
- The publisher of the standard, if different to the ratification body.
- The elements in the architecture which conform to the standard.
- Any other supporting information (e.g. service area), which would be captured as a comment on the standard in M3.

So, for all intents and purposes, most TV-2 products will look very similar to TV-1. It may, however be useful to present the TV-2 according to the dates (e.g. corresponding to Enterprise Phases) in which the standards are expected to be ratified. An example of this is shown below:

TRM Category	**Standards Forecast**		
	Short Term (1 Year)	**Mid Term (3 Years)**	**Long Term (5 Years)**
Application Platform			
Data Interchange Document Interchange	Security Marking DTD – in CAPCO coordination (proposed IC standard)		
Mapping	Geography DTD 2.0 – accepted by GIS Consortium	Commercial products that use the standard become available	
	Geospatial XSD – in coordination Open GIS		Geospatial XSD – accepted by Open GIS
Communications Electronic Mail		IETF RFC2060 Internet Mail Access Protocol (IMAP) – accepted, replaces *de facto* standard	
World Wide Web Services	IETF – Common Gateway Interface (CGI) 1.2 – becomes proposed standard		IETF – Common Gateway Interface (CGI) 1.2 – becomes *de facto* standard
			IETF – RFC 2818 HTTP Over TLS – accepted, replaces RFC 2616
Communications Transport Services		IETF – Wireless Extensions to TLS – becomes proposed standard	
		IETF – RFC 2002 IP Mobility Support – accepted	IETF – Ipv4 Mobile IP Protocol – becomes proposed standard
Security			IETF – RFC 2246 The Transport Layer Security (TLS) Protocol Version 1.0 – accepted; replaces SSL

Example Standards Forecast

This document is no longer extant and has been withdrawn.

Defence |

about defence

Acquisition view (AcV) viewpoint

A description of and guidance for the use of the MOD Architecture Framework (MODAF)Â Acquisition ViewÂ Viewpoint.

The Acquisition Views (AcVs) describe programmatic details, including dependencies between projects and capability integration across the Defence Lines of Development (DLODs). The Views identify interaction between programmes and projects, and integrate acquisition activities across all of the DLODs. The AcVs provide important programmatic information for those involved in capability management and acquisition. Since they also address the maturity across all of the DLODs to deliver an integrated military capability, the AcVs also form an important interface between the acquisition IPT and its Lead User community.

- **AcV viewpoint PDF [263.3 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

ViewpointsÂ andÂ views
Parent page

Views summary documents

All views viewpoint

Strategic view viewpoint

Operational view viewpoint

System view viewpoint

Technical standards view viewpoint

Service oriented view viewpoint

MODAF - detailed guidance

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page not yet rated

The MODAF Acquisition Viewpoint

Viewpoint Summary

The Acquisition Viewpoint depicts programmatic information, including project/programme structure, project ownership, dependencies between projects, and capability integration across the Defence Lines of Development (DLODs).

The Acquisition Viewpoint consists of 2 Acquisition Views (AcVs) that support capability management and acquisition by identifying the interactions between programmes, projects and integration activities across all of the DLODs.

Views

There are 2 AcVs Views that make up the Acquisition Viewpoint:

AcV-1 Acquisition Clusters

AcV-1 provides an organisational-based perspective on how projects can be grouped together to achieve coherent programmes.

Background

The AcV-1 depicts logical groupings of projects (into "clusters"), mapped to the organisations that manage them, in order to describe the portfolio of projects that contribute to a coherent acquisition programme. It provides a means of identifying the main dependencies between acquisition elements.

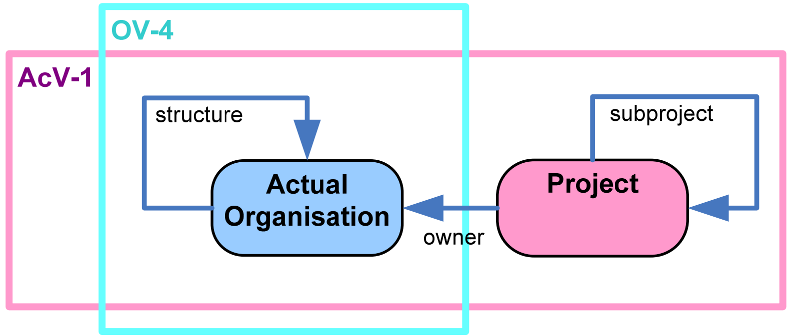
Usage

- Programme management (specified acquisition programme structure).
- Project organisation.

Data objects

The data in an AcV-1 can include:

- Project.
- Project Owner.
- Enterprise Phase.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Topological (Connected Shapes).
- 'Nested box' Diagram.
- UML Composite Structure Diagram.
- UML Class Diagram.

Detailed Product Description

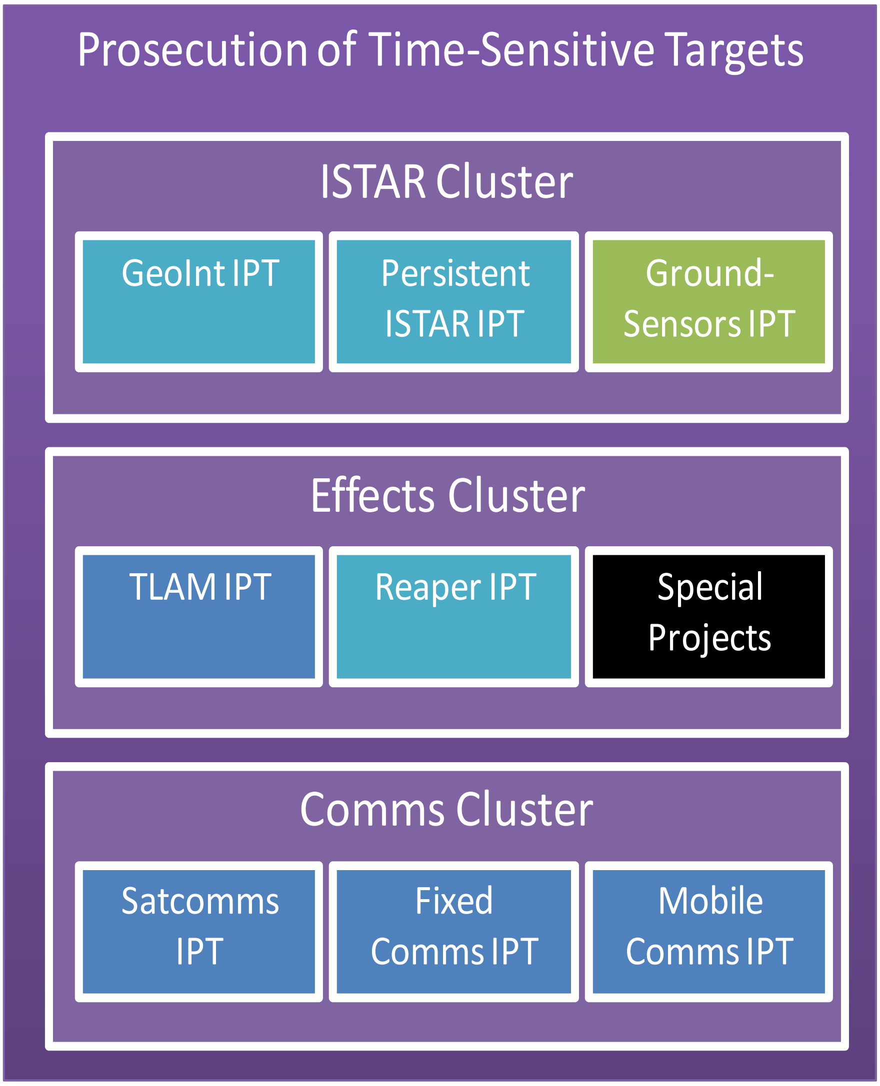
The AcV-1 provides a way of describing the organisational relationships between multiple acquisition projects that deliver individual systems or capabilities as part of a wider programme. Consequently, it generally will not be developed by those building Architectures for an individual project.

In essence, AcV-1 is an organisational breakdown consisting of actual organisations (see OV-4, Operational Relationships Chart) in a hierarchical structure which groups the projects they manage to form acquisition clusters.

The view is strongly linked with StV-4, Capability Dependencies, which shows capability clusters and dependencies.

The intent of an AcV-1 is to show:

- All of the acquisition projects delivering capability within the programme of interest.
- The structure of acquisition projects.



Example of AcV-1

An AcV-1 is specific to a particular Enterprise Phase – i.e. the structure of programmes may change over time as projects are merged, axed and the capability portfolio rebalanced. Hence, it is possible that an acquisition programme could have more than one AcV-1, each showing how the acquisition clusters are arranged for relevant Enterprise Phases.

AcV-2 - Programme Timelines

The AcV-2 provides a time-based perspective of programmes.

Background

The AcV-2 is primarily intended to support the acquisition process across multiple projects or programmes, including the management of dependencies between projects and the integration of all the DLODs to achieve a successfully integrated military capability.

Use of AcV-2 should support the management of capability delivery and be aligned with the StV-3, Capability Phasing.

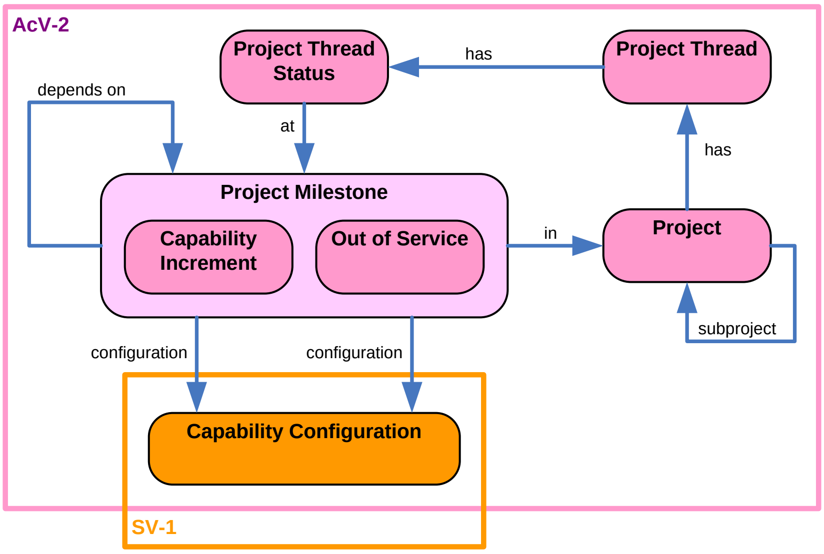
Usage

- Project management and control (including delivery timescales).
- Project dependencies and the identification of associated risk.
- Portfolio management.
- Through Life Management Planning (TLMP).

Data objects

The data in an AcV-2 can include:

- Projects.
- Project Milestones (which may indicate capability increments or capability configurations going out of service).
- Project Threads (e.g. DLOD).
- Project Dependencies.
- Capability Configurations.



Relationships between Key Data Objects (Simplified from M3)

Representation:

- Timeline View.
- Augmented Gantt Chart

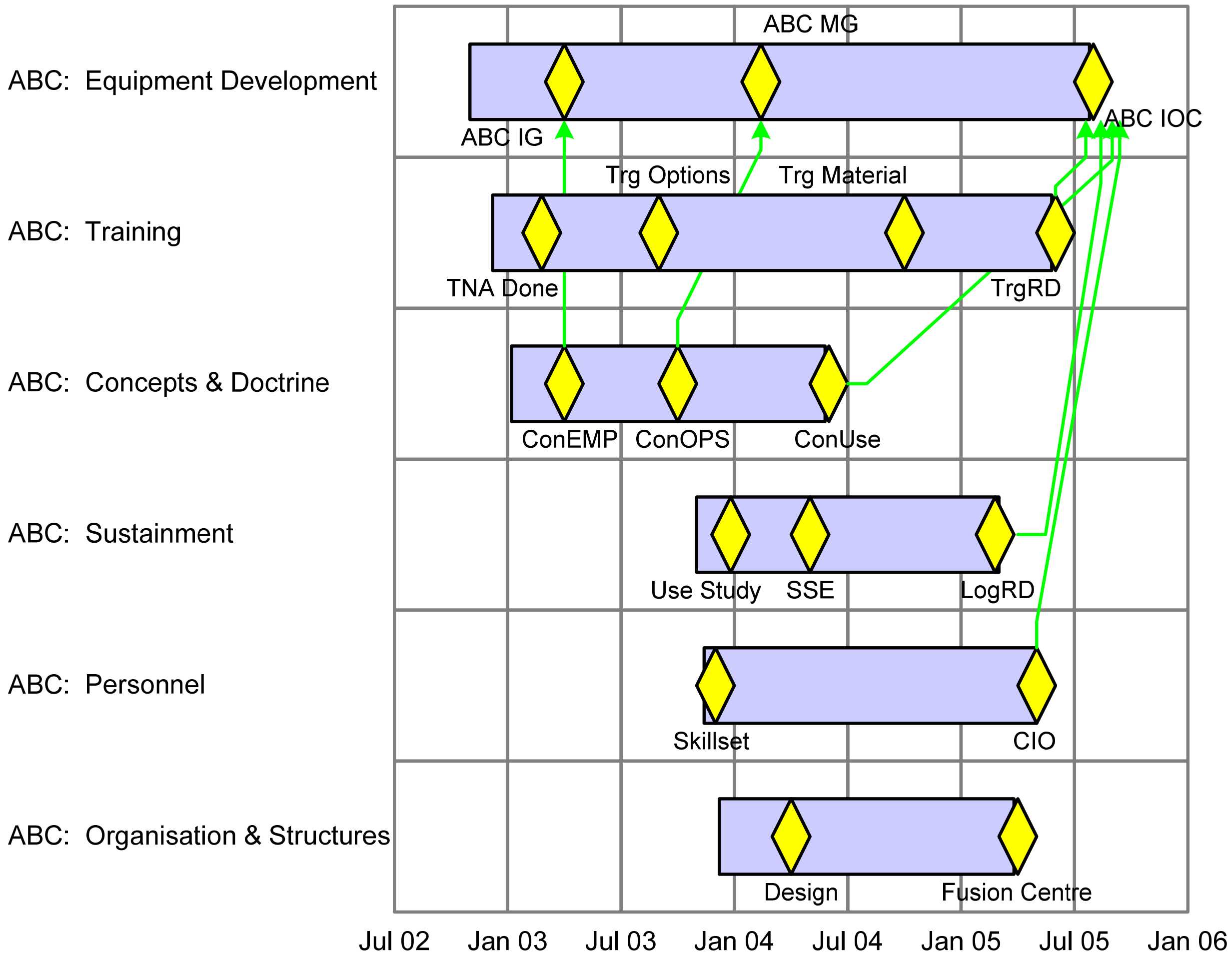
Detailed Product Description

The AcV-2 provides an overview of a set of individual projects (e.g. within a given programme), based on a time-line. Projects may be broken into threads (work streams) to show the dependencies at a lower level. For capability-based acquisition, these threads could conveniently be equated with MOD's Defence Lines of Development (DLODs) – though MODAF does not mandate the use of DLODs.

Where appropriate, the AcV-2 may also summarise the level of maturity achieved across the threads at given milestones in each project.

The information provided by the AcV-2 can be used to determine the impact of either planned or unplanned programmatic changes, and highlight opportunities for optimisation across the delivery programme. The inclusion of the DLOD information enables the Delivery Team to identify potential delays which impact on the delivery of capability. Action to address areas of concern identified in one or more thread, (e.g. a shortfall in training resource) can be co-ordinated across a programme or group of projects. In particular, the knowledge of the dependencies between project milestones can quickly inform a programme manager when a slippage in one project is likely to cause delays in other projects.

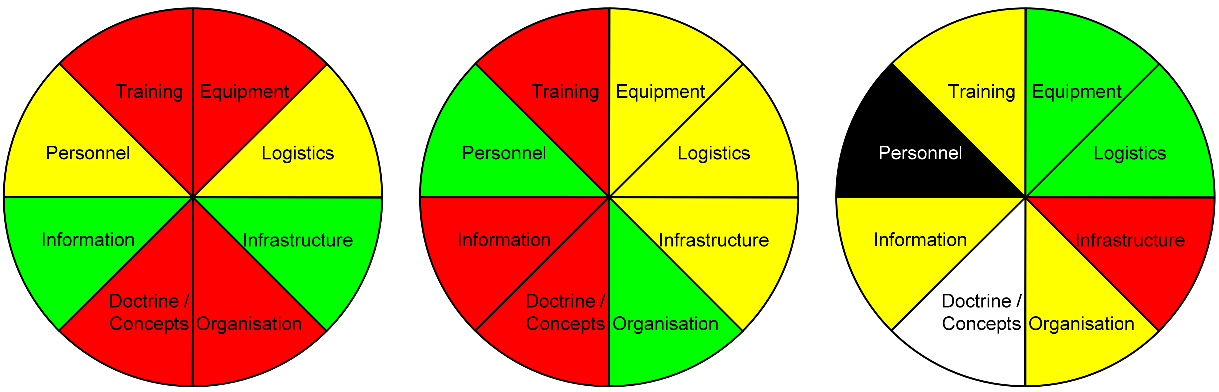
The presentational format for an AcV-2 Product is a Gantt chart optionally augmented with symbols that show the status of each thread at given milestones, and the dependencies between them. The example below shows an AcV-2 depicting dependencies between project milestones, but *not* showing the thread status icons for the milestones.



Example AcV-2

Where AcV-2 differs significantly from a standard Gantt chart is in its use of project thread status indicators. The architect can identify a set of standard threads that run through all projects – e.g. the DLOD. For each of these threads, there can be a status indicator at any given project milestone. This is usually achieved using colour-coding so that stakeholders can tell, at a glance, the status of a given project at different points in time. In the example below, based on the MOD DLOD, colour coding has been used as follows:

- Green cells indicate that there are no outstanding issues or areas of concern and that the DLOD is at a level of maturity appropriate to the stage of the lifecycle.
- Yellow cells indicate that there are outstanding issues or areas of concern, but that there are planned activities that will provide resolution in the required timescale and the DLOD is at a level of maturity appropriate to the stage of the lifecycle.
- Red cells indicate that there are outstanding issues or areas of concern, for which there are no planned activities that will provide resolution in the required timescale, or that the DLOD is below the level of maturity appropriate to the stage of the acquisition lifecycle.
- White cells indicate that the DLOD relevance is not known at this time.
- Black cells indicate that the DLOD is not required.

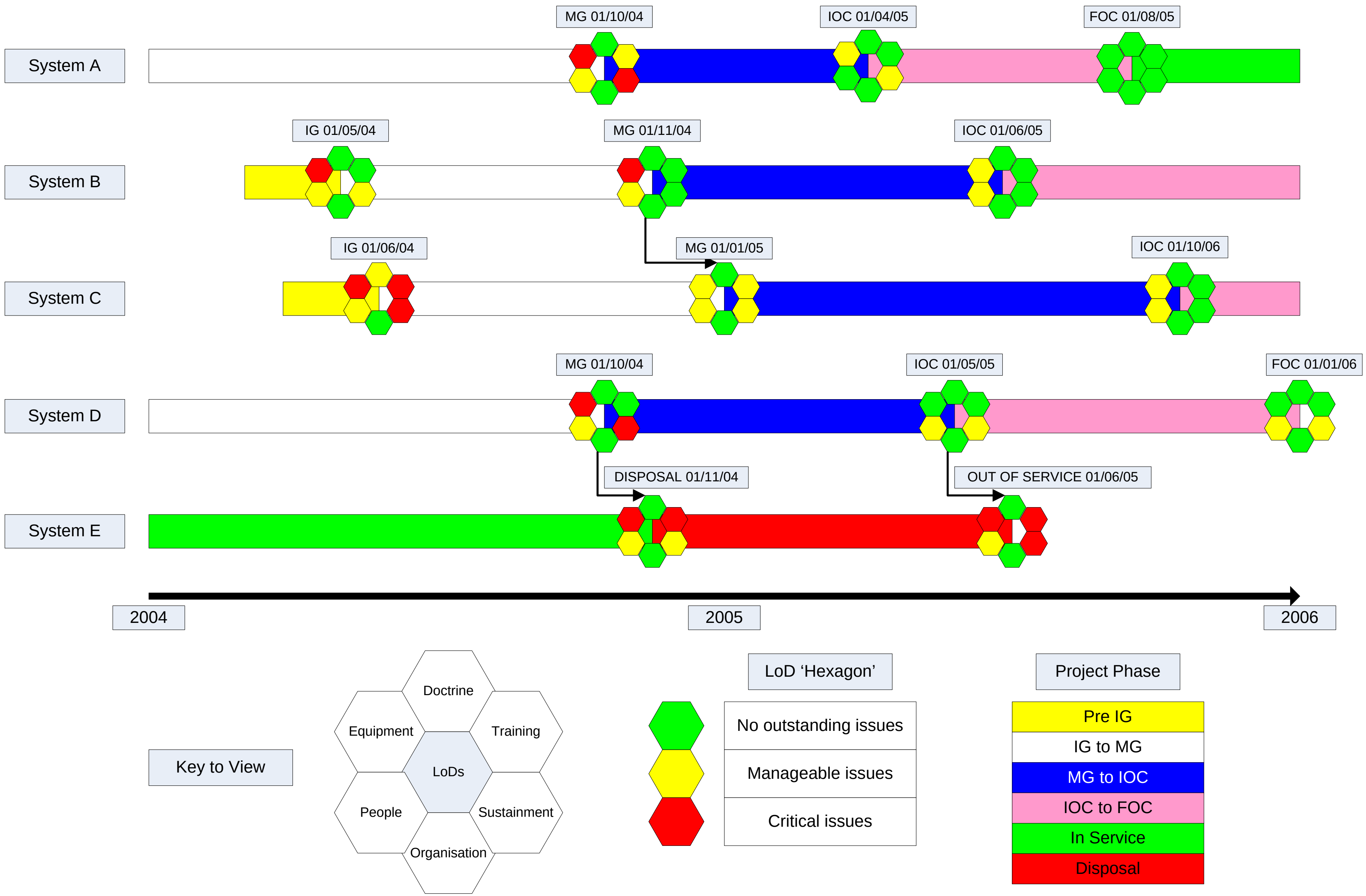


DLOD Status for AcV-2

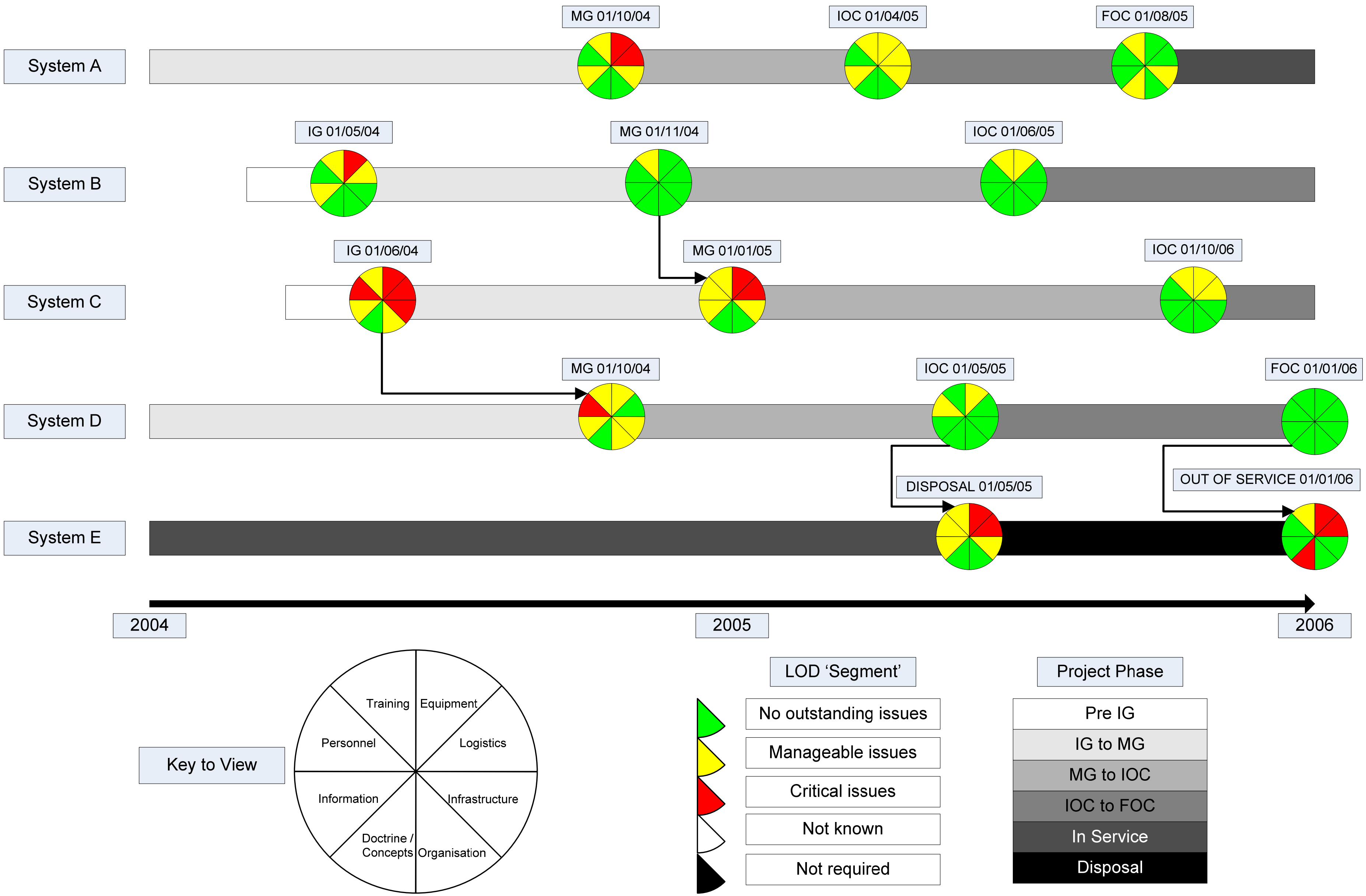
AcV-2 does not mandate a standard set of colour codes or even standard threads or statuses; however, it is likely that UK MOD acquisition projects will use the DLODs.

Example templates for this form of AcV-2 can be found on pages 7 and 8.

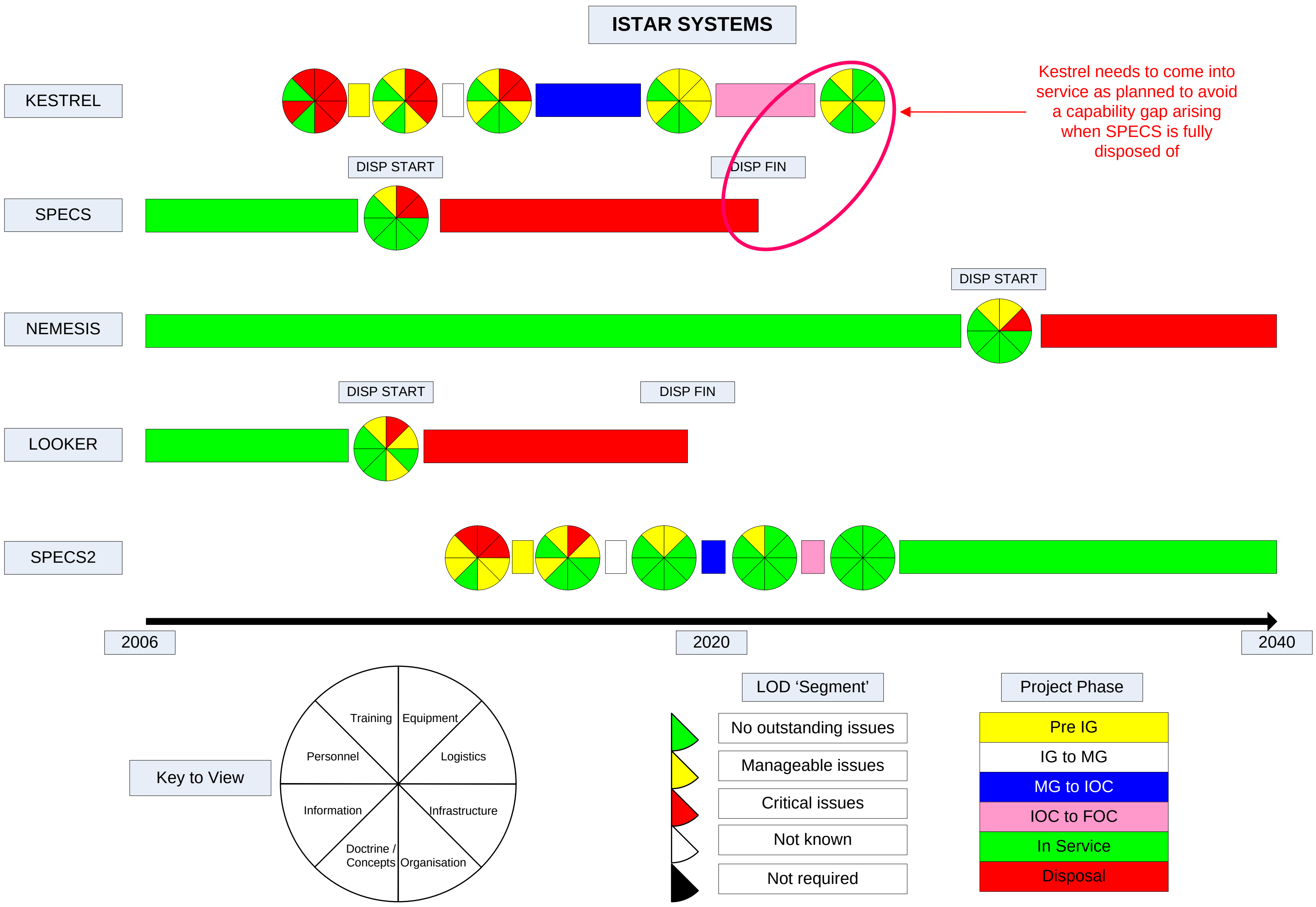
In support of the management of a programme of acquisition projects, it is desirable to have a set of common milestones at the programme level against which the maturity of each constituent projects is judged. MODAF does provide the flexibility to define individual milestones at the project level if required. An example of programme synchronisation is given at page 9. It is expected that UK MOD acquisition programmes will at least feature the standard OGC gateways as milestones on an AcV-2.



AcV-2 Template (with DLOD *Status)*



Example AcV-2 (with DLOD Status)



Example AcV-2 (Programme Synchronisation)

Service oriented view (SOV) viewpoint

A description of and guidance for the use of the MOD Architecture Framework (MODAF) Service Oriented View Viewpoint.

The Service-Oriented Views (SOVs) are a set of views that specify services that are to be used in a Service-Oriented Architecture (SOA). In MODAF terms, services are an implementation-independent specification of a packaged element of functionality. The views describe the specification of these services, how services are orchestrated together for a purpose, the capabilities that services deliver and how services are implemented. Note that the views do not focus on the detailed design of the service, rather on the requirement the service fulfils.

- **SOV viewpoint PDF [352.8 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

ViewpointsÂ andÂ views
Parent page

Views summary documents

All views viewpoint

Strategic view viewpoint

Operational view viewpoint

System view viewpoint

Technical standards view viewpoint

Acquisition view viewpoint

MODAF - detailed guidance

Related links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page not yet rated

★★★★★

The MODAF Service Oriented Viewpoint

Viewpoint Summary

The Service Oriented Viewpoint provides a perspective that enables the specification of services[1] which are to be used in a Service Oriented Architecture (SOA).

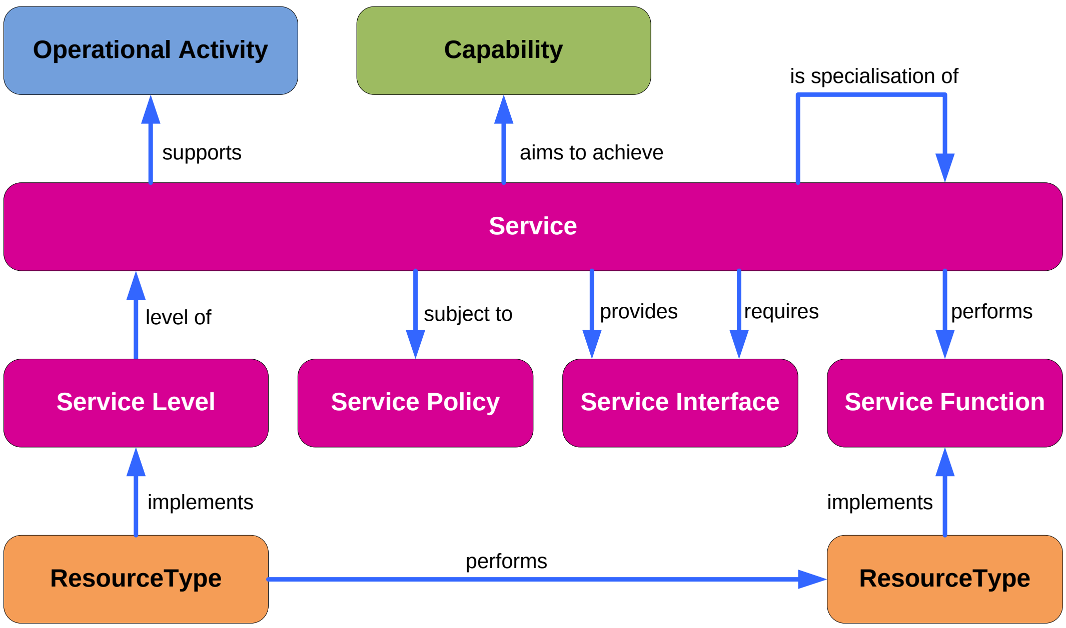
The viewpoint consists of 7 Service Oriented Views (SOVs) that specify the services used in an architecture (their behaviour and the interfaces they provide and require), the capabilities that the services deliver, and the policy governing the use of the services. It should be noted that the views do not focus on the detailed implementation of the service, but on the requirement the service fulfils – i.e. there may be many different implementations of a given service that is specified in the SOVs.

The SOVs are based on the same meta model elements as the NATO Architecture Framework (NAF) v3.1. However, a different set of views has been defined for MODAF. The definition of the MODAF SOVs include equivalency statements to indicate where the MODAF views differ from those in NAF.

In addition, service elements have been added to some existing MODAF views, specifically:

- OV-2, Operational Node Relationship Description.
- OV-5, Operational Activity Model – used for service orchestration.
- OV-6c, Operational Event-Trace Description.
- SV-5, Function to Operational Activity/Service Function Traceability Matrix.
- SV-12a&b, Service Composition and Implementation.

The diagram below summarises how services are linked to other elements in MODAF.



It should be noted that the SOVs are intended for specifying services for use in an SOA (i.e. loose-coupled, opaque service specifications). Services should not be used for simple interface management – SV-1 provides the concept of Resource Interface for this purpose.

[1] In MODAF terms, a service is an implementation-independent specification of a packaged element of functionality.

Views

There are 7 SOVs, including sub-views, that make up the Service Oriented Viewpoint:

SOV-1 - Service Taxonomy

The SOV-1 specifies a hierarchy of services. The elements in the hierarchy are M3 Services (i.e. service specifications rather than service implementations), and the relationships between the elements are specialisations – i.e. one service is a special type of another. Service attributes, interfaces and constraints are inherited down a service taxonomy – e.g. if Service A is a specialisation of Service B then it also inherits all the features of Service B.

NAF V3 Equivalency

The equivalent NAF v3 view to SOV-1 is NSOV-1, Service Taxonomy.

Background

The purpose of an SOV-1 is to provide a governance structure for a Service-Oriented Architecture. Along with SOV-2, Service Interface Specification, it defines a standard library of service specifications for an enterprise, which service implementers are expected to conform to.

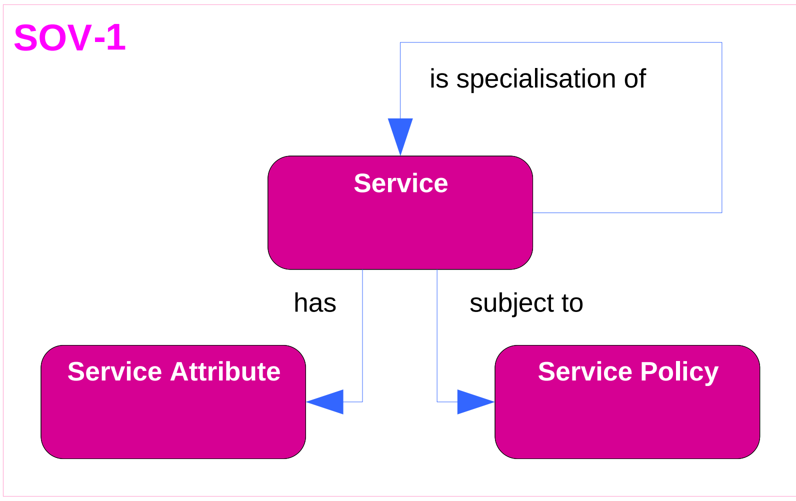
Usage

- SOA governance.
- Identification of services.
- Service planning.
- Service audit.
- Service gap analysis.
- Providing reference services for architectures.
- Tailoring generic services for specific applications.

Data objects

The data in an SOV-1 can include:

- Service.
- Service Generalisation (the specialisation relationship).
- Service Attribute.
- Service Policy (optional, also shown in SOV-4a, Service Constraints).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabulation.
- Hierarchical (connected shapes).
- UML class diagram.

Detailed Product Description:

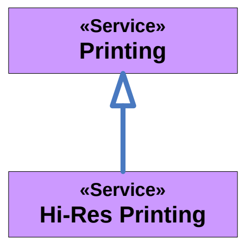
In MODAF terms, a service is an implementation-independent specification of a packaged element of functionality and/or capability.

There is, however, potential for confusion between services and capabilities. To help clarify this:

> The key indicator of a service is that it provides a standard interface to consumers. This means that services may be used as "wrappers" for one or more capability in order to provide a standard method of access to the capability. A well-designed service taxonomy provides a set of specifications for capability providers to adhere to.

An SOV-1 depicts services, specialisation relationships between services, service attributes and service policy (i.e. constraints). A service taxonomy persists over time (an architect may wish to specify historical, current or future services) and may be referenced by multiple architectures.

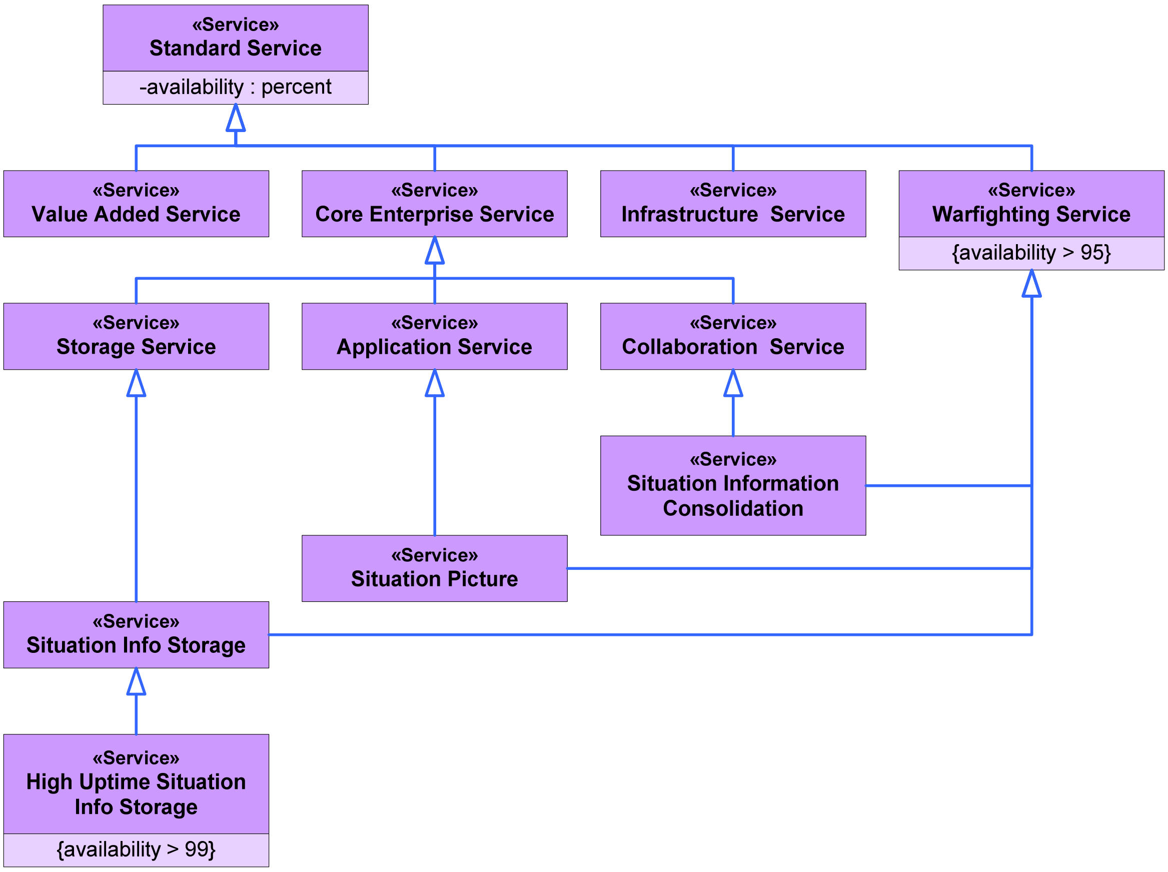
In SOV-1, services are only defined in the abstract, i.e. SOV-1 does not specify how a service is to be implemented. An SOV-1 is structured as a specialisation hierarchy of services, with the most general at the root and most specific at the leaves.



Example of a Specialisation Relationship

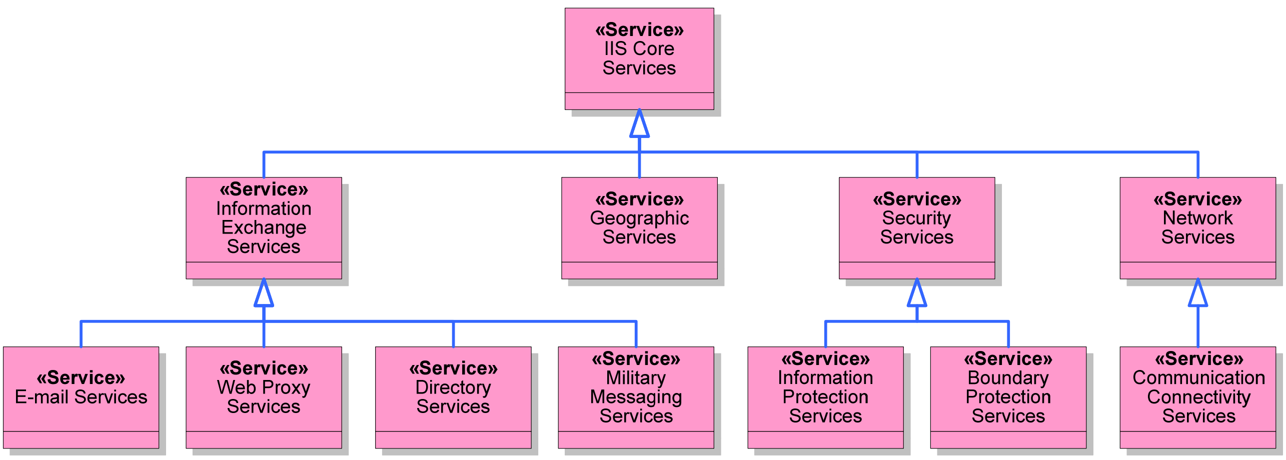
In contrast to AV-2, Integrated Dictionary, an SOV-1 is structured using only one type of specialisation relationship between elements: super-subtype. A super-subtype relationship is a relationship between two classes with the second being a pure specialisation of the first. Any service that specialises from another must implement all the functionality of its parent, and provide all the same input and output interfaces of its parent; in other words, any specialised service shall be entirely compatible with its parent (however, it may add functionality and interfaces). For example, if a service, "Printing", requires input of paper size and ink colours, a service, "Hi-Resolution Printing", that specialises from it must also accept these parameters and produce equivalent behaviour when initiated.

Services may have attributes and constraints (service policy) defined against them. Attributes are inherited by specialised services. Where an attribute is specified for a service, implementations of that service shall specify their values for the attribute. The example below shows an availability attribute defined against the top service. All other services inherit that attribute, and the WarfightingService sets a constraint (service policy) that the availability shall be greater than 95%. This policy is then inherited by the three situational awareness services. Note that policy may be overridden in specialised services.



Sample Service Hierarchy

UML is a useful modelling language in which to develop service taxonomies as the object oriented approach naturally includes the concept of generalisation-specialisation.



UML Sample Service Hierarchy

SOV-2 - Service Interface Specification

The SOV-2 defines the interfaces provided and required by a service.

NAF V3 Equivalency

The equivalent NAF v3 view to SOV-2 is NSOV-2, Service Definitions. However the MODAF view is more restricted in that it only specifies the service interfaces (attributes are specified in SOV-1).

Background

A service presents one or more interfaces to consumers (a "consumer" being any agent capable of using the service; i.e. a person, an organisation, a system or another service). A service may also be capable of using interfaces exposed by other services, and the architect may specify these as used interfaces.

Specifying the interfaces that a service provides and requires defines compatibility between services - e.g. if Service A provides interface X, and Service B can use Interface X, then Service B can call upon at least some of the functionality of Service A.

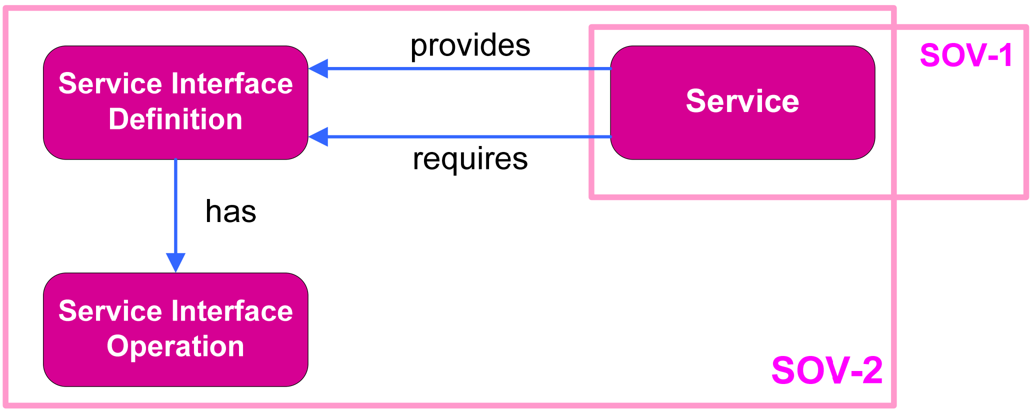
Usage

- SOA governance.
- Detailed service specification.
- Service interoperability.

Data objects

The data in an SOV-2 can include:

- Service.
- Service Interface.
- Service Interface Operation.
- Service Interface Parameter.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabular.
- UML.

Detailed Product Description

Service interfaces are defined in terms of their operations (methods of access) and parameters (data that must be passed to the service, or produced by the service). The interfaces and their operations all have names. Parameters may be simple types (text, numbers, Boolean) or typed by an entity that shall be specified in a Physical Schema (SV-11).

SOV-2 products can be presented in tabular form, or as diagrams (usually UML). If a tabular approach is used, the first five columns are mandatory and should be in the following order:

1) Service Name.
2) Interface Name.
3) Provided / Used (note that if the service both provides and uses a particular interface, this should be recorded as two entries in the table).
4) Operation Name.
5) Parameters.

Service	Interface	Provided / Used	Operation	Parameters
Situation Information Consolidator	Situation Information Submission	P	setAreaOfInterest	Geographic location
			submitLocationOfResource	Resource, geographic location, track
	Situation Information request	P	requestPictureForArea	Geographic location, situation picture
			requestLocationOfResource	Resource, geographic location
	Store	U	storeInformation	Situation info package
	Retrieve	U	retrieveInformation	Geographic location, time, situation info package
Situation Information Storage	Store	P	storeInformation	Situation info package
	Retrieve	P	retrieveInformation	Geographic location, time, situation info package
Situation picture	…			

SOV-2 Tabular Representation

If presented in UML format, the information above would look like this:

SOV-2 UML Representation

SOV-3 - Capability to Service Mapping

The SOV-3 specifies the capabilities that services provide.

NAF V3 Equivalency

The NAF v3 equivalent view is NCV-7, Capability to Service Mapping.

Background

An SOV-3 presents a simple mapping of services to capabilities, showing which capabilities a given service provides.

Note that in MODAF v1.2.003, the semantics of the relationship between services and capabilities was unclear. If more than one service mapped to a given capability, it was not clear if each of those services provided the capability in and of itself, or if all the services were required together in order to deliver the capability.

From v1.2.004, the relationship between capability and service indicates that the service provides the capability. If more than one service maps to a capability, each of those services provides the capability in and of itself. Should the architect wish to express the need to bring together multiple services to deliver a capability, this should be expressed in OV-2, Operational Node Relationship Description, (mapping the capability to a node) and OV-5, Operational Activity Model (orchestrating the services against the activities performed by the node).

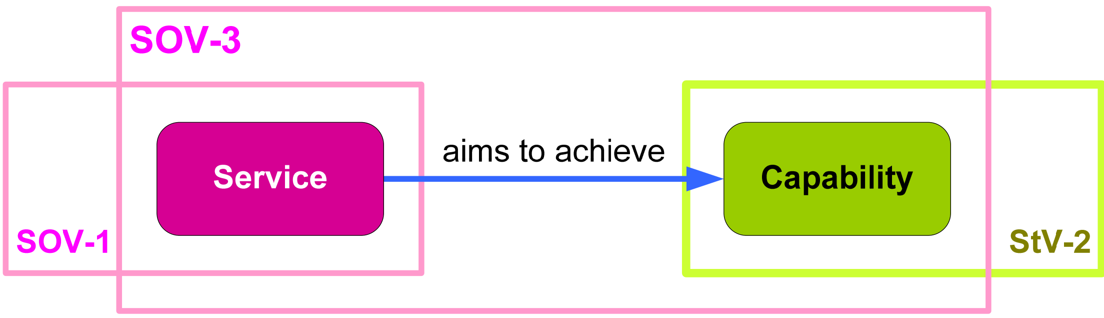
Usage

- Service specification & planning.
- Governance.

Data objects

The data in an SOV-3 can include:

- Service.
- Capability.
- Service Aims to Achieve (relationship from Service to Capability).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabular.
- UML.

Detailed Product Description

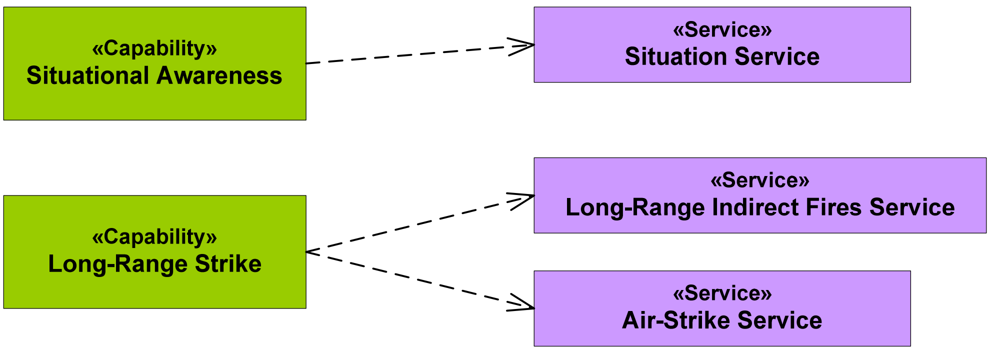
An SOV-3 can be presented as a matrix with capabilities on one axis and services on the other.

	Services		
Capabilities	Long Range Indirect Fires Service	Air-Strike Service	Situation Service
Long-Range Strike	X	X	
Situational Awareness			X

Tabular Representation of SOV-3

The relationship between capability and service is many-to-many. A given service may provide one or more capabilities, or a given capability may be provided by more than one service. Note that if a combination of services are required deliver a certain capability, this should be modelled using a combination of OV-2 (to map the capability to a node) and OV-5 (to orchestrate the services against activities performed by the node).

Alternatively, SOV-3 can be presented as a diagram showing tracing relationships from capabilities to services.



UML Representation of SOV-3

SOV-4a - Service Constraints

The SOV-4a specifies constraints that apply to implementations of services.

NAF V3 Equivalency

SOV-4a has no direct equivalent in NAF v3, though service policy constraints can be shown in NSOV-1, Service Taxonomy.

Background

To better enable consistency and re-use of service specifications, it is important to set constraints on how a service should behave. An SOV-4a specifies constraints against services to which implementations of must conform.

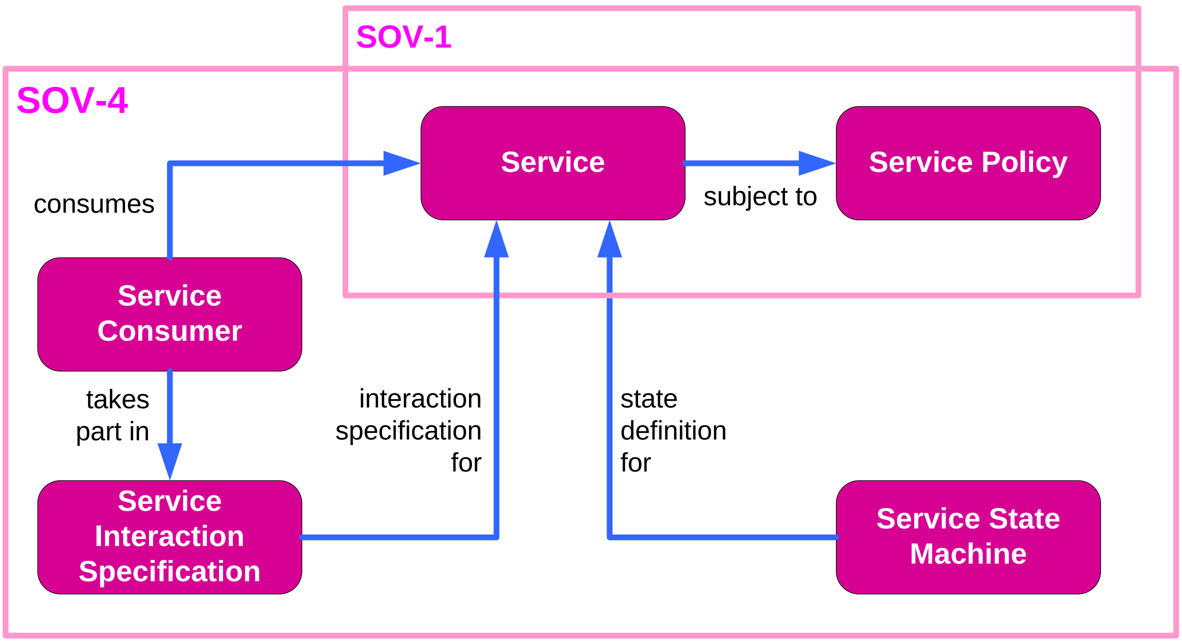
Usage

- Service specification.
- Service governance.

Data objects

The data in an SOV-4a can include:

- Service.
- Service Policy.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabular.
- UML.

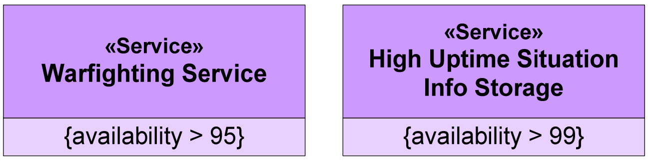
Detailed Product Description

SOV-4a products are usually tabular, with services as rows and constraints as columns.

	«ServiceConstraint»	
«Service»	availability	
Warfighting Service	> 95%	
High Uptime Situation Info Storage	> 99%	

SOV-4a tabular representation

It is also possible to present the constraints as adornments to services in a diagram (e.g. a compartment in a UML class).



SOV-4a UML representation

SOV-4b - Service State Model

The SOV-4b specifies the possible states a service may have, and the possible transitions between those states.

NAF V3 Equivalency

SOV-4b has no direct equivalent in NAF v3.

Background

In specifying a service, it is often necessary to specify the allowable states so as to constrain how implementations of the service will behave. SOV-4b is a specification of those states, and the possible transitions between them.

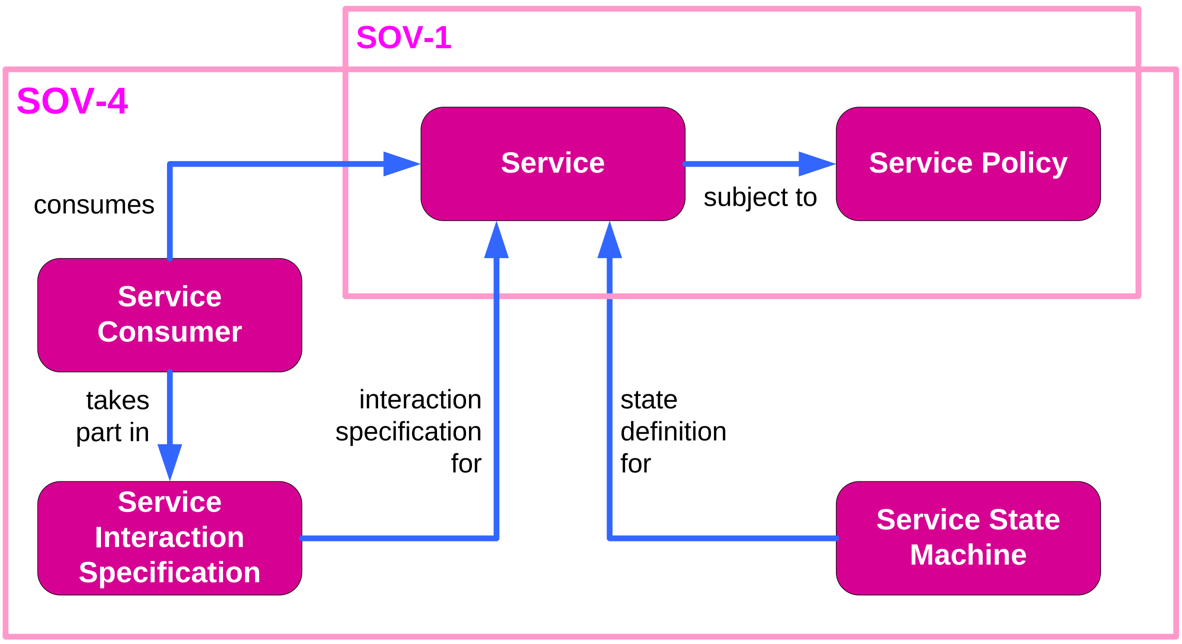
Usage

- Service specification.

Data objects

The data in an SOV-4b can include:

- Service.
- Service State Machine.



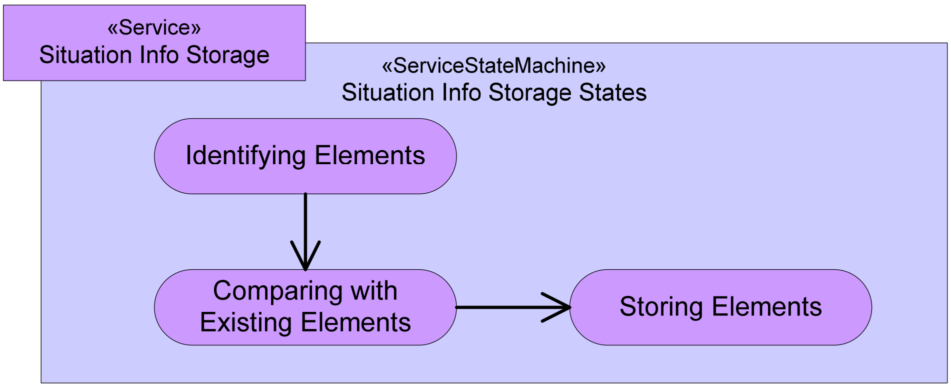
Relationships between Key Data Objects (Simplified from M3)

Representation

- UML.
- Other state transition models.

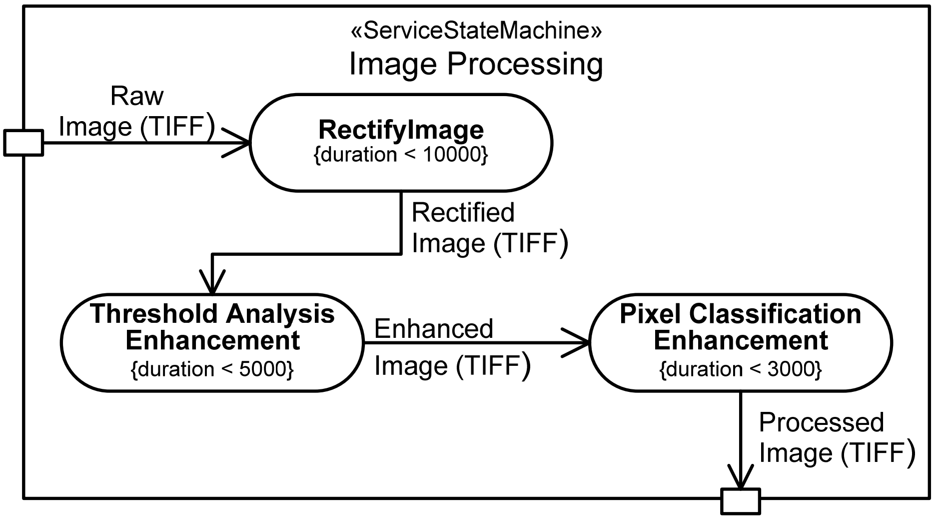
Detailed Product Description

SOV-4b products are usually UML (or similar) state transition models.



SOV-4b state transition model representation

An SOV-4b may also specify performance constraints (ie the maximum duration a service may be in a particular state):



SOV-4b showing performance constraints

SOV-4c - Service Interaction Specification

The SOV-4c specifies how a service interacts with external agents, and the sequence and dependencies of those interactions.

NAF V3 Equivalency

The equivalent NAF view is NSOV-5, Service Behaviour.

Background

The purpose of the SOV-4c is to specify the general sequence of interactions that are possible for a given service.

Usage

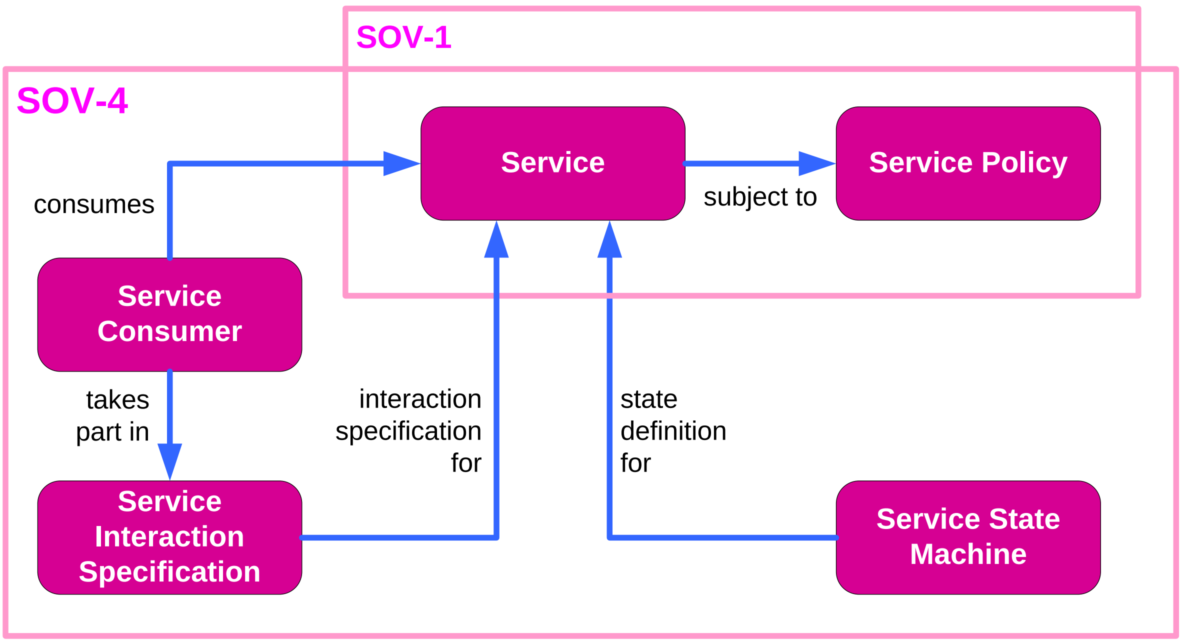
- Service specification.

Data objects

The data in an SOV-4c can include:

- Service.
- Service Interface.
- Service Lifeline.
- Service Consumer

.



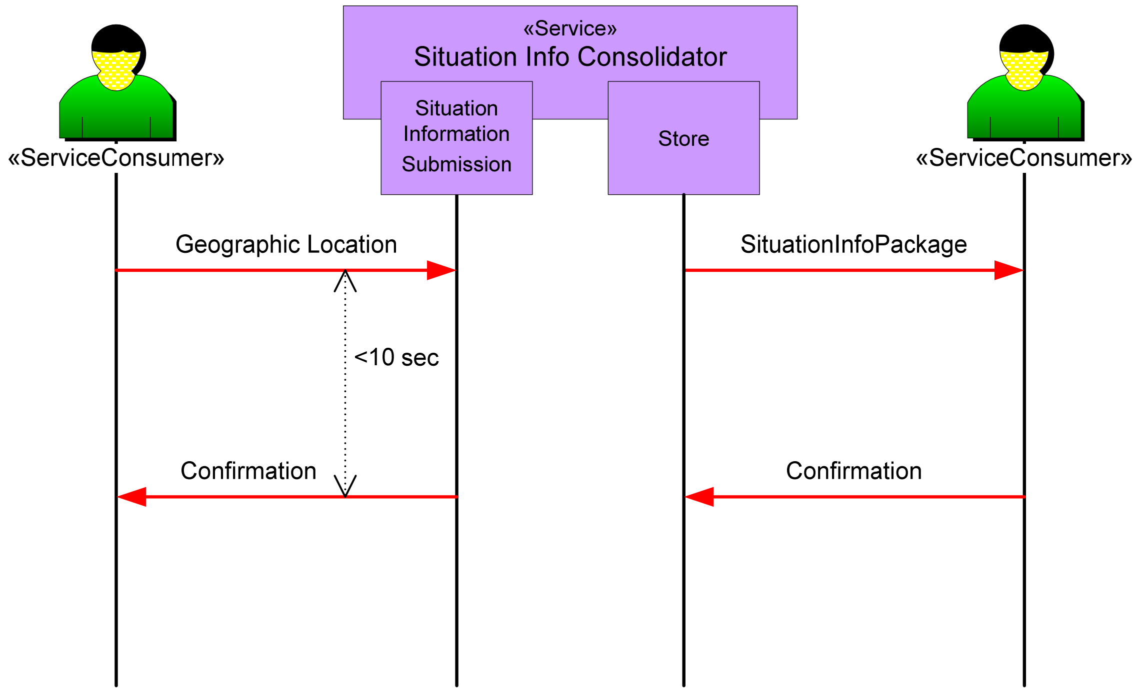
Relationships between Key Data Objects (Simplified from M3)

Representation

- UML.

Detailed Product Description

SOV-4c specifies how a service interacts with any given agent that has compatible interfaces. The representation is usually a UML Sequence Diagram.



SOV-4c Sequence Diagram representation

The product shows a service and the interfaces it exposes – quite often the diagrams can be cluttered if all interfaces are shown, so it is advisable to consider producing multiple SOV-4c products for a given service. Each interface in an SOV-4c has a “lifeline” to which messages are shown passing. It is also possible to show timing constraints between messages.

SOV-5 - Service Functionality

The SOV-5 defines the behaviour of a service in terms of the functions it is expected to perform.

NAF v3 Equivalency

The NAF v3 equivalent view is NSOV-5, Service Behaviour.

Background

SOV-5 is the key behavioural specification for services. Equivalent in nature to OV-5, Operational Activity Model, and SV-4, Functionality Description, it specifies a set of functions that a service implementation is expected to perform.

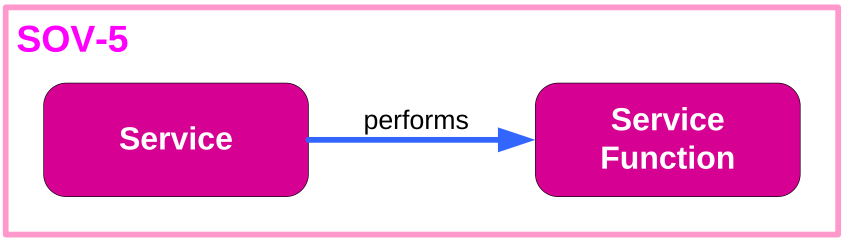
Usage

- Service specification.
- Functional requirements definition.

Data objects

The data in an SOV-5 can include:

- Service.
- Service Function.



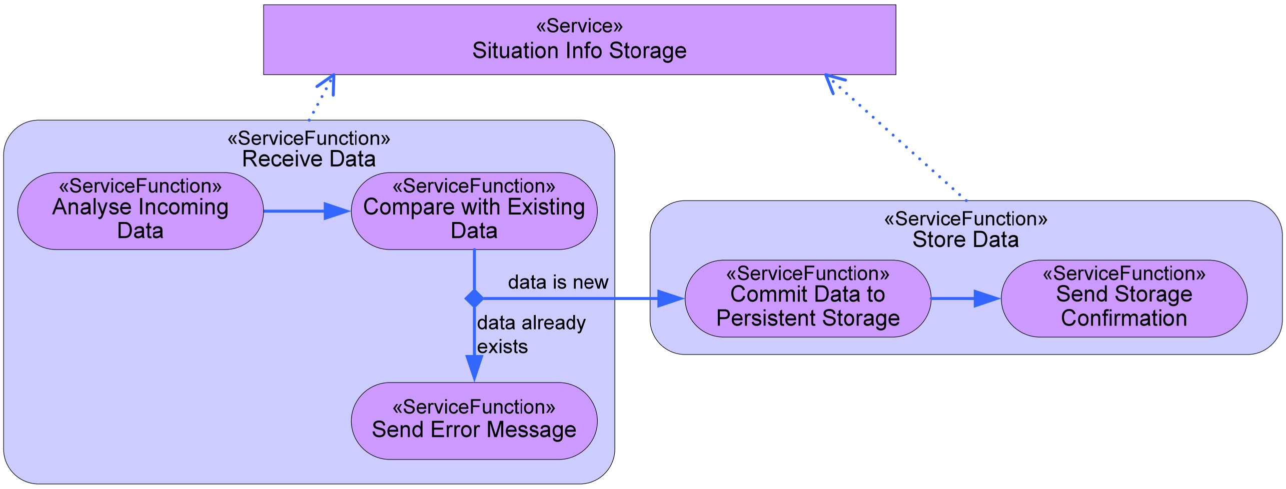
Relationships between Key Data Objects (Simplified from M3)

Representation

- Diagram.
- UML.

Detailed Product Description

An SOV-5 specifies the required functionality that an implementation of a service is expected to have; (the implementation of that behaviour is represented in SV-4, Functionality Description and SV-5, Function to Operational Activity / Service Function Traceability Matrix). An SOV-5 is usually presented as a functional diagram, with optional flows. An SOV-5 product should also show which service the functions correspond to.



Representation of SOV-5

Note that an SOV-5 should be a statement of what a service implementation is to do rather than how it is to do it. The functions specified in an SOV-5 should be essential to the service rather than an attempt to steer a particular implementation approach.

This document is no longer extant and has been withdrawn.

Use and examples of MODAF

General guidance for the architecting process and application of the MOD Architecture Framework (MODAF).

The MODAF Architecting Process

There are many different approaches to architecting which can be taken depending on the situation. MODAF does not prescribe an â€˜officialâ€™ architecting process.Â However, for someone new to architecting with MODAF, some direction is helpful â€“ this document suggests a way to go.

Uses and Examples

The other documents on this page present a number of examples of how MODAF can be used to support activities associated with the delivery of military capability.

- **The MODAF architecting process PDF [129.3 KB]**
- **MODAF support to analysis of capability integration in the context of Defence Lines of Development (DLODs) PDF [29.2 KB]**
- **MODAF support to user requirements definition PDF [32.7 KB]**
- **MODAF support to system requirements definition PDF [32.4 KB]**
- **MODAF support to dependency analysis PDF [48.5 KB]**
- **MODAF support to gap analysis PDF [107.8 KB]**
- **Creating capability architectures with MODAF PDF [156.8 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

MODAF - Guidance
Detailed Guidance on MODAF

Viewpoints and views
Overview of the differentÂ viewsÂ used in MODAF

MODAF Meta Model (M3)
link to the Meta Model

MODAF - FAQ's
Frequently asked questions

Configuration control
Version history

Information Coherence Authority for Defence (ICAD)
Related team

MODAF - detailed guidance

External links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

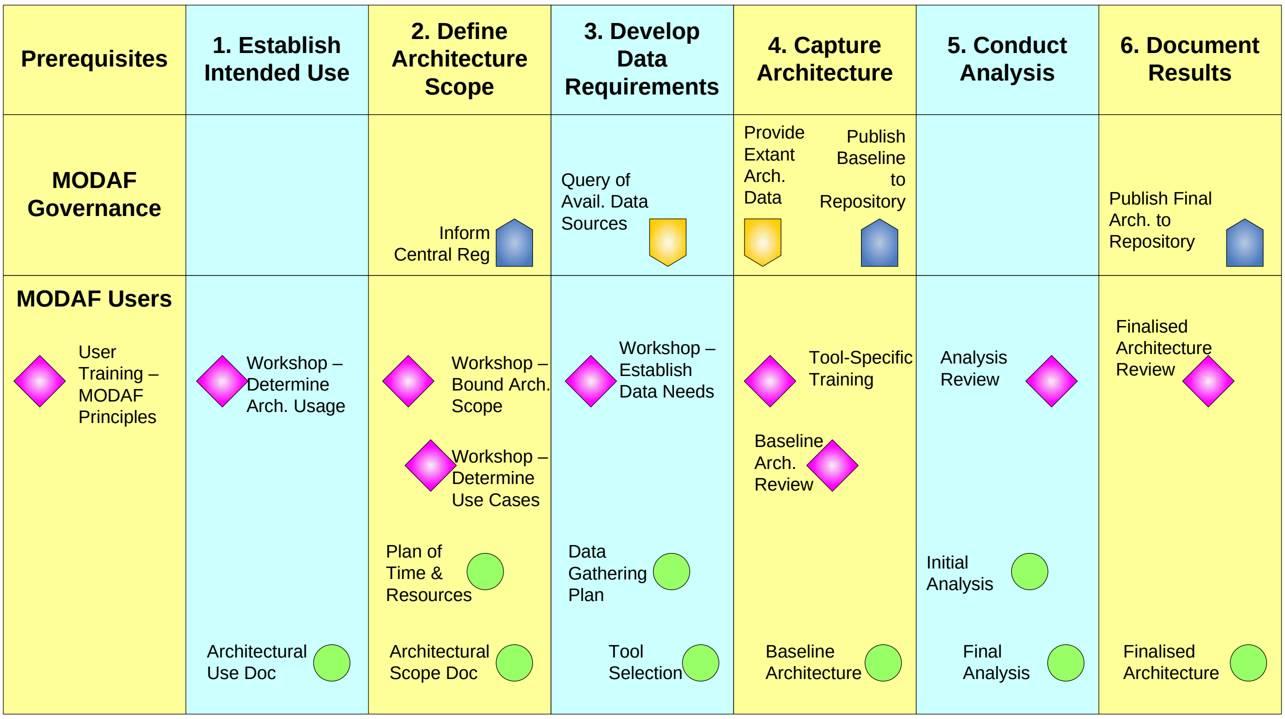
Page not yet rated
★★★★★

THE MODAF ARCHITECTING PROCESS

What is the MODAF Architecting Process?

The overall approach to developing a MODAF compliant architecture is broadly the same regardless of which MOD community is doing the work, or the MODAF views that are being generated. However, the MOD does not prescribe a "MODAF Method" for architecting and creating MODAF views. What this document presents is an example of one approach to take; there are many different ways to approach the architecting process.

In reality, few, if any, teams within the MOD will simply follow the general six-step process outlined above from start to finish once only and then not utilise the architectures again. In practice there will be a wide variety of approaches to conducting architectural work that will involve various iterations and variations around this general process.



In addition to showing the steps that a MODAF user should follow in this example method, the diagram also highlights the key interactions that are required with the MODAF governance processes. Amongst the MODAF governance mechanisms is the Architectural Repository that is run by the System Engineering Integration Group (SEIG)[1]. This can be used to run queries and extract existing architectural data; such as information on the systems that a new capability has to interface with. It is also important that all new architectures are recorded with the appropriate repository to inform others and are available for re-use by other architectures. Furthermore, for the acquisition community the SEIG also provides additional integration services that assist in modelling end-to-end performance and interoperability assurance.

Taking each of the columns from the Diagram above:

Prerequisites

Before commencing a MODAF architecture it is important that the team concerned agree on an approach to creating the MODAF architecture, are familiar with the available views and the

[1] The SEIG were previously known as the Integration Authority (IA)

expected nature of architectural activities associated with their Community Of Interest (COI). Although all of this information is available through this online repository of MODAF guidance and support information, it may be appropriate for the affected team to undertake an introductory course regarding the use of MODAF within their COI. At this point it is probably not appropriate to undertake training regarding the use of any particular MODAF architecture tools as subsequent architectural scoping work may influence the team's final tool selection.

Step 1 – Establish Intended Use

It is essential that any architectural activities are conducted with a clear purpose in mind; the production of a suitable abstraction of complex real world situations that are amenable to detailed analysis. Therefore, step 1 of the architecture development process is aimed at determining and documenting the intended usage of the architecture which can subsequently be used to test whether the developed architecture is fit for purpose. It is often useful to elicit statements of intended use for the architecture through a workshop that includes all of the potential stakeholders who are expected to provide data to and / or utilise the resulting architecture.

Some examples of the "exam questions" that MODAF architectures might address for different COIs include:

- Identification of capability gaps and overlaps – Sponsor[2].
- Develop and trade-off capability options in order to optimise the overall Equipment Programme – Sponsor.
- Develop a clear understanding of the operational context and use cases in support of URD production – Sponsor, Acquisition Integrated Project Team (IPT), Core User[3].
- Establish system boundaries and interfaces, including interoperability analysis – Acquisition IPT.
- Documentation of applied concepts (CONUSE, CONEMP, CONOP) – Concepts and Doctrine organisations.

Step 2 – Define Architecture Scope

The key outcome of this stage is a clear definition of the content and boundaries of the architecture that is to be developed. This will include a definition of the architectural scope in relation to many dimensions, examples of which may include:

- Process scope.
- Organisational scope.
- Systems / platforms scope – including those that have to be interfaced with.
- Geographic scope.
- Coverage of the Defence Lines of Development.
- Timescales that are to be considered (eg 'as-is', 'to-be', 'circa 2015').
- Degree of granularity that is to be modelled (eg system, subsystem or component).

---

[2] Previously known as "Customer One".

[3] Previously known as "Customer Two".

During this stage the team should also start to consider how the architectural information is likely to be presented so as to address the "exam questions" developed during Step 1. This would normally include a list of the key MODAF views that are expected to be produced.

In some cases modified MODAF views may be desirable in order to enhance the required analysis or presentation of results. For example, modified MODAF views may include the addition of overlays to enhance understanding. However, there is a risk that modified views my not be compatible with other tools / repositories. Therefore, advice should be sought through the SEIG to ensure maximum compatibility.

At this stage it is also important to inform the MODAF governance processes of the intended architectural activities. This will help ensure that architecture developers can be made aware of all extant architectural data sources before they commence work and can also be put in touch with other teams that may be developing architectures with similar or overlapping scopes. As repositories become more densely populated this will considerably ease the burden of developing architectures – whole elements could be cut-and-pasted from extant models.

Step 3 – Develop Data Requirements

Before commencing data gathering in order to populate the architecture, it is good practice to establish a data gathering plan. This should include the definition of what data is required, the level of granularity of data that is required, identification of multiple / redundant data sources to provide data validation and / or back-up sources. The data gathering plan should also consider data formats, pre-processing and data migration issues.

Over time, architectural repositories should become a valuable source of existing architectural data which could be re-utilised with little, if any, translation effort required. This is why it is important to inform the MODAF governance processes of the architecture's intended scope; to enable a central register of all the MOD's architectural activities to be built. Based upon this scope information, the repository team(s) can provide a summary of the available architectural data that may be of value to the new architecture.

An important consideration associated with the data gathering plan is conducting an assessment of the security aspects of the populated architecture. This needs to consider not only the classification of the individual data sources, but also the potential for a higher classification if certain combinations / aggregations of lower classification data are presented through the architecture. Consideration should also be made of the security implications for accessing the published architectural data and conducting the required analyses.

Tool Selection

This is probably also the most appropriate stage of the overall process in which to consider tool selection. MODAF does not require a particular tool / suite of tools to be implemented; definitive guidance as to tool availability and fit with different COIs is not available.

Architecting teams should, however, consider the following when selecting a tool / suite of tools:

- Does the tool enable modelling of the architecture at the right level (eg is it modelling at the business level or the technical level? Can it provide the right level of detail?)
- Does / can the tool support the MODAF Meta Model (M3)?
- Can architectural models created in the tool be easily shared with other tools or with the SEIG repository?
- Can the tool exploit existing architectural models?

Note: Although it is intended to set a model interchange standard between a (to be defined) set of tools, there will often be an advantage to edit models within the native format that they were developed in – which maintains the intended graphical layout and potentially additional architectural data that goes above and beyond the MODAF specification.

Having made the tool selection it may be necessary to provide tool-specific training to those who are going to be deeply involved in capturing and editing the architectural models. It is expected that there will be a variety of tool-specific MODAF course available through tool vendors and their intermediaries.

Step 4 – Capture Architecture

It is during this stage of the process that the bulk of the architecture development actually takes place: importing and editing extant architectural models, capturing additional data and entering it into the architecture. This is likely to include extracting data from existing architectures via the SEIG or other repositories.

When building the architecture it is important that it is only constructed in accordance with the MODAF Meta Model and MODAF Taxonomy[4]. These constraints underpin the MODAF tool interoperability mechanisms and compliance with them ensures that the architecture will be compatible with the SEIG and other repositories and that others will be able to re-use the content in the future. Help on how to achieve this will be available through the CIO MODAF team, The Information Coherence Authority for Defence or the SEIG.

It is important that before the resulting architecture is baselined for publication and analysis its accuracy and validity is confirmed. This should include a review of the entire architecture by the subject matter experts who have provided key inputs. It may also be advisable to consult the MODAF governance processes / SEIG during the review process to ensure that any dependent architectures (eg with details of interfacing processes or systems) have not changed or are not in the process of changing.

At this point in the architecture development process the baseline (ie pre-analysis) architecture should be published to an appropriate repository in order to provide visibility to others across the MOD.

In order to facilitate the searching and query of architectures it is essential that the All Views (AV-1 with meta data regarding the architecture and AV-2 with the architecture's object dictionary) are completed thoroughly for all architectures before they are published. It may even be appropriate to start the documentation of the AVs during an earlier stage and to refine them as the scope of the architecture evolves.

Step 5 – Conduct Analyses

Given the validated baseline architecture delivered through step 4 of the process, all of the required data should now be available to conduct the analyses that were identified during step 1. These analyses are likely to be COI-specific, and may include a variety of analytical techniques, including but not limited to:

- Static analysis, such as a gap / overlap analysis against the Strategic Views in order to identify capability issues.
- Dynamic analysis such as network traffic / bandwidth analysis based upon network configurations from SV-1 and traffic data from OV-2 / OV-3.
- Experimentation. Using information developed from the architectural analysis to establish the use cases / context for experimentation campaigns such as those run through NITEworks.
- Trials. Using architectures to provide use case / context information for exercises and trials at a variety of scales from battlelabs to full brigade or division level exercises.

---

[4] See also the document, "*20090203-Ontologies and their Use in MODAF-U*" elsewhere on this MODAF guidance website.

As with the review of the baseline architecture, it would be good practice to conduct a review on the initial analyses and if necessary to revise the analyses before issuing the final product(s).

Step 6 – Document Results

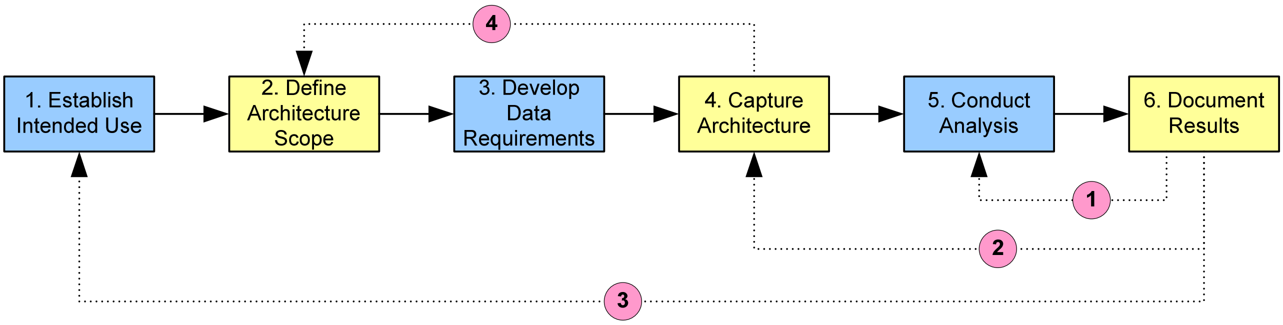
Having conducted the required analyses, changes to the baseline architecture will often be identified. Examples might include:

- Capability analysis may have highlighted a serious capability gap which has been developed into an EP option. The capability, timing and other details of which should then be entered into the finalised architecture
- System interoperability analyses may identify interface problems that have to be rectified by means of changes to the applicable standards or introduction of a gateway equipment, which need to be included in the finalised architecture

When the architecture has been updated with the relevant changes it should again be subjected to a further review and the resulting finalised architecture published to the appropriate repository.

Approaches to Iterative Development

There is no right way of conducting iterations around this general architecting process, but some practical examples are highlighted in this diagram.



The first common type of iteration (1) is where having generated the architecture there are periodic analysis / update cycles without any major refresh of the architecture itself. This approach may apply for example to the development and detailing of a number of capability options within the Sponsor's processes of finalising the Equipment Programme.

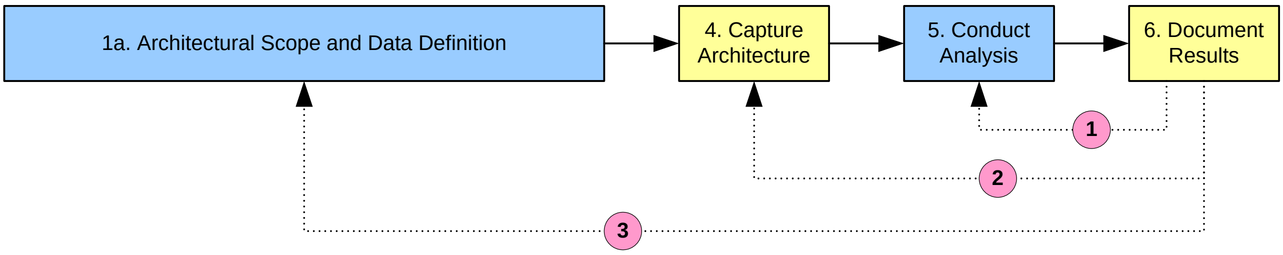
Another type of iteration (2) would be where the architecture is refreshed with more up to date data before the analysis is repeated. This approach may apply for example to the update of the Strategic Views each time the capability audit is conducted within the Sponsor's processes.

In some cases (3) it may be appropriate to periodically return right back to the start of the architecture processes to review the purpose, scope and data sources. A good example of where this may apply is within an acquisition IPT as it moves between stages in the CADMID / CADMIT cycle; where there are different stage objectives, the solution boundaries may have changed and new data sources may be available. These review activities of the early architectural activities can usually be conducted quite rapidly, possibly covering the review of steps 1 to 3 in a single workshop.

Sometimes, as the data is being gathered and entered into the architecture it may become apparent that it is not going to be possible to achieve the desired results using the elements being considered. In this case (4) it may be necessary to re-visit the architecture scope and / or data gathering plan in order to develop an architecture that will satisfy the original objectives.

Approaches to Rapid Architectural Update

In some cases the team will be working with an architecture that is largely pre-existing (eg from elsewhere within the IA repository) and against a well defined task and scope definition. In these cases it may be possible to abbreviate the process and conduct steps 1 to 3 in a single quick pass through the definition of desired outcomes, architectural scope and data sources as shown here.

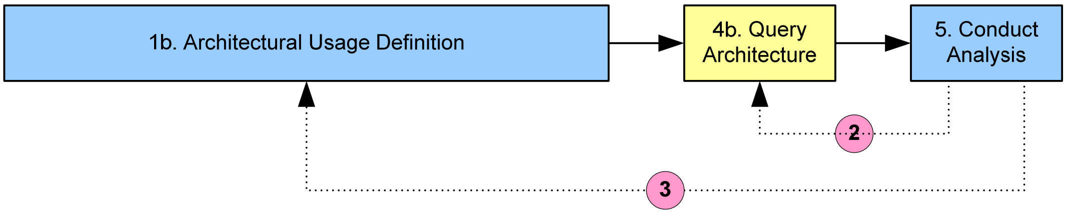


It is still good practice to document the key deliverables of each of these architectural stages even if they are in a single document that has been captured during a single workshop.

It should be noted that similar iterative options could still exist with this rapid update approach.

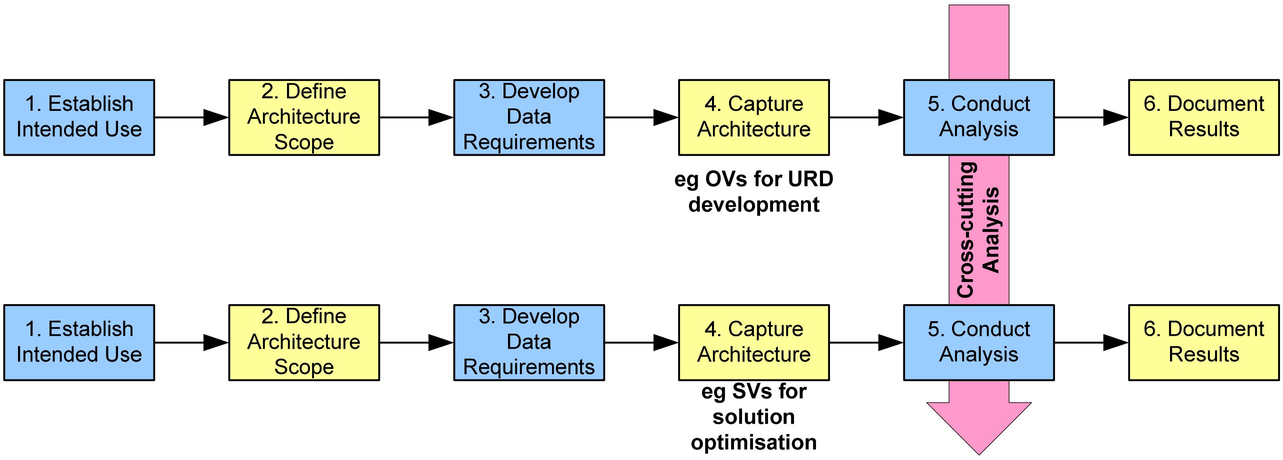
Read-Only Architectural Usage

In some cases particular groups of MOD architecture users will not need to create architectures of their own but will be conducting analysis on the architectures produced by others. For instance, this may apply to the assurance and scrutiny communities who want to examine the adequacy and maturity of architectural activities conducted by IPTs at various stages of the acquisition lifecycle. In this case, a rather abbreviated version of the six-step process may apply; there will be no update or publication of the architecture, as shown in this diagram.



Parallel Architectural Activities

Another common situation in the MOD will be where there are a number of parallel streams of architectural activities being conducted in relation to the same overall project. For example, within the concept stage of the acquisition cycle there will be refinement activities on the User Requirement Document (URD) being conducted largely using the OV suite of MODAF views while simultaneously a high level suite of SVs will be in the process of being developed for the purpose of optimising different system solutions. In some cases these parallel streams of architectural activity may be being conducted by quite separate teams. However, in most cases these various architectural streams will need to converge at certain points in the project when joint / cross-cutting analyses are required (see the diagram below), such as an IPT conducting an overall risk assessment using elements from both the OVs and SVs to assess issues such as the clarity of use cases and the degree of interface definition.



Defence Lines of Development analysis with MODAF

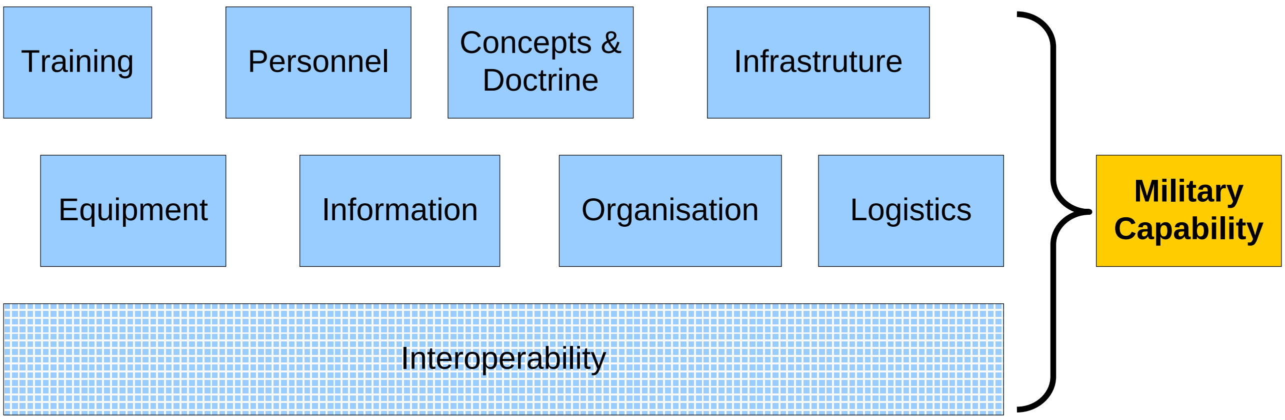
This article will illustrate the use of MODAF with reference to common challenges for capability integration based on analysis of the Defence Lines of Development DLODs).

Within MOD, the DLODs provide a mechanism for co-ordinating the parallel development of different aspects of capability that need to be brought together to create a real military capability:

- Training.
- Equipment.
- Personnel.
- Information.
- Concepts & Doctrine.
- Organisation.
- Infrastructure.
- Logistics (ie Sustainability).

Interoperability is regarded as an overarching theme.

The co-ordination of the DLODs is sometimes referred to as Capability Integration. This is illustrated in the diagram below.



Improving framework support to the DLODs has been one driver for the evolution of MODAF from just being concerned with the Equipment line (like the original DoDAF-based framework).

The first observation is that the Information LOD has been addressed within MODAF: information within the Operational Viewpoint is addressed through a related suite of views (OV-2, OV-3, OV-5, OV-7) and the same is true for data within the System Viewpoint (SV-1, SV-4, SV-6, SV-11).

The second observation is that in addition to the Equipment LOD, the Training and Logistics LODs probably merit full architecture development in their own right. This is because the full appreciation of these DLODs in capability terms will involve understanding a combination of process, information and people.

The area that has received most attention in the changes leading to the current baseline have focused on the features of MODAF relating to the human element of capability. Specifically:

- Within the operational viewpoint, it is possible to model types of organisations and individual posts within an organisation; it is also possible to model the roles to which these posts may be assigned and define the functional responsibilities of those roles.
- Types of organisation and post are collectively referred to as 'organisational resources' within MODAF.
- An organisational resource that has responsibility for the performance of an activity or set of activities is known as the process owner; an organisational resource having such a responsibility must have the competence to undertake that role
- Within the system viewpoint, the focus is on the human roles that contribute to a capability configuration (that is a combination of elements that in combination fulfil an operational requirement)
- Some roles (operator roles) directly operate systems; a distinction is now made in MODAF between system functions and manual functions
- An organisational resource undertaking an operator role must have a competence to perform the associated manual functions.

These changes are intended to make it easier to specify the solution required at different levels of abstraction. The organisation LOD is covered by these changes.

Using capability configurations alone, it is possible to specify a solution capability in terms of the overall functions that are needed without specifying the human contribution to the delivery of those functions. This would apply to the early stages of procurement using a capability-based procurement approach. Such an approach enables the exploration of new organisational structures as part of the solution space so that there may be genuine trade-off between manual and automated functions.

The competencies referred to above will provide an input to training analyses. Among other things, such analyses will look at the gap between the actual and required competencies in formulating options for training. Competencies relate directly to the Personnel LOD.

The remaining DLODs are addressed by MODAF as follows:

- The Infrastructure LOD primarily relates to facilities; the concept of the 'physical asset' that is introduced into MODAF at this baseline has been defined in such a way as to cover both physical platforms (eg tanks or aircraft) and facilities (eg the Land System Reference Centre), the defining characteristic of a physical asset being that systems can be deployed to it
- The role of Concepts & Doctrine within MODAF is reinforced through of the logical nature of Nodes in the current baseline; there are increased opportunities to capture operational concepts within MODAF operational views. For example, low level concepts may be captured as business rules in OV-6a.

User Requirements Analysis with MODAF

This article is not an exhaustive description of the way in which MODAF supports user requirements definition and analysis within MOD. That subject is described in detail within the "*Interoperability for Communication and Information Services*" section in the Acquisition Operating Framework. What this article does do is describe how the MODAF concepts support the key principles of requirements definition for defence systems.

A companion article addresses MODAF support to system requirements definition[1].

At the higher level of requirements (eg capabilities, systems of systems), the strategic views provide the platform for establishing the capability needs. It is important to recognise that, in a managed enterprise, the capability needs will change over time; these are captured in the StV-3 view. At a given point in time within the enterprise timeline, the capability needs, as represented in a column within a StV-3 view, can be expressed in terms of the level of capability performance required. The OV-1c tabular view can be used to capture these levels of capability as they evolve over time.

The models within the operational viewpoint then reflect the capability needs at a given point in the enterprise timeline. They refine the needs expressed at the strategic level in three complementary ways:

- The functional needs can be represented in terms of a model of the business process (which will focus on the OV-5 activity diagrams).
- Nodes (captured in the OV-2 view) provide a focus for the operational requirements which can cover both functional and non-functional aspects.
- Behavioural models (addressed using the OV-6 views) provide a focus for operational requirements covering operational behaviour (e.g the need for the capability to respond in a particular manner to specific events).

As an example, there may be distinct nodes modelled that represent the air and ground segment of a capability that is known to have a significant airborne component. The air segment node will be the focus for non-functional requirements that will express the mobility and survivability needs of this segment (among others) which will be quite distinct from those associated with the ground segment. This use of models (to reflect the principal operational concepts) is not prematurely defining any solution because having an airborne component may be a fundamental element of the capability need.

One important way that architectural modelling supports requirements definition is in terms of boundary definition. Boundary definition is a process that often requires a significant degree of stakeholder engagement; the standardised views provided by MODAF provide ideal support for this interactive process. MODAF now provides support to the concept of the 'ProblemDomain' which provides a bounding shape on an OV-2 operational node relationships view; those aspects of the capability that are not subject to change during the acquisition programme are left outside the boundary. These need to be modelled, however, so that the boundary interoperability requirements can be captured.

[1] A link to the document covering 'MODAF Support to System Requirements' topic can be found on the same 'Uses of MODAF and Examples' page that contained the link to this document.

Definition of user-level interoperability requirements is another subject for which there is guidance through the Acquisition Operating Framework. Within the operational viewpoint, MODAF supports interoperability analysis in a number of ways:

- An OV-3 enables information exchange requirements (linked to the needlines between nodes shown on the OV-2) to be tabulated.
- An OV-2 enables capture of node-to-node interactions that are based on exchange of energy or materiel, not just information.
- OV-2, OV-3, OV-5 and OV-7 form a linked set of views that support a coherent model of operational information and boundary-crossing exchanges; when behavioural models are included this set can be extended to include OV-6c.

The functional boundary can also be described in terms of the activities within the OV-5 activity model that the capability of interest will enhance support for.

Another aspect of requirements definition is the construction of test scenarios. The OV-6 behavioural models provide opportunities here for the development of test scenarios that are linked in with the operational models that support the user requirements. Such models both reflect operational requirements (the capability must do X once event E happens) and provide tests of it.

Use of MODAF in this way should improve the quality of requirements definitions by:

- Explicitly tying user requirements to strategic level capability needs.
- Enabling early agreement to be reached on the capability boundary.
- Providing a validated ‘reference’ model of the business against which the completeness of a requirements definition can be assessed (visualisation aids validation).
- Ensuring that functional requirements are clearly linked to a validated model of business processes.
- Ensuring that the information-related requirements (not just IERs) are captured in a coherent manner and in a way that really reflects the user collaboration needs.
- Providing test scenarios linked to the user requirements.

Associating an architectural model with a requirements definition also has several benefits:

- It is often easier to gauge the most appropriate level of abstraction in an architectural model (because it tends to be easier to identify when modelling decisions are solutioneering) than is the case with text-based requirements
- The models can then provide a means of testing the completeness of the requirements set.

System Requirements Analysis With MODAF

This article does not attempt an exhaustive description of the way in which MODAF supports system requirements definition and analysis within MOD. Many of the observations contained in the article on user requirements analysis[1] also apply to system requirements definition.

The Acquisition Operating Framework (AOF) should be consulted for full details of system requirements definition in the MOD.

Models in the system viewpoint represent alternate realisations in terms of equipment capability of the operational capabilities expressed through models in the Operational Viewpoint and in the User Requirements.

In MODAF, the system viewpoint primarily addresses the specification of the system capability needed (rather than implementation details). However, recent changes have improved the ability for MODAF modellers to represent configuration of capability that include people as well as systems and platforms; see the article on Defence Lines of Development[2] for more information. This article will focus on equipment specification.

System specification models refine the expression of needs at the operational level in the following three ways:

- The functional needs can be represented in terms of a model of the system functionality (which will focus on the SV-4 functional flow diagrams).
- The structural requirements may be represented using the SV-1 view.
- The performance and behavioural requirements may be represented using behavioural models (captured in the SV-10 suite of views).

Maintaining a trace to the models reflecting the user requirements is essential. The key mechanisms for achieving this within MODAF are:

- SV-5 which maps functions in the System Viewpoint to activities in the Operational Viewpoint (for functional traceability).
- SV-1 which maps capability configurations in the System Viewpoint to nodes in the Operational Viewpoint (for structural traceability).

One important way that architectural modelling supports system specification is in terms of system boundary definition. Boundary definition is a process that often requires a significant degree of stakeholder engagement; the standardised views provided by MODAF provide ideal support for this interactive process. The system boundary can be represented using overlays on views such as SV-1 (structural boundary) and SV-4 (functional boundary). More detail can be provided of the boundary in views such as SV-2 (interface specification), SV-3 (system-to-system mapping) and SV-6 (system data exchange requirements).

---

[1] A link to the document covering 'MODAF Support to User Requirements' can be found on the 'Uses of MODAF and Examples' page that contained the link to this document.

[2] A link to the document covering 'MODAF Support to DLOD Analysis' can be found on the 'Uses of MODAF and Examples' page that contained the link to this document.

Definition of system-level interface requirements is another subject for which there is guidance through the Acquisition Operating Framework. Within the System Viewpoint, MODAF supports interface analysis in two ways:

- SV-2 views can add detail to the structural specification (SV-1) in regard to interface specification; specifically, details of the required protocols can be captured.
- SV-1, SV-4, SV-6 and SV-11 views form a linked set of views that support a coherent model of system data and boundary-crossing exchanges; when system behavioural models are included this set can be extended to include SV-10c.

In respect of these interface requirements, the linkage between SV-6 and OV-3 provides a further degree of traceability to the operational models that provide capability context to the system requirements.

System specification is not just a top-down process of decomposing the relevant parts of the required operational capability. Particularly in the context of Network Enabled Capability, there will be constraints and legacy capabilities that will influence the specification. MODAF helps to address this by using models of legacy capability to specify where the system boundary should lie and what the data and services are that cross the boundary.

Another aspect of system specification is the construction of test scenarios. The SV-10 behavioural models provide opportunities here for the development of test scenarios that are linked in with the system models that express system behaviour.

Use of MODAF in this way should improve the quality of system specifications by:

- Explicitly tying system requirements to the (models of) user requirements which provide context for those specifications.
- Enabling early agreement to be reached on the system boundary.
- Understanding the specification implications of constraints and legacy capability.
- Ensuring that the information-related requirements are captured in a coherent manner.
- Providing test scenarios linked to the system requirements.

In addition to these benefits, the specification models may support stakeholder engagement in respect of system trade-offs.

Dependency analysis with MODAF

This article illustrates the use of MODAF with reference to dependency analysis. Examples of dependencies that are of interest to MOD include capability dependencies, programmatic dependencies, technology dependencies etc. Analysis of dependencies of this type is considered a key use of an enterprise architecture.

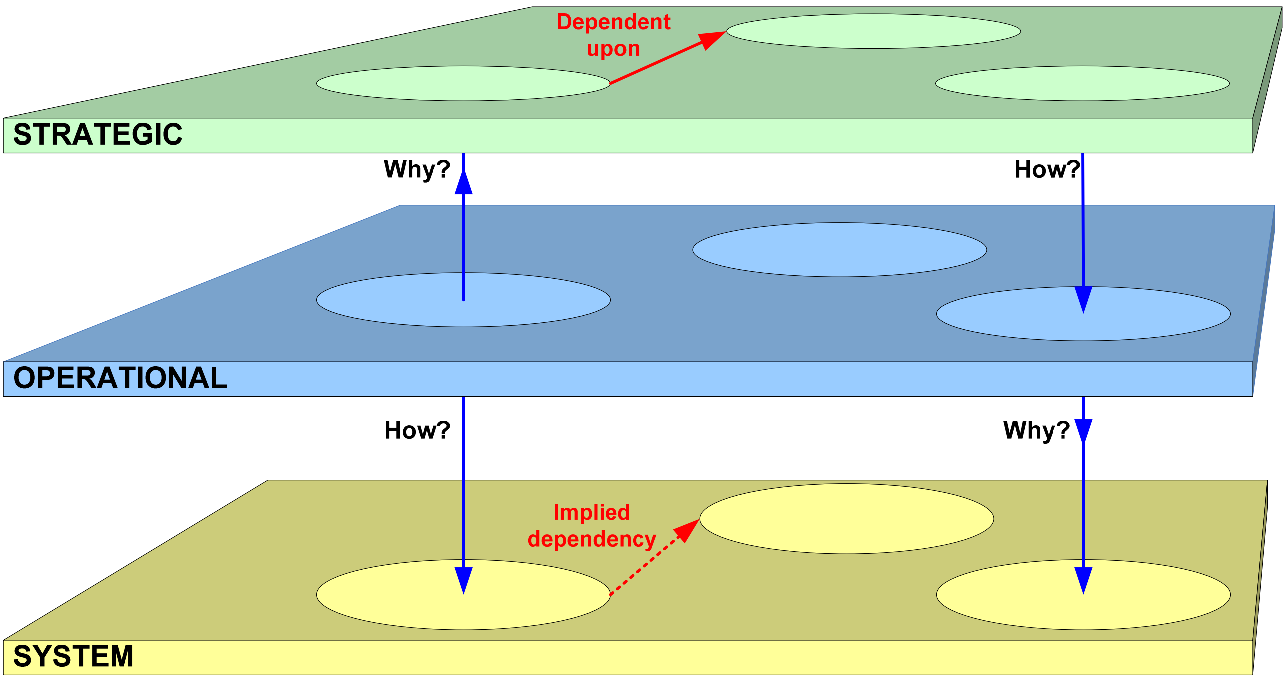
The following guidance depends upon the modelling layers and viewpoint linkages covered in other articles. The assumption is made that an enterprise architectural model has been created using MODAF.

Technical dependency analysis

This form of analysis relates to the investigation of dependencies between technical capabilities. In MODAF terms, these capabilities can be at the strategic, operational or system level. MODAF does not deal with dependencies at the technology component level.

The diagram below illustrates a situation in which a capability dependency has been identified in the strategic viewpoint (perhaps through modelling in support of a StV-4 view). By tracing through the operational and system 'layers', an analyst can identify a potential difficulty in the case of a system dependency, which is implied by the capability dependency but does not manifest itself. System-level dependencies might be reflected in system interactions shown on an SV-1 and / or might be reflected in functional linkages shown on an SV-4.

Assuming that the model is accurate, this analysis might be the cue for rectification action, for example, the introduction of an interface between the systems that support the dependent capabilities.

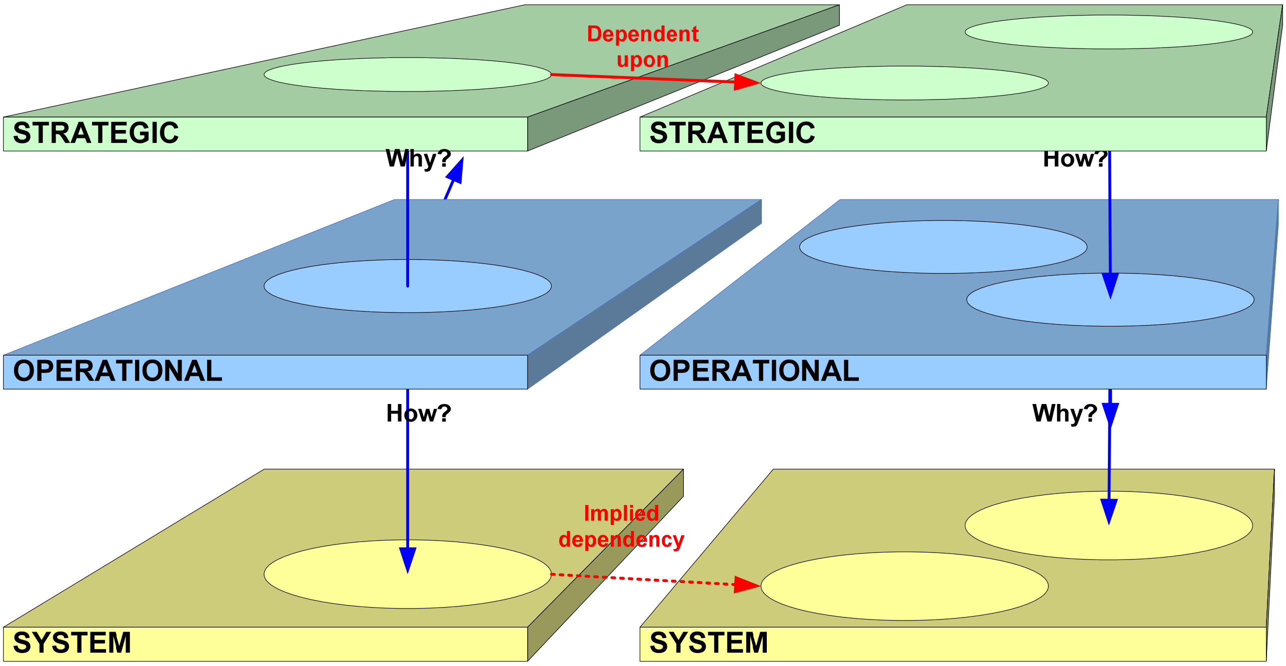


Programmatic dependency analysis

A similar situation occurs when two projects have been set up to address the capability needs of each of the two dependent capabilities. Analysis based on the MODAF architectural model might (as illustrated below) then determine that there should be a dependency between the projects that may not have been captured (project dependencies are addressed in the AcV-1 view and, from a timeline perspective, in the AcV-2 view).

This analysis might be a cue for rectification action in terms of setting up engagement between the projects concerned to investigate what the dependency actually is. This situation might easily occur where there are capability dependencies that cross Programme boundaries.

Projects might well have dependencies that reflect sub-system interactions in the system architecture, for instance, and may not always manifest themselves (going up through layers) in terms of capability dependencies. That is a different situation to the one represented here, but the principle is the same.



In general, the ability to conduct dependency-based analysis depends upon the coherence of the models generated using the framework. That in turn depends upon the coherence of the framework (between views, viewpoints and at the data modelling level). Considerable attention has been given, during the development of the framework, to achieving this coherence.

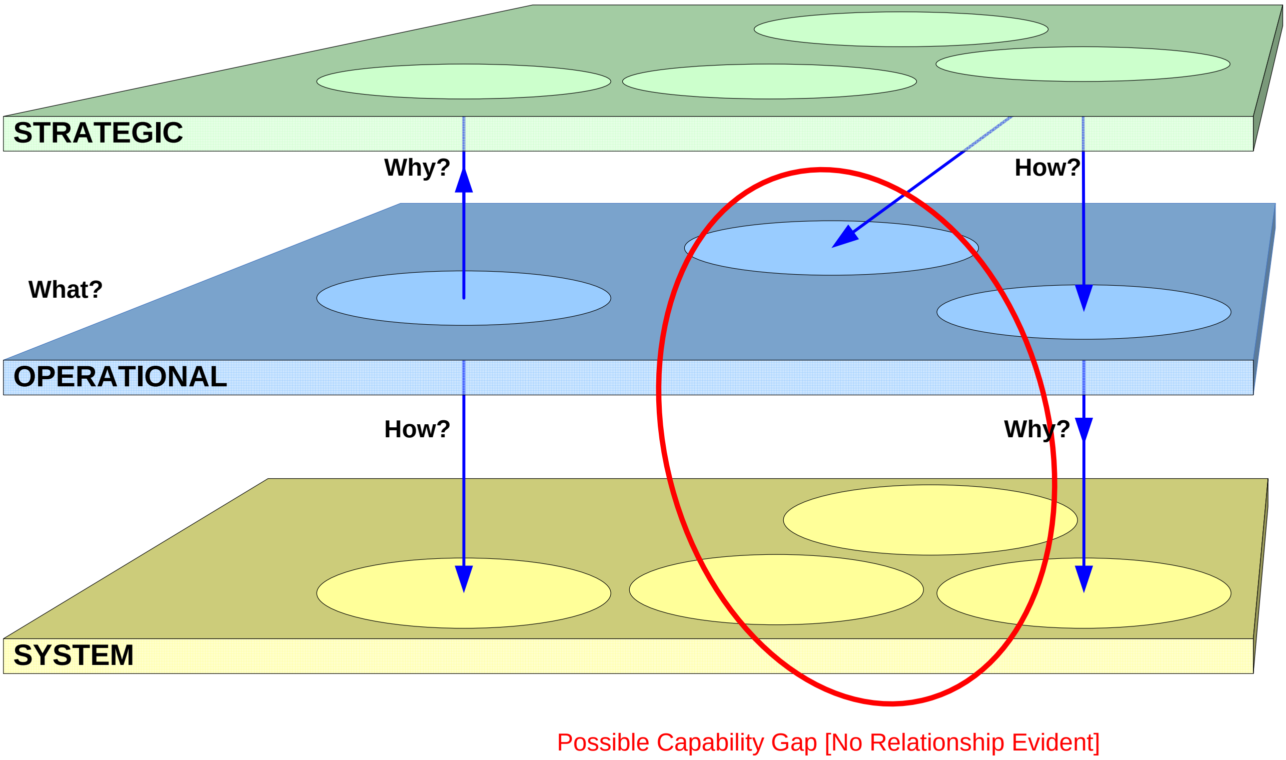
Use of MODAF in this way supports its role as an Enterprise Architecture framework. The ability to undertake this type of analysis is one benefit from wider use of the framework within MOD. Early identification of problems such as those illustrated above should reduce the degree of rework associated with lack of coherence in the equipment programme.

Gap analysis with MODAF

This article illustrates the use of MODAF with reference to gap analysis. Gap analysis is a way that enterprise knowledge can be used to inform decision-making. The gaps of interest to MOD can include gaps in the realisation of military capability (based on the linkage between Strategic and Operational Viewpoints) and gaps in the realisation of equipment capability (based on the linkage between Operational and System Viewpoints).

The discussion below depends upon the modelling layers and viewpoint linkages covered in other articles. The assumption is made that an enterprise architectural model has been created using MODAF.

The diagram below illustrates use of MODAF in the conduct of gap analysis.



In this example, a model of the operational capability required has been created together with a partial specification of the associated solution. An operational capability has been identified through analysis for which there appears to be no solution capability.

A specific example of this is the use of SV-5, which traces system functions to operational activities. This tabular view might be used to identify an operational activity which lies within the scope of the operational boundary but for which there is no prescribed system functionality.

Note that the technique used to analyse gaps in capability can also be used to identify overlaps, ie redundancy in the provision of operational capability from more than one system capability. For example, based on an architectural repository that contains models of a number of system capabilities, it is possible to perform an analysis which identifies all the systems that are able to support a particular type of operational activity.

Gap analysis, like dependency analysis, makes use of the expected linkages between capabilities expressed at different modelling levels. Typically the analysis is focused on identifying realisation gaps (ie requirements that are not yet fulfilled). One difference is that the analysis can also be focused on identifying gaps in the model by looking upwards through the modelling layers.

As an example, when modelling the ‘As-Is’ capability it is not uncommon for models of existing systems to be captured using MODAF and the business requirements for them (in terms of models at the operational and perhaps the strategic levels) reverse-engineered. Dependencies (such as interfaces or the common use of a technology component) between the As Is systems might then manifest themselves as dependencies at a higher level (higher modelling layer).

The reason that this type of gap filling is important is that any incompleteness or inaccuracies in the model associated with the enterprise architecture might lead to incorrect decisions being taken at a later date

Creating Capability Architectures With MODAF

Creating a capability architecture allows users to express capability requirements. Any architect starting to use MODAF for the first time should acquaint themselves with the terminology used for modelling capabilities. These are described at various points in this article:

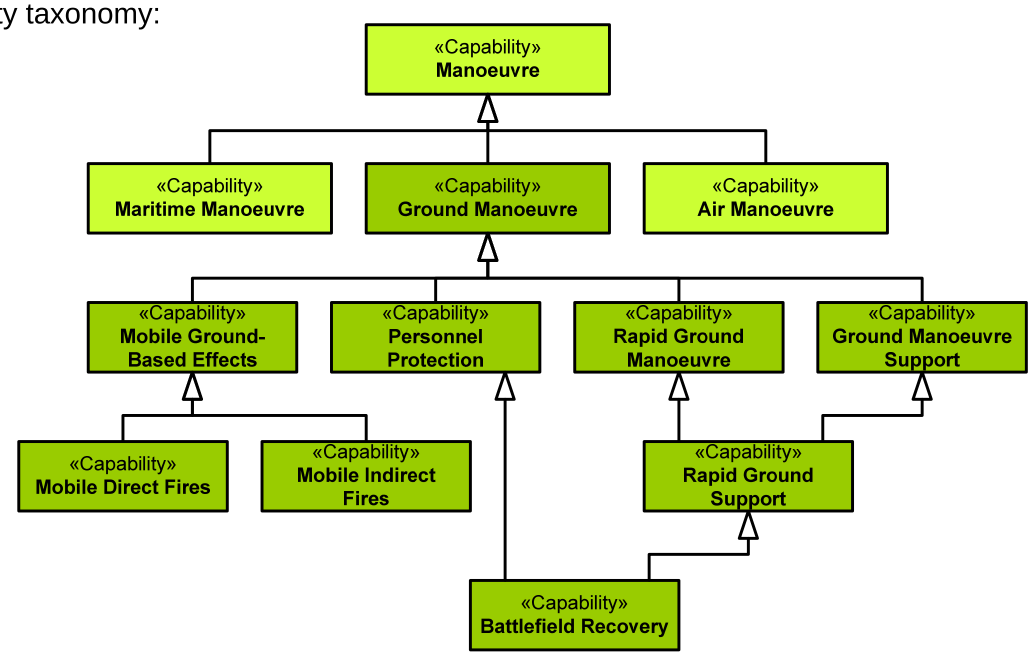
Capability Vision

> "*The overall aims of an enterprise over a given period of time. Although not strictly speaking an architectural element, this document is the "capping paper" providing provenance for the capability architecture and is usually a strategic policy statement or operational concept.*"

Capability

> "*A high level specification of the enterprise's ability. A capability is a classification of some ability – and can be specified regardless of whether the enterprise is currently able to achieve it. For example, one could define a capability "Manned Interplanetary Travel" which no-one can currently achieve, but which may be planned or aspired to. Capabilities in MODAF are not time-dependent – once defined they are persistent. MODAF allows the architects to develop a formal taxonomy of capabilities which can be re-used across multiple architectures.*"

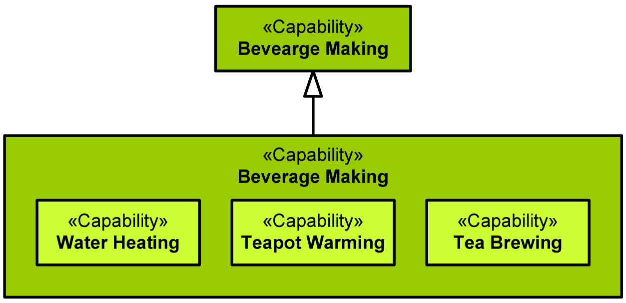
Assuming the Capability Vision already exists, the first task in developing a capability architecture is to establish the capability taxonomy. It is likely that capabilities are already defined to some degree, and the main task in creating a MODAF-compliant capability architecture is to organise these capabilities into a formal taxonomy. The example below shows a fictitious ground manoeuvre capability taxonomy:



The diagram above (which uses a UML notation) shows a specialisation hierarchy of capabilities. Note that some capabilities specialise from more than one parent, for example, **Rapid Ground Support** is both a **Rapid Ground Manoeuvre** and a **Ground Manoeuvre Support** capability.

- The capabilities defined in a MODAF capability taxonomy are used throughout the architecture, and are often used by more than one architecture. MODAF operational architectures refer to capabilities – i.e. they define what capabilities are required for a given scenario or operation. Systems architectures define the personnel, platforms, equipment and processes needed to fulfil capabilities. The capability taxonomy is, therefore, one of the most important parts of a MODAF architecture, so it is vital that the taxonomy is produced according to certain guidelines:
- The capabilities should not "solutioneer". That is, they should not pre-suppose how a capability is to be achieved. This provides maximum design freedom in acquisition, and also allows ops planners maximum flexibility in operational architectures.

True specialisation should be used. It is quite natural to mistake a component capability for a specialised capability. Using a simple example, "Tea Making" is a specialised "Beverage Making" capability, whereas "Water Heating" is a component capability of "Tea Making" rather than a specialisation of it. MODAF uses a different notation for component capabilities:

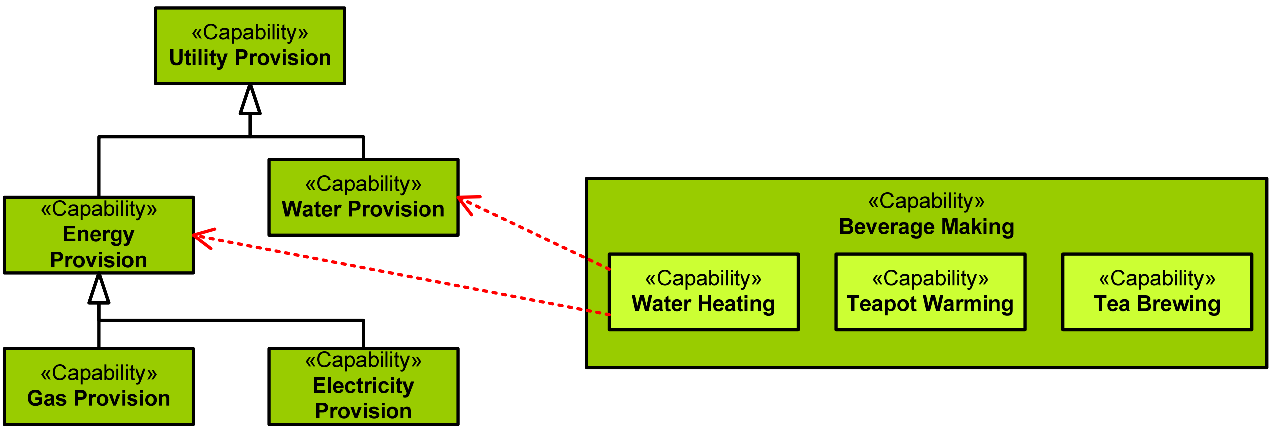


The ability to model capabilities as being components of other capabilities is part of the more general MODAF approach to modelling *dependencies* between capabilities.

<u>Capability Dependency</u>

"*A relationship that asserts a capability is dependent on another capability – i.e. one capability has to exist before the other can be achieved.*"

Sticking with the tea-making example, this can be illustrated thus:



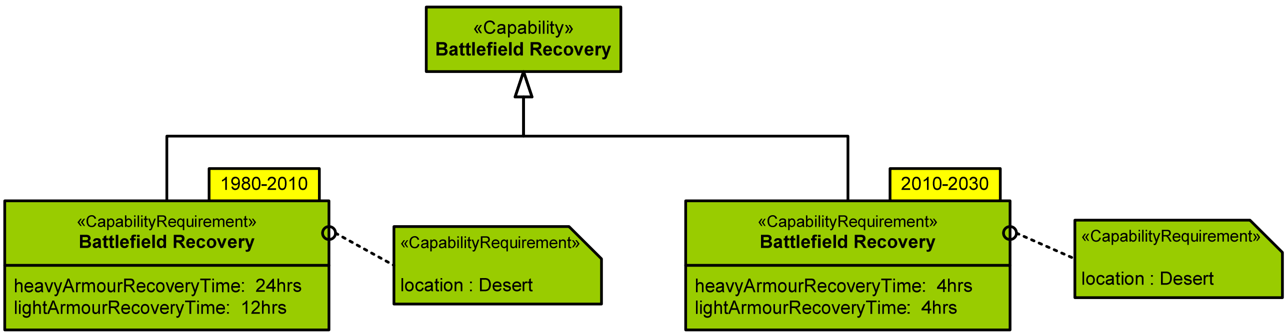
The dotted arrow points to the capability upon which another capability depends; ie the dotted line should read from tail to head: "depends on".

Up to this point capabilities have only been discussed in general terms, regardless of whether they are required or not. The capabilities described in a taxonomy may not all be required by the enterprise; some may be capabilities exhibited by coalition partners or opposing forces. To make use of the capabilities in an operational or acquisition architecture, it is necessary to state to what level the capability is required and by when.

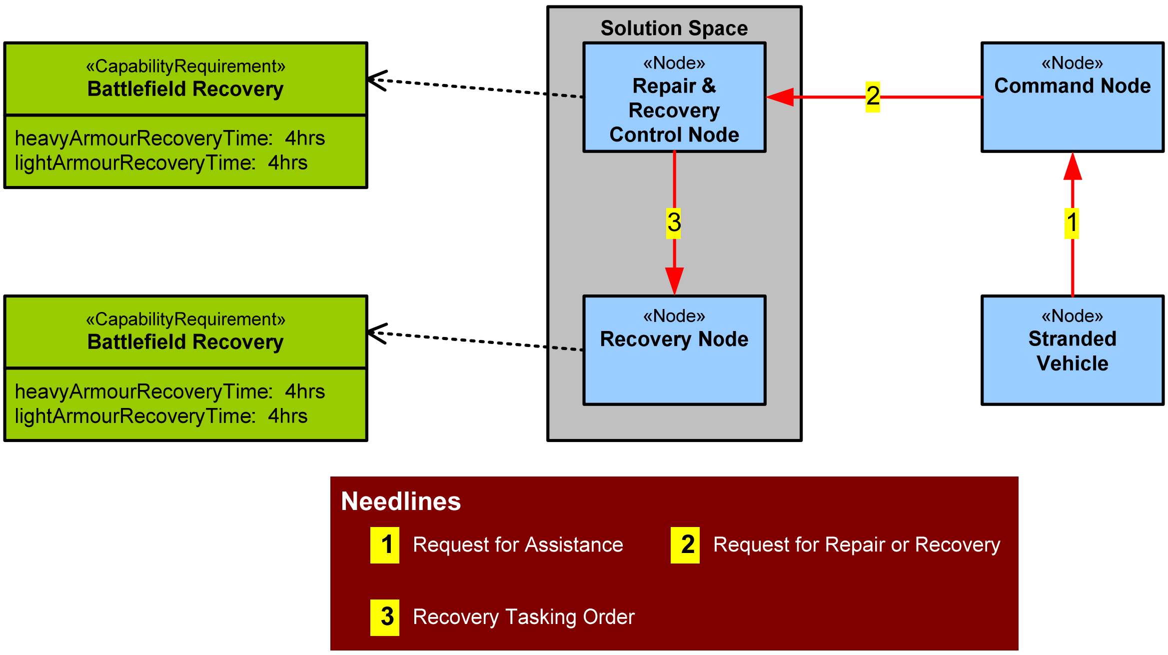
Capability Requirement

> "*A time-dependent requirement for a capability. A capability requirement is a statement that a Capability is required to a certain level (specified by formal metrics and natural language assertions) within a specified time frame. Taking the interplanetary travel capability from before, we could create a capability requirement stating that a space agency intends to achieve the capability by 2020, with a required journey time to Mars of 6 months.*"

Capability requirements are usually shown as special cases of the capabilities they describe. In the example below, the *Battlefield Repair & Recovery* capability is extended to show the requirements for the 1980-2010 and 2010-2030 time-spans (or "epochs"):



The capability requirements specify performance metrics – in this case, the required recovery time is greatly reduced. Once a taxonomy of capabilities and capabilities requirements have been defined, these can be re-used across a number of architectures. Operational architectures can apply capabilities to scenarios and plans:

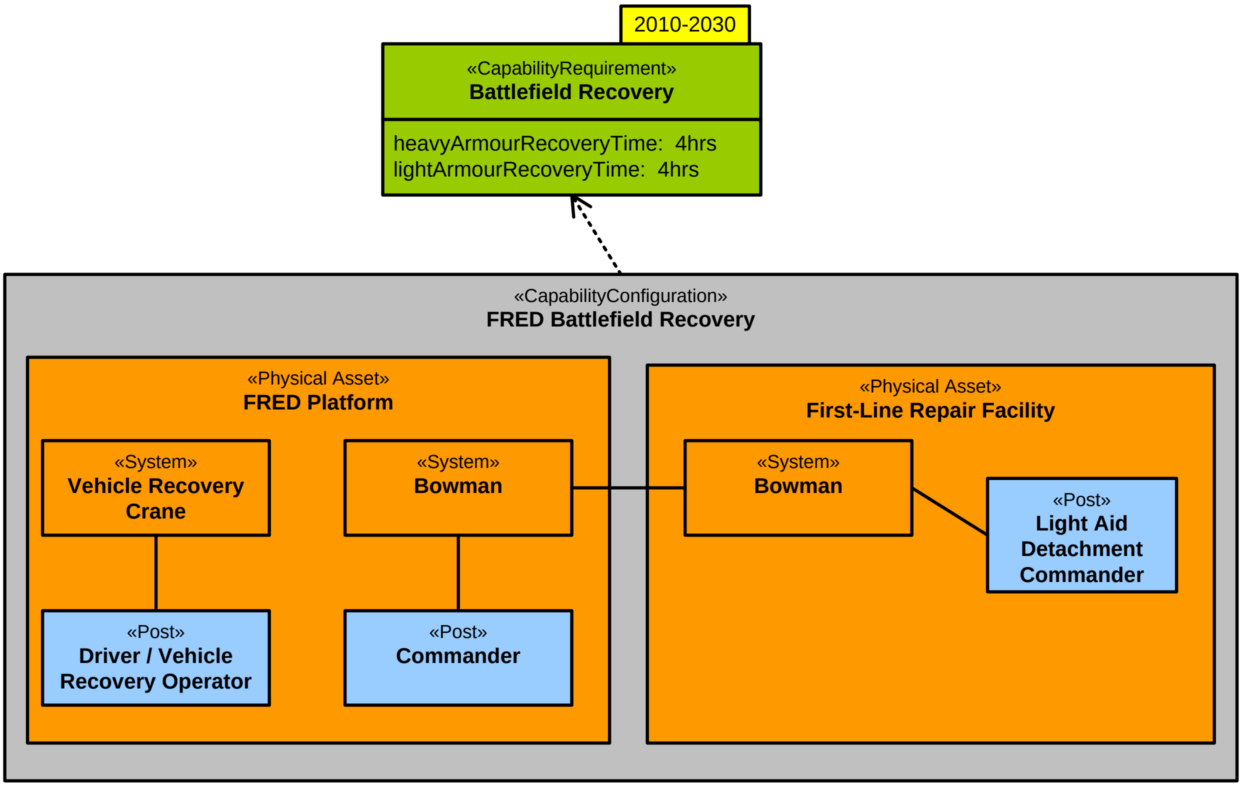


MODAF also specifies how combinations (configurations) of equipment and trained staff can realise capability.

Capability Configuration

> "*A combination of organisational aspects (with their competencies) and equipment that combine to provide a capability. A Capability Configuration is a physical asset, organisation or post configured to provide a capability.*"

Capability configurations are represented as a collection of assets, systems and people that are inter-connected. The root of a configuration must be a platform, facility, person (post) or organization. People may be deployed to platforms and may use systems. The example below shows a fictitious configuration that realises the battlefield recovery capability from the previous example:



The final aspect of capability that is covered by MODAF is the concept of actual, physical capability, ie a real world instance of a capability configuration.

Fielded Capability

> "*An actual, fully-realised capability – eg HMS Daring in fully operational condition, with a trained crew.*"

Fielded Capabilities are not often used in architectures, but sometimes it is necessary to refer to an existing capability, for example, referring to Permanent Joint Headquarters.

Summary

Capabilities are at the core of all MODAF architectures. They provide strategic context and high-level requirements for acquisition architectures, and provide ops planners with an easy way to specify how capabilities are required to interact in an operational architecture. A capability architecture may support any number of other architectures, ie the capability taxonomy and requirements may be re-used over and over again. This level of re-use helps to ensure a common understanding across projects and reduces the amount of repeated effort.

This document is no longer extant and has been withdrawn.

Defence | about defence

MODAF Meta Model

The MOD Architecture Framework (MODAF) Meta Model (M3) is the information model for MODAF, defining the structure of the underlying architectural information that is presented in the views. The goal is that MODAF tools are "model-driven" - ie the views that are presented to the user are snapshots of underlying architectural data which is stored in the tool or in a repository.

As well as an overview of the M3, there are three versions of the detailed M3 available to download in XMI, Sparx and HTML formats. For RLI users an online version of the M3 is provided under Applications & Tools/Information Policy & Services.

- **M3 Introduction PDF [123.4 KB]**
- **M3 in Sparx format (zipped) ZIP [2.8 MB]**
- **M3 in HTML format (zipped) ZIP [3.2 MB]**
- **M3 in XMI format (zipped) ZIP [700.4 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

MODAF - detailed guidance

External links

Access online conversion tools for Adobe PDF documents

Page not yet rated

What is the MODAF Meta-Model?

The MODAF Meta Model (M3) is the reference model that underpins MODAF. It, defines the structure of the underlying architectural information that is presented in the MODAF views. The goal is that MODAF tools are 'model-driven'; ie, the views that are presented to the user are snapshots of underlying architectural data which is stored in the tool or in a repository.

Individually, views can only provide consistency in terms of the type of information produced; ie, it can be recognised that one view is a systems model, whilst another is a process model. However, the same information may be represented in more than one view, and there may be important relationships between the information in different views that should be captured. This consistency between views is provided by a reference model which identifies all the types of architectural elements represented across all the views, and the relationships between those concepts. The reference model (or Meta Model in the case of MODAF) therefore provides semantic rigour for the architectural framework.

Many of the benefits from using an architectural approach will ultimately come about from the ability to share, integrate, search and re-use architectural information across an enterprise. In order for the architectural information to be stored, managed and queried electronically, the reference model that underpins the views needs to support the sharing of architectural products between tools and the implementation of an architectural repository that stores those products and the metadata relating to those products.

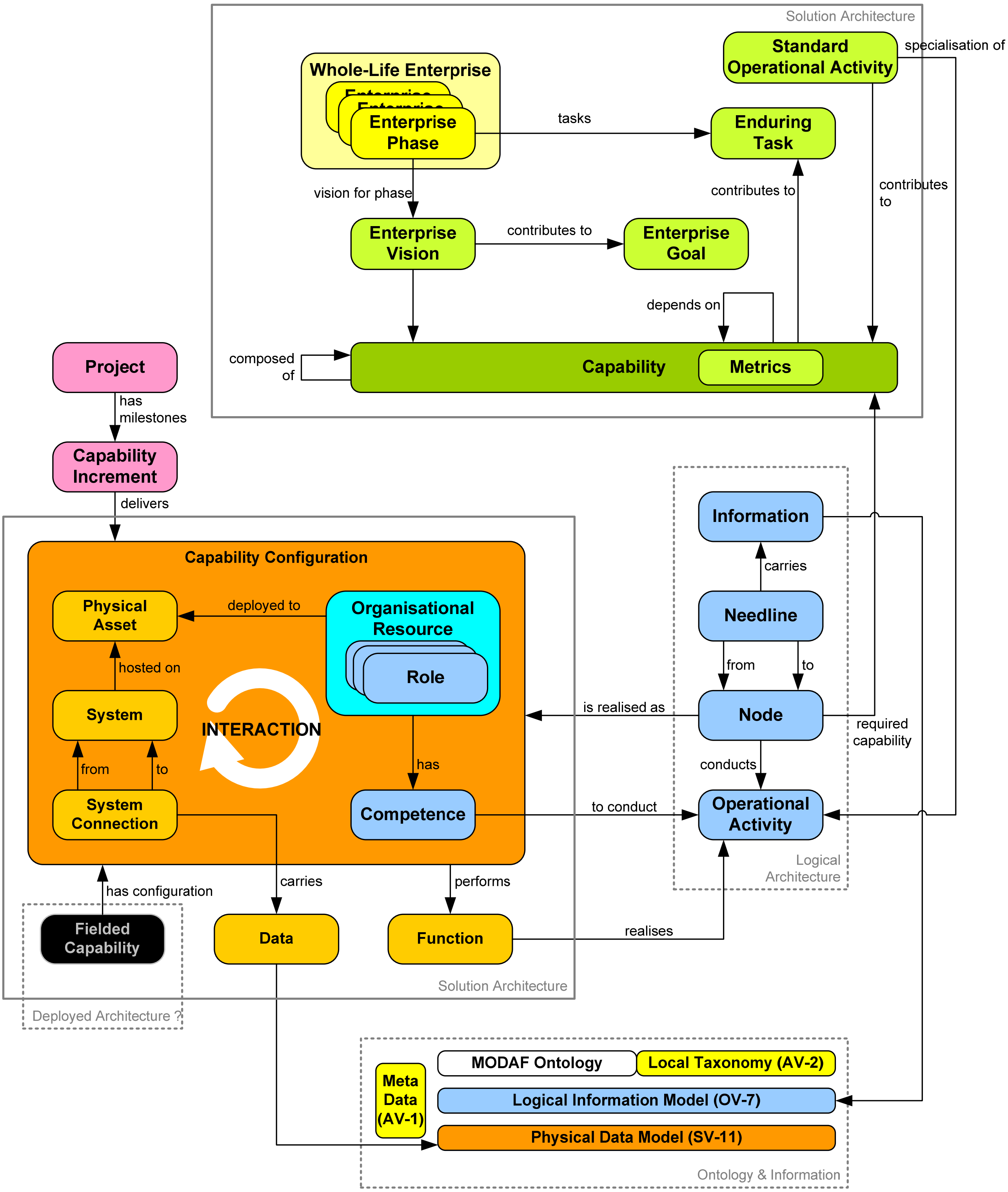
The diagram on Page 2 presents a simplified overview of the M3. This diagram, however, excludes services. The representation of services can be found in the SOA Views document available elsewhere on the MODAF guidance site.

The following table presents the relationship between some of the key elements within the M3 in a different way, showing the relationship between "things" at different levels within the enterprise / node / capability configuration hierarchy and the corresponding activities, entities and flows. In the context of this table, it should be noted that organisational resources and systems are essentially at the same level as capability configuration, which represents the coming together of systems and organisational resources in order to manipulate data and carry out functions.

Thing	Activity	Entity	Flow
Enterprise *eg ISTAR*	**Capability** *eg Imagery intelligence*		
Node *eg Analysis cell*	**Operational Activity** *eg Exploit imagery*	**Information** *eg Intelligence request*	**Activity / Information Flow** *eg IR input to CCIRM*
Capability Configuration *eg Watchkeeper unit*	**Function** *eg Exploit imagery*	**Data** *eg Imagery report*	**Function / Data Flow** *eg Intelligence Reporting*
Organisation, Post, Role *eg Imagery analyst within Watchkeeper troop*	**Human Function** *eg Analyse image*	**Data** *eg Annotated image*	**Data Flow** *eg Pilot - analyst*
System *eg Watchkeeper exploitation terminal*	**System Function** *eg Pre-process image*	**Data** *eg RAW Image*	**Data Flow** *eg Sensor - terminal*

This document is no longer extant and has been withdrawn.

Simplified overview of the MODAF Meta Model



Detailed M3 Description

The M3 is fully described within a dedicated section of this website, in which the information is browsable within a three-frame layout. The introductory page to the M3 description goes on to say that the purpose of the MODAF Meta-Model (M3) is to specify the data exchange format for MODAF architectures. The chosen file format is the Object Management Group's XMI specification (v2.1). In order to make maximum re-use of the XMI interfaces that tool vendors may already have, the M3 is an extension of the Universal Modelling Language (UML) 2.0 Meta-Model. In UML terminology this means that the M3 defines an abstract syntax for a UML profile. Each element defined in the M3 specifies a UML stereotype. The M3 does not provide the concrete syntax (the visual representation of the stereotypes that would appear in a UML diagram) because UML is not the preferred modelling approach for MODAF products – only an abstract syntax is required in order to specify the XMI usage.

Also on the M3 page can be found links to downloadable versions of the M3 in XMI V2.0, Native Sparx and HTML formats.

This document is no longer extant and has been withdrawn.

Frequently asked questions

The following links provide information on ontologies and their use in the MOD Architecture Framework (MODAF), a glossary and abbreviations list and frequently asked questions.

- **Ontologies and their use in MODAF PDF [157.2 KB]**
- **MODAF glossary PDF [41.9 KB]**
- **Comparison of MODAF with other frameworks PDF [131.1 KB]**
- **How MODAF can reflect security concerns PDF [47.6 KB]**
- **Unified Modeling Language (UML) relation to MODAF PDF [24.6 KB]**
- **Coherency across models with MODAF PDF [456.4 KB]**
- **Sharing architecture data PDF [35.4 KB]**
- **Other frequently asked questions PDF [32.5 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

MODAF - Guidance
Detailed Guidance on MODAF

Viewpoints and views
Overview of the differentÂ viewsÂ used in MODAF

Examples and use of MODAF
Including an example of a generic architecting process

MODAF Meta Model (M3)
link to the Meta Model

Configuration control
Version history

Information Coherence Authority for Defence (ICAD)
Related team

MODAF - detailed guidance

External links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page not yet rated

ONTOLOGIES AND THEIR APPLICATION IN MODAF

This document is intended to provide a brief overview of the concepts surrounding the use of ontologies and then to show how they may support a MODAF based architecting process.

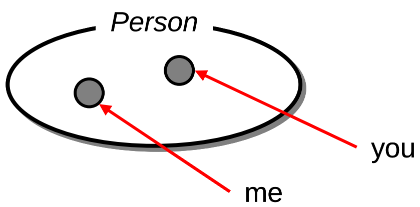
What is an Ontology? [1]

Ontology is the study of what exists. As a discipline, it has been honed over thousands of years by the finest minds in philosophy and mathematics. In recent years, the topic of ontology has come to some prominence in the domain of software engineering. The formal principles of ontology used by mathematicians and logicians have been shown to be useful in enabling software systems to better represent the physical world, and so more accurately support users' requirements.

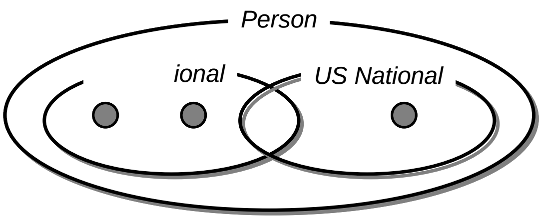
Developing a proper formal ontology is not a task to be taken lightly. New ontologies appear daily on the web, which are little more than traditional data models represented in OWL (the W3C's Web Ontology Language). Where true ontologies do exist, they are the result of years of hard work by academics and software professionals. Good examples are SUMO , ISO15926 and Dolce.

The main benefit of an ontology for an organization like the MOD is that, if properly designed, it can offer great benefits of interoperability. This is because a true, formal ontology aims to describe what exists rather than what is perceived – in other words, it is not slanted towards any particular stakeholder's view of the world. This is sometimes described as a "view from nowhere", and it is this feature that makes an ontology particularly useful in enabling parties with very different views to come to an agreement on meaning. This feature is also the reason that ontologies are difficult to develop properly – each new term in the ontology must be adequately analysed to assess its true meaning and establish how it relates to other parts of the ontology. Great care has to be taken not to "model the entire world", and a practical ontology should remain focussed on the domain it is to support.

Most formal ontology development is based around set theory or similar branches of mathematics and logic such as category theory or type theory. The fundamental components of an ontology are classes and individuals. Individuals are things which have spatial and temporal extent: me, you, the computer you're using, the Eiffel Tower, etc. Classes are categories of things: people, organizations, computers, monuments, etc. This is easily illustrated with a Venn Diagram:



Aside from these fundamental concepts, there are a number of important relationships. The first to consider is the relationship between class and individual (as illustrated in the previous diagram). The second is specialisation; ie one class being a subset of another:



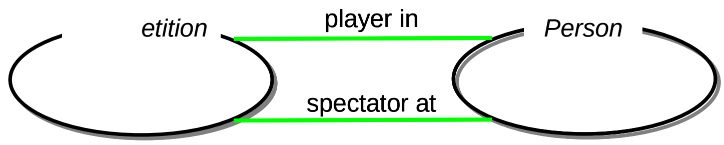
The example above shows that there are two subsets (specialisations) of person which are UK nationals and US nationals. Note that the sets overlap to cover the case of dual nationality.

---

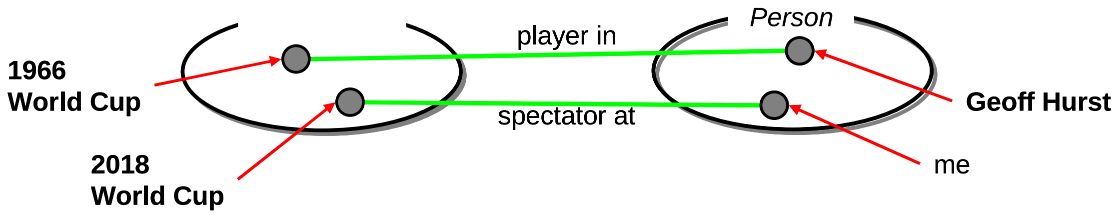
[1] This section was first published in a report on Ontology for the Information Coherence Authority for Defence (ICAD) and the Integration Authority (IA) (now known as the System Engineering Integration Group, the SEIG).

Mistaking class-individual for specialisation is a common mistake; you are not a type of person, you are a person. This degree of semantic precision is essential in developing an ontology and is at the root of what makes an ontology useful and also what makes it difficult to develop.

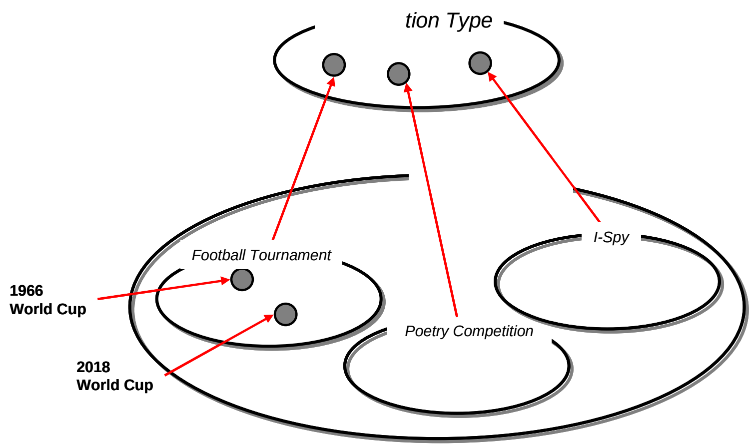
Other than class-individual and specialisation, there is the general case of relationships between classes (sometimes called predicates). These establish common patterns in the ontology, such as:



The example above establishes two relationships that assert people can both play in competitions and be spectators at them. These relationships are classes themselves, that is, there can be actual relationships between individuals:



Another important aspect of formal ontology development is the fact that classes can be classified; it's not just individuals that belong to classes. This is often ignored by ontology developers but it has to be realised that a very large proportion of the information that businesses work with is classification information. To be able to manage this information properly, it has to be classified; ie classification of classes. The example below shows members of the class competition type which are themselves classes:



Any ontology that is to be useful must deal with classes of classes. In formal terms, this is called a higher-order ontology (an ontology which only has classes whose members are individuals is first-order). A practical problem of software implementation exists with higher-order ontologies; a machine reasoner is not guaranteed to resolve an answer from a higher-order ontology in a finite amount of time. This presents something of a dilemma to ontology developers, the real world is higher-order but the reasoners and inference engines can only practically work with first-order ontologies. Two solutions are possible. The first is to develop a higher-order ontology and implement without using reasoners (the commercial benefits of reasoning and inference are still as yet confined to niche applications). The second is to "compress" the higher-order concepts into a first-order framework (this usually means replacing the class-individual relationships with simple predicates).

What is the MODAF Ontology?

Although the MODAF Meta Model describes generic types of architectural information and their relationships, if re-use and integration of architectural products is required, those products must also utilise a common terminology and library of standard elements across architectures. The MODAF Ontology serves this purpose and ensures that each instance of an architectural element (organisation, system, activity, etc) uses a commonly agreed and shared definition for its name. By providing a standard set of terminology and reference data, the MODAF Ontology supports:

- Architectural coherence across the MOD. This is achieved through ensuring all MODAF users employ the same terminology to describe the elements in their Architectures.
- Architectural comparison. Using the same base definitions for standard organisations, systems, activities, etc allows comparison of different aspects of the business.
- Data exchange clarity. Information exchanged between architectural tools can be fully defined using the MODAF Ontology.

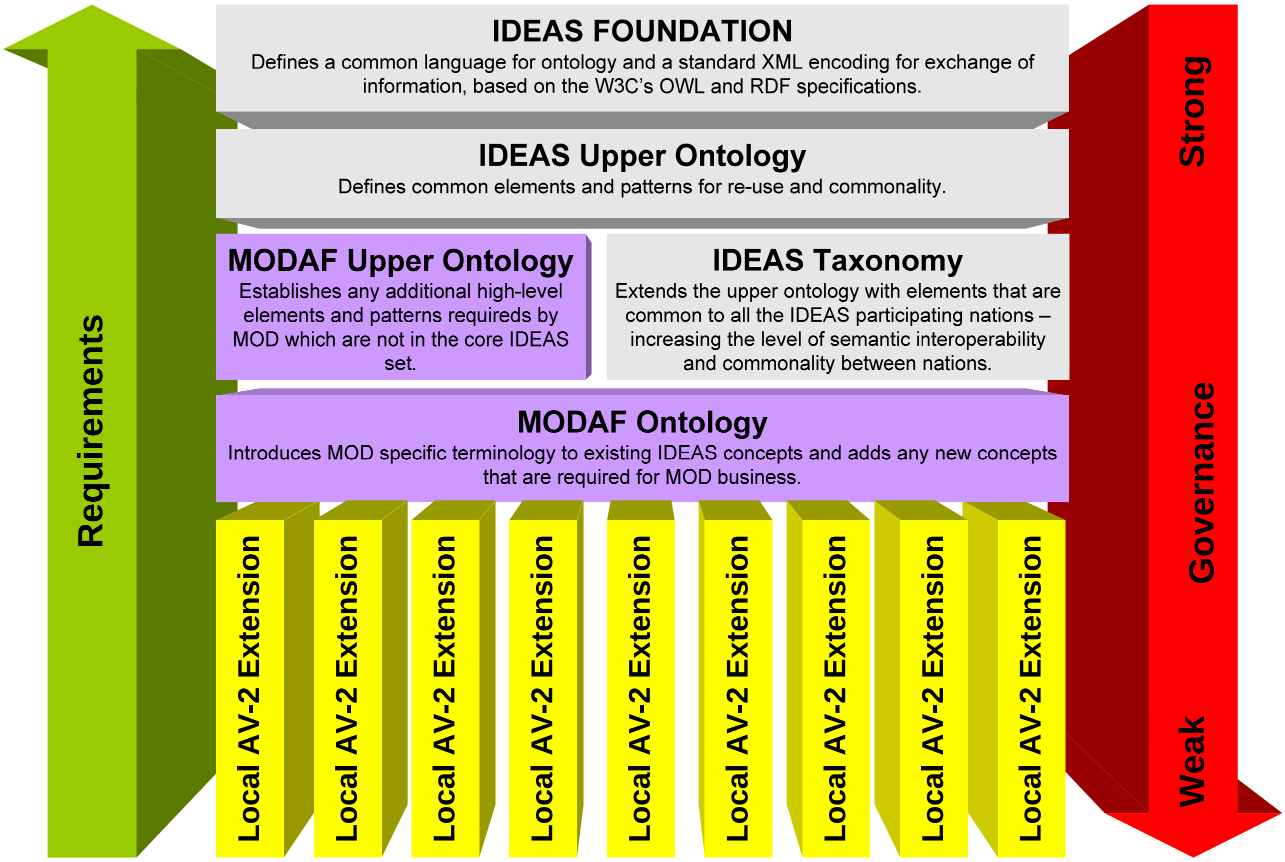
At the time of publishing the MODAF v1.1 document, the MODAF Ontology was at the feasibility stage. ICAD (Information Coherence Authority for Defence – part of DG Info) is responsible for the MODAF Ontology. The approach being investigated is based on the following premises:

- The BORO Methodology[2] is to be used in developing the Ontology
- The MODAF Ontology will extend the IDEAS[3] model – an AUSCANUKUS effort to develop a common ontology for defence enterprise architecture
- The UK Defence Taxonomy and Thesaurus will be the starting point for development, with inputs from other sources of reference data such as the Defence Data Repository, PLCS, BMS, etc.

A governance regime for the Ontology is yet to be formalised, but will be based on a tiered principle, with IDEAS at the top and individual architectures' AV-2s at the bottom:

[2] BORO consulting are currently working on the construction of the Core Enterprise Ontology (CEO) - an 'industrial strength' ontology to be used as a tool by enterprises to significantly improve the semantic aspects of their information systems.

[3] The International Defence Enterprise Architecture Specification for exchange. The purpose of the project is to develop a data exchange format for military Enterprise Architectures to allow seamless sharing of architectures between the partner nations.

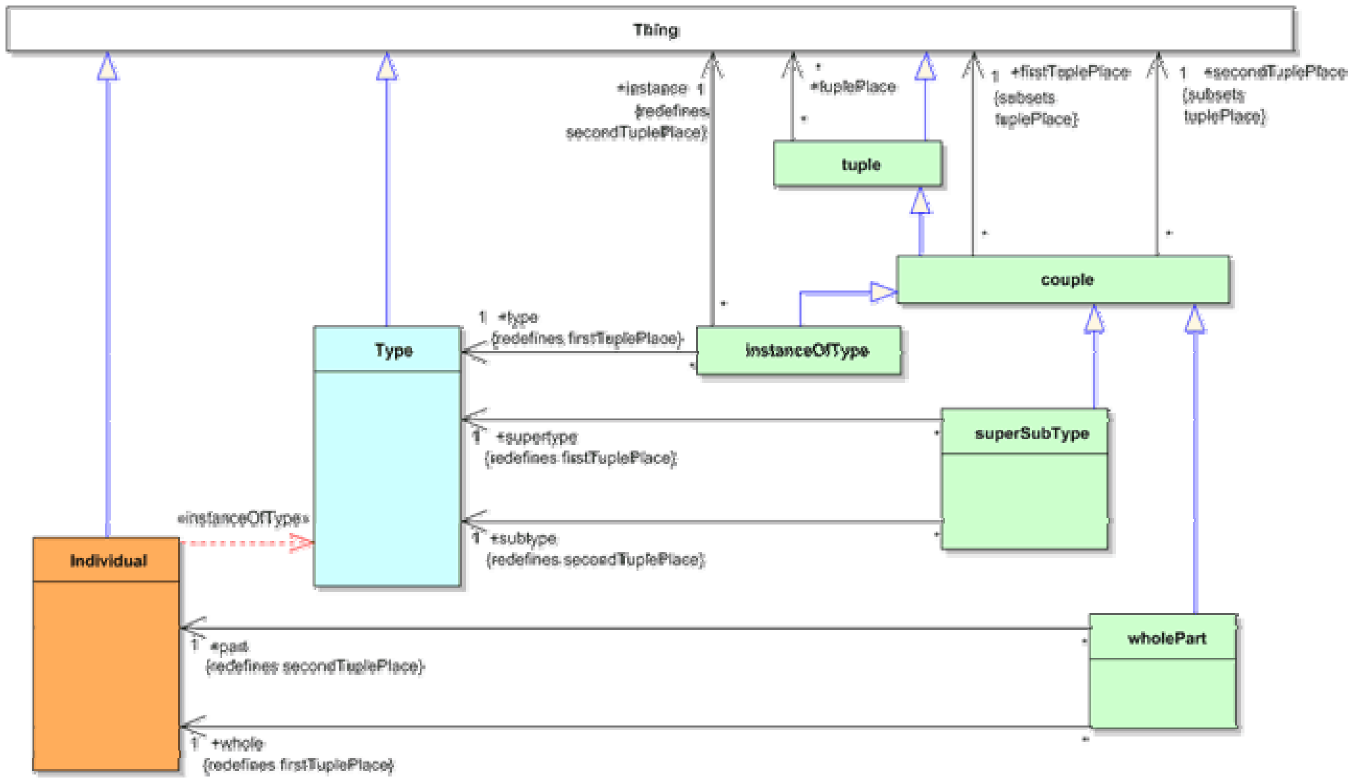


This model allows architects the freedom to make local extensions to the ontology provided the extension is made formally in an AV-2 and linked back to the ontology. The upper layers are, however, more strictly governed. For example, a change to the IDEAS Foundation would require consensus from the four nations and a change to the MODAF ontology would go through the appropriate MOD information management body (eg ICAD).

It is in the nature of architectures that they tend to reference each other and use common elements. If an architect introduces a new element (eg a new system) then it is likely that it may need to be referenced in other architectures. For this reason, there needs to be a way to migrate elements up the tiers as time goes by; this could even happen at the national level, where more than one nation adopts a new technology or way of working.

The approach being taken in IDEAS allows multiple names (with context) to be applied to all the elements. In other words, elements are created once, but may be named several times by different stakeholders. This mechanism allows communities, nations, etc. to work with their favoured terminology but still allows for commonality between the parties.

The IDEAS model is still in development. In the mean-time, some early parts of the IDEAS Foundation have been made public (but these may change before formal publication):



The Key Elements of the IDEAS Foundation

The suite of MODAF documents on this web site also includes a glossary and acronym list. Whilst these will be consistent with the Ontology, their purpose is only to support the readability of the MODAF documents themselves.

MODAF Glossary v1.2

This glossary defines terms that have been widely used in the context of MODAF and Enterprise Architecture. Where terms are not defined, the ordinary dictionary usage should be assumed. Where applicable comments have been added, identifying known sources for the definition and expansions on the definitions.

This glossary is for the MODAF v1.2 release and some elements have been updated. MOD policy is for alignment of terms with the MOD Defence Terminology as maintained by Information Coherence Authority for Defence (ICAD)[1] and the MOD Abbreviations list. In future, only where there are specific definitions for MODAF use will these definitions be listed.

Architectural Product	A connected and coherent set of **Architectural Elements** which conform to a **View**	
Architecture	The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time	Source IEEE 1471
Artefact	Any element in the physical domain that is not **software** or **organisational** (where organisational includes people)	Systems, physical assets, components, hosting systems etc are all artefacts
Attribute	A property or characteristic, or a testable or measurable characteristic, that describes an aspect of an **entity** or an **object**	
Capability	Capabilities in the MODAF sense are specifically not about equipment but are a high level specification of the **enterprise**'s ability. A capability is a classification of some ability – and can be specified regardless of whether the **enterprise** is currently able to achieve it. For example, one could define a capability "Manned Interplanetary Travel" which no-one can currently achieve, but which may be planned or aspired to. Capabilities in MODAF are not time-dependent – once defined they are persistent. It is only the Capability Requirement that changes.	
Capability Configuration	A composite structure representing the physical, human and software **resources** (and their interactions) in an enterprise. A **capability configuration** is a set of artefacts or an organisation configured to provide a capability, and should be guided by doctrine and policy, and should take account of all the Defence Lines of Development (DLODS). These replace the system node concept in DODAF.	

[1] Link to ICAD can be found on the Glossary & Abbreviation 'home' page.

Term	Definition	Comment
Data	A representation of individual facts, concepts or instructions in a manner suitable for communication, interpretation or processing by humans or by automatic means.	Source IEEE 610.12 Used in the MODAF System Views
Enduring Task	A type of behaviour recognised by an enterprise as being essential to achieving its goals – i.e. a strategic specification of what the enterprise does. Examples include Financial Management and Human Resource Management. These equate to Business Functions in the commercial world.	
Enterprise	The term enterprise can be defined in two ways. The first is when the entity being considered is tightly bounded and directed by a single executive function. The second is when organisational boundaries are less well defined and where there may be multiple owners in terms of direction of the resources being employed. The common factor is that both entities exist to achieve specified outcomes. (1) An organisation(or cross organisational entity) supporting a defined business scope and mission that includes interdependent resources (people, organisations and technologies) that must coordinate their functions and share information in support of a common mission (or set of related missions). (US Federal CIO Council) (2) A systematic, purposeful set of activities whose primary purpose is focussed on achieving a set of clearly defined objectives that may transcend organisational boundaries and consequently require integrated team working under the direction of a governing body of resource providers.	
Enterprise Architecture	The formal description of the structure and function of the components of an enterprise, their inter-relationships, and the principles and guidelines governing their design and evolution over time. (Note: 'Components of the enterprise" can be any elements that go to make up the enterprise and can include people, processes and physical structures as well as engineering and information systems.	
Enterprise Architecture Framework	A logical structure for classifying, organising and presenting complex information relating to **Enterprise Architectures** in a uniform manner.	
Entity	A representation of an **object**, with characteristics (or **attributes**) and **relationships**, that exists in one or more a**rchitectures**. It is a 'blueprint' for creating **objects** for a particular **architecture**. An **entity** could be a building, ship or organisation amongst other things.	

Term	Definition	Comment
Function	A function is performed by a resource. For IT systems, it tends to refer to data transformations. However it also covers human functions and software functions where these are manipulating data or physical elements.	
Information	The refinement of **data** through known conventions and context for purposes of imparting knowledge to individuals.	Used in the MODAF Operational Views
Information Exchange	The collection of **information** elements that are bundled together to form a **needline**.	
Information Exchange Requirement	Defines the requirement for an exchange of information in terms of who has the need, what is exchanged and the quality of service.	
MetaModel	Strictly this means a model of a model. In the MODAF context this means a representation of the **entities** (and **data elements**) pertinent to an **architecture**, including the **relationships** amongst **entities** and their **attributes** or characteristics.	
Needline	A requirement that is the logical expression of the need to transfer **information** between nodes. A needline bundles **information exchanges**.	
Node	A logical entity that performs operational activities. Note: nodes are specified independently of any physical realisation.	
Object	An instance of an **entity** that forms part of a particular **architecture**. It has **attributes** that are specific to its instance, but characteristics and **relationships** that are common to other **entities**.	With reference to parts of an aircraft, it could be ‘left wing’, ‘tail plane’ or ‘Eurofighter undercarriage’
Ontology	Ontology is the study of what exists. In architecture terms, it allows us to model the ‘things’ we see in the real world without the confusion of the different names applied to them. Ontologies are related to but different from **taxonomies**.	
Operational Activity	An activity is an action performed in conducting the business of an **enterprise**. It is a general term that does not imply a placement in a hierarchy (e.g. it could be a process or a **task** as defined in other documents and it could be at any level of hierarchy). It is used to portray operational actions not hardware/software system functions. **Operational Activity** may include either military operations or business processes.	

Term	Definition	Comment
Relationship	A connection that asserts how an **entity** (or **object**) can permissibly interact with the outside world (i.e. with other **entities** or **objects**).	
Requirement	A need, demand or constraint.	
Resource	An abstract element that in the physical or solution space. The top level resources are **artefacts**, **organisational resources** or **software**.	Seen as Resource Type in the MODAF meta model
Role	A function or position filled by a person or post.	
System	MODAF 1.2 specific.	
System of Systems	A set or arrangement of interdependent **systems** that are related or connected to provide an enhanced measure of **capability**. The loss of one component **system** will degrade the performance of the whole but not affect the performance of the individual **systems**.	
Taxonomy	Provides the standard structured dictionary for the architecture. A **taxonomy** has the ability to constrain the diversity of an **architecture** to ensure consistency across the **enterprise**.	
View	A specification of a way to present an aspect of the **architecture**. Views are defined with one or more purposes in mind e.g. show the local enterprise topology, describe a process model, define a data model, etc.	
Viewpoint	A collection of **Views**. Viewpoints are usually categorised by domain i.e. in MODAF we have Acquisition, Strategic, etc.	

UNCONTROLLED COPY WHEN PRINTED

FAQs

How does MODAF relate to other Architecture Frameworks?

One of the MOD aims for MODAF is to preserve an appropriate level of international alignment. This is because there is a degree of multinational co-operation in respect of architectures, implying that it is highly desirable that there is compatibility between architectural frameworks, the tools that support their use and the skills and knowledge employed by architects in different nations.

This article refers to the relationship between MODAF and:

US DoD Architectural Framework (DoDAF)

NATO Architectural Framework (NAF). (Access the Framework)

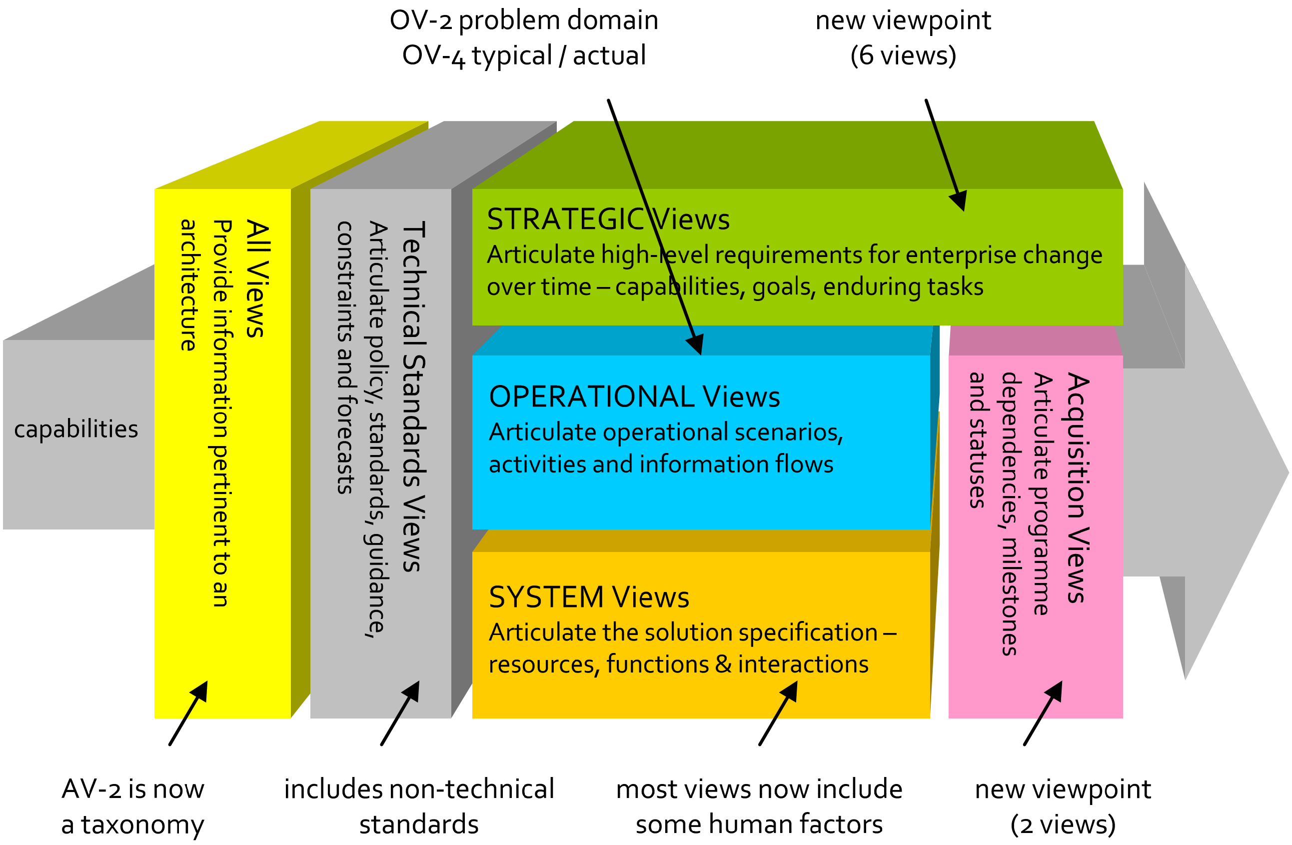
The Open Group Architectural Framework (TOGAF)

Zachman Framework

Relationship to DoDAF

MODAF is based on the DoDAF version 1 baseline, and this section summarises the main distinctions between MODAF – as represented by the MODAF Meta Model (M3) – and the current version of DoDAF. It is recognised that DoDAF is in a state of evolution so the contents of this section should not be taken as indicating significant divergence from the evolving DoDAF.

The effects of the move to DoDAF v1.5 (released in March 2007) have not yet been assessed. Note also that the introduction into MODAF of two new viewpoints has resulted in the need for review of the ex-DoDAF views; some changes have been made to MODAF (at version 1.1) as a result of this review.



In summary, the following factors have led to differences between core MODAF at version 1.1 compared with DoDAF version 1:

- The need to model incremental acquisition programmes as these represent an increasingly common form of defence procurement
- The need to model transformational programmes and their inter-dependencies
- The need to model capabilities as the outcome from force development and capability integration programmes
- The need to model solution resources in terms of people as well as technical system resources
- The need to model physical attributes and capabilities and, by extension, flows of personnel, energy and materiel not just information
- The need to integrate programme models into traditional architecture models in order to meet the needs of enterprise architects
- A drive towards a more coherent object oriented underpinning for the Architectural Framework.

None of these factors are believed to be specific to the UK procurement regime or UK defence architecture requirements. It is therefore expected that, over time, existing defence architectural frameworks like DoDAF will evolve to accommodate the changing needs of defence architects.

Strategic Viewpoint

The Strategic Viewpoint was introduced into MODAF to address the concerns of Capability Managers. In particular, strategic views describe capability taxonomy and capability evolution. The Viewpoint is an essential component of an enterprise architectural framework, since the Enterprise view is all about strategic change. In DoDAF, it could be argued that this Viewpoint was not needed because, at the time of writing version 1 of DoDAF, it was envisaged that architecture models would be written in one of only two states: ‘As Is’ (capturing the current capability) and ‘To Be’ (capturing the intended target capability).

MOD increasingly employs incremental acquisition to help to manage the risks of complex procurements, and there is consequently a need to provide visualisations of the evolving capabilities so that Capability Managers can synchronise the introduction of capability increments across a Programme of Projects. The views included within MODAF’s Strategic Viewpoint are based on the programme and capability visualisation techniques that are used by Capability Managers to capture the increasingly complex relationships between inter-dependent projects and capabilities.

Another justification for the Strategic Viewpoint within MODAF is the increasing importance of transformational programmes within the MOD (e.g. NEC, Logistics Transformation). These types of programme do not conform to the standard form of project and tend to be benefit-driven rather than capability delivery focused. An ability to model these transformational programmes, and their inter-dependencies, provides a potentially powerful tool for defence Enterprise Architects.

Acquisition Viewpoint

The Acquisition Viewpoint was introduced into MODAF to address the concerns of Acquisition Managers. In particular, acquisition views describe projects, how those projects deliver capabilities, the organisations contributing to the projects and dependencies between projects.

DoDAF takes a traditional view of architecture in which programme development is considered outside scope; to compensate for this, various DoDAF views represent the evolution of systems, technologies and standards (e.g. SV-8, SV-9 and TV-2).

The integration of acquisition views (organisational and project oriented views) with the more traditional architecture views is a characteristic aspect of MODAF-based enterprise architecture.

This approach provides most benefit when time-based views are accepted as being needed at all levels within an enterprise architecture.

Model Concepts

The following DoDAF model concepts have been amended during the development of MODAF:

- Needline: Node Connections (a new construct) enable modelling of flows of energy, material and personnel flows as well as the information flows that are addressed by Needlines (the M3 has deliberately avoided extending the definition of Needline to retain compatibility with DoDAF).
- Node: MODAF reasserts the logical nature of an operational Node. What DoDAF calls System Nodes are Physical Assets in M3.
- Organisation: MOD requires the preservation of the distinction between organisations and posts; this has been accomplished by introducing the Organisational Resource (in effect Organisational Resource plays the same role in MODAF as Organisation does in DoDAF). MODAF also makes a clear distinction between actual organisations and types of organisation.
- System Function: In order to enable more refined modelling of information-rich equipment capabilities, a MODAF System Function may act on a particular set of Data Elements. Similarly, a MODAF Operational Activity may act on a particular set of Information Elements. In version 1.1, the more general Function has been introduced – this can be provided by any functional resource (capability configuration, system, role).

Specific Views

The DoDAF views relate to the Operational, Systems and Technical Viewpoints in MODAF. MODAF has changed the names of the following ex-DoDAF views to more accurately describe their content: OV-2, OV-7, SV-4, SV-5, TV-1, TV-2.

In addition to the changes associated with the revised model elements, the following DoDAF views have been amended during the development of MODAF:

- SV-1 (resource interaction specification): introduction of Capability Configuration and human solution resources
- SV-2 (system connection specification): refinement of views to address Protocols and Protocol stacks
- SV-11 (data model): greater integration with SV-4 and OV-7.

The MODAF form of OV-5 strives to combine support for object-oriented (UML) and structured methods (IDEF0).

Finally several of the DoDAF views have been amended to reflect integration with the Strategic and Acquisition Viewpoints in MODAF. These changes focus on the relationship between Capability Configurations and Capability Increment (milestones).

Relationship with NATO Architectural Framework (NAF)

NATO has adopted M3 and MODAF (at NAF Revision 3) and has add several new views. In particular, NAF is the source for the System Orientated views being incorporated into MODAF v1.2. The final published version of NAF 3 is available at this site.

Relationship with The Open Group Architectural Framework (TOGAF)

A comparison between MODAF and TOGAF has been carried out as part of the DLO SOA study work. This is available here. (requires registration)

Relationship with Zachman Framework

Whilst considering the relationship between MODAF and DoDAF, it is also worth looking briefly at how MODAF might map onto the widely-used Zachman Framework.

The following points should be considered with respect to this mapping:

- The Strategic Viewpoint in MODAF is about Enterprise Planning, and maps on well to the Zachman Scope/Planner layer.
- The Operational Viewpoint in MODAF is about the business at any one time, and maps on well to the Zachman Business Model / Owner layer.
- The System Viewpoint MODAF is about Capability Specification (effectively, designing and putting together Capability Configurations) and therefore maps on well to the Zachman System Model / Designer layer.
- The Technical Standards Viewpoint in MODAF is about standards applicable to the enterprise as a whole, rather than being specifically about technology. The implementation technology aspects of an architecture are not currently covered in MODAF (apart from the communications details provided by the various parts of SV-2). TV-1 and TV-2 therefore fit in well up in the Zachman Scope/Planner layer.
- The Acquisition Viewpoint in MODAF is really about looking in more detail at the Enterprise Planning process. AcV-2 therefore also fits in well up in the Zachman Scope/Planner layer.
- The MODAF StV-6 and SV-5 views are mappings between layers, and have therefore been shown as such.

FAQs

How does MODAF represent security?

Capabilities are subject to a variety of threats to the integrity, availability and confidentiality of their operation. These threats range from failures of equipment, attempts to gain unauthorised access to their services and data, through to sabotage of their functions. Security engineering is concerned with identifying the potential threats to a capability, and then, using a risk management approach, devising a set of measures which reduce the known and potential vulnerabilities to an acceptable level. In general the measures that can be applied fall into the following categories:

- Physical – measures such as guards, guard dogs, fences, locks, sensors, including CCTV, strong rooms, armour, weapons systems, etc.
- Procedural – the specification of procedures, including vetting (which tests that personnel have a sufficient level of integrity and trust to be given responsibility to access and use a capability's services and data) that will reduce the likelihood of vulnerabilities being exploited.
- COMSEC –using encryption and other techniques to ensure that data transmission is available at sufficient bandwidth, that the traffic pattern and content of data in transit are indecipherable to a third party who might intercept the data, and that its integrity is protected.
- TEMPEST – measures to ensure that the electromagnetic transmissions from equipment can't be intercept in order to derive information about the equipment's operation and the data it processes.
- INFOSEC – ensuring the integrity, availability and confidentiality of data and IT-based services.

In general, the measures employed to protect a capability will have undesirable impacts on all of the capability's lines of development, and in particular on its deployability, usability and procurement and maintenance costs. It is therefore desirable to minimise the strength of the measures to be employed in a fashion commensurate with the value of the assets being protected. This requires a risk-managed approach based on the assessment of the likely threats posed to the asset. The UK undertakes this risk assessment by considering the following characteristcs:

- Environment – The level of hostility of the environment the asset is being deployed to.
- Asset Value – this is denoted by a protective marking which indicates the impact of the loss or disclosure of the asset would have on the effective operation of the UK government and its departments of state.
- Criticality – an assessment of the criticality of the asset to enabling the UK government to undertake its activities.
- Personnel Clearance – a measure of the degree of trust that the UK government is willing to put in the personnel that will have (direct or indirect) access to the asset.

The Defence Manual of Security, JSP 440, formulates MOD's policies for protecting its assets and those of other government departments and nations with whose protection it is entrusted. JSP 440 calls on other HMG policies, particularly for communications and information security those of CESG. Security policies and procedures must also be compliant with various legislation such as the Data Protection Act and Regulation of Investigative Powers Act.

The aim of this guidance for representing security considerations is to enable sufficient information to be recorded for interested parties (accreditors, security advisors, users, system managers) to understand the potential security exposure of capabilities so that security can be managed effectively throughout the life of a capability. It is not the aim to provide an alternative for a formal security policy constructed in accordance with JSP 440, although the information provided using this guidance should provided the starting point for the necessary analysis required to derive such a policy, and the views created could be used as part of a security policy.

The table below shows the MODAF scheme for assigning security characteristics and protective measures to elements of MODAF. There is not a specific "security view" in MODAF: security information can be shown on views using annotations and call -outs, UML features or styling of symbols and edges. An appropriate key should be provided. A model library is provided with the MODAF Meta-Model to underpin the representation of security characteristics in a consistent way between models. Protective Measures are captured in MODAF using sub-types of SysML::Requirement. A non-normative extension to the MODAF Meta-Model is also provided containing these sub-types.

Viewpoint	Element	Security Characteristics	Protective Measures	Notes
Strategic	Capability Requirement	Security Marking Criticality Environment User Security Profile		The security characteristics of a capability requirement provide the security envelope for the capability during a particular epoch.
Operational	Node	User Security Profile Environment		The USP is the lowest clearance of users who will constitute a realised node. The environment identifies the most hostile conditions a node will be realised in. Nested nodes can be used to represent security domains, with sub-nodes in a 'domain' deriving their characteristics from the most immediate owning 'domain.'
	Operational Activity	Security Marking Criticality		The security marking identifies the highest security marking of information that will be processed by a realised Operational Activity, and the Criticality measures the impact on Government operations of the disruption of the activity.
	Node Connector Type	Security Marking		The security marking identifies the highest security marking that will be exchanged across a node connector of this type.
	Organisation/Post	User Security Profile Environment		The minimum clearances, etc of members of the organisation/post.
System	Capability Configuration	Environment* User Security Profile* Criticality* Security Marking*		The security characteristics for a capability configuration are to be derived from the constituents.
	System	Environment* User Security Profile* Criticality* Security Marking*	Physical TEMPEST COMSEC	The environment of a system is derived from the Physical Asset to which is deployed. The USP is derived from the Organisation which uses the system, its Criticality and Security Marking from its Functions.
	Physical Asset	Environment	Physical TEMPEST	The environment identifies the worst environment to which the Physical Asset will be deployed.

	Function	Security Marking Criticality	INFOSEC Procedural	Security Marking identifies the maximum security marking of the data the Function will process, and its criticality represents the degree of harm to Government operations if it is disrupted.
	Resource Interaction Specification	Security Marking	COMSEC	The Security Marking represents the maximum security marking of information transversing the interaction.
	Role	User Security Profile	Procedural	The USP is the lowest clearance, etc of the user who will undertake the role. This should be derived from Organisations and Posts who can undertake the Role, if that information exists.

FAQs

How does MODAF relate to UML?

The M3 defines a Unified Modelling Language (UML) profile by extending the UML 2.0 meta model, which in turn specifies the structure and content of XML Metadata Interchange (XMI) files used for exchanging information between MODAF tools. The appropriate elements of the MODAF Meta Model (M3), that are needed to exchange the information for a MODAF View, are described within the definition for each View within this web site. It should be noted that the elements shown for one View may also be used in several other Views.

The classes defined in the M3 specify the allowable UML stereotypes that may be exchanged in an XMI file. As it is a Meta Model, all relationships that feature in a View are also modelled as classes. Rather than define a class for every conceivable item that could appear in a View, the meta model defines generic classes and allows references to the MODAF Ontology. For example, the MOD would be represented in XMI as an Organisation stereotype, with a tagged value referring to the element in the Ontology which says “Ministry of Defence”.

In addition to the M3 being underpinned by UML, many of the Views are capable of being expressed in the form of UML diagrams. Such cases are made clear within the individual View descriptions, but in summary they are:

- Use of UML class and/or assembly diagrams for StV-4, OV-2, OV-4, OV-7, SV-1, SV-2c, SV-11 and AcV-1
- Use of UML activity diagrams for OV-5,
- Use of UML class and/or activity diagrams for SV-4
- Use of UML ports notation for SV-2a and SV-2b
- Use of UML constraints notation for OV-6a and SV-10a
- Use of UML state charts for OV-6b and SV-10b
- Use of UML interaction (sequence) diagrams for OV-6c and SV-10c

M3 itself is a UML profile. The specific way in which the Meta-Model has been specified in terms of the UML meta-model is described in the M3 introduction (link tbd). In particular, this explains the role of Composite Structure Diagrams in MODAF.

SysML is also relevant here, since Architecture Modelling can be regarded as being a branch of Systems Engineering, and SysML represents a Systems Engineering extension to UML. As time goes on, it is likely that SysML will become more applicable than UML.

FAQs

How does MODAF help to ensure Model Coherency?

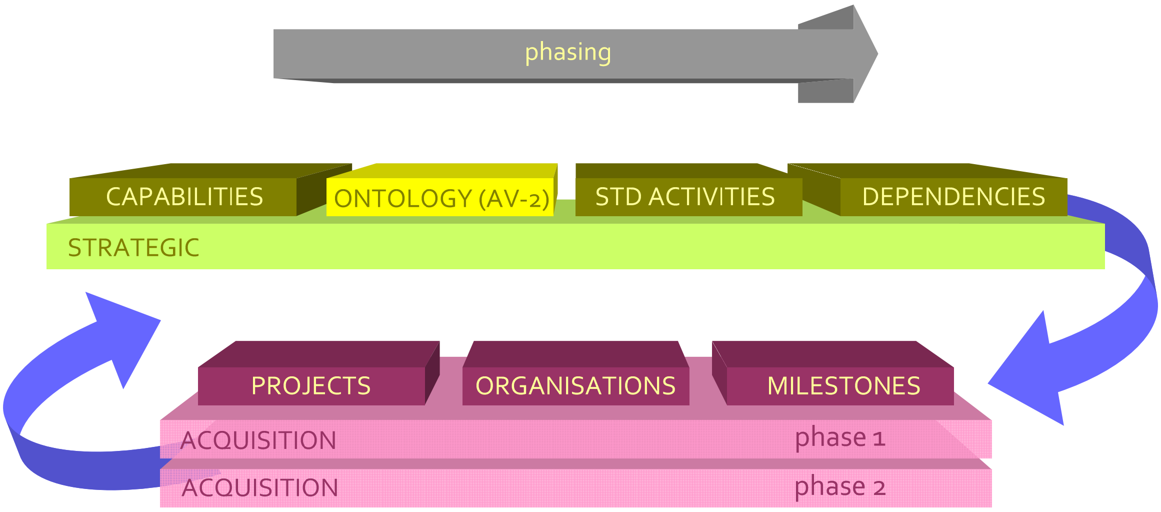
The MOD Architectural Framework has been designed to be highly integrated. Key to this is the role played by the MODAF Meta-Model. Integration has been achieved at the view presentation level and the data level.

Because an architectural description is inevitably multi-faceted, modelling coherence is an essential ingredient of the model-driven development approach. The framework supports the achievement of modelling coherence in a number of ways that are described in this article.

Temporal Coherence

The framework has a much richer temporal component than other architectural frameworks because the representation of time (in the enterprise sense) is addressed by two viewpoints, the Strategic and Acquisition Viewpoints, that are unique to MODAF.

Temporal coherence is based on integration of these two viewpoints, which is illustrated below.



The time dimension in the Strategic Viewpoint represents the fundamental time dimension of the Enterprise; this is normally measured in years and may have a span of several decades with a granularity of perhaps several years.

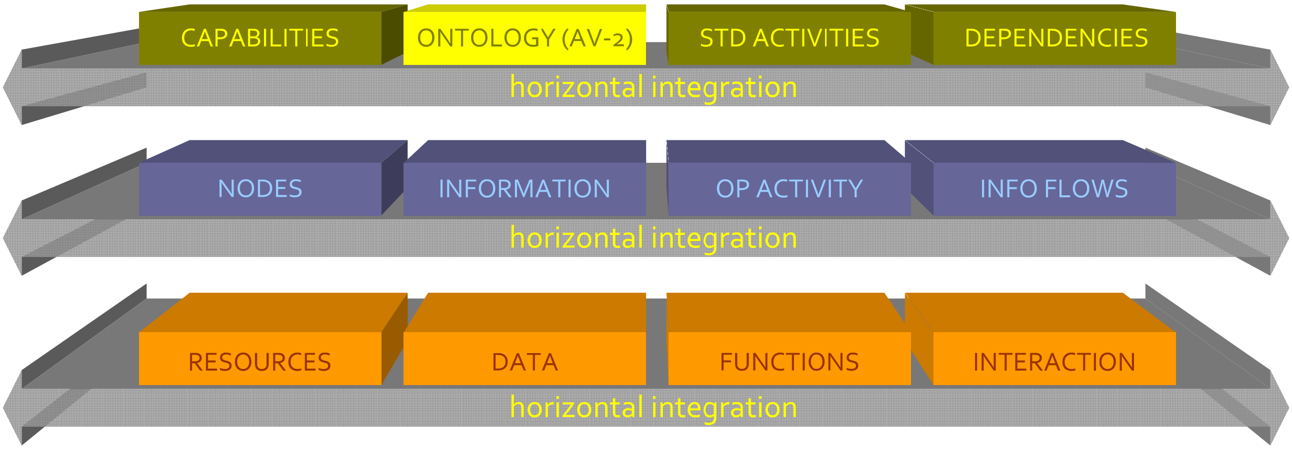
The time dimension in the Acquisition Viewpoint may have nearly as broad a scale as that of the strategic Viewpoint but is finer grained. Time gradations of interest may be months or years.

The time-based views within these Viewpoints (specifically StV-3 and AcV-2) need to be considered together if Enterprise goals are to be met. StV-3 provides a time-based representation of capability phasing. AcV-2 provides a timeline at the programme or project level. Matching these up is important if the strategic planning aspiration to realise capability at the enterprise level at a particular time is going to be fulfilled through action at the project level. AcV-2 can be used to perform programme synchronisation actions that manage the risks associated with the co-ordination of a related set of capability deliveries. But StV-3 is needed to ensure that the overall programme of capability delivery actually meets the longer term aspirations of the Enterprise.

In addition to the views in the Strategic and Acquisition Viewpoints, there are also a small number of views looking at evolution at the system level including SV-8 (system evolution), technology forecast (SV-9) and standards forecast (TV-2). The timescales used in these views should mirror those in StV-3 or AcV-2 depending upon the scale of the evolution timeline.

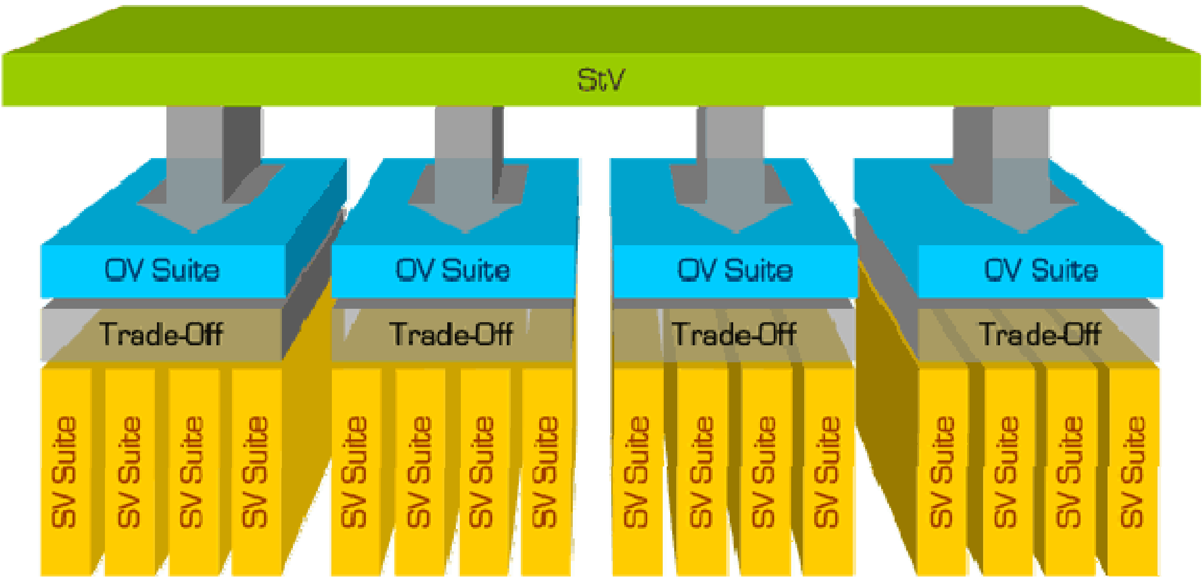
Horizontal Integration

Horizontal integration refers to the achievement of modelling coherence within a Viewpoint (or modelling layer). This is based on the support to view production provided by a coherent set of data objects underpinned by the MODAF Meta-Model. This is illustrated below.

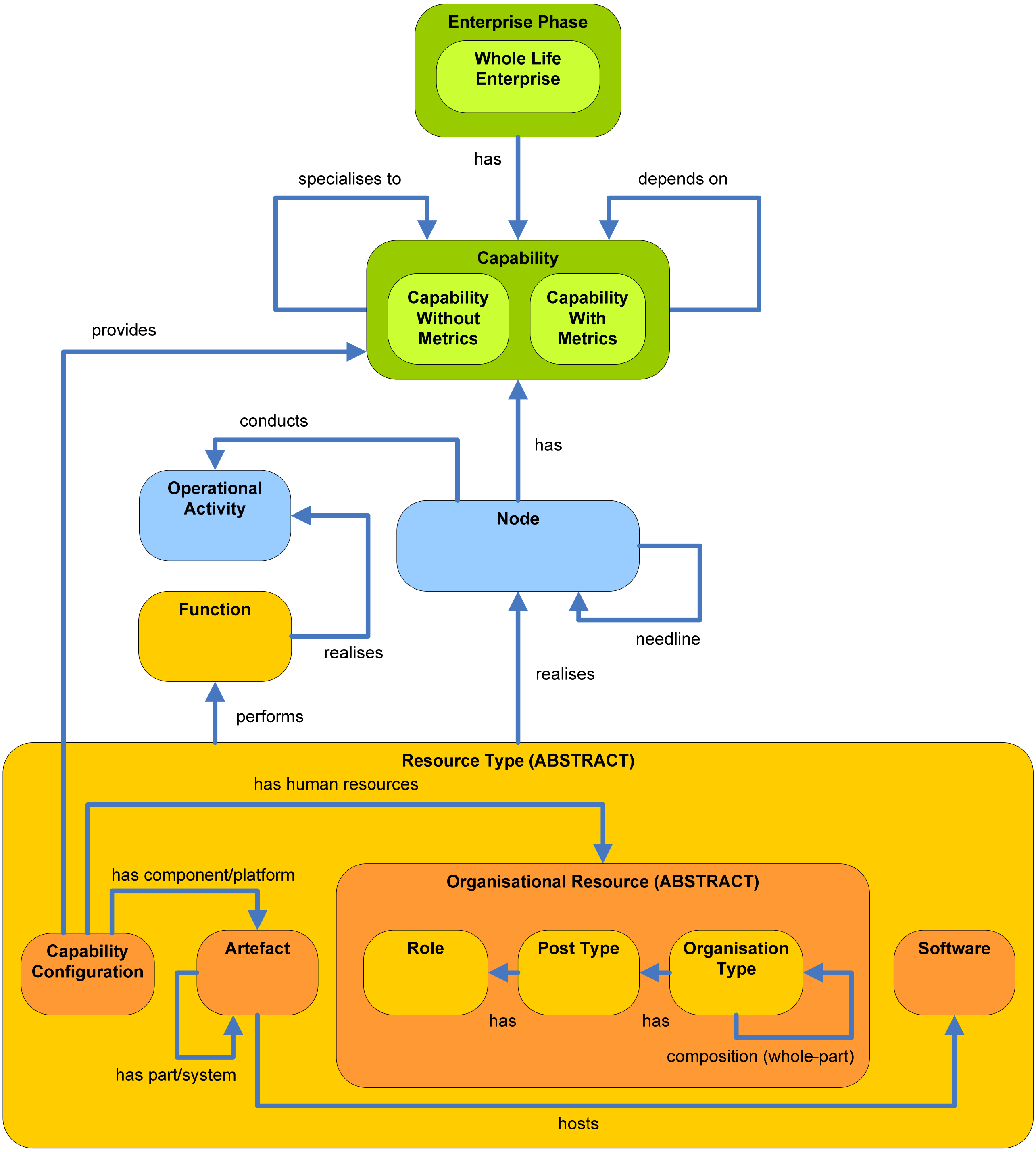


Vertical Integration

Finally vertical integration refers to the achievement of modelling coherence between Viewpoints as illustrated below.



The elements and relationships defined in the MODAF Meta-Model provide the foundation for this. The diagram below summarises the key structural and behavioural elements in the M3, and the relationships between them:



Simplified Representation of Key M3 Concepts and Relationships

Architecture Governance

While modelling coherence is supported by use of MODAF, especially use of the MODAF Meta-Model, it should be recognised that architecting within the MOD environment is inevitably a collaborative activity. This implies that architecture governance is needed to ensure success. This is the responsibility of DG Info, who is discharging this responsibility through the Enterprise Architecture programme (within which the MODAF development programme sits).

The primary elements of governance relate to the direction to the MOD community to actually use MODAF, arrangements for assurance of architecture products, and support to the MOD community from an architectural repository (*see EA policy* – link tbd). The Meta-Model (link tbd) and MODAF Ontology (link tbd) are technical enablers that support the governance arrangements.

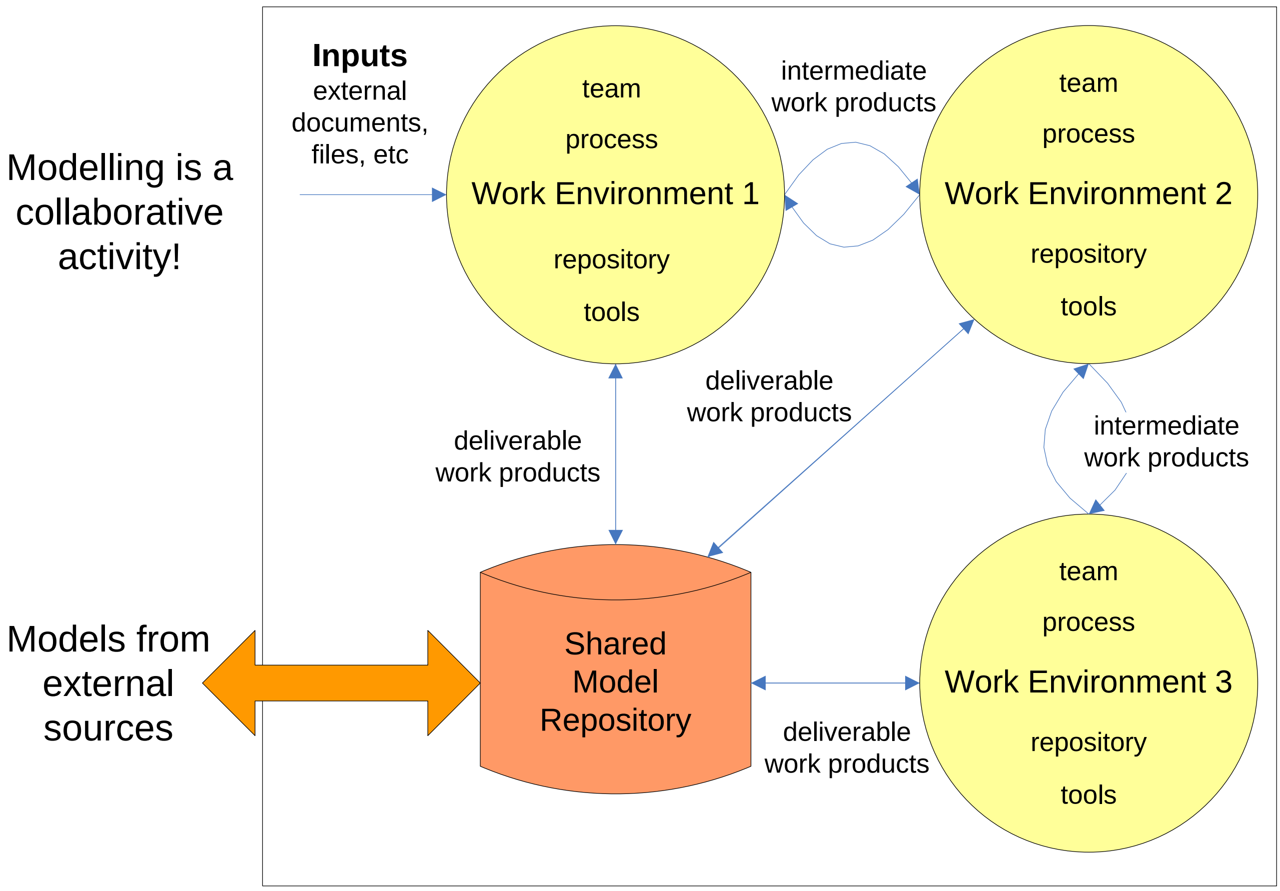
Disciplined application of MODAF (within the parameters prescribed by governance) through collaborative architecting is essential if the vision of NEC is to be achieved.

FAQs

How is architecture data shared between users?

It is important to appreciate that architecture modelling is a collaborative activity, and cannot be undertaken by individuals working in isolation. Various individuals may be responsible for developing models of specific aspects of an enterprise, for a specific purpose, and these individual pieces of work can have some value in answering specific questions. However, the full value of architecture models is only realised when they are collected together within a shared model repository, and more particularly are linked together in a way that recognises common shared elements and relationships.

The diagram below illustrates this point. The diagram shows a number of separate working environments (which could be co-located or undertaken at different locations), each responsible for developing their own MODAF-compliant architecture work products. There may be an informal exchange of intermediate work products, as a means of sharing knowledge and attaining a degree of consistency. Ultimately, however, there will come a point where it is necessary for finished architecture work products to be collated within a shared repository (such as the IA's architecture repository). A number of practical matters need to be dealt with in order for this collation to be undertaken and maintained, including issues such as naming convention, ontology, use of modelling construct, and configuration management. These are currently being looked into by the IA and other organisations involved in developing and maintaining architecture repositories.



The diagram also makes the point that there may be other valid sources of model data, in addition to the products of the collaborating architecture teams, that can usefully be collated in some way with the growing repository of MODAF-compliant architecture models. Depending on the standards used for developing these external models, there may be limits to how well they can be formally integrated into the repository. However, this should not prevent useful relationships being established between common or related elements, which could support a wider and richer range of queries than would be possible if limited only to the core MODAF-compliant set. This is likely to involve the wider use of semantic web technology to create equivalences and relationships between modelling domains that adhere to related but different ontologies.

MODAF supports the sharing of architectural data in the following ways:

- at the presentation level, the standardised views help MODAF users to understand the models created by other people
- at the data level, the MODAF Meta-Model (when used appropriately in conjunction with the XMI2.1 open standard), provides the data format standard aimed at ensuring passage of data in a tool-agnostic manner.

Use of architectural repositories within MOD will formalise the sharing of information – subject to the appropriate governance being in place.

This section aims to answer some of the more common MODAF queries and misconceptions.

MODAF and Tools

The MODAF policy on tools is contained within the MODs Enterprise Architecture policy. If, however, the FAQs do not provide the answer you need, please contact us via the e-mail address given.

What is an Enterprise?

An Enterprise is any collection of organizations that has a common set of goals and principles and/or a single bottom line. Therefore an enterprise can be a whole corporation, a division of a corporation, a government organization, a single department, or a network of geographically distant organizations linked together by common objectives

What is the status of MODAR?

When MODAF was originally developed, there was a vision of a single repository for MODAF architectures. The MOD Architecture Repository (MODAR) was the name proposed for this central repository. Over time it has become apparent that this approach is not sustainable and a federated approach is preferable. The US DOD also appear to be moving in this direction with their Defence Architecture Repository System (DARS) repository, where the 'R' is now thought of in many areas as 'Registry'.

The MOD is still determining the best way forward for this federated approach. The US DOD are looking to adopt the federated approach for their DODAF 2 development.

Key MODAF Definitions

Capability

Capabilities in the MODAF sense are specifically not about equipment but are a high level specification of the enterprise's ability. A capability is a classification of some ability – and can be specified regardless of whether the enterprise is currently able to achieve it. For example, one could define a capability "Manned Interplanetary Travel" which no-one can currently acheive, but which may be planned or aspired to. Capabilities in MODAF are not time-dependent – once defined they are persistent. It is only the Capability Requirement that changes.

MODAF allows the architects to develop a formal taxonomy of capabilities which can be re-used across multiple architectures.

MODAF operational architectures refer to capabilities – i.e. they define what capabilities are required for a given scenario or operation. Systems architectures define the personnel, platforms, equipment and processes needed to fulfil capabilities.

Capability Requirement

A time-dependent requirement for a Capability. A capability requirement is a statement that a Capability is required to a certain level (specified by formal metrics and natural language assertions) within a specified time frame. Taking the interplanetary travel capability from before, we could create a Capability Requirement stating that a space agency intend to achieve the capability by 2020, with a required journey time to Mars of 6 months.

Capability Configuration

A combination of organisational aspects (with their competencies) and equipment that combine to provide a capability. A Capability Configuration is a physical asset, organisation or post configured to provide a capability.

Operational Node

An Operational Node is a logical element of the operational Architecture that may produce, consume, or process information, energy, materiel or people. It is possible to think of Nodes as a container for a set of coherent operational activities.

What constitutes an Operational Node can vary among Architectures, including:

- a logical or functional grouping (e.g. Logistics Node, Intelligence Node)
- the headquarters for an organisation (e.g. Command HQ) or an organisation type (e.g. Joint Task Force HQ)
- the base for an operational capability or other facility of importance to the business

Needline

A Needline documents the exchange (required or actual) of information between Nodes. A needline is a conduit for one or more information exchanges – i.e. it represents a logical bundle of information flows. The Needline does not indicate how the information transfer is implemented

What MODAF views are mandated?

Mandation of MODAF views has always been a confusing issue. In a sense, it shows a lack of understanding of the primary purpose of MODAF, which is to present complex relationships in a simple, consistent manner. If this thought process is followed through, the question therefore becomes 'which MODAF views do I, as a presenter, need to produce to show my customer the results of my work'.

The maturity and scope of the architecture will also help determine which views are produced. In the early stages of a small project, it is not reasonable to expect a wide range of views. Even a major project may use only a small set of views to communicate on a particular issue.

This document is no longer extant and has been withdrawn.

Defence |

about defence

Version: 1.2.004

Configuration control

Configuration control policy and version history.

The MODAF documentation is version controlled. The link provides information on major, minor and editorial revisions.

- **20090518-MODAF_1_2_Version_History_V1_0_U.pdf PDF [49.1 KB]**
- **20100426-MODAF1_2_004ChangeLog.pdf PDF [42.0 KB]**

Information Management

in this section:

- ICAD
- MODAF

Related pages

MODAF - Guidance
Detailed Guidance on MODAF

Viewpoints and views
Overview of the differentÂ viewsÂ used in MODAF

Examples and use of MODAF
Including an example of a generic architecting process

MODAF Meta Model (M3)
link to the Meta Model

MODAF - FAQ's
Frequently asked questions

MODAF - Home Page

External links

International Defence Enterprise Architecture Specification (IDEAS) Group

NATO Architcture Framework (NAF)

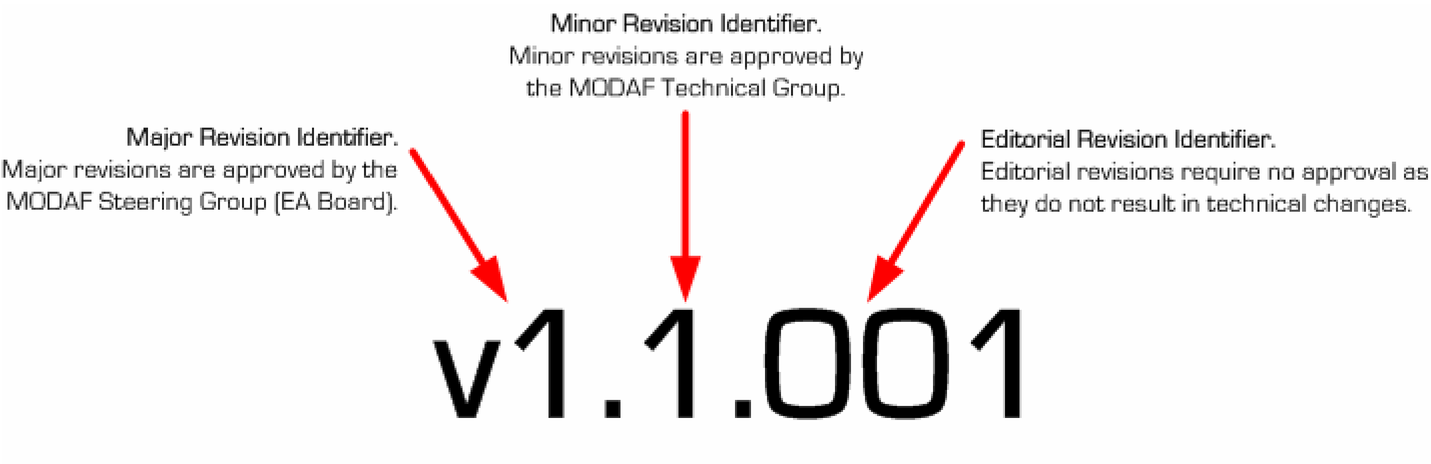
Department of Defence Architecture Framework

Access online conversion tools for Adobe PDF documents

Page not yet rated

MODAF Configuration Control Policy and History

The MODAF documentation is version controlled. The revision identifier is in three parts – major, minor & editorial.



Major Revisions

Major Revisions can only be approved by the MODAF Steering Group, and involve changes to the structure or usage of the MODAF specification. Changes that would typically result in a Major Revision are:

- Addition of Views or Viewpoints
- Removal of Views or Viewpoints
- Renaming of Viewpoints
- Change of meta-model approach – e.g. deciding to no longer use UML/XMI

Minor Revisions

Minor Revisions can be approved by the MODAF Technical Working Group or the MODAF Steering Group. These revisions involve changes to the MODAF documentation and meta-model that are not structural, and do not significantly change the purpose of given views or viewpoints. Changes that would typically result in a Minor Revision are:

- Renaming Views
- Adding or removing allowable architectural elements from a view
- Additions, removals and changes to the Meta-Model that do not significantly alter the purpose of views
- Expanding or narrowing the purpose of a view to provide better coherence across the framework, but without significantly altering the scope of the entire framework
- Changes specifically requested by the user community which have been passed to the MODAF Technical Working Group for scrutiny

Editorial Revisions

Editorial Revisions are changes to the documentation of the framework or Meta-Model that either fix typographic errors, or clarify a statement that was unclear. Changes that would typically result in an Editorial Revision are:

- Changes to the framework text that do not change the scope or semantics of a view or the framework itself
- Addition of new examples of views
- Additional text to provide clarification of an existing point
- Changes to unclear statements
- Fixing typos and grammatical errors

Revision 1.2.003

Minor update of the M3 and corresponding views

M3 changes:

- DataElement added to SV-2b diagram (was missing, no model change)
- InformationElement added to SV-11 diagram (was missing, no model change)
- hasChildren tagged value added to DataElement and InformationElement, to allow decomposition of elements

Revision 1.2.002

Minor update of the M3 and corresponding views

M3 changes:

- Added missing Software on SV-11
- requiredLevel tagged value added on SV-12 to show that when a resource uses a service it requires it to be at a given level
- requiredLevel tagged values added on OV-2 – when a node is required to provide a certain level of service, or requires a service at a certain level
- hasVision tagged values added to StV-1 to show that an enterprise has a vision

Revision 1.2.001

Minor update of the M3 and corresponding views

M3 changes:

- Provide a link between Functions and Service Functions (to match the link between Functions and OperationalActivities).
- Provide a reciprical link to Resources provide Services so Resources can use Services

Corresponding View changes:

- SV-5 Alternate matrix showing Resource to Service Function relationships
- SV-12 shows how a Resource depends on a Service being available in order to function correctly

Revision 1.2.000

Major revision to version 1.2 of MODAF. Note that changes specific to the M3 are documented within the M3 specification. Pages significantly updated to the new version are now labelled 1.2 in their titles.

Highlight changes:

- Alternate names for the viewpoints are suggested more in line with their use in MODAF.
- Introduction of Service-Orientated Views (SOVs)
- Use of services in OV-2, OV-5 and OV-6c
- Known Resources are now allowed in OV-2
- Software added as a Resource Type
- Top level element Artefact added as a Resource Type. Systems, components, physical asset, parts etc are now all stereotypes of Artefact
- Information Exchanges (Bundled as Needlines) are now just one example of a Logical Flow between Nodes (others are Energy Flow, Movement of People and Material Flow)
- Addition of an interactive FAQ section

Revision 1.1.004

- Pages now print without Navigation Links. For the best output use Landscape

Revision 1.1.003

- Revised Navigation look-and-feel

Revision 1.1.002

- Addition of Configuration Control Policy Section
- Fixed ordering of sections in side menu
- Note added to M3 page about delays in releasing M3 v1.1 final
- Changed layout on front page

Revision 1.1.001

- Minor correction to text of OV-2 section, which referred to the previous name for OV-2 (Operational Node Connectivity Description).
- Site history section added (this section)

Change Log for MODAF v1.2.003 to 1.2.004

Effective 01 April 2010

Background

Since the publication of MODAF v1.2 in April 2008 (last updated to 1.2.003 in September 2008), MOD has constrained the changes to MODAF in order to provide MODAF users and architecture software tool vendors with a stable framework to develop against. MOD has, however, continued to receive change proposals from interested parties including:

- MOD's Systems Engineering and Integration Group (SEIG);
- Swedish Armed Forces (SwAF);
- Unified Profile for DoDAF and MODAF (UPDM) Group;
- NATO Architecture Framework (NAF) Management Syndicate (NMS).

A total of 59 change proposals were received, catalogued, rationalised into 16 composite proposals ("Change Proposal A" through to "Change Proposal P"), assessed for impact, and prioritised.

In January 2010 the MOD Sponsor of MODAF decided that the time was right to implement the priority changes that provided minimum impact on the structure of the framework, including the MODAF Meta Model (M3), but provided the stakeholders with most benefit.

The overall impact of selected change proposals was considered minimal so not to merit a major version change to MODAF, hence the change is from v1.2.003 to 1.2.004. The log below describes the changes made against the composite Change Proposal.

In addition, the opportunity was taken to review the MODAF documentation and make cosmetic changes to the text to make it clearer, particularly for those of a non-technical background. These textual changes to the documentation are minor and have not been logged.

View and Documentation Changes

Change Proposal A - Ports on SV-1

Documentation for SV-1 amended appropriately. No new examples provided as ports are optional.

Change affects SV-2 (also note fix to way protocol stacks are managed in M3).

Change Proposal B – Numerous Fixes to Examples

Examples brought in line with M3 and disambiguated wherever an issue was raised against a specific example. Some other examples were also updated to reflect changes to M3 resulting from other issues.

Change Proposal C - Minor fixes and additions to M3

Simplified M3 excerpts updated in documentation. Examples modified to reflect M3 changes where necessary.

Change Proposal D – Numerous Clarifications and Fixes to Documentation

Mostly driven by specific issues, many small changes made to documentation. Documentation was also given a cursory review and any glaring errors or contradictions that became apparent were corrected.

StV-6 documentation and example were updated to reflect true purpose of the view.

Change Proposal G – New View for Service Composition

Given that composition of services indicates a method of implementation, it was deemed appropriate (in consultation with SwAF) to use an SV for this purpose. SV-12b was added, along with a new resource type - <<ServiceImplementation>>

Change Proposal K – Add Service Aspects to MODAF

Based on recommendations from CBM-J6 and SEIG, some rudimentary security modelling features have been added to OV-2. New examples added to documentation also.

Change Proposal M – Add materiel, energy and people flows to OV-3, SV-1, OV-5, SV-4, SV-6, OV-6c, SV-10c

This change has impacted the documentation for several views, but is mostly an M3 change. No new examples have been prepared for this, as it is not a core aspect of any of the views.

Meta-Model Changes

The changes to MODAF 1.2.004 were mostly in the Meta-Model. These changes are listed below:

Change Proposal A - Ports on SV-1

- PortType (SV-2) removed.
- ResourcePort added (SV-1, SV-2).
- SoftwarePort added (SV-2).
- InteractionEnd(ABSTRACT) added to allow ResourceInteractions to also go between ResourcePorts (SV-1).
- SystemPortConnector renamed to ResourcePortConnector and now connects resource ports (SV-2).
- SystemPortConnectorEnd renamed to ResourcePortConnectorEnd.

Change Proposal C - Minor fixes and additions to M3

- OperationalStateDescription now only applicable to Nodes (OV-6).
- NodeContextUsage on OperationStateDescription now displayed as taggedValue relationship instead of attribute (no semantic change) (OV-6).
- NodeContextUsage removed for OperationalNodeLifeline (lifelines can only refer to properties anyway) (OV-6).

- v1.2 was supposed to change StV-4 so that aggregation relationships were used for composition instead of composite class. This never ended up in M3 though.
    - Addition of PartEnd and WholeEnd (properties).
    - CapabilityComposition extends UML::Association.
    - Also affects StV-3.
- tasks relationship is now a redefined ownedBehaviour (was tagged value) in StV-1.
- LogicalFlowItem (ABSTRACT) added to enable info flows to/from KnownResources (OV-2).
- Process added as ABSTRACT supertype of EnduringTask and OperationalActivity.
- EnterpriseStructure and EnterpriseTemporalPart were missing from StV-1 diagram – fixed.
- ProcessOwner (already there in OV-4) added to StV-1.
- Link from ActualOrganisation to EnterprisePhase added (StV-1).
- Added isProject and isOrganisation tagged values in StV-2 to allow architects to show that their enterprise is either a Project or an ActualOrganisation.
- ResourceStateMachine owner removed - state machines only applicable to resources, not functions.
- SystemStructureModel removed (it was not connected to anything).
- subGoals tagged value added to EnterpriseGoal in StV-1 - used to provide a goal structure (parent-child).
- Tagged value between ConfigurationDeployed and CapabilityConfiguration renamed to "configuration" from "fromTime" (StV-5).
- ProjectTypeSpecialisation (extends UML::Generalization) added (AcV-1).
- LastEdited and Architect taggedvalues added to ArchitecturalProduct in AV-1 in anticipation of MODAF/SOSA style guides.
- ArchitectureRealisation (extends UML::Realization) added to trace between LogicalArchitecture and PhysicalArchitecture (AV-1 Architecture Product).
- ElementOfEnvironment (abstract) added to clean up environment model in AV.
- SupportingActivities tagged value added to EnduringTask (StV-6).
- SupportingCapabilities tagged values added to Enduring Task (StV-1/6).
- ResourceInterface added to support SEIG requirement for interface-based connections (SV-1).
- Adding missing <<extends>> between servicesupportsactivity and UML::Dependency (OV-5).
- ProtocolImplementation and ImplementsProtocol removed.
- ImplementedProtocol, ProtocolLayer and ImplementedOn added to show how protocols can be implemented for a particular purpose.
- ProtocolStack removed.
- RunsOn added to show which protocols *can run* on other protocols (cf ImplementedOn).
- Definitions cleaned up in SOV elements.

Change Proposal G - Add Service Composition View

- New View added SV-12b (old SV-12 renamed to SV-12a).
- USedService renamed to RequiredService.
- ProvidesService renamed to ProvidedService, and taggedvalue added so that it matches UsedService.

Change Proposal K - Add Security Aspects to MODAF

- SecurityDomain (subtype of Node) added to OV-2.
- SecurityPolicy (subtype of OperationalConstraint) added to OV-6 and OV-2.
- OperationalConstraint added to OV-2 (this was allowed in MODAF anyway).
- Trustline (extends UML::Dependency) added to OV-2.

Change Proposal M - Add materiel and people flows to OV-3, SV-1, OV-5, SV-4, SV-6, OV-6c, SV-10c

- MaterielFlow, EnergyFlow and MovementOfPeople added to OV-3.
- ResourceCommunicaiton added to SV-1 and OV-4 Typical.
- FunctionCommunication added to SV-4.
- MaterielFunctionFlow added to SV-4.
- PeopleFunctionFlow added to SV-4.
- EnergyFunctionFlow added to SV-4.
- exchangeProperties tagged value now on LogicalFlow - was on InformationExchange) (OV-3).
- MaterielFlow, EnergyFlow and MovementOfPeople added to OV-6c.
- InformationExchangeMessage renamed NodeInteraction (OV-6c).
- carriedInfoElement tagged value renamed to carried (OV-5).
- OperationaActivityMaterielFlow added (OV-5).
- OperationaActivityPeopleFlow added (OV-5) - any jokes about their being only one type of activity that produces people should be addressed to the Swedish Armed Forces, who requested this addition.
- OperationaActivityEnergyFlow added (OV-5).
- Energy (extends Class) added to OV-2,3,5 to provide compatibility with NAF 3.1.
- ResourceMessage (extends UML::Message) added to SV-10c - allows clearer link back to materiel, human, etc. flows.