Date: April 2011

Unified Profile for DoDAF and MODAF (UPDM)

Version 1.0.1

OMG Document Number: formal/2011-04-01
Standard document URL: http://www.omg.org/spec/UPDM/1.0.1
Associated Schema Files*: http://www.omg.org/spec/UPDM/20090501
http://www.omg.org/spec/UPDM/20090501/model-library.xmi
http://www.omg.org/spec/UPDM/20090501/updm-profile.xmi

---

* original file(s): dtc/09-05-11 (model library and XMI)

NOTE: This document supersedes formal/2009-09-01 due to a minor update of the copyright data.

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

LICENSES

PATENTS

GENERAL USE RESTRICTIONS

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page *http://www.omg.org*, under Documents, Report a Bug/Issue (*http://www.omg.org/technology/agreement.htm*).

Table of Contents

List of Figures

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A Specifications Catalog is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

Specifications within the Catalog are organized by the following categories:

OMG Modeling Specifications

- UML
- MOF
- XMI
- CWM
- Profile specifications.

OMG Middleware Specifications

- CORBA/IIOP
- IDL/Language Mappings
- Specialized CORBA specifications
- CORBA Component Model (CCM).

Platform Specific Model and Interface Specifications

- CORBAservices
- CORBAfacilities
- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications.

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
140 Kendrick Street
Building A, Suite 300
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

`Courier - 10 pt. Bold:` Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/technology/agreement.htm*.

Part I - Overview of the UML Profile for DoDAF and MODAF

This part contains the following Clauses:

1.Scope

2.Compliance

3.Normative References

4.Terms and Definitions

5.Symbols and Acronyms

6.Additional Information

1 Scope

The authors believe that this specification for a Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF) will

- enhance the quality, productivity, and effectiveness associated with enterprise and system of systems architecture modeling,
- promote architecture model reuse and maintainability,
- improve tool interoperability and communications between stakeholders, and
- reduce training impacts due to different tool implementations and semantics.

The purpose of this document is to specify a UML 2, and optional SysML, profile to enable practitioners to express DoDAF and MODAF model elements and organize them in a set of specified viewpoints and views that support the specific needs of stakeholders in the US Department of Defense and the United Kingdom Ministry of Defence. The profile defined in this specification has been imported by at least four tool vendors including Artisan, EmbeddedPlus, No Magic, and Visumpoint, and the implementation is actively under development. They all plan to release a commercially available product supporting this version of UPDM within the next year. Currently, partial implementations are being used actively on projects.

UPDM defines a set of UML and optional SysML stereotypes and model elements and associations to satisfy the requirements of the UPDM RFC. This specification documents the language architecture in terms of the parts of UML 2 that are reused and the extensions to UML 2 and SysML. The specification includes the concrete syntax (notation) for the complete language. The reusable portion of the UML 2 and SysML specification are not included directly in the specification but are included by reference. The specification also provides an example of how the language can be used to represent a UPDM architecture.

The scope of UPDM includes the language extensions to enable the extraction of specified and custom views from an integrated architecture description. These views include a system's viewpoint (DoDAF Systems View) along with associated systems implementation standards (DoDAF/MODAF Technical View) within the context of the business or enterprise viewpoint (DoDAF/MODAF Operational View). The DoDAF/MODAF AllViews is also included. In addition, UPDM allows the architecture model to include representation of an enterprise capability and strategic intent (MODAF Strategic Viewpoint) and the process steps associated with the procurement of conformant systems (MODAF Acquisition Viewpoint). Finally, the MODAF Services View is included to model Service Oriented Architectures. UPDM also includes mechanisms for designing ad hoc custom views and more formal extensions of new views of the model.

UPDM will support the capability to:

- Model architectures for a broad range of complex systems, which may include hardware, software, data, personnel, and facility elements.
- Model consistent architectures for system-of-systems down to lower levels of design and implementation.
- Model service oriented architectures support the analysis, specification, design, and verification of complex systems.
- Improve the ability to exchange architecture information among related tools that are UML based and tools that are based on other standards.

The profile provides the modeling of operational capabilities, services, system activities, nodes, system functions, ports, protocols, interfaces, performance, and physical properties and units of measure. In addition, the profile enables the modeling of related architecture concepts such as DoD's doctrine, organization, training material, leadership & education, personnel, and facilities (DOTMLPF) and the equivalent UK Ministry of Defence Lines of Development (DLOD) elements.

UPDM, as illustrated in Figure 1.1, will address DoDAF and MODAF Viewpoints as well as enabling extensions to new architecture perspectives (e.g., Services views, Custom views, Logistics views).

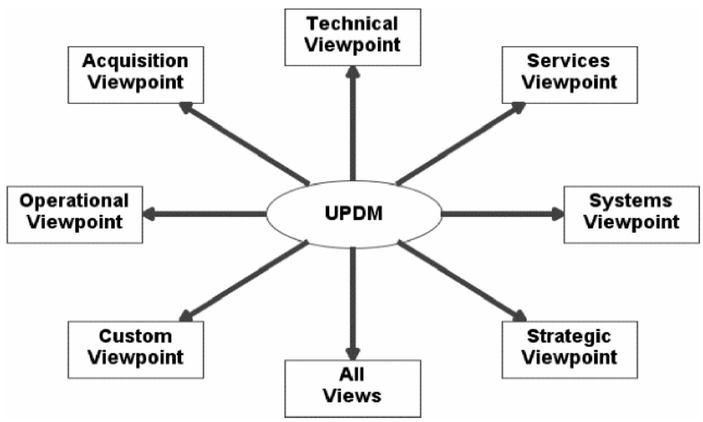


Figure 1.1 - UPDM Viewpoint Support Illustration

2 Compliance

2.1 Compliance Levels

UPDM specifies two compliance levels corresponding to supporting a UML-based profile and a UML+ OMG SysMLTM profile.

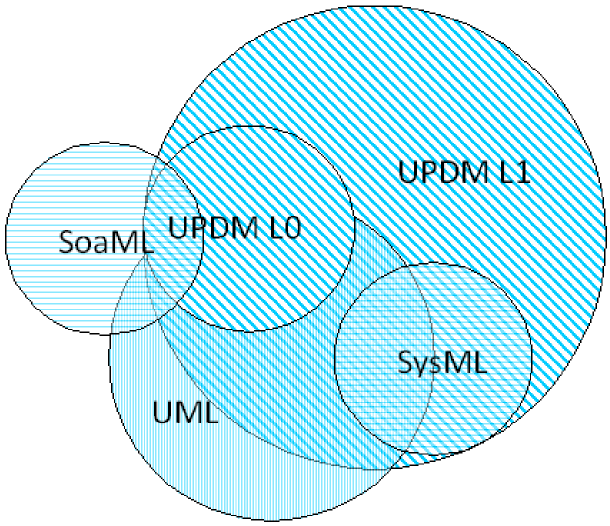


Figure 2.1 - UPDM Compliance Levels 0 and 1

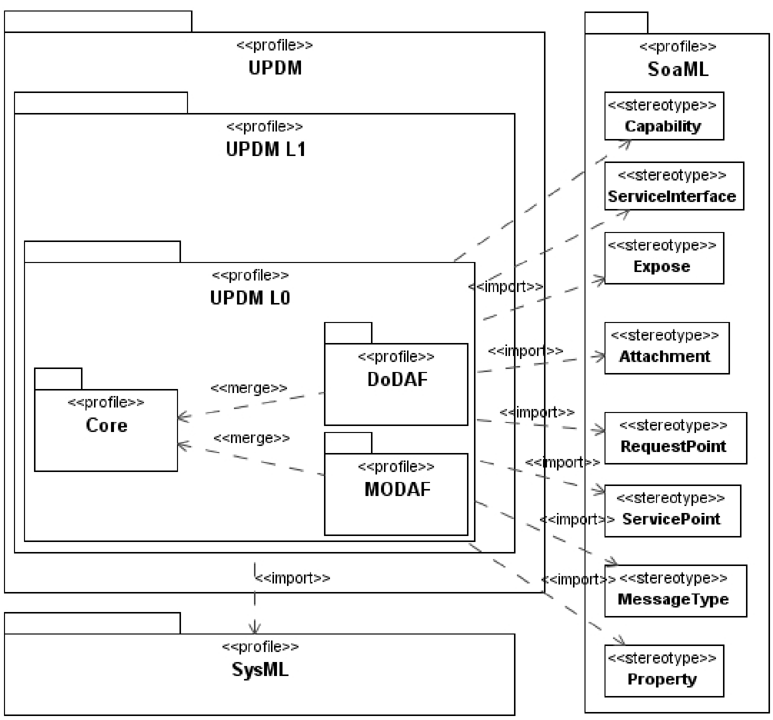


Figure 2.2 - L0 and L1

2.1.1 Level 0 : Based on UML 2 and Partial SoaML Import

Figure 2.2 illustrates that UPDM Compliance Level 0 is an implementation of UPDM extending UML 2 and importing several SoaML stereotypes – namely Expose, Attachment, RequiresPoint, ServicePoint, MessageType, Property. In order for a tool to be considered as compliant with L0, the following must be true:

- All stereotypes, classes, attributes, constraints, associations, and package structures that are scoped to the L0 package (including sub-packages) must exist and be compliant with this specification.
- XMI import and export of the user model and profile must be supported.
- A Level 0 compliant implementation must be able to import and export Level 0 UPDM models with 100% fidelity (i.e., no loss or transforms).
- A Level 0 compliant implementation must be able to import Level 1 UPDM models with only minimal losses.

2.1.2 Level 1 : Based on UML 2 and Full SysML Import

Figure 2.2 illustrates that UPDM Compliance Level 1 includes everything in Level 0, imports the rest of the SysML sub-profiles and defines constraints that pair together the application of SysML and UPDM stereotypes. This provides a UPDM implementation that can be seamlessly taken forward into SysML modeling. For a tool to be considered as compliant with L1, the following must be true:

- All stereotypes, classes, attributes, constraints, associations and package structures that are scoped to the L1 package (including sub-packages) must exist and be compliant with this specification.
- XMI import and export of the user model and profile must be supported.
- A Level 1 compliant implementation must be able to import and export Level 1 UPDM models with 100% fidelity (i.e., no loss or transforms).
- A Level 1 compliant implementation must be able to import Level 0 UPDM models with no loss, and transformations where necessary.

3 Normative References

The following normative documents contain provisions, which through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- Unified Modeling Language: Superstructure version 2.1.2 (http://www.omg.org/docs/formal/07-11 -02.pdf)
- Unified Modeling Language: Infrastructure version 2.1.2 (http://www.omg.org/docs/formal/07-11-04.pdf)
- MOF 2.0/XMI Mapping Specification, v2.1 (http://www.omg.org/cgi-bin/doc?formal/2005-09-01)
- UML 2.0 OCL Specification (http://www.omg.org/docs/ptc/03-10-14.pdf)
- OMG Systems Modeling language (OMG SysML) V1 .0 (http://www.omg.org/docs/formal/07-09-01.pdf)
- DoD Architecture Framework Version 1.5, Volume I: Definitions and Guidelines, 23 April 2007, DoD Architecture
- Framework Working Group (http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf)
- DoD Architecture Framework Version 1.5, Volume II: Product Descriptions, 23 April 2007, DoD Architecture

Framework Working Group (http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_II.pdf)

- DoD Architecture Framework Version 1.5, Volume III: Product Descriptions, 23 April 2007, DoD Architecture Framework Working Group (http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_III.pdf)
- MODAF The MOD Architectural Framework Version 1.2.002 August 2008 (http://www.modaf.org.uk http://www.modaf.org.uk/M3)

4 Terms and Definitions

No new terms and definitions have been required to create this specification. All terms should be available in the normative references or bibliographic citations for detailed explanation.

5 Symbols and Acronyms

AcV-*	Acquisition View
AV-*	All View
BPMN	Business Process Modeling Notation
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
COI	Communities of Interest
DoD	United States Department of Defense
DoDAF	Department of Defense Architecture Framework
DOTMLP	Doctrine, Organization, Training, Material, Leadership, Personnel, Facilities
EIE	Enterprise Information Environment
GIG	Global Information Grid
IDEF	Integrated DEFinition Methods
JCIDS	Joint Capabilities Integration and Development System
JETL	Joint Essential Task List
MOD	United Kingdom Ministry of Defence
MODAF	Ministry of Defence Architecture Framework
NEC	Network Enabled Capability
NCW	NetCentric Warfare
NCAT	NetCentric Assessment Tool
NCOIC	NetCentric Operations Industry Consortium
OV-*	Operational View
POC	Proof of Concept

SoS	System of Systems
SOV	Service Oriented View
StV-*	Strategic View
SV-*	System & Services View
TPPU	Task, Post, Process, and Use
TV-*	Technical View
UPDM	Unified Profile for DoDAF and MODAF

6 Additional Information

6.1 Additional Materials

Accompanying this specification are XMI files and requirements documents, as listed below.

Title	OMG Document Number	Supercedes
UPDM Profile RFC List of Requirements	c4i/2008-08-01	N/A
UPDM Profile Submission	c4i/2008-08-13	c4i/2008-08-02
UPDM Profile Submission - ERRATA	c4i/2008-08-12	N/A
UPDM Beta 2 specification without change notes	dtc/2009-05-08	N/A
Inventory List	dtc/2009-05-09	N/A
Final Report	dtc/2009-05-10	N/A
UPDM XMI Document for UML	dtc/2009-05-11	c4i/2008-08-03
UPDM Requirements Traceability Document	c4i/2008-09-01	c4i/2008-08-05
UPDM Requirements Traceability Document - ERRATA	c4i/2008-09-02	N/A

6.1.1 Statement of support from the DoD representative

From: Leonard F. Levine DoD/DISA/GE33 1/703-681-2614 Subject: Summary of US DoD Support for UPDM

DoD support for the Unified Profile for DoDAF and MODAF (UPDM) has been strong since 2005 and has not wavered. The DoD promotes the use of international, national, and industry-wide open standards to the extent feasible. It looks forward to a rapid adoption of the UPDM by the Object Management Group (OMG) as an industry standard for architecture tools and to its submission as an international standard.

In the United States, support for the UPDM development comes primarily from the DoD Chief Information Officer (DoD CIO -- formally, The Office of the Assistant Secretary of Defense (Networks and Information Integration) (OASD(NII)), specifically the Directorate of Architecture & Interoperability. Mr. Brian Wilczynski addressed this subject at the UPDM working group in Orlando on 17 April 2008 during the annual DoD Enterprise Architecture Conference, and Mr. Walt

Okon similarly apprized the DoD IT Standards Committee (ITSC) on 18 June 2008. Both reconfirmed the DoD CIO's support for the current UPDM development process generally and, in particular, the goal of submitting domain model, profile, and related documentation for UML as required underpinning for a RFC in time for the September 2008 quarterly meeting of the OMG. To bolster this commitment, the DoD CIO has requested that a representative of the DoD Executive Agent for IT Standards work with the UPDM group to assure that it will generate a product conforming to our current DoD Architecture Framework (DoDAF Version 1.5) and coordinating with our continued development of the DoDAF. Also, the DoD CIO has generously made available the time of the chief architect of the developmental version of the architectural framework (DoDAF Version 2.0). The architects on the UPDM working group and the DoDAF have met frequently by electronic collaboration and recently face-to-face in both Orlando and Ottawa to exchange detailed modeling concepts and to promote convergence. The DoD CIO anticipates that the UPDM working group will continue to refine the profile after the September 2008 submission and, as required, will revise the resulting profile during the next calendar year.

The DoD CIO looks forward to the emergence of tools from vendors supporting DoDAF through the standardized UPDM profile including architecture exchanges through the XML Metadata Interchange (XMI), and to UPDM extension to the systems engineering discipline. The mapping of UPDM to the Unified Profile for SysML has also received support of the Office of the Director, Systems and Software Engineering (S SE), Office of the Deputy Under Secretary of Defense (Acquisition and Technology)(DUSD(A&T)). The DoD IT Standards Registry (DISR) currently "mandates" UML 2.0 and XMI 1.1 for system acquisition, and a request will be submitted this summer recommending the profile as an "emerging" standard as soon as a stable URL is available. In the normal lifecycle of the DISR, a standard such as UPDM must be formally adopted by a recognized body such as OMG before being advanced to "mandated."

Leonard F. Levine
Standards Development Branch (GE33 1)?IT Standards Division
Defense Information Systems Agency?PO Box 4502
Arlington, VA 22204-0502

6.1.2 Statement of support from the MOD representative

Matthew and all,

I am happy to confirm Matthew's statement that UK MOD fully support the work of this UPDM task force.

We appreciate the amount of the effort that the team are putting into this and, notwithstanding Adrian's, Ian's and Fariba's contribution to date, my only regret is that we are unable to allocate more resources to help you.

Kind regards Patrick

Patrick Gorman
Framework Development Manager
Information Exploitation Enterprise Architecture Team?Ministry of Defence
Main Building, G.B.32
Whitehall
LONDON, SW1A 2HB

The people referred to above are the following:

Dr Adrian Pearson
IA8b, Architecture Framework Technical Authority
Systems Engineering and Integration Support Group, MOD

Ms. Fariba Hozhabrafkan
SERCO Consulting, MOD Consultant

Dr Ian Bailey
Model Futures Limited, MOD Consultant

6.2 Overview of this Specification

6.2.1 Intended Audience

This specification will be of interest to end users who expect to use this profile, and to tool vendors interested in developing tool support for the development of enterprise and system of systems architectures, and that can satisfy contract documentation requirements for DoD and MOD customers. Tool vendors will also be able to use this specification to support Model Driven Development of systems based on the architectural descriptions based on this profile. Developers and reviewers of the views will have a clearer understanding of the semantics behind specific views and viewpoints, which will support more precise evaluation and comparison.

6.2.2 Organization of this Specification

DoDAF and MODAF are specifically organized around a set of viewpoints and views that address the concerns of a well defined set of stakeholders. This specification organizes the presentation of the UPDM abstract and concrete syntax around those viewpoints, so that the discussion is well-connected to their domain expertise. (See Section 1.5 for a more detailed description.)

The rest of this document contains the technical content of this specification. As background for this specification, readers may wish to review the UML and OMG SysMLTM specifications that complement this specification. Although the chapters are organized in a logical manner and can be read sequentially, this is a reference specification that can be read in a non-sequential manner.

Part I of the specification describes the details of the specification.

Part II provides the technical details essential to understanding the specification:

The specification of the Profile language. The profile includes both a Compliance Level 0 that extends UML and a Compliance Level 1 that extends UML and OMG SysMLTM. The elements of the profile are organized by the specific viewpoints required by DoDAF and MODAF. Within each of the viewpoint-specific sections (e.g., Operational Views (OVs)), the elements are presented in alphabetical order.

Annex A presents a non-normative view of various diagrams that document the Domain Metamodel (DMM) that document the MoDAF 1.5 and MoDAF 1.2 integrated model. This model was used as a basis for creating the UPDM profile.

Annex B presents the traceability among UPDM stereotypes and DODAF/MODAF elements. Please note that not all DoDAF/MODAF elements have corresponding UPDM stereotype. Those DoDAF/MODAF elements are modeled by UML artifacts directly, which shows in the Metaclass column.

Annex C Sample Problem illustrating UPDM concepts.

Annex D contains the bibliography providing a listing of additional consulted artifacts.

6.3 Acknowledgements

The following individuals submitted parts of this specification and/or have assisted the UPDM team in the development of the specification:

• Adaptive Inc	Pete Rivett
• Advanced System Management Group	Michael Abrahamson
• Artisan Software Tools Ltd.	Phil Astle
• Artisan Software Tools Ltd.	Matthew Hause
• BAE Systems	J. D. Baker
• BAE Systems	David C. Putman
• Decisive Analytics Corp	Charles Johnson
• DOD	Dwayne Hardy
• DOD	Chuck Johnson
• DOD	Leonard Levine
• EmbeddedPlus Engineering	Kumar Marimuthu
• EmbeddedPlus Engineering	Paula Obeid
• Generic AB	Lars-Olof Kihlstrom
• International Business Machines	Graham Bleakley
• Lockheed Martin	Sanford Friedenthal
• Malina Software	Bran Selic
• Mitre	Fatma Dandashi
• MOD	Patrick Gorman
• MOD	John Keefe
• MOD	Adrian Pearson
• Model Futures	Ian Bailey
• No Magic, Inc.	Daniel Brookshier
• No Magic, Inc.	Jim Rice
• No Magic, Inc.	Andrius Strazdauskas
• Raytheon	Wally Lee, PhD
• Raytheon	Ron Williamson
• Rolls Royce	Francis Thom
• SERCO	Fariba Hozhabrafkan
• Silver Bullet	Dave McDaniel
• Sparx Systems	Sam Mancarella
• Visumpoint	Robert Lario
• Visumpoint	Ginna Yost

The team would like to express their thanks to all of the above individuals and many others who are not listed.

Once again, it is important to stress that UPDM is not a new framework. Instead, UPDM is a specification for modeling DoDAF and MODAF architectures using UML and SysML. As such, it could not have been produced without taking concepts, structures and descriptions, etc, from the DoDAF and MODAF documentation and specifications, particularly the M3. The main authors of the M3 were:

- V1.0 – Dave Mawby (PA Consulting), Paul King (Vega/ Detica), and Ian Bailey.
- V1.1 – Adrian Pearson (MOD), Paul King, and Ian Bailey.
- V1.2 – Adrian Pearson (MOD), Patrick Gorman (MOD), and Ian Bailey.

The authors of this UPDM specification are therefore greatly indebted to organizations and authors who have contributed to all the DoDAF and MODAF specifications over the years. Some of these are listed above. To list all of them would not be possible.

Part II - Language Architecture

This part contains the following Clause and subclauses:

7. Language Architecture

7.1 Introduction
7.2 Philosophy
7.3 Core Principles
7.4 Profile structure
7.5 Representing stereotype constraints
7.6 Important areas of the architecture

7 Language Architecture

7.1 Introduction

The UPDM specification reuses a subset of UML 2 and provides additional extensions needed to address requirements in the UPDM RFC Mandatory Requirements. We have used those requirements as the basis for this specification. This specification documents the language architecture in terms of the parts of UML 2 that are reused and the extensions to UML 2, as well as defining how to implement UPDM in SysML. This chapter explains design principles and how they are applied to define the UPDM language architecture.

7.2 Philosophy

The UPDM was developed using a model-driven approach. A simple description of the work process is:

- The Domain Metamodel (DMM) was created using UML Class models to represent the concepts in DoDAF and MODAF. Concepts common to both DoDAF and MODAF were captured in a Core package.
- The DMM concepts were mapped to corresponding stereotypes in the Profile.
- The Profile was analyzed and refactored to reflect language architecture, tool implementation, and reuse considerations.
- The conformance levels were finalized including mapping to SysML.
- The Profile diagrams, stereotype descriptions, and documentation were added.
- The specification was generated from the profile model.

This approach allowed the team to concentrate on architecture issues rather than documentation production. Consistency was automatically maintained by the UML tool.

The UML tool also enabled traceability to be maintained between the profile and the DMM where every stereotype is linked to the DMM element using UML Abstraction relationship.

7.3 Core Principles

The fundamental design principles for UPDM are:

- **Requirements-driven**: UPDM is intended to satisfy the requirements of the UPDM RFC Mandatory Requirements.
- **Domain meta model (DMM) driven**: The DMM was created first by domain experts and it served as a foundation for profile development.
- **Reuse of existing specifications**: UPDM reuses UML/SysML wherever practical to satisfy the requirements of the UPDM RFC and leverage features from both UML and SysML to provide a robust modeling capability. Consequently, UPDM is intended to be relatively easy to implement for vendors who support UML 2. The UPDM team intended to reuse UPMS. However, since UPMS had not been formally adopted at the time of this specification, a separate service profile in UPDM was developed that used similar concepts, with the intent to replace it with UPMS in the future.
- **Partitioning**: The package is the basic unit of partitioning in this specification. The packages partition the model elements into logical groupings that minimize circular dependencies among them.

- **Compliance levels**: UPDM includes two compliance levels. L0 is a UML only profile and L1 extends L0 to enable seamless integration with SysML modeling and to leverage the features of SysML in UPDM modeling.
- **Interoperability**: UPDM inherits the XMI interchange capability from UML.

7.4 Profile Structure

7.4.1 Top Level

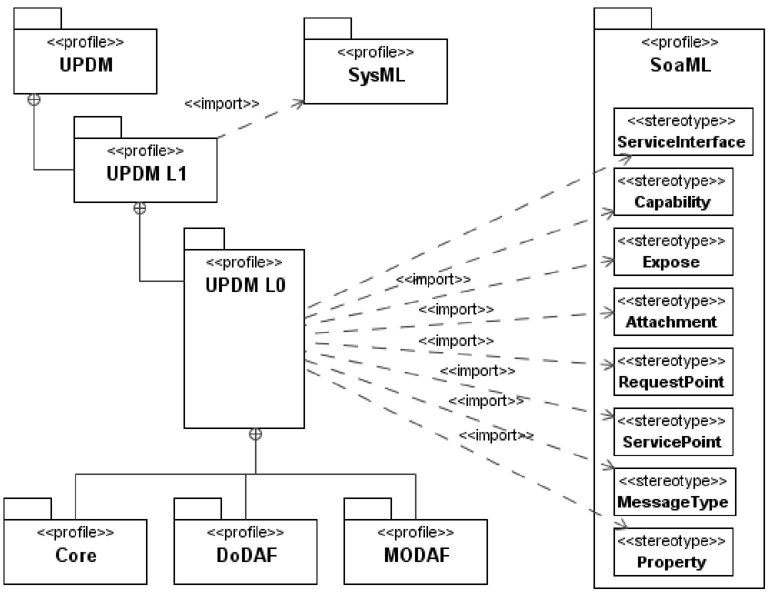


Figure 7.1 - Top Level Profile Structure

All the core elements for UPDM are in the UPDM L0 profile. The UPDM L0 profile has 3 top level profiles:

- **Core -** Elements shared by DoDAF and MODAF
- **DoDAF** - DoDAF specific elements
- **MODAF** - MODAF specific elements

7.4.2 Middle Level

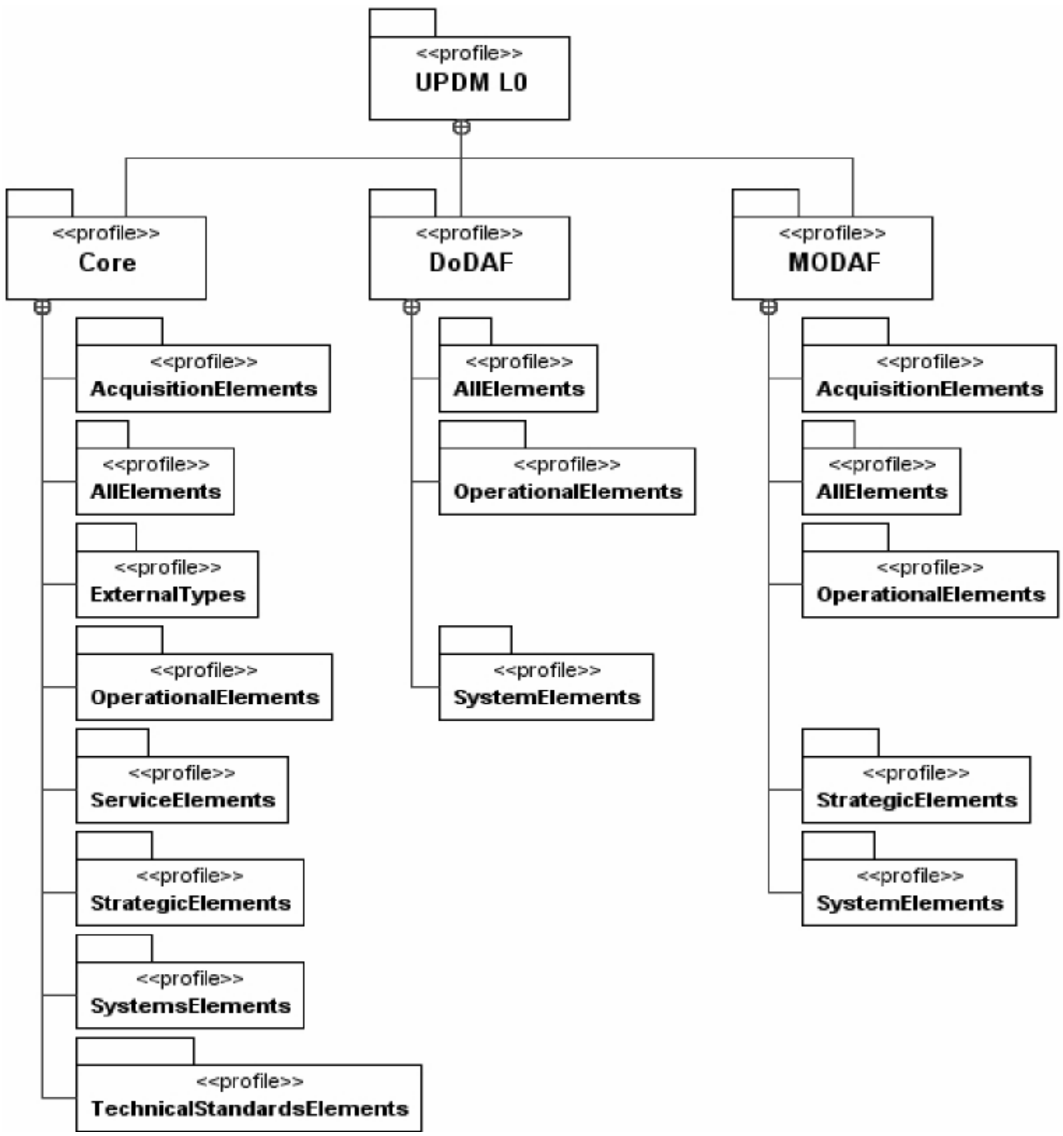


Figure 7.2 - Middle Level Profile Structure

Every top level profile may have the following subprofiles:

- **AllElements** - Cross-cutting elements.
- **AcquisitionElements** - Elements relating to Acquisitions.
- **ExternalTypes** - External types.
- **OperationalElements** - Elements relating to Operational models.
- **ServiceElements** - Elements relating to Service models.
- **StrategicElements** - Elements relating to Strategic models.

- **SystemsElements** - Elements relating to Systems models.
- **TechnicalStandardsElements** - Elements relating to Technical Standards models.

7.4.3 Low Level

Each of these subprofiles may be further decomposed into low-level profiles:

- **Behavior -** Stereotypes for modeling behavior
- **Data -** Stereotypes for modeling data
- **Environment -** Stereotypes for modeling environment
- **Flows -** Stereotypes for modeling flows
- **Measurements -** Stereotypes for modeling measurements
- **Milestone -** Stereotypes for modeling milestones
- **Structure -** Stereotypes for modeling structure
- **Views -** Stereotypes for modeling views

7.5 Representing Stereotype Constraints

The profile uses a non-standard notation to represent stereotype constraints in the profile to improve readability of the profile.

«metaconstraint» dependency

«metaconstraint» is a stereotype that extends the Dependency metaclass. It is used to specify constrained elements within the profile.

A sample of the «metaconstraint» dependency is a diagram for stereotype extending the Dependency metaclass. See the following example:

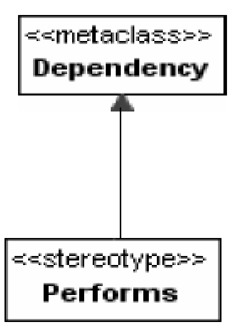


Figure 7.3 - Performs Stereotype

Performs is a stereotype that extends Dependency. The constraint on this stereotype is that its client end must be stereotyped by a Performer and its supplier end must be stereotyped by Activity. But as this constraint is not visible; therefore, the diagram does not communicate the needed information. We are using the “metaconstraint” dependency to visualize the constraint.

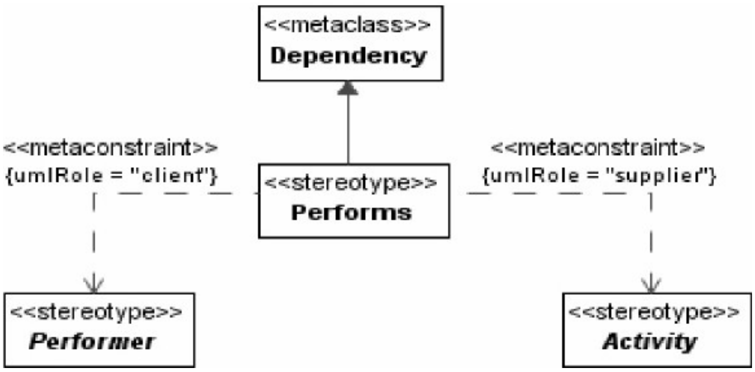


Figure 7.4 - Performs Hierarchy

This diagram should be read as follows:

> Performs is a stereotype extending the Dependency metaclass and is used for modeling a relationship between a Performer (or its specializations) and an Activity (or its specializations). A Dependency stereotyped Performs must have its values for the client property stereotyped as Performer, and its values for the supplier property must be stereotyped Activity.

The «metaconstraint» dependency will appear only in the specification diagrams, but not the profile XMI.

Note – When stereotype extends Connector, the stereotype property umlRole has values "end[0].role" and "end[1].role."

For example:

This is done because Connector has no direct "linkage" to the connected element; it links to the Connector Ends, which references the linked element. So, end[n] gives the reference to the ConnectorEnd, and role gives the reference to the linked element.

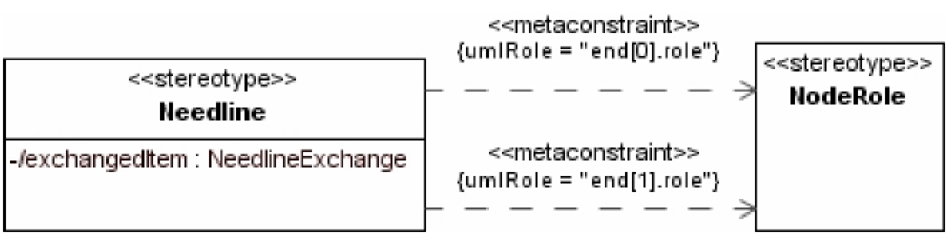


Figure 7.5 - Connector Extension

"metarelationship" dependency

"metarelationship" is a stereotype for dependency, showing that certain domain concepts will be implemented using regular UML relationships.

For example: A Capability may depend on other Capabilities, but this concept cannot be visualized on the diagram:

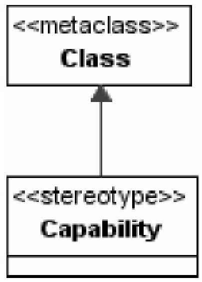


Figure 7.6 - Capabilities Generalization

We are using the “metarelationship” dependency to visualize the dependency concept.

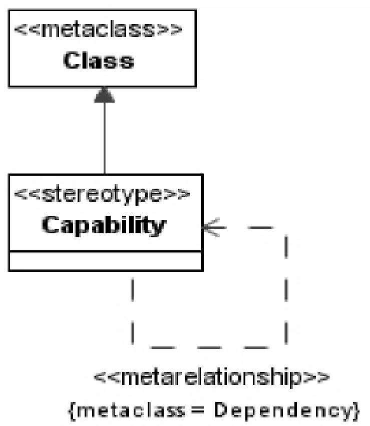


Figure 7.7 - Visualizing «metarelationship»

This diagram should be read as follows:

> Capability may have other Capabilities related to it, using the UML Dependency metaclass.

The “metarelationship” dependency will appear only in the specification diagrams, but not the profile XMI.

"stereotyped relationship" dependency

The “metaconstraint” dependency creates a good way to show the constrained ends of the stereotyped relationship, however, it also creates some overhead when showing the relationship between two stereotypes.

For example, the diagram needs to show that Node may require a Capability.

Figure 7.8 - UML Dependency metaclass

The “stereotyped relationship” dependency is used as follows:

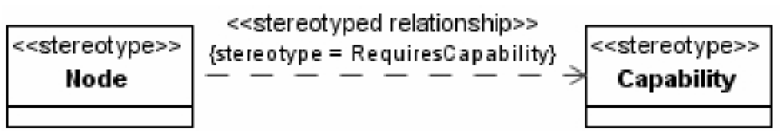


Figure 7.9 - “stereotyped relationship” dependency

The “stereotyped relationship” dependency will appear only in the specification diagrams, but not the profile XMI.

7.6 UML Constraint Representation

The specification uses the Object Constraint Language (OCL), as defined in Clause 6, “Object Constraint Language Specification” of the UML specification, for expressing well-formedness rules. The following conventions are used to promote readability:

- Self - which can be omitted as a reference to the metaclass defining the context of the invariant, has been kept for clarity. UML Infrastructure Specification, v2. 1.2 25
- In expressions where a collection is iterated, an iterator is used for clarity, even when formally unnecessary. The type of the iterator is usually omitted, but included when it adds to understanding.
- The ‘collect’ operation is left implicit where this is practical.
- The context part of an OCL constraint is not included explicitly, as it is well defined in the sub clause where the constraint appears.

The OCL constraints are stored with the profile and can be interchanged via XMI standard. Below is the pattern to represent constraint for stereotyped relationship in OCL as per UML 2.1:

- To constraint the client of the stereotyped relationship that should be a particular stereotyped element:
  self.client->forAll(getAppliedStereotype(CLIENT_STEREOTYPE)-> notEmpty()
- To constraint the supplier of the stereotyped relationship that should be a particular stereotyped element:
  self.supplier->forAll(getAppliedStereotype(SUPPLIER_STEREOTYPE)-> notEmpty()
- The constraint represented in Figure 7.7 can be represented in OCL as follows:
  self.client->forAll(getAppliedStereotype('UPDM: :AllElements: :Behavior : :Performer')-> notEmpty()
  self. supplier->forAll(getAppliedStereotype("UPDM: :AllElements: :Behavior : :Activity')->notEmpty())

7.7 Important Areas of the Architecture

7.7.1 Aliases

Although there are similar concepts in DoDAF and MODAF, they are not named the same. To keep interoperability and to fit the needs of both audiences, the UPDM specification used generalizations as a way to alias concepts.

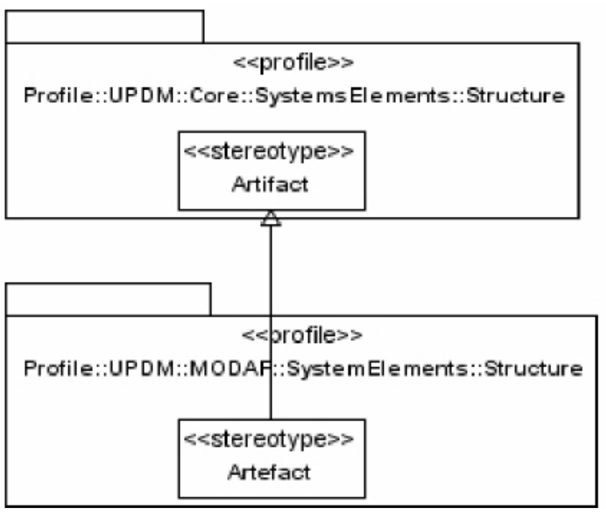


Figure 7.10 - Aliases

7.7.2 DoDAF 1.5 Conformance

Compliance with UPDM Profile including metadata should assist the tool vendor in adhering to DoDAF 1.5 because the UPDM Core and DoDAF-specific metadata models in UPDM adhere to the metadata model inherent in DoDAF 1.5 Conceptual and Logical data models. In developing the UPDM, domain meta-modelers have also consulted the corresponding Physical data model in DoDAF 1.5 and the CADM [Core Architecture Data Model] to resolve questions of general conformance with enterprise-level architectural elements. Nevertheless, tool vendors are advised to consult DoDAF Version 1.5 (especially Volume I, page 2-6; Volume II, page 2-6; and Volume III, page 1-2) before claiming DoDAF 1.5 compliance. While conformance with UPDM Core and DoDAF-specifics should greatly facilitate conformance with DoDAF 1.5, each tool vendor is still responsible for the tool's ultimate conformance with the documented architecture framework.

7.7.3 SoaML Reuse in L0

SoaML is quickly becoming the standard modeling choice for capturing and creating service oriented architectures. By importing the SoaML stereotypes, a UPDM model gains access to these powerful features. They can be used and viewed in a UPDM model using the standard SoaML approach and as such have not been further documented.

7.7.4 SysML Reuse in L1

Defining an architectural framework in UPDM provides the highest level abstraction of what will one day become integrated pieces of hardware and software. Being able to trace from the architectural framework to the various levels of implementation is critical for ensuring the initial goals have been reached. By including the full SysML profile inside UPDM, a modeler can have all of the architectural, system, and software design in the same place. This provides huge benefits in analysis, cross abstraction level communication, traceability, and reuse. As in L0, all of the stereotypes contained in SysML can be used and displayed using standard SysML approaches while still being able to be connected to UPDM elements such as Nodes and Artifacts.

Part III - UPDM Profile

This part contains the following Clause and sub clauses:

8 UPDM Profile

UPDM L1 contains UPDM L0 and imports the entire SysML profile. This compliance level contains a set of constraints that specify which SysML stereotypes are applied to the L0 elements. The use of this compliance level is intended to provide more seamless integration with system modeling using SysML and to be able to fully leverage the capabilities of SysML in UPDM.

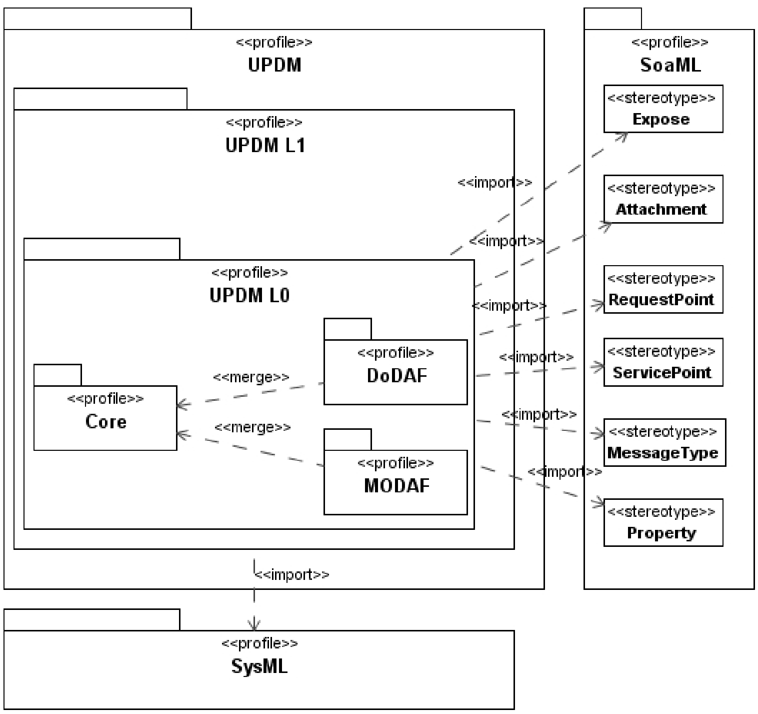


Figure 8.1 - UPDM Profile

8.1 UPDM L1

These constraints require that if element is stereotyped with particular UPDM stereotyped, appropriate SysML stereotype must be applied as well. The constraints are written in OCL. The name of the constraint is followed by the constraint specification.

Artefact

context Class inv:
UPDM: :Artefact: :allInstances()->exists(n|n.base _Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

ResourceArtifact

context Class inv:
UPDM: :ArtifactResourceArtifact::allInstances()->exists(n|n.base _Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

Capability

context Class inv:
UPDM: :Capability::allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

CapabilityConfiguration

context Class inv:
UPDM: :CapabilityConfiguration::allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

Climate

context DataType inv:
UPDM: :Climate::allInstances()->exists(n|n.base_Class=self) implies SysML: :ValueType::allInstances()->exists(b| b.base_Class = self)

Commands

context InformationFlow inv:
UPDM: :Commands: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ItemFlow: :allInstances()->exists(b| b.base_Class = self)

Controls

context InformationFlow inv:
UPDM: :Controls::allInstances()->exists(n|n.base _Class=self) implies SysML: :ItemFlow: :allInstances()->exists(b| b.base_Class = self)

DataExchange

context InformationFlow inv:
UPDM: :DataExchange: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ItemFlow: :allInstances()->exists(b|
b.base_Class = self)

Energy

context Class inv:
UPDM: :Energy::allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

EnergyExchange

context InformationFlow inv:
UPDM: :EnergyExchange: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ItemFlow: :allInstances()->exists(b| b.base_Class = self)

EnterpriseGoal

context Class inv:
UPDM: :EnterpriseGoal: :allInstances()->exists(n|n.base_Class=self) implies
SysML: :Requirement: :allInstances()->exists(b| b.base_Class = self)

Entity

context Class inv:
UPDM: :Entity: :allInstances()->exists(n|n.base _Class=self) implies SysML: :ValueType: :allInstances()->exists(b|
b.base_Class = self)

Environment

context Class inv:
UPDM: :Environment: :allInstances()->exists(n|n.base _Class=self) implies SysML: :ValueType: :allInstances()-
>exists(b|
b.base_Class = self)

ExternalType

context Class inv:
UPDM: :ExternalType: :allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b|
b.base_Class = self)

HighLevelOperationalConcept

context Class inv:
UPDM: :HighLevelOperationalConcept::allInstances()->exists(n|n.base_Class=self) implies SysML: :Block:
:allInstances()->exists(b| b.base_Class = self)

Information Exchange

context InformationFlow inv:
UPDM: :InformationExchange: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ItemFlow::allInstances()-
>exists(b| b.base_Class = self)

LightCondition

context DataType inv:
UPDM::LightCondition: :allInstances()->exists(n|n.base _Class=self) implies SysML: :ValueType::allInstances()-
>exists(b| b.base_Class = self)

Location

context DataType inv:
UPDM: :Location: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ValueType::allInstances()->exists(b|
b.base_Class = self)

MaterielExchange

context InformationFlow inv:
UPDM: :MaterielExchange: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ItemFlow::allInstances()-
>exists(b| b.base_Class = self)

MeasurementSet

context DataType inv:
UPDM: :MeasurementSet: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ValueType::allInstances()-
>exists(b| b.base_Class = self)

Node

context Class inv:
UPDM: :Node::allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b|
b.base_Class = self)

NodePort

context Port inv:
UPDM: :NodePort: :allInstances()->exists(n|n.base_Class=self) implies SysML: :FlowPort::allInstances()->exists(b| b.base_Class = self)

OperationalNode

context Class inv:
UPDM: :OperationalNode: :allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

OrganizationalExchange

context InformationFlow inv:
UPDM: :OrganizationalExchange: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ItemFlow: :allInstances()->exists(b| b.base_Class = self)

PhysicalLocation

context DataType inv:
UPDM: :PhysicalLocation: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ValueType::allInstances()->exists(b| b.base_Class = self)

ResourceInteraction

context InformationFlow inv:
UPDM: :ResourceInteraction: :allInstances()->exists(n|n.base_Class=self) implies SysML: :ItemFlow: :allInstances()->exists(b| b.base_Class = self)

ResourcePort

context Port inv:
UPDM: :ResourcePort: :allInstances()->exists(n|n.base_Class=self) implies SysML: :FlowPort: :allInstances()->exists(b| b.base_Class = self)

Software

context Class inv:
UPDM: :Software: :allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

System

context Class inv:
UPDM: :System: :allInstances()->exists(n|n.base_Class=self) implies SysML: :Block: :allInstances()->exists(b| b.base_Class = self)

8.1.1 UPDM L1::UPDM L0

UPDM L0 contains all the Core, DoDAF and MODAF elements, and imports parts of SysML - Requirements and ModelElements namely. This compliance level is primarily based on UML and reuse of a minimum of SysML elements. This includes Requirements and Views/Viewpoints. As one of the core principles is reuse, cloning/recreating of these existing SysML structures was considered as inappropriate.

8.1.1.1 UPDM L1::UPDM L0::Core

The Core contains most of the elements of UPDM profile. These elements are common to both DoDAF and MoDAF or are critical to a complete model of core concepts. The Core is always associated with either the DoDAF or MoDAF profiles.

8.1.1.1.1 UPDM L1 ::UPDM L0::Core::AcquisitionElements

The AcquisitionElements describe project details, including dependencies between projects and capability integration. These Views guide the acquisition and fielding processes.

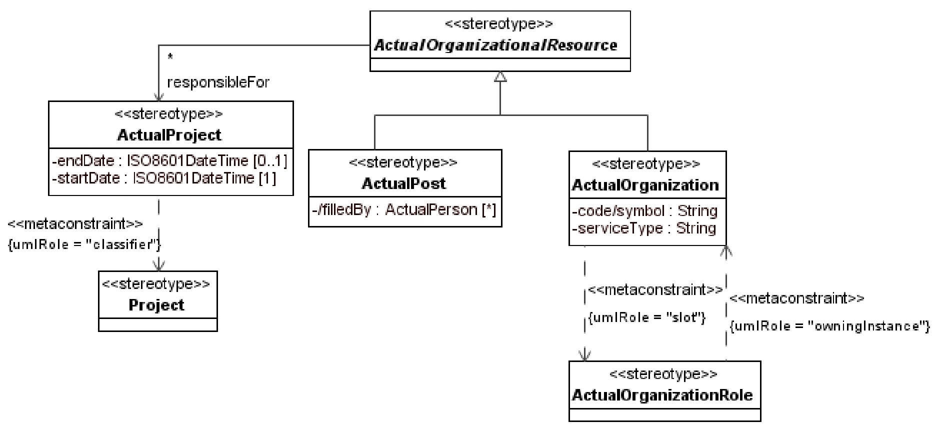


Figure 8.2 - Elements related to the AcV-1 Product

AcV-1 view products represent an organizational perspective on programs.

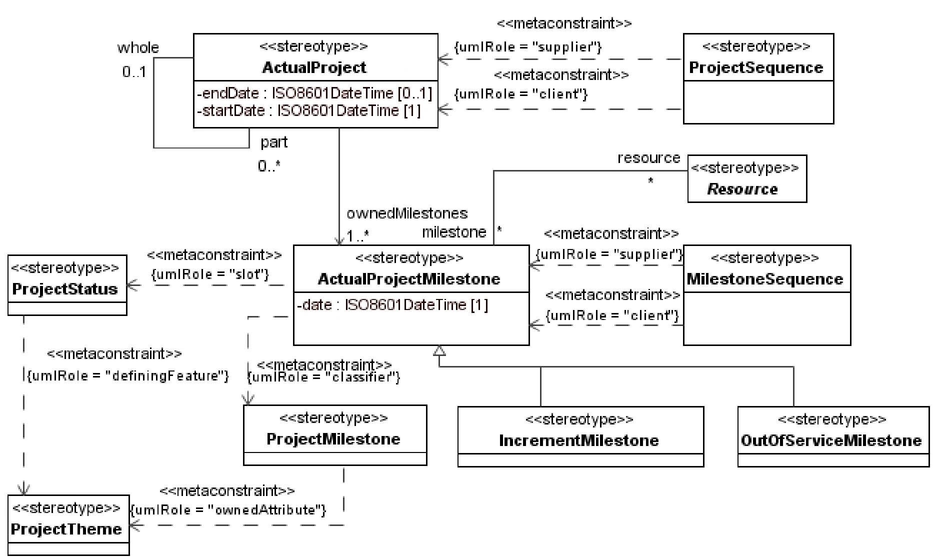


Figure 8.3 - Elements related to the AcV-2 Product

AcV-2 view products provide a timeline perspective on projects and their relationship to CapabilityConfigurations.

8.1.1.1.1.1 UPDM L1 ::UPDM L0::Core::AcquisitionElements::Milestone

Milestone elements from the acquisition section of the profile.

8.1.1.1.1.1.1 ActualProjectMilestone

MODAF: (ProjectMilestone): An event in an ActualProject (MODAF::Project) by which progress is measured. Note: In the case of an acquisition project, there are two key types of milestones that shall be represented using subtypes - IncrementMilestone (MODAF::CapabilityIncrement) and OutOfServiceMilestone (MODAF::OutOfService).

DoDAF: N/A.

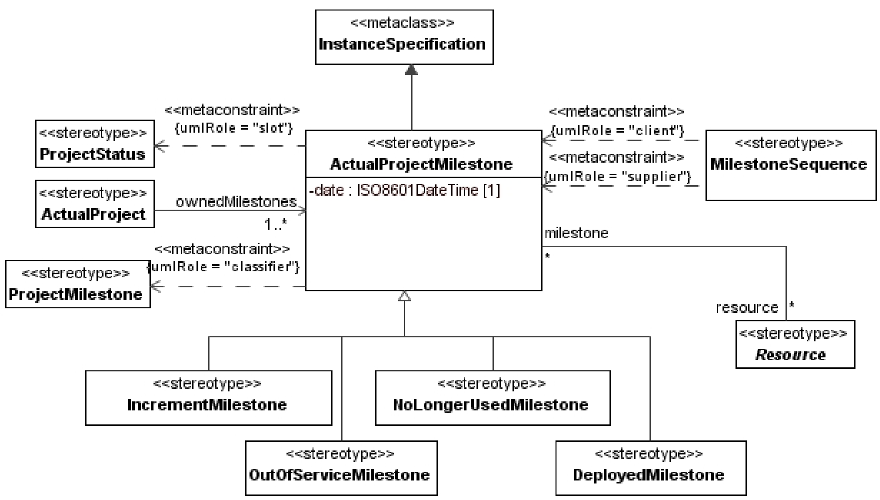


Figure 8.4 - Elements related to the ActualProjectMilestone stereotype

Constraints

The following are constraints for ActualProjectMilestone:

- ActualProjectMilestone.classifier - Value for the classifier property must be stereotyped «ProjectMilestoneType» or its specializations.
- ActualProjectMilestone. slot - Slot values have to be stereotyped «ProjectStatus» or its specializations.

Attributes

The following are attributes of ActualProjectMilestone:

- endTime: ISO8601DateTime[0..1] - End time for this Project.
- startTime: ISO8601DateTime[1] - Start time for this Project.
- Resource: Resource[*] - Affected resource.

Extensions

The following are extensions for ActualProjectMilestone:

- InstanceSpecification

Generalizations

The following are generalization relationships for ActualProjectMilestone:

- UPDMElement

8.1.1.1.1.1.2 MilestoneSequence

MODAF: A MilestoneSequence (MODAF::MilestoneRelationship) is a relationship between two milestones. DoDAF:NA.

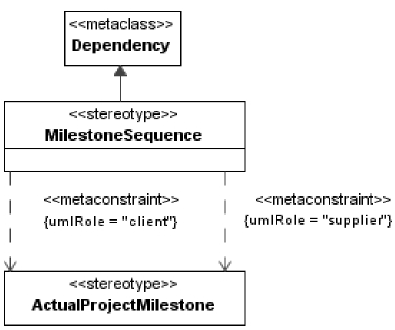


Figure 8.5 - Elements related to the MilestoneSequence stereotype

Constraints

The following are constraints for MilestoneSequence:

- MilestoneSequence.client - Client must be “ProjectMilestone.”
- MilestoneSequence.supplier - Supplier must be “ProjectMilestone.”

Extensions

The following are extensions for MilestoneSequence:

- Dependency

Generalizations

8.1.1.1.2 UPDM L1 ::UPDM L0::Core::AllElements

The AllElements are elements that are part of the All View. The All-Views (AVs) provide 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. They also provide a place to record any findings arising from the architecting 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. Since the AVs provide critical information for the future access and exploitation of an architectural model their population is essential whenever an architecture is created or modified. The AVs provide a critical input into the processes that provide architectural governance.

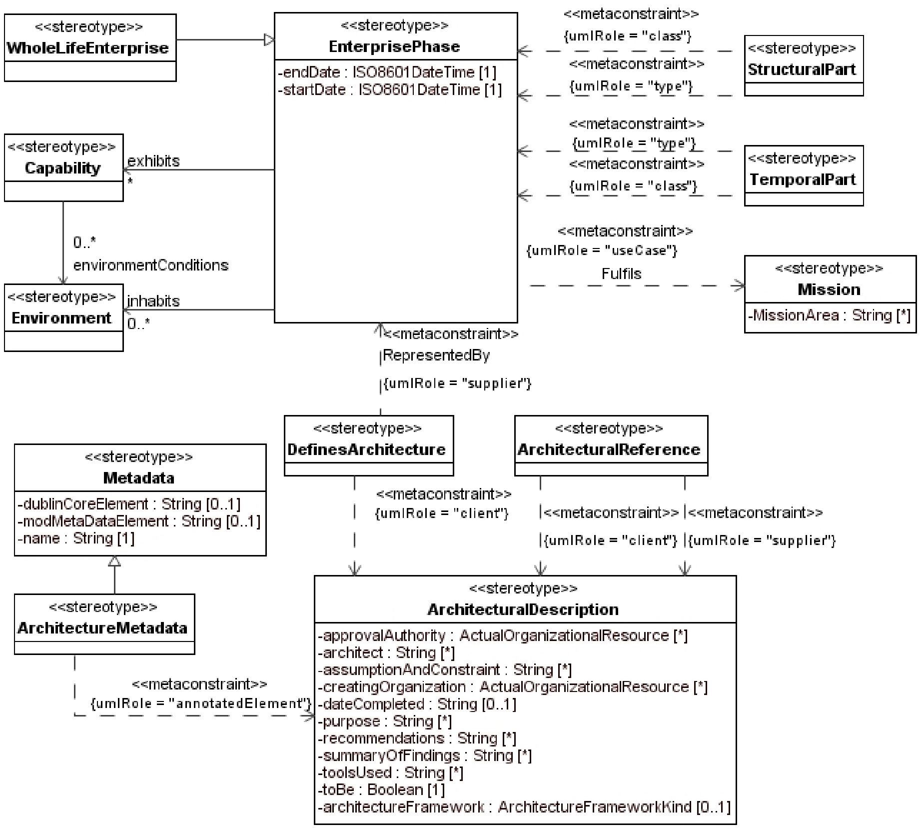


Figure 8.6 - Elements related to the AV-1 Product

The Overview & Summary Information identifies the architecture goals, viewpoint, findings, and recommendations. The AV-1 contains sufficient textual information to enable a reader to select one architecture from among many to read in more detail. AV-1 serves two additional purposes:

1. In the initial phases of architecture development, it serves as a planning guide.
2. Upon completion of an architecture, the AV-1 provides summary textual information concerning that architecture.

The AV-1 comprises a textual executive summary of a given architecture and documents the following descriptions:

Architecture Project Identification - identifies the architecture project name, the architect, and the organization developing the architecture. It also includes assumptions and constraints, identifies the approving authority and the completion date, and records the level of effort and costs (projected and actual) required to develop the architecture.

Scope - identifies 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, and so forth. Scope also identifies the organizations and Communities Of Interest (COI) that fall within the scope of the architecture. The scope also includes the COI that are related to the architecture.

Purpose and Viewpoint - explains the need for the architecture, what it should demonstrate, the types of analyses (e.g., Activity-Based Costing) that will be applied to it, who is expected to perform the analyses, what decisions are expected to be made on the basis of an analysis, who is expected to make those decisions, and what actions are expected to result. The viewpoint from which the architecture is developed is identified (e.g., planner or decision maker).

Context - describes the setting in which the architecture exists. It includes such things as mission, doctrine, relevant goals and vision statements, concepts of operation, scenarios, information assurance context (e.g., types of system data to be protected, such as classified or sensitive but unclassified, and expected information threat environment), other threats and environmental conditions, and geographical areas addressed, where applicable. Context also identifies authoritative sources for the rules, criteria, and conventions that were followed. The tasking for the architecture project and known or anticipated linkages to other architectures are identified.

Tools and File Formats Used - identifies the tool suite used to develop the architecture and file names and formats for the architecture and each product.

Findings - states the findings and recommendations that have been developed based on the architecture effort. Examples of findings include identification of shortfalls, recommended system implementations, and opportunities for technology insertion.

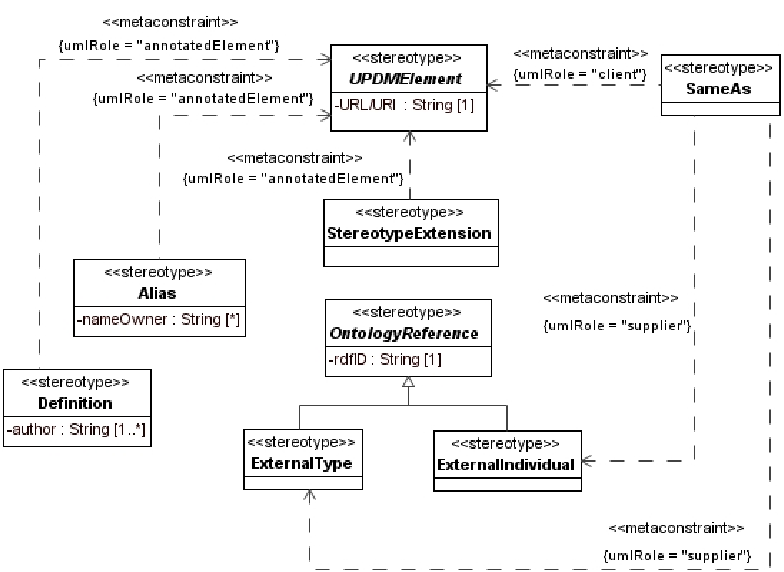


Figure 8.7 - Elements related to the AV-2 Product

The AV-2 contains definitions of terms used in the given architecture. It consists of textual definitions in the form of a glossary, a repository of architecture data, their taxonomies, and their metadata (i.e., data about architecture data), including metadata for tailored products, associated with the architecture products developed. Metadata are the architecture data types, possibly expressed in the form of a physical schema. In this document, architecture data types are referred to as architecture data elements.

AV-2 provides a central repository for a given architecture's data and metadata. AV-2 enables the set of architecture products to stand alone, allowing them to be read and understood with minimal reference to outside resources. AV-2 is an accompanying reference to other products, and its value lies in unambiguous definitions. The key to long-term interoperability can reside in the accuracy and clarity of these definitions.

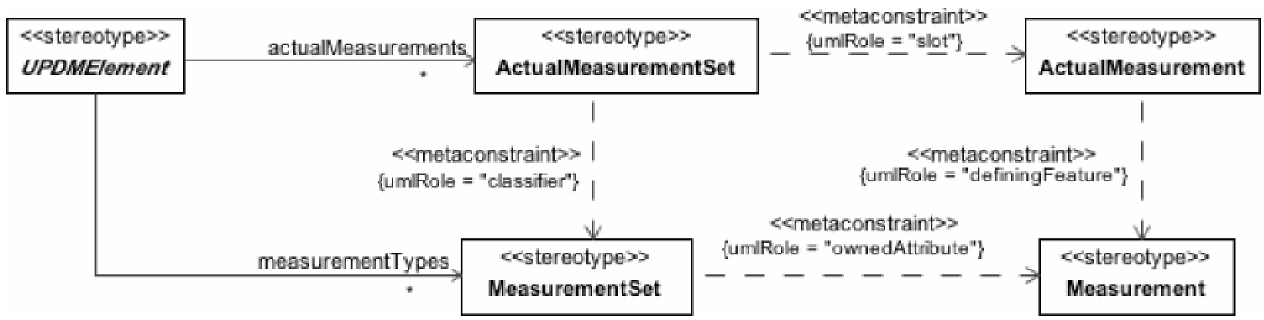


Figure 8.8 - Elements related to the AV-3 product

The AV-3 defines the measureable properties of something in the physical world, expressed in amounts of a unit of measure that can be associated with a UPDMElement.

8.1.1.1.2.1 UPDMElement

UPDM Artifact: Super type for many of the UPDM elements. It provides a means of extending UPDM elements in a common way. With links to the measurement set, it also allows quantitative metrics to be associated with structural and behavioral elements.

Note – UPDMElement is abstract.

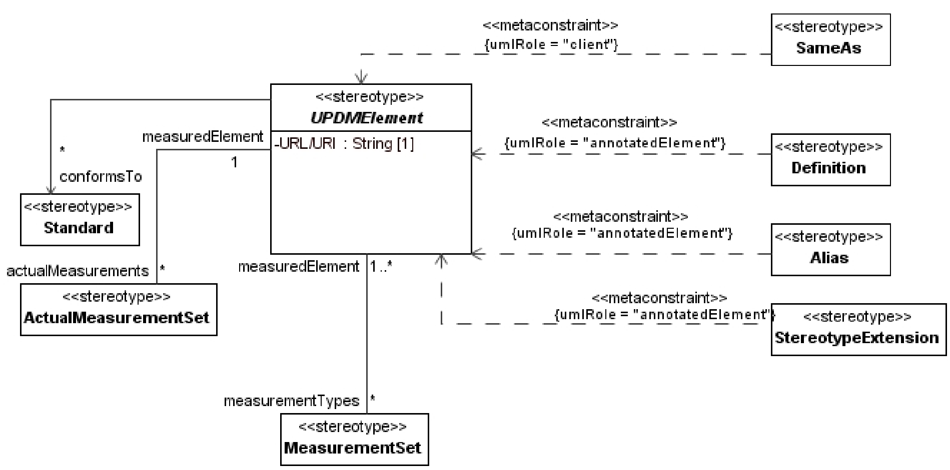


Figure 8.9 - Elements related to the abstract UPDMElement stereotype

Attributes

The following are attributes of UPDMElement:

- conformsTo : Standard[*] - Standard that this UPDM element is conforming to.
- URL/URI : String[1] - Unique identifier for the element.

- measurementTypes : MeasurementSet[*] - Types of measurements corresponding to the actual measurements.
- actualMeasurements : ActualMeasurementSet[*] - The actual measurements to which the element must conform.

8.1.1.1.2.2 UPDM L1 ::UPDM L0::Core::AllElements::Behavior

The behavioral portion of the AllElements profile.

8.1.1.1.2.2.1 PerformedActivity

UPDM: An abstract element that represents a behavior (i.e., a Function or OperationalActivity) that can be performed by a Performer.

DoDAF:

Note – PerformedActivity is abstract.

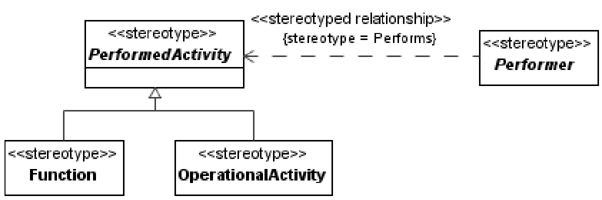


Figure 8.10 - Elements related to the abstract Activity stereotype

Generalizations

The following are generalization relationships for PerformedActivity:

- UPDMElement

8.1.1.1.2.2.2 Performer

UPDM: An abstract element that represents a structural element that can perform behaviors (i.e., PerformedActivity).

DoDAF.

Note – Performer is abstract.

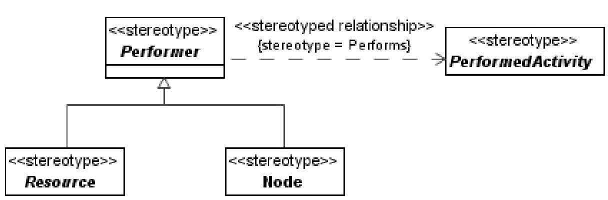


Figure 8.11 - Elements related to the abstract Performer stereotype

Generalizations

The following are generalization relationships for Performer:

- UPDMElement

8.1.1.1.2.2.3 Performs

UPDM: Links a Performer to the behavior that it can perform.

DoDAF: The Performs (DoDAF::activityPerformedByPerformer) relationship is an overlap between a Performer and a PerformedActivity (DoDAF::Activity) wherein the activity is performed by the Performer.

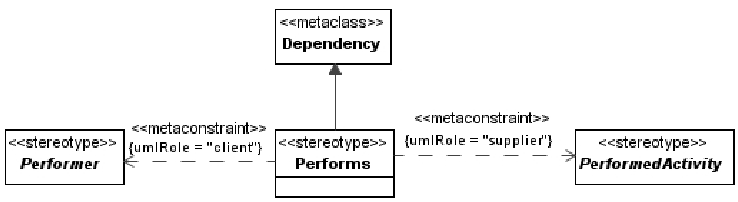


Figure 8.12 - Elements related to the Performs stereotype

Constraints

The following are constraints for Performs:

- Performs.supplier - Values for the supplier property must be stereotyped «PerformedActivity» or its specializations.
- Performs.client - Values for the client property must be stereotyped «Performer» or its specializations.

Extensions

The following are extensions for Performs:

- Dependency

Generalizations

The following are generalization relationships for Performs:

- UPDMElement

8.1.1.1.2.2.4 ImplementsOperational

UPDM: Relationship between a system element that implements an operational element.

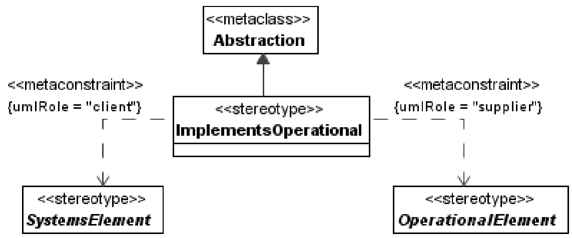


Figure 8.13 - Elements related to the ImplementsOperational stereotype

Constraints

The following are constraints for ImplementsOperational:

- ImplementsOperational.supplier - Value for the supplier property must be stereotyped «OperationalElement» or its specializations.
- ImplementsOperational.client - Value for the client property must be stereotyped «SystemsElement» or its specializations.

Extensions

The following are extensions for ImplementsOperational:

- Abstraction

Generalizations

The following are generalization relationships for ImplementsOperational:

- UPDMElement

8.1.1.1.2.3 UPDM L1 ::UPDM L0::Core::AllElements::Data

The data portion of the AllElements profile.

8.1.1.1.2.3.1 EntityItem

MODAF: (MODAF::Entity): A definition (type) of an item of interest.

DoDAF: NA

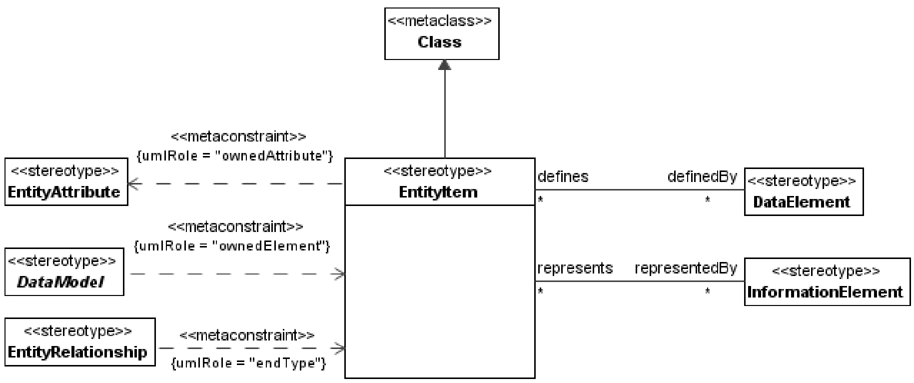


Figure 8.14 - Elements related to the EntityItem stereotype

Attributes

The following are attributes for EntityItem:

- representedBy : InformationElement[*] - The list of InformationElements that represent the EntityItems during exchanges.
- definedBy : DataElement[*] - The list of DataElements that define the EntityItems during exchanges.

Constraints

The following are constraints for EntityItem:

- EntityItem.ownedAttribute - Value for the slot property must be stereotyped «EntityAttribute» or its specializations.

Extensions

The following are extensions for EntityItem:

- Class

8.1.1.1.2.4 UPDM L1 ::UPDM L0::Core::AllElements::Environment

The environmental aspects of the AllElements profile.

8.1.1.1.2.4.1 Climate

MODAF: A type of weather condition, or combination of weather conditions (e.g., high temperature & dry).

DoDAF: The state of an environment or situation in which a Performer performs.

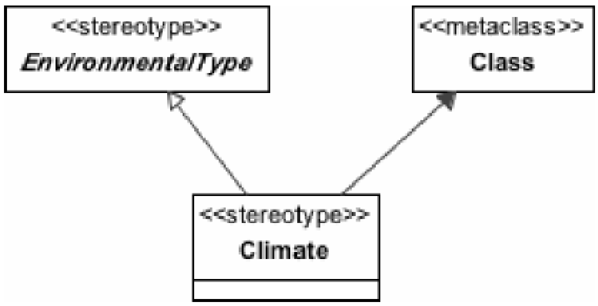


Figure 8.15 - Elements related to the Climate stereotype

Extensions

The following are extensions for Climate:

- Class

Generalizations

The following are generalization relationships for Climate:

- EnvironmentalType

8.1.1.1.2.4.2 Environment

Anything outside the context of the Enterprise that has an influence on the Enterprise in terms of where the Enterprise is located or what function it is supposed to do. A complete description of the Environment will take into account both the physical and logical environment including weather conditions, ambient light, climate and terrain; any security factors including perceived or actual threats and any mechanism of interoperating with other Enterprises.

MODAF: A definition of the conditions in which the “Enterprise” exists or functions. An “Environment” may be specified in terms of “Location” (e.g., terrain), “WeatherCondition” (e.g., cloud, rain) and “Climate” (e.g., tropical), and “LightCondition” (e.g., dark, light, dusk).

DoDAF: An object that encompasses meteorological, geographic, and control features mission significance.

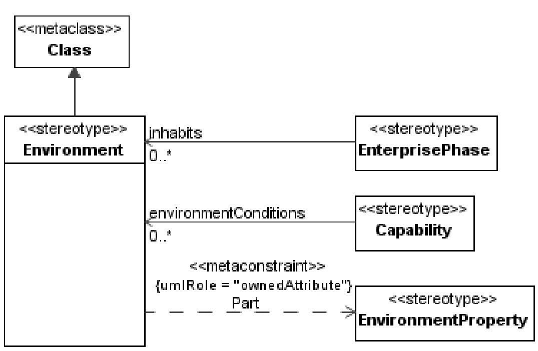


Figure 8.16 - Elements related to the Environment stereotype

Constraints

The following are constraints for Environment:

- Environment.part - Values for the ownedAttribute property must be stereotyped «EnvironmentProperty» or its specializations.

Extensions

The following are extensions for Environment:

- Class

Generalizations

The following are generalization relationships for Environment:

- UPDMElement

8.1.1.1.2.4.3 EnvironmentalType

UPDM Artifact, Abstract Element: A type of the environment.

Note – EnvironmentalType is abstract.

Generalizations

The following are generalization relationships for EnvironmentalType:

- UPDMElement

8.1.1.1.2.4.4 EnvironmentProperty

MODAF:Asserts that an “Environment” has one or more properties. These may be “Climate,” “Location,” or “LightCondition.”

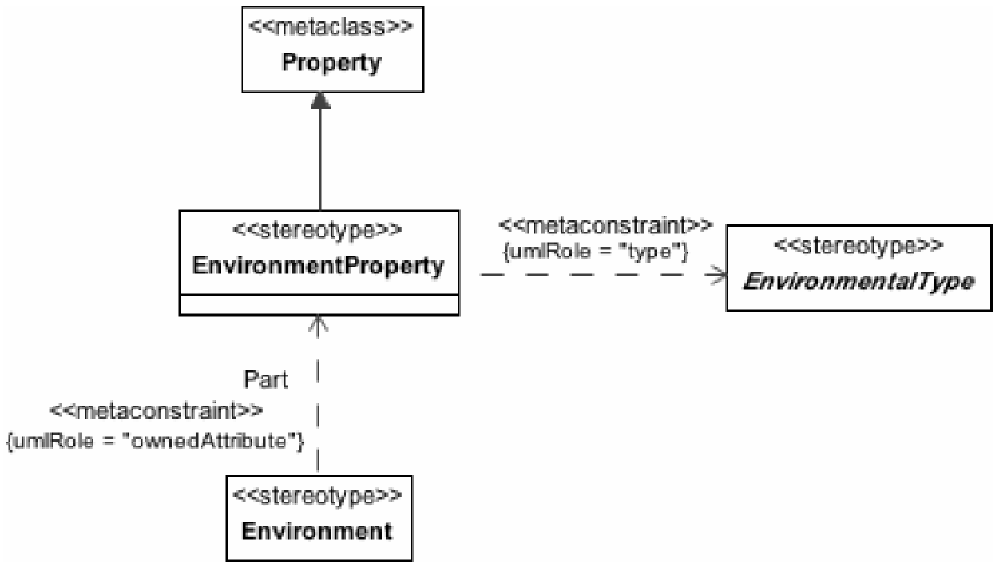


Figure 8.17 - Elements related to the EnvironmentProperty stereotype

Constraints

The following are constraints for EnvironmentProperty:

- EnvironmentProperty.type - Value for the type property must be stereotyped «EnvironmentalType» or its specializations.

Extensions

The following are extensions for EnvironmentProperty:

- Property

Generalizations

The following are generalization relationships for EnvironmentProperty:

- UPDMElement

8.1.1.1.2.4.5 LightCondition

MODAF: A specification of environmental lighting conditions.

DoDAF: NA – this is a specialization of EnvironmentalType (DoDAF::GeoFeature).

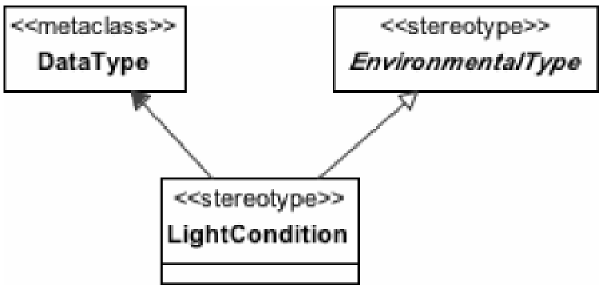


Figure 8.18 - Elements related to the LightCondition stereotype

Extensions

The following are extensions for LightCondition:

- DataType

Generalizations

The following are generalization relationships for LightCondition:

- EnvironmentalType

8.1.1.1.2.4.6 Location

MODAF: A general specification of the surroundings / scenario in which an operation may take place. Examples would be: “desert,” “arctic,” “at sea.”

DoDAF: A point or extent in space that may be referred to physically or logically. Includes concepts such as: Facility, Installation, RealProperty, Site, and instances of conditions such as underwater (as specified in UJTLs).

Extensions

The following are extensions for Location:

- DataType

Generalizations

The following are generalization relationships for Location:

- ReferredLocation

8.1.1.1.2.4.7 ReferredLocation

MODAF: Abstract Element: Either an actual location, or a type of location (i.e., environment) at/in which operations may be conducted.

Note – ReferredLocation is abstract.

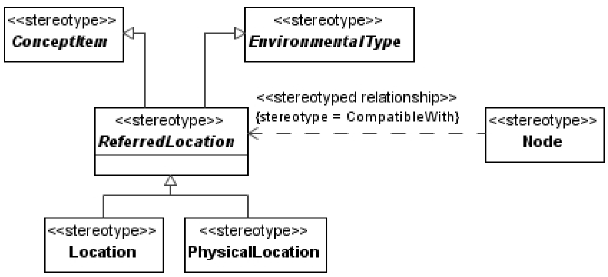


Figure 8.19 - Elements related to the abstract ReferredLocation stereotype

Generalizations

The following are generalization relationships for ReferredLocation:

- ConceptItem
- EnvironmentalType

8.1.1.1.2.4.8 PhysicalLocation

PhysicalLocation is anywhere that can be specified. The means of describing the location is a string (locationDescription). The information contained in that string is governed by the Location.

MODAF: A PhysicalLocation (MODAF::ActualLocation) is a location anywhere on the earth. The means of describing the location is a string (locationDescription). The information contained in that string is governed by the taxonomy reference (e.g., if the PhysicalLocation is a “GPS reference”) the string will contain the GPS coordinates.

Note – this has been extended in UPDM to include non-earth locations.

DoDAF: All subtypes of «IndividualType» Location, such as Facility, Site.

CADM: (343/2) (A) A specific place.

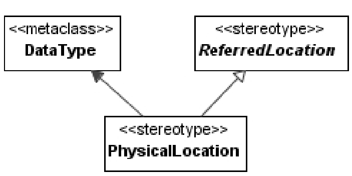


Figure 8.20 - Elements related to the PhysicalLocation stereotype

Attributes

The following are attributes of PhysicalLocation:

- locationDescription : String[1] - The description of the PhysicalLocation.

Extensions

The following are extensions for PhysicalLocation:

- DataType

Generalizations

The following are generalization relationships for PhysicalLocation:

- ReferredLocation

8.1.1.1.2.5 UPDM L1 ::UPDM L0::Core::AllElements::Measurements

The measurement portion of the AllElements profile.

8.1.1.1.2.5.1 ActualMeasurement

UPDM: An actual value of the Measurement.
MODAF: NA
DoDAF: NA

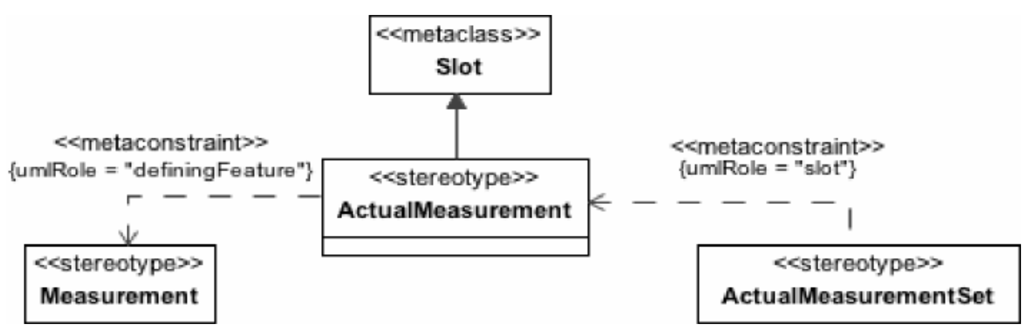


Figure 8.21 - Elements related to the ActualMeasurement stereotype

Constraints

The following are constraints for ActualMeasurement:

- ActualMeasurement.definingFeature - Value for definingFeature property must be stereotyped «Measurement» or its specializations.

Extensions

The following are extensions for ActualMeasurement:

- Slot

Generalizations

The following are generalization relationships for ActualMeasurement:

- UPDMElement

8.1.1.1.2.5.2 ActualMeasurementSet

UPDM: A set or collection of ActualMeasurement(s). A date of measurement can be set. An intent of ActualMeasurementSet can be “Result,” “Required,” or “Estimate.”

MODAF: NA

DoDAF: NA

ActualMeasurement holds all actual measurement defined by connected MeasurementSet.

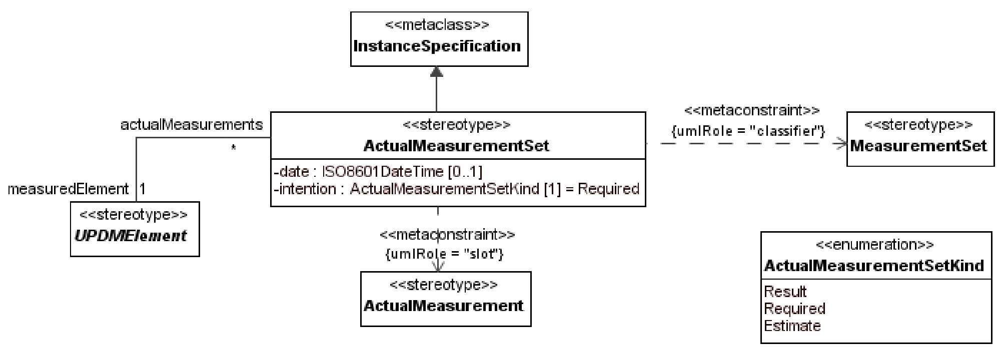


Figure 8.22 - Elements related to the ActualMeasurementSet stereotype

Attributes

The following are attributes of ActualMeasurementSet:

- measuredElement : UPDMElement [1] - Measured element
- date : ISO8601 DateTime[0..1] – Time of the measurement
- intention : ActualMeasurementSetKind[1] – Intention of the measurement. Default value is “Required.”

Constraints

The following are constraints for ActualMeasurementSet:

- ActualMeasurementSet.slot - Value for the slot property must be stereotyped «ActualMeasurement» or its specializations.
- ActualMeasurementSet.classifier - Value for the classifier property must be stereotyped «MeasurementSet» or its specializations.

Extensions

The following are extensions for ActualMeasurementSet:

- InstanceSpecification

Generalizations

The following are generalization relationships for ActualMeasurementSet:

- UPDMElement

8.1.1.1.2.5.3 Measurement

MODAF: MeasurableProperty: A property of something in the physical world, expressed in amounts of a unit of measure. The property may have a required value - either specified by the [defaultValue] from UML::property attribute, or the [minValue] and [maxValue] to specify a required range.

DoDAF: Measure: A Measurement (DoDAF::Measure) is the magnitude of some attribute of an individual.

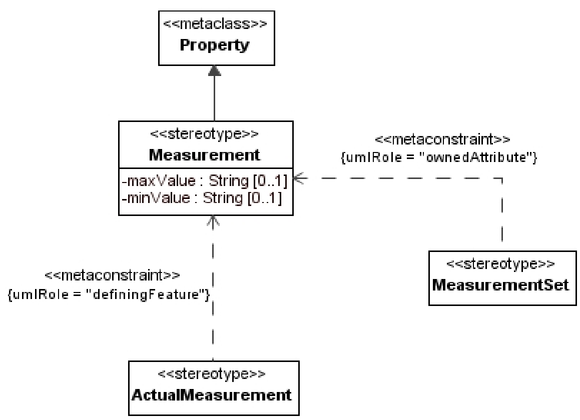


Figure 8.23 - Elements related to the Measurement stereotype

Attributes

The following are attributes of Measurement:

- maxValue : String[0..1] - The maximum value of the measurement.
- minValue : String[0..1] - The minimum value of the measurement.

Extensions

The following are extensions for Measurement:

- Property

Generalizations

The following are generalization relationships for Measurement:

- UPDMElement

8.1.1.1.2.5.4 MeasurementSet

UPDM: A set or collection of Measurement(s).

MODAF: NA

DoDAF: NA

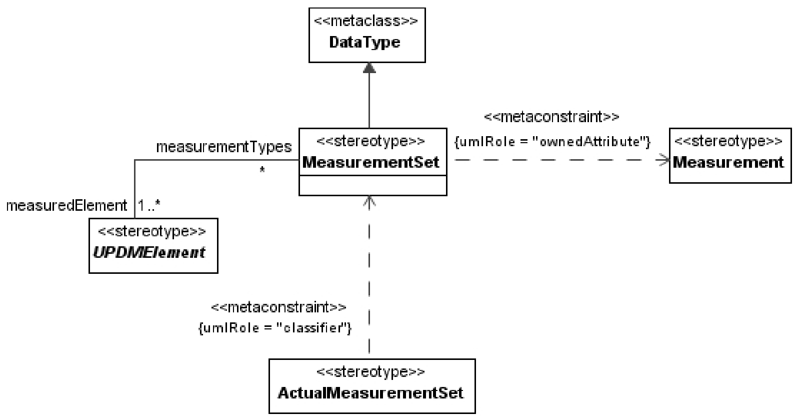


Figure 8.24 - Elements related to the MeasurementSet stereotype

Attributes

The following are attributes of MeasurementSet:

- measuredElement : UPDMElement[1 ..*] - Measured element.

Constraints

The following are constraints for MeasurementSet:

- MeasurementSet.ownedAttribute - Values for the ownedAttribute property must be stereotyped «Measurement» or its specializations.

Extensions

The following are extensions for MeasurementSet:

- DataType

Generalizations

The following are generalization relationships for MeasurementSet:

- UPDMElement

8.1.1.1.2.5.5 Enumeration ActualMeasurementSetKind

Possible kinds of ActualMeasurementSet intention

EnumerationLiterals

- Result
- Required
- Estimate

8.1.1.1.2.6 UPDM L1 ::UPDM L0::Core::AllElements::Views

The views section of the AllElements profile.

8.1.1.1.2.6.1 ArchitecturalDescription

An enterprise has an architecture, which is manifested through an ArchitecturalDescription. That ArchitecturalDescription consists of a number of Views each of which corresponds to a ViewPoint. ArchitecturalDescriptions can also reference other ArchitecturalDescriptions.

MODAF: A specification of a system of systems at a technical level that also provides the business context for the system of systems.

DoDAF: Information describing architecture.

IEEE1471 describes an architectural description as “a collection of products to document the architecture of a system.” This is something of a circular definition (as product in this sense is an architectural product), and also assumes a technical system, whereas MODAF compliant architectures describe an enterprise - i.e., the system of systems and the human processes they support.

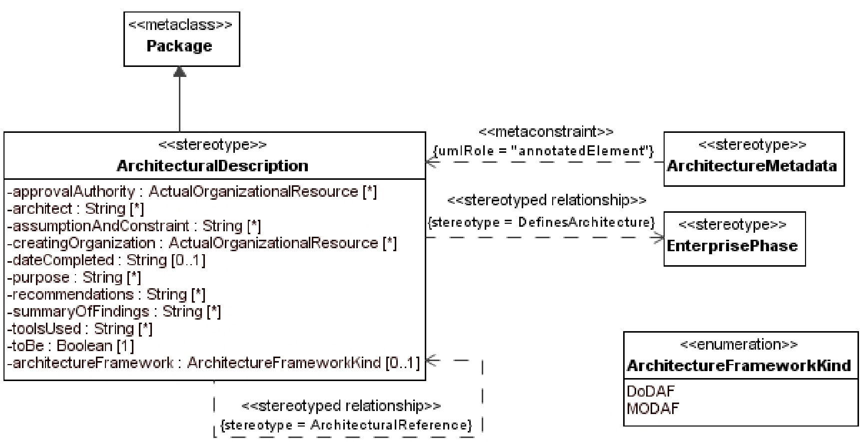


Figure 8.25 - Elements related to the ArchitecturalDescription stereotype

Attributes

The following are attributes of ArchitecturalDescription:

- approvalAuthority : ActualOrganizationalResource[*] - References the actual organizational resource that has the authority to approve the architectural description.
- architect : String[*] - The name of the architect responsible for the ArchitecturalDescription.
- assumptionAndConstraint : String[*] - Any assumptions, constraints, and limitations contained in the ArchitecturalDescription, including those affecting deployment, communications performance, information assurance environments.
- creatingOrganization : ActualOrganizationalResource[*] - Describes the ActualOrganizationalResource creating the ArchitecturalDescription.

- dateCompleted : String[0.. 1] - Date that the Architectural Description was completed.
- purpose : String[*] - Explains 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 are expected to be made on the basis of each form of analysis, who is expected to make those decisions, and what actions are expected to result.
- recommendations : String[*] - States the recommendations that have been developed based on the architecture effort. Examples include recommended system implementations, and opportunities for technology insertion.
- summaryOfFindings : String[*] - Summarizes the findings that have been developed so far. This may be updated several times during the development of the ArchitecturalDescription.
- toolsUsed : String[*] - Identifies any tools used to develop the ArchitecturalDescription as well as file names and formats if appropriate.
- toBe : Boolean[1] - Indicates whether the ArchitecturalDescription is existing or future.
- architectureFramework : ArchitectureFrameworkKind [0 ..1] – The architectureFramework tag identifies the subset of aliases to use within the context of the ArchitectureDescription (i.e., DoDAF, MODAF, or none).

Constraints

The following are constraints for ArchitecturalDescription:

- ArchitecturalDescription.ownedElement - All owned packages must be stereotyped «View» or its specializations.
- ArchitecturalDescription .architectureFramework - If the property is set to DoDAF, only aliases scoped under the DoDAF profile can be used – if set to MODAF, then only MODAF aliases can be used. Should the property be set to nothing, none of the aliases can be used.

Extensions

The following are extensions for ArchitecturalDescription:

- Package

Generalizations

The following are generalization relationships for ArchitecturalDescription:

- UPDMElement

8.1.1.1.2.6.2 ArchitecturalReference

MODAF: Asserts that one architectural description (referrer) refers to another (referred).

DoDAF: NA

A reference from one ArchitecturalDescription to another where one is the client and the other is the supplier. ArchitecturalDescriptions can reference each other to provide a complete picture of the Architecture.

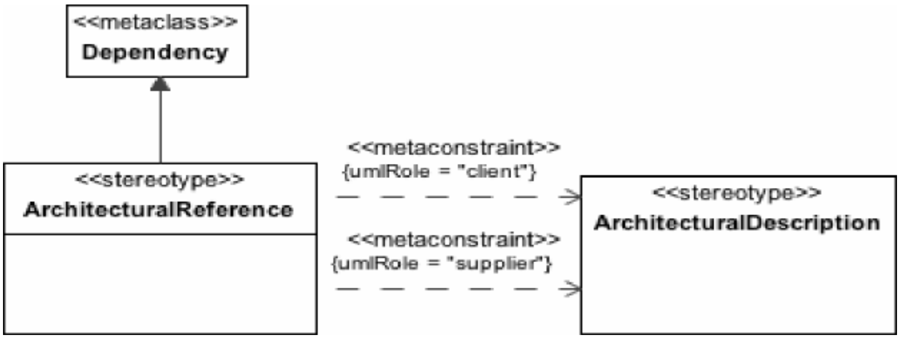


Figure 8.26 - Elements related to the ArchitecturalReference stereotype

Constraints

The following are constraints for ArchitecturalReference:

- ArchitecturalReference.client - Value for the client property must be stereotyped «ArchitecturalDescription» or its specializations.
- ArchitecturalReference. supplier - Value for the supplier property must be stereotyped «ArchitecturalDescription» or its specializations.

Extensions

The following are extensions for ArchitecturalReference:

- Dependency

Generalizations

The following are generalization relationships for ArchitecturalReference:

- UPDMElement

8.1.1.1.2.6.3 ArchitectureMetadata

UPDM: Information on ArchitecturalDescription. It states things like what methodology was used, notation, etc.

MODAF: A Metadata element that applies to the whole architecture.

DoDAF: NA

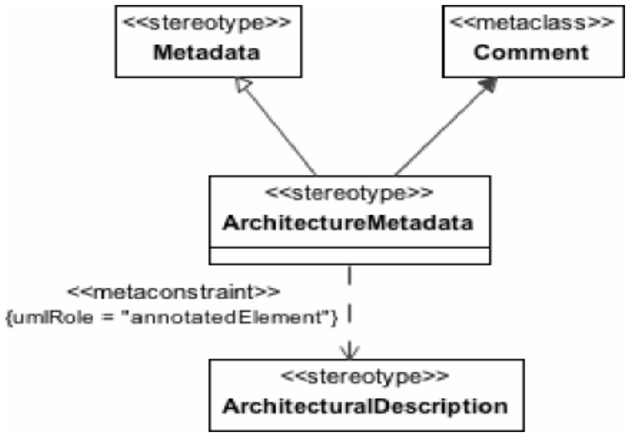


Figure 8.27 - Elements related to the ArchitectureMetadata stereotype

Constraints

The following are constraints for ArchitectureMetadata:

- ArchitectureMetadata.annotatedElement - Value for the annotatedElement property must be stereotyped «ArchitecturalDescription» or its specializations.

Extensions

The following are extensions for ArchitectureMetadata:

- Comment

Generalizations

The following are generalization relationships for ArchitectureMetadata:

- Metadata

8.1.1.1.2.6.4 DefinesArchitecture

UPDM: An ArchitecturalDescription describes the architecture for an EnterprisePhase. The DefinesArchitecture stereotype establishes a relationship between ArchitecturalDescription and EnterprisePhase.

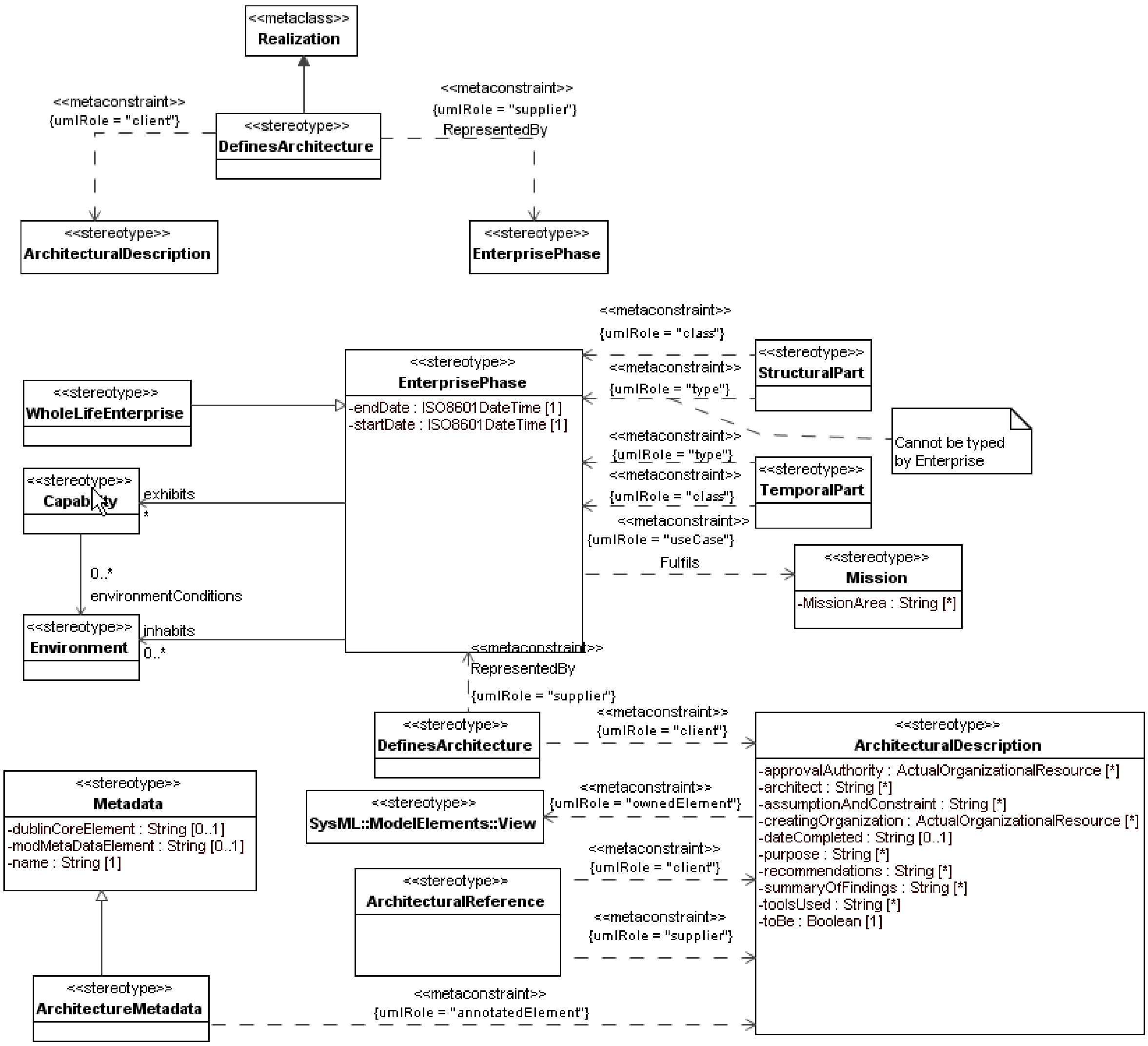


Figure 8.28 - Elements related to the DefinesArchitecture stereotype

Constraints

The following are constraints for DefinesArchitecture:

- DefinesArchitecture.client - Value for the client property must be stereotyped «ArchitecturalDescription» or its specializations.
- DefinesArchitecture.supplier - Value for the supplier property must be stereotyped «EnterprisePhase» or its specializations.

Extensions

The following are extensions for DefinesArchitecture:

- Realization

Generalizations

The following are generalization relationships for DefinesArchitecture:

- UPDMElement

8.1.1.1.2.6.5 Metadata

MODAF: Annotation that can be applied to any element in the architecture.

DoDAF: NA

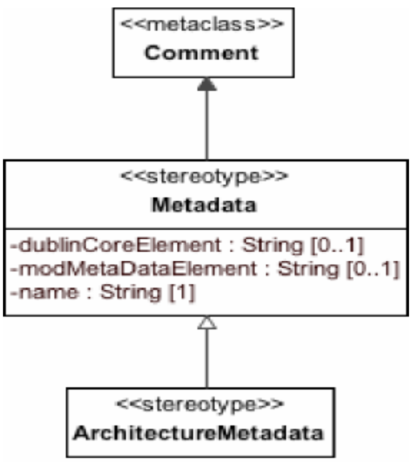


Figure 8.29 - Elements related to the Metadata stereotype

Attributes

The following are attributes of Metadata:

- dublinCoreElement : String[0.. 1] - If the meta data corresponds to the Dublin Core Meta-Data Standard, then the metadata element name should be listed here.
- modMetaDataElement : String[0.. 1] - If the meta data corresponds to the MOD Meta-Data Standard, then the metadata element name should be listed here.
- name : String[1] - The name of the Metadata.

Extensions

The following are extensions for Metadata:

- Comment

Generalizations

The following are generalization relationships for Metadata:

- UPDMElement

8.1.1.1.2.6.6 Enumeration ArchitectureFrameworkKind

Architecture Framework Kind

Enumeration Literals

- DoDAF
- MODAF
- NAF

8.1.1.1.3 UPDM L1 ::UPDM L0::Core::ExternalTypes

A type defined by an external ontology. This may be higher-order - i.e., a type of a type.

Figure 8.30 - Elements related to the ISO8601DateTime stereotype

Elements related to the ISO8601DateTime stereotype.

8.1.1.1.3.1 ISO8601 DateTime

MODAF: A date and time specified in the ISO8601 date-time format including timezone designator (TZD): YYYY-MMDDThh:mm:ssTZD So, 7:20pm and 30 seconds on 30th July 2005 in the CET timezone would be represented as “2005- 07-30T19:20:30+01 :00.”

DoDAF: NA

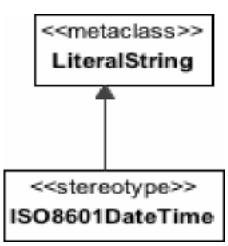


Figure 8.31 - Elements related to the ISO8601DateTime stereotype

Extensions

The following are extensions for ISO8601DateTime:

- LiteralString

Generalizations

The following are generalization relationships for ISO8601DateTime:

- UPDMElement

8.1.1.1.4 UPDM L1 ::UPDM L0::Core::OperationalElements

OperationalElements group elements used to model product for Operational View. Operational Views can either be “as is” architectures, in which case it is a high level description of the current behavior/problems of the architecture at an abstract level or a “to-be” architecture, for a future EnterprisePhase, in this case it represents a set of requirements that need to be met. An Operational View (OV) describes the tasks and activities, operational elements, and information exchanges required to conduct operations. A pure OV is material independent. However, operations and their relationships may be influenced by new technologies such as collaboration technology, where process improvements are in practice before policy can reflect the new procedures. There may be some cases, as well, in which it is necessary to document the way processes are performed given the restrictions of current systems, in order to examine ways in which new systems could facilitate streamlining the processes (as-is). In such cases, an OV may have material constraints and requirements that must be addressed. For this reason, it may be necessary to include some high-level Systems View (SV) architecture data as overlays or augmenting information onto the OV products.

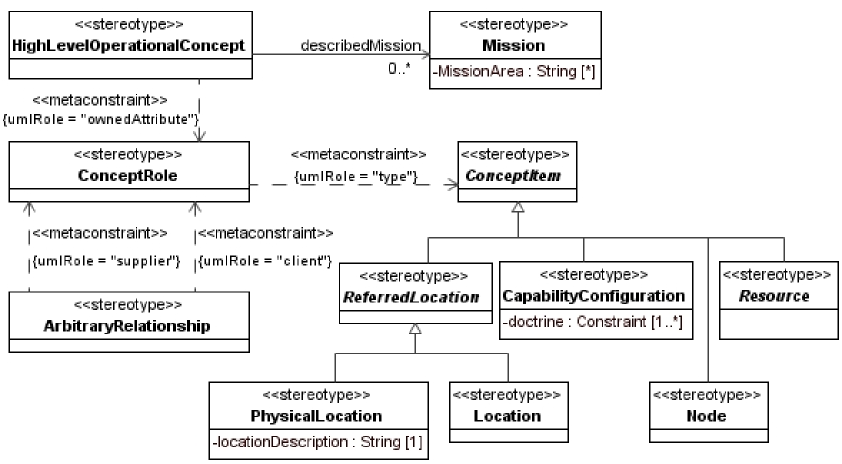


Figure 8.32 Elements related to the OV-1 product

MODAF: OV-1 addresses the high level operational concepts related to one or more missions. An OV-1 describes a mission, class of mission, or scenario; and highlights the main operational elements and interesting or unique aspects of operations.

The OV-1 has two purposes. First, it provides a means of organizing the operational architecture models into distinct groups based on scenario context. Second, it communicates the essence of the scenario context in an essentially graphical form.

DoDAF: The OV-1 DoDAF-described View describes a mission, class of mission, or scenario. It shows the main operational concepts and interesting or unique aspects of operations. It describes the interactions between the subject architecture and its environment, and between the architecture and external systems. A textual description accompanying the graphic is crucial. Graphics alone are not sufficient for capturing the necessary architecture data.

The purpose of High-level Operational Concept Graphic is to provide a high-level graphical and textual description of operational concept (high level organizations, missions, geographic configuration, connectivity, etc.) of what the architecture is supposed to do, and how it is supposed to do it. The OV-1, along with the corresponding AV-1 product is intended to serve as an executive summary of the architecture.

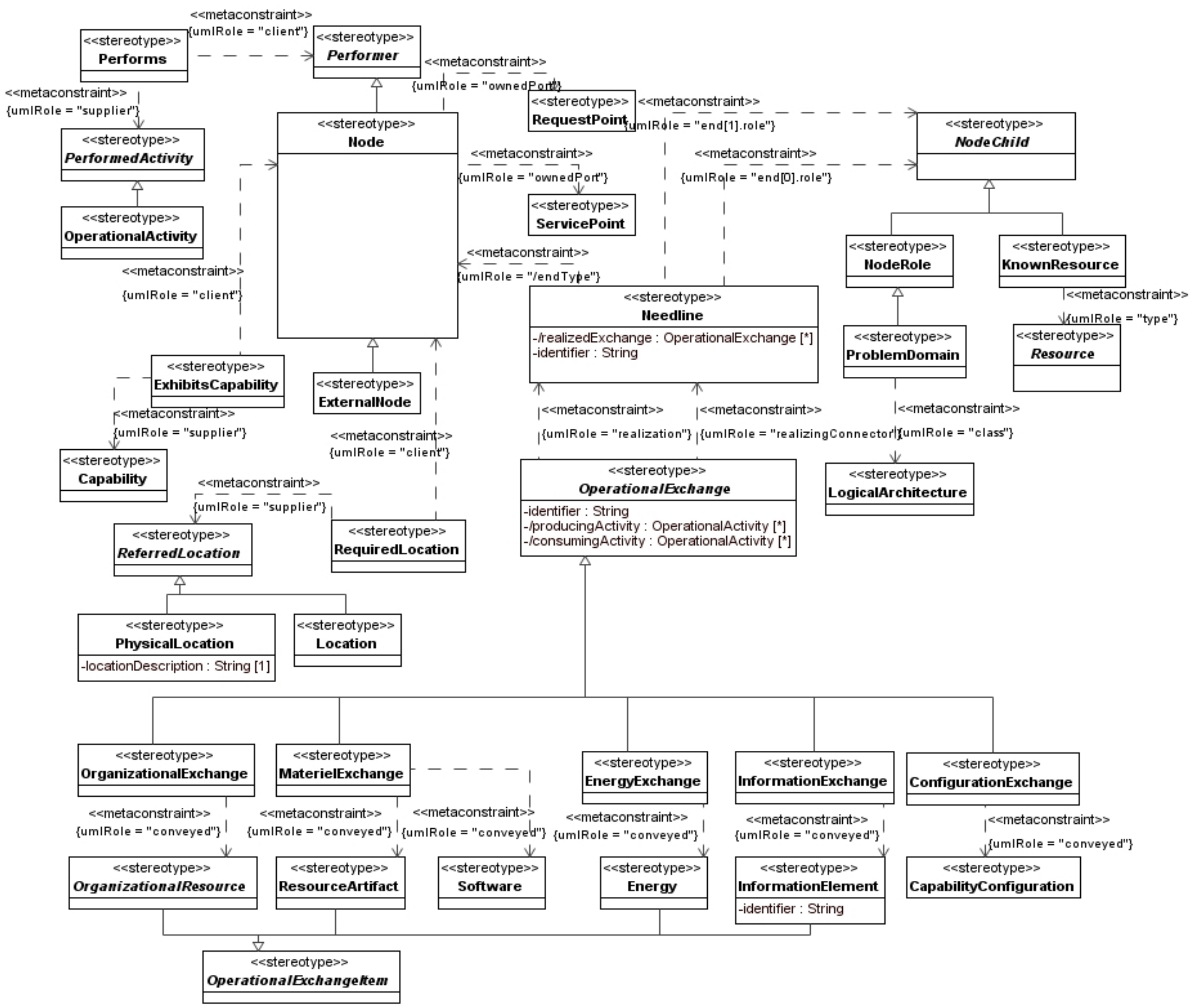


Figure 8.33 - Elements related to the OV-2 product

The Operational Node Connectivity Description is intended to track the need to exchange information from specific operational nodes (that play a key role in the architecture) to others. MoDAF modifies the OV-2 is in two ways. First of all, it recommends that an OV-2 diagram shows the platforms or geographic locations at which operational nodes are deployed. Secondly, it provides additional information about each needline in the form of a requirements specification.

There are now four types of needlines identified as follows:

1. InformationExchange
2. EnergyFlow

3. MaterialFlow

4. MovementOfPeople

In addition, MoDAF permits service-oriented architectures. Instead of needlines between nodes, it is possible simply to show which services the nodes provide and consume. Finally, MoDAF again permits known resources to be shown in an OV-2. However, this must be clearly shown as a KnownResource in an OV-2 model. The concept of a LogicalArchitecture is introduced; this is the container class for all the nodes and KnownResources.

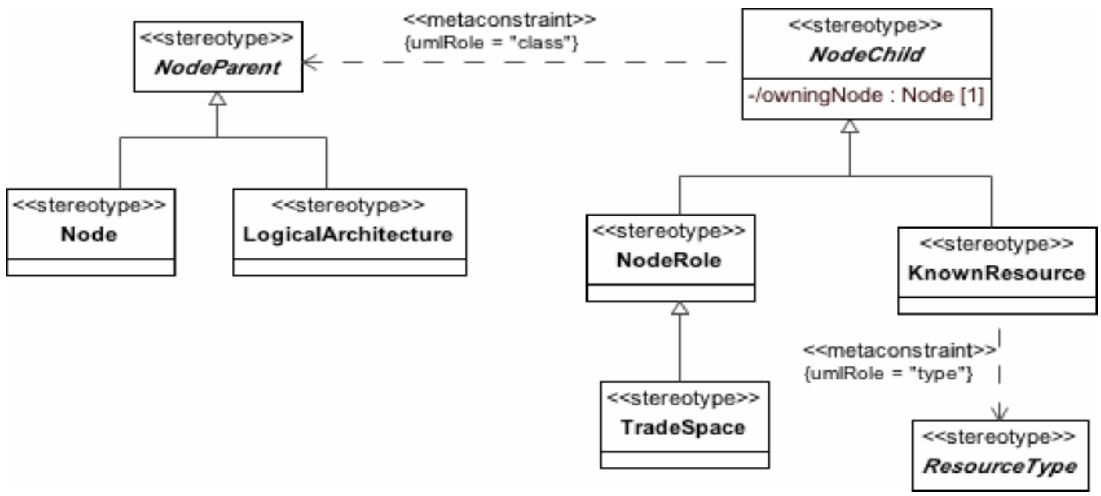


Figure 8.34 - Elements related to the OV-2 Nodes stereotype

Nodes related to OV-2 View.

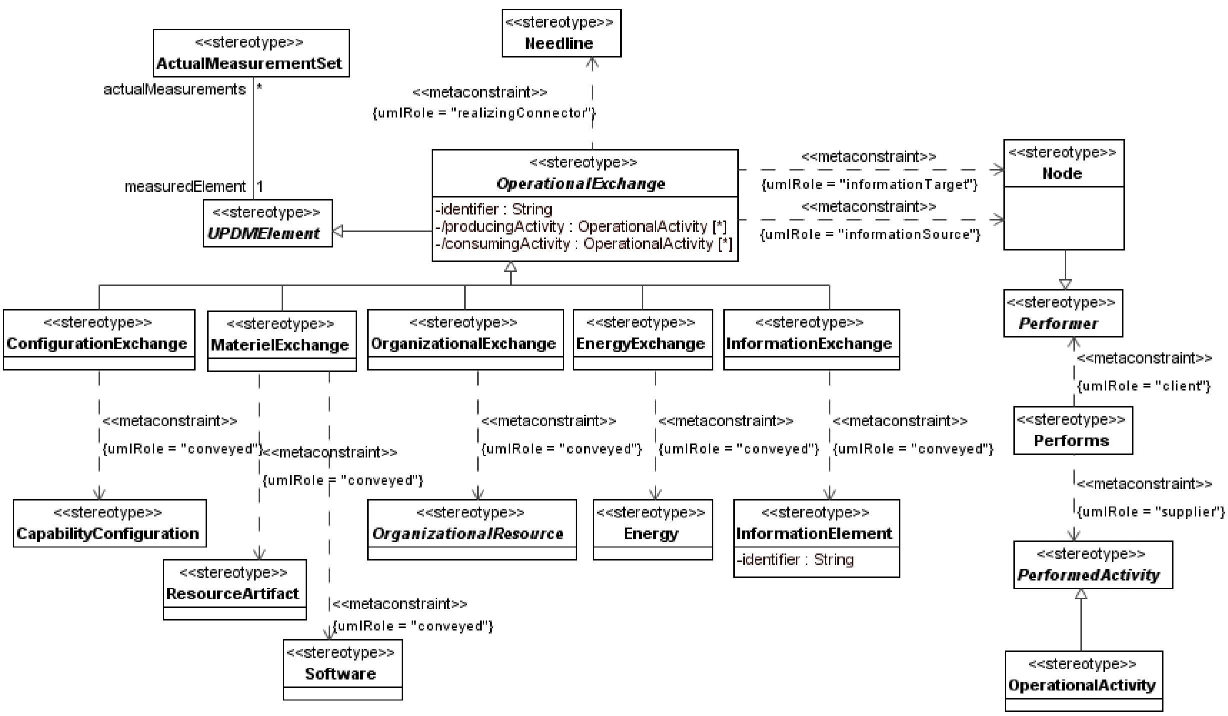


Figure 8.35 - Elements related to the OV-3 product

MODAF: The Operational Information Exchange Matrix (OV-3) addresses operational information exchanges between nodes.

DoDAF: The Operational Resource Flow Matrix (OV-3) DoDAF-described addresses operational resource flows exchanged between Operational Activities and locations.

Information exchanges express the relationship across the three basic architecture data elements of an OV (operational activities, operational nodes, and information flow) with a focus on the specific aspects of the information flow and the information content.

The Information Exchanges of the OV-3 should remain at a high level of aggregation to represent actual information workflow products that are used at the operational nodes shown in OV-2 (and not their subordinate operational nodes).

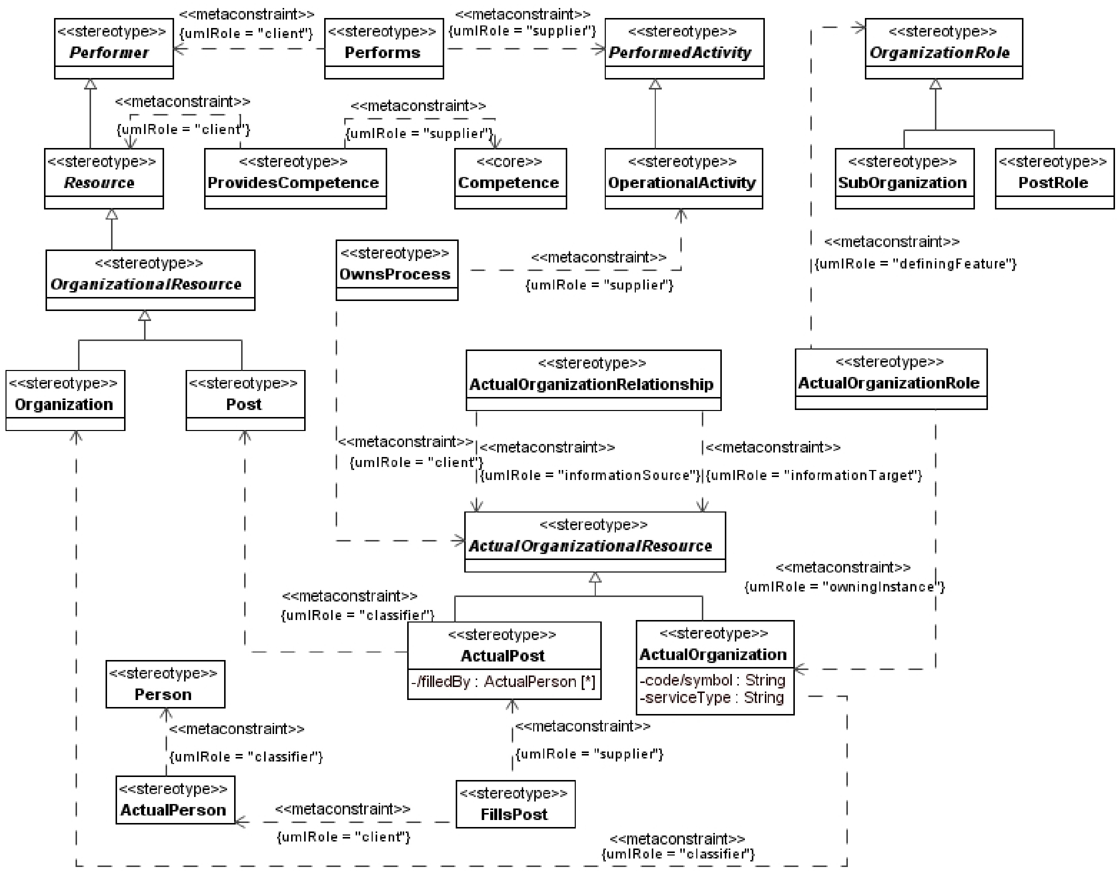


Figure 8.36 - Elements related to the OV-4 Actual product

This is the OV-4 Actual View. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. MoDAF divides the OV-4 two views, an OV-4 Typical and an OV-4 Actual. The former is exactly as the DoDAF OV-4, while the latter is a special form of the SV- 1; where the resources are restricted to being organizational.

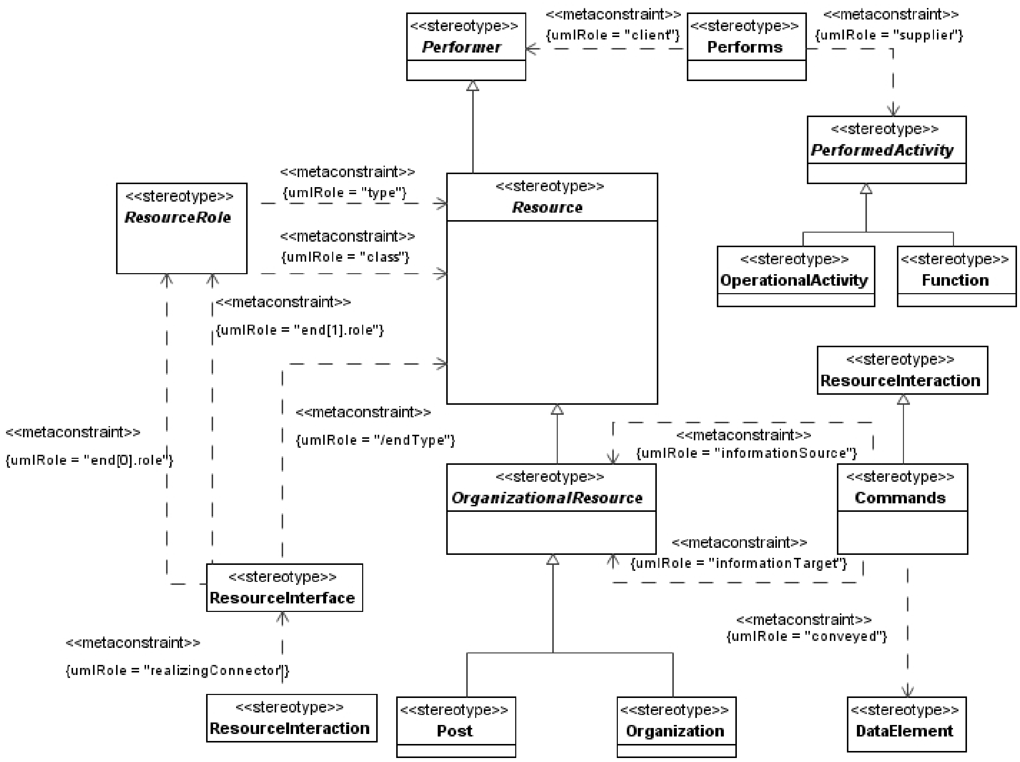


Figure 8.37 - Elements related to the OV-4 Typical product

This is the OV-4 Typical View. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. MoDAF divides the OV-4 two views, an OV-4 Typical and an OV-4 Actual. The former is exactly as the DoDAF OV-4, while the latter is a special form of the SV-1; where the resources are restricted to being organizational.

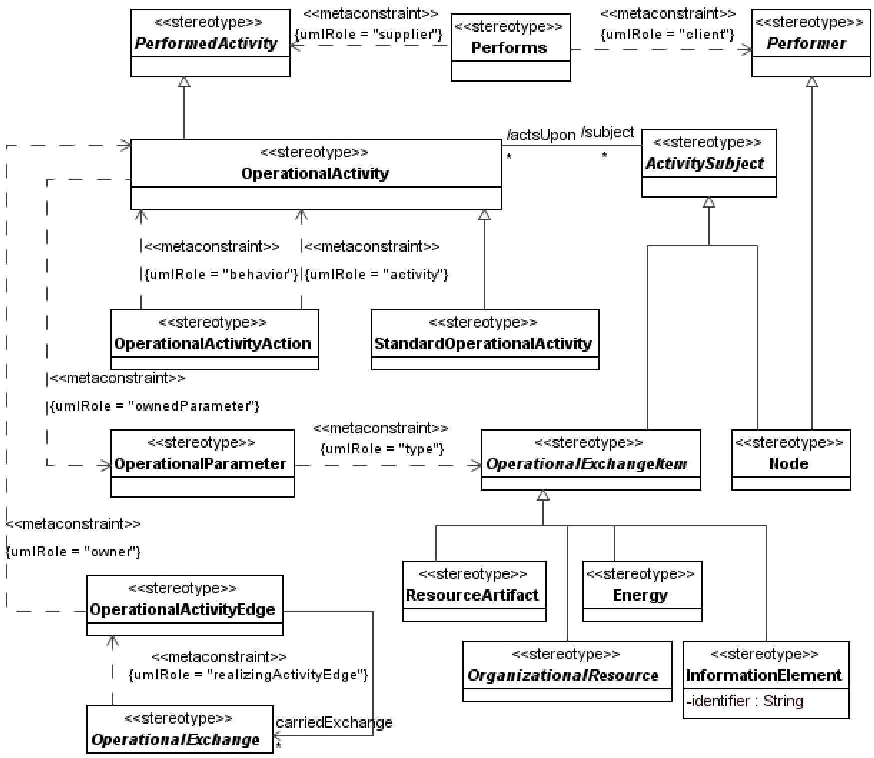


Figure 8.38 - Elements related to the OV-5 product

MODAF: The Operational Activity Model (OV-5) describes the operations that are normally conducted in the course of achieving a mission or a business goal. It describes operational activities (or tasks), Input/Output flows between activities and to/from activities that are outside the scope of the Architecture.

DoDAF: The Operational Activity Model DoDAF-described View describes the operations that are normally conducted in the course of achieving a mission or a business goal. It describes operational activities (or tasks); Input/Output flows between activities, and to/from activities that are outside the scope of the Architecture.

The Operational Activity Model describes the operations that are normally conducted in the course of achieving a mission or a business goal, from a net-centric perspective. It describes capabilities, operational activities (or tasks), input and output (I/O) flows between activities, and I/O flows to/from activities that are outside the scope of the architecture. It is imperative that the levels-of-detail between the OV-2, OV-3, and OV-5 remain cohesive. For example, if one diagram of OV-2 operational nodes is developed that shows aggregated organizations only, then it is imperative that the corresponding OV-5 product be developed to show only those operational activities that are meaningful with respect to these operational nodes. Similarly, the information exchanges of OV-3 should remain at a high level of aggregation to represent actual information workflow products that are used at the operational nodes depicted in OV-2 (and not their subordinate operational nodes). The net-centric OV-5 may be used in the following ways:

- Delineate lines of dependency on external activities when coupled with OV-2.
- Highlight information flows to depict the status of the information's refinement (raw, pre-processed, fused).
- Provide the critical foundation for depicting Task, Post, Process, and Use (TPPU) activity sequencing and timing in the OV-6a, OV-6b, and OV-6c.

- Identify critical mission threads and operational information exchanges by annotating which activities are critical (i.e., identify the activities in the model that are critical).

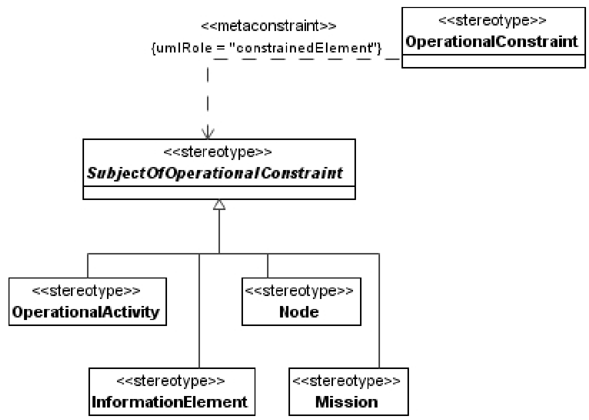


Figure 8.39 - Elements related to the OV-6a product

The Operational Rules Model specifies operational or business rules that are constraints on an enterprise, a mission, operation, business, or an architecture. While other OV products (e.g., OV-1, OV-2, and OV-5) describe the structure of a business-what the business can do-for the most part, they do not describe what the business must do, or what it cannot do. At the mission level, OV-6a may consist of doctrine, guidance, rules of engagement, and so forth. At the operation level, rules may include such things as a military Operational Plan (OPLAN). 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).” At a top level, rules should at least embody the concepts of operations defined in OV- 1, and should provide guidelines for the development and definition of more detailed rules and behavioral definitions that will occur later in the architecture definition process.

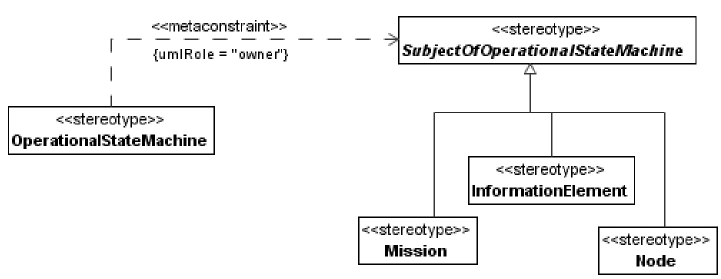


Figure 8.40 - Elements related to the OV-6b product

MODAF: OV-6b: The Operational State Transition Description is a graphical method of describing how an Operational Node or activity responds to various events by changing its state. The diagram represents the sets of events to which the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

DoDAF: The Operational State Transition Description (OV-6b) DoDAF-described View is a graphical method of describing how an Operational Activity responds to various events by changing its state. The diagram represents the sets of events to which the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

Alternatively, OV-6b can be used to reflect the explicit sequencing of actions internal to a single operational activity or the sequencing of operational activities with respect to a specific operational node. In a net-centric architecture, the OV6b is used to describe the set of state transitions for providers and consumers in the Net-Centric Environment (NCE) in response to the posting of information to the NCE or retrieving of information from the NCE.

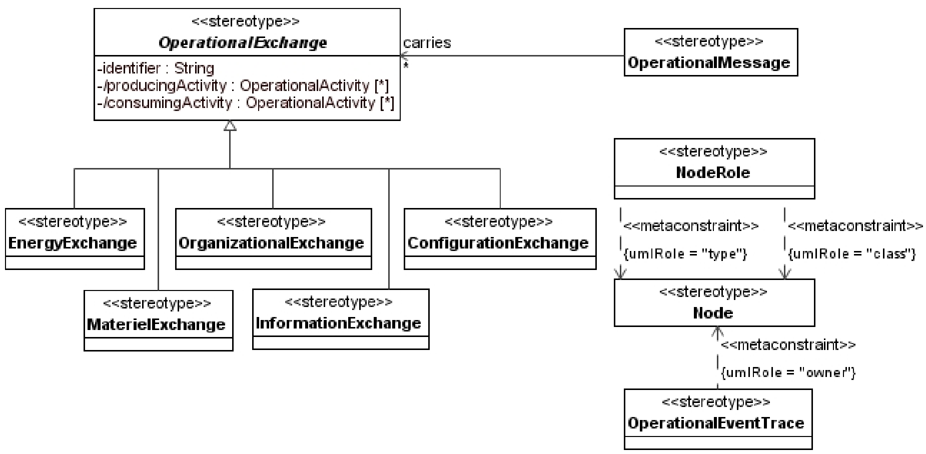


Figure 8.41 - Elements related to the OV-6c product

MODAF: OV-6c: The Operational Event-Trace Description provides a time-ordered examination of the information exchanges between participating Operational Nodes as a result of a particular scenario. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

DoDAF: The Operational Event-Trace Description (OV-6c) DoDAF-described View provides a time ordered examination of the resource flows as a result of a particular scenario. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

The Operational Event- Trace Description provides a time-ordered examination of the information exchanges between participating operational nodes as a result of a particular operational thread or scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation and represent a specific capability. The OV-6c is also used in conjunction with an OV-5 to depict process flow (such as an IDEF3 model). A process flow model captures precedence and causality relations between situations and events by providing a structured method for expressing knowledge about how a process or organization works. A process flow model should be annotated with the names of the operational nodes responsible for conducting those activities.

The net-centric OV-6c describes the business and mission processes that need to be executed to achieve Net-Centric Operations (NCO). The ability to discover, access, and understand information and capabilities from the NCE, where and when they are needed, is supported by the OV-6c and can be decomposed to the level of specificity required for the subject architecture. In the NCE, the OV-6c may depict the following:

- Exchanges between the Service Functionality Providers and Service Consumers, the Service Consumers and external Service Functionality Providers, and between the Service Functionality Providers and Unanticipated Users.

- Sequences that describe the timeline for the availability of information for any of its refinement states (raw, preprocessed, fused).
- Handling, methodologies, and the Enterprise Information Environment (EIE) infrastructure components that support the operational concepts of post before processing.
- Illustration of one-to-many, many-to-one, and many-to-many exchanges between Service Functionality Providers and Service Consumers found in the net-centric OV-3.

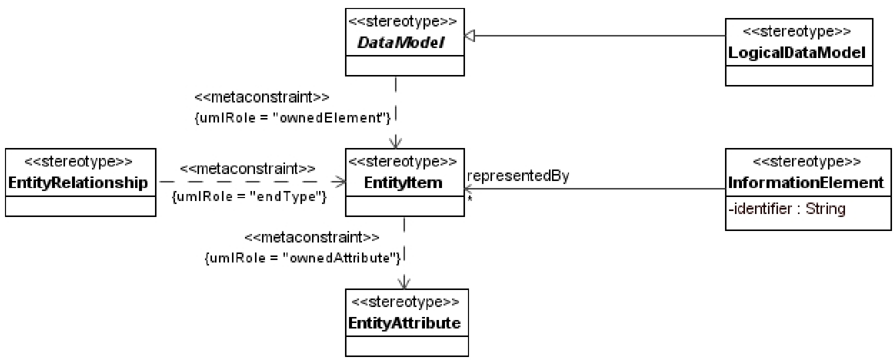


Figure 8.42 - Elements related to the OV-7 product

MODAF: Information Models (OV-7) address the information perspective on an operational architecture.

DoDAF: The Conceptual Data Model (DIV-1), a new DoDAF-described View in DoDAF V2.0, addresses the information concepts at a high-level on an operational architecture. The Logical Data Model (DIV-2) DoDAF-described View allows analysis of an architecture’s data definition aspect, without consideration of implementation specific or product specific issues.

8.1.1.1.4.1 UPDM L1 ::UPDM L0::Core::OperationalElements::Behavior

Behavioral section of the OperationalElements Profile.

8.1.1.1.4.1.1 OperationalActivity

MODAF:A logical process, specified independently of how the process is carried out.

Note – An OperationalActivity may only be carried out by a logical Node.

DoDAF: 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 the hierarchy of the OV-5). It is used to portray operational actions not hardware/software system functions. (DoDAF)

Note – This is also a specialization of Activity.

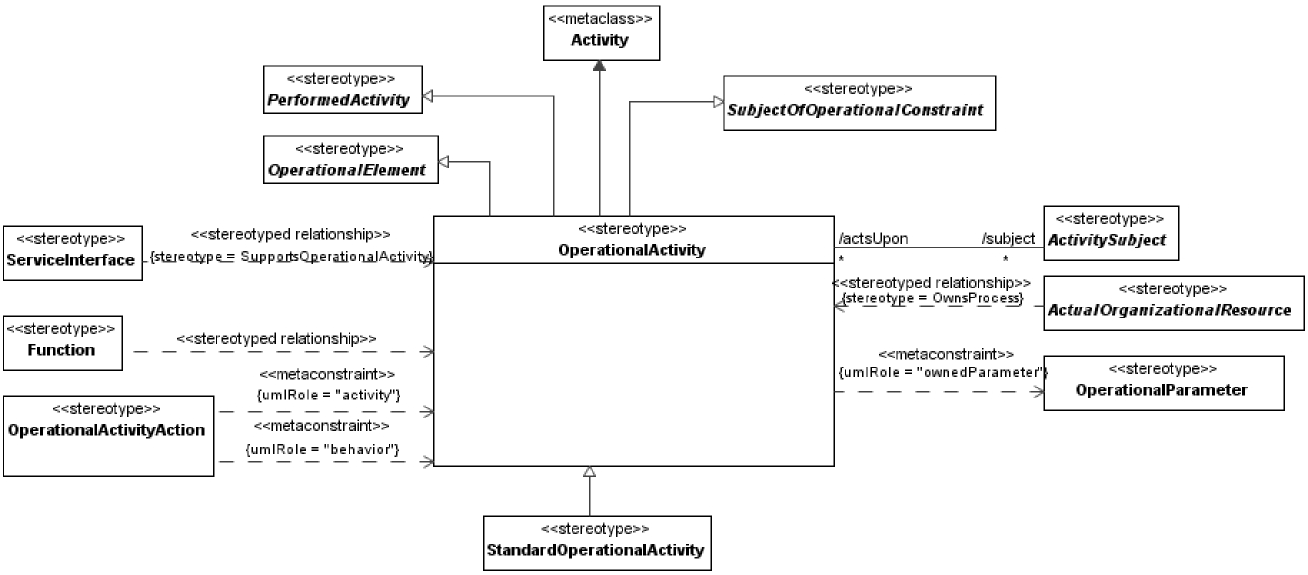


Figure 8.43 - Elements related to the OperationalActivity stereotype

Constraints

The following are constraints for OperationalActivity:

- OperationalActivity.ownedParameter - The values for the ownedParameter property must be stereotyped “OperationalActivityParameter» or its specializations.

Attributes

- The following are attributes of OperationalActivity:
  - / subject : ActivitySubject[*] - A list of all the ActivitySubjects that are acted/functioned upon by the OperationalActivity/Function. The list is derived by looking at all the ObjectNodes scoped within the OperationalActivity (directly or indirectly via StructuredActivityNodes, etc.) and getting a unique list of their types.

Extensions

The following are extensions for OperationalActivity:

- Activity

Generalizations

The following are generalization relationships for OperationalActivity:

- PerformedActivity
- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine

8.1.1.1.4.1.2 OperationalActivityAction

UPDM The OperationalActivityAction is defined as a call behavior action that invokes the activity that needs to be preformed.

MODAF: Used to relate an OperationalActivity to its sub-activities.

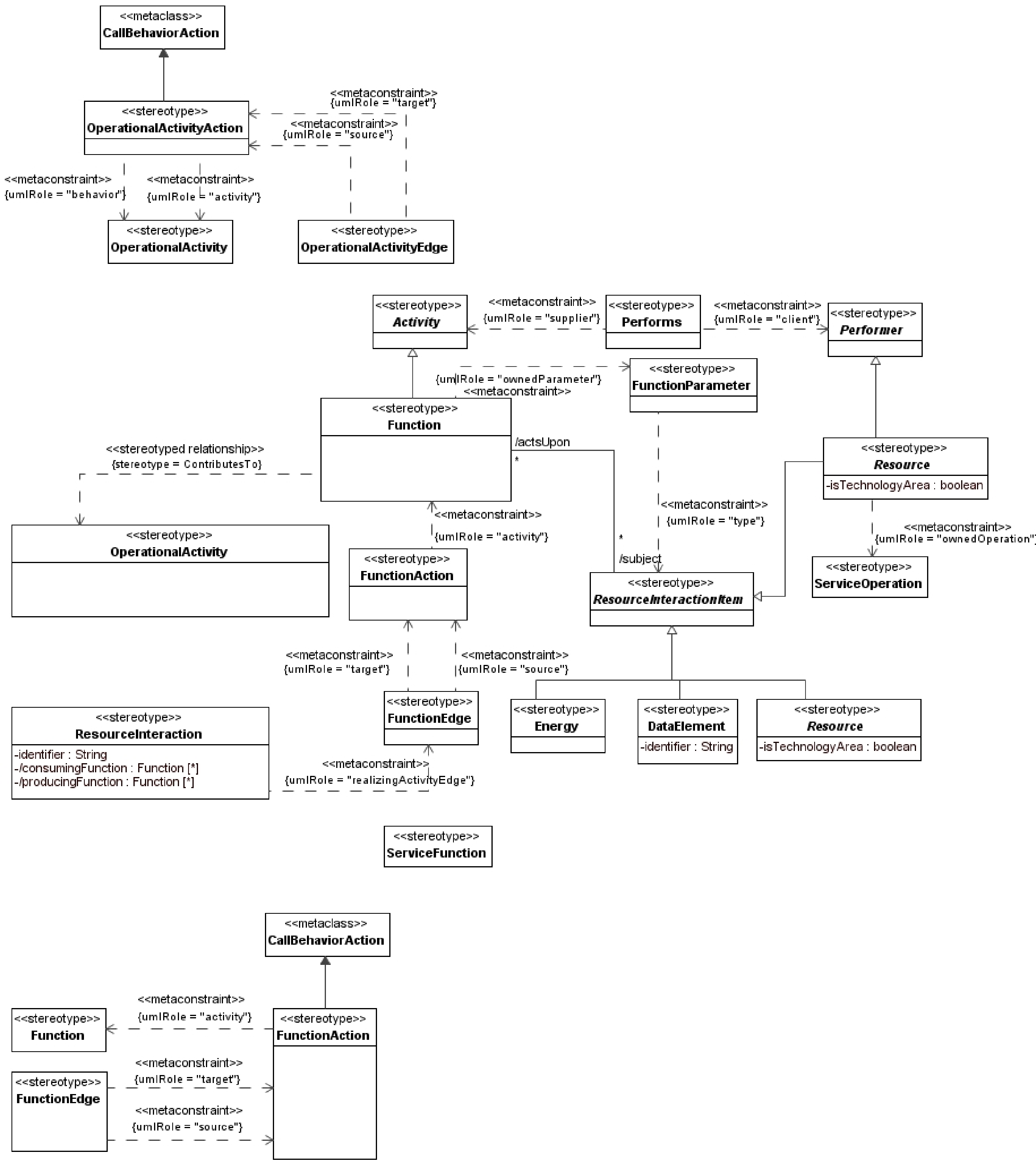


Figure 8.44 - Elements related to the OperationalActivityAction stereotype

Constraints

The following are constraints for OperationalActivityAction:

- OperationalActivityAction.behavior - Value for behavior property must be stereotyped «OperationalActivity» or its specializations.
- OperationalActivityAction.activity - Value for activity property must be stereotyped «OperationalActivity» or its specializations.

The following are constraints for FunctionAction:

- FunctionAction.activity - Value for the activity property must be stereotyped «Function».

Extensions

The following are extensions for OperationalActivityAction:

- CallBehaviorAction

Generalizations

The following are generalization relationships for OperationalActivityAction:

- UPDMElement

8.1.1.1.4.1.3 OperationalActivityEdge

UPDM An extension of «ActivityEdge» that is used to model the flow of control/objects through an OperationalActivity.

MODAF: An OperationalActivityEdge (MODAF::OperationalActivityFlow) is a flow of information, energy, or material from one activity to another.

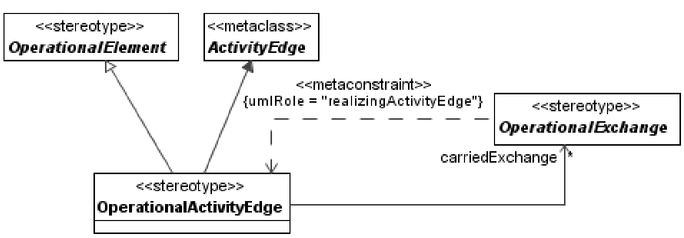


Figure 8.45 - Elements related to the OperationalActivityEdge stereotype

Attributes

The following are attributes of OperationalActivityEdge:

- carriedExchange : OperationalExchange[*] - Exchange carried by OperationalActivityEdge.

Constraints

The following are constraints for OperationalActivityEdge:

- OperationalActivityAction.source - Value for the source property must be stereotyped «OperationalActivityAction» or its specializations.
- OperationalActivityAction.target - Value for the target property must be stereotyped «OperationalActivityAction» or its specializations.
- OperationalActivityAction.target.owner - «OperationalActivityEdge» must be owned directly or indirectly by «OperationalActivity».

Extensions

The following are extensions for OperationalActivityEdge:

- ActivityEdge

Generalizations

The following are generalization relationships for OperationalActivityEdge:

- UPDMElement

8.1.1.1.4.1.4 OperationalParameter

UPDM: OperationalParameter represents inputs and outputs or OperationalActivity. It is typed by OperationalExchangeItem.

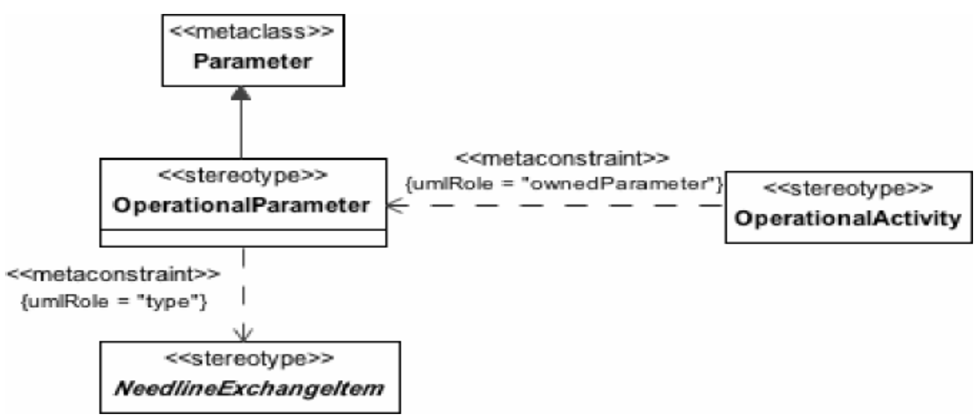


Figure 8.46 - Elements related to the OperationalParameter stereotype

Constraints

The following are constraints for OperationalParameter:

- OperationalParameter.type - Value for the type property must be stereotyped by specialization of «OperationalExchangeItem».

Extensions

The following are extensions for OperationalParameter:

- Parameter

Generalizations

The following are generalization relationships for OperationalParameter:

- UPDMElement

8.1.1.1.4.1.5 OperationalStateMachine

UPDM: A state machine describing an operational behavior or property.

MODAF: An OperationalStateMachine (MODAF::OperationalStateDescription) is a rule governing an operational behavior or property.

DoDAF: The Operational State Transition Description (OV-6b) DoDAF-described View is a graphical method of describing how an Operational Activity responds to various events by changing its state. The diagram represents the sets of events to which the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

Extensions

The following are extensions for OperationalStateMachine:

- StateMachine

Generalizations

The following are generalization relationships for OperationalStateMachine:

- UPDMElement

8.1.1.1.4.1.6 SubjectOfOperationalStateMachine

UPDM Abstract Element: The element being described by the state machine.

Note – SubjectOfOperationalStateMachine is abstract.

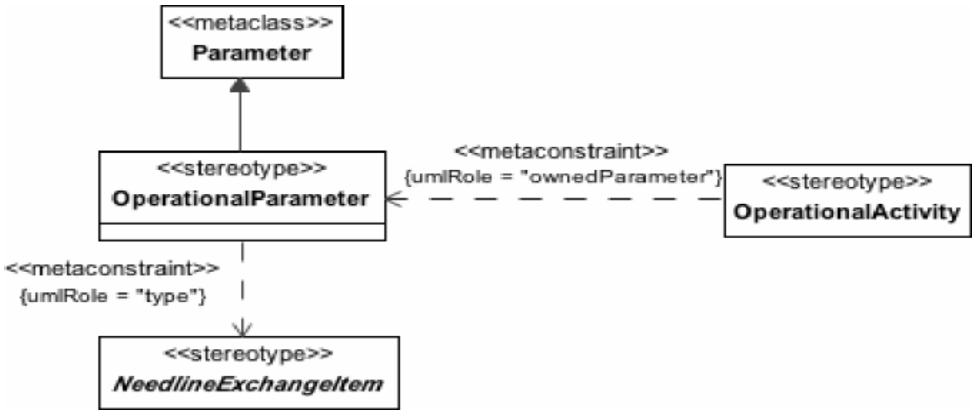


Figure 8.47 - Elements related to the abstract SubjectOfOperationalStateMachine stereotype

Constraints

The following are constraints for SubjectOfOperationalStateMachine:

- SubjectOfOperationalStateMachine.ownedBehavior - If elements, that have applied stereotypes that are specializations of “SubjectOfOperationalStateMachine” have StateMachines as owned behaviors, then those behaviors must be stereotyped «OperationalStateMachine» or its specializations.

Generalizations

The following are generalization relationships for SubjectOfOperationalStateMachine:

- UPDMElement

8.1.1.1.4.2 UPDM L1 ::UPDM L0::Core::OperationalElements::Data

The Data Profile is used to document the business information requirements and structural business process rules of the architecture. It describes the information that is associated with the information exchanges of the architecture. Included are information items, their attributes or characteristics, and their inter-relationships.

8.1.1.1.4.2.1 InformationElement

MODAF: A relationship specifying the need to exchange information between nodes.

DoDAF: NA - this is a specialization of OperationalExchange (DoDAF::Interface).

CADM: (4199/2) (A) A FORMALIZED REPRESENTATION OF DATA SUBJECT TO A FUNCTIONAL PROCESS

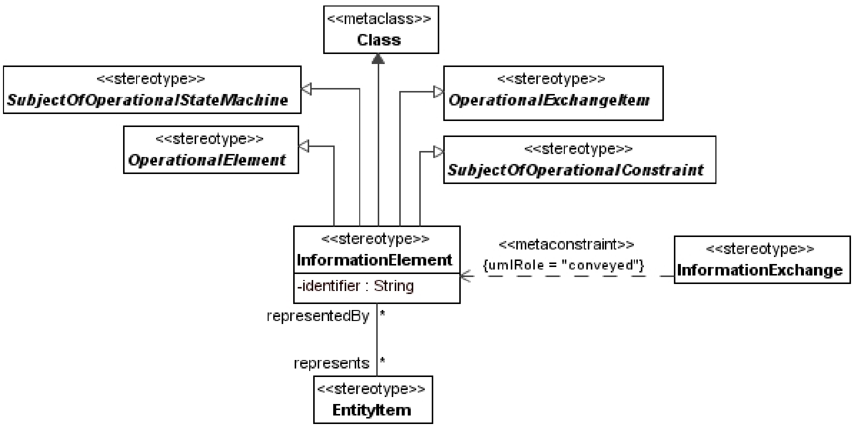


Figure 8.48 - Elements related to the InformationElement stereotype

Attributes

The following are attributes for InformationElement:

- Identifier : String - Identifier.
- Represents : EntityItem[*] – This list of EntityItems that are represented by the InformationElement.

Extensions

The following are extensions for InformationElement:

- Class

Generalizations

The following are generalization relationships for InformationElement:

- OperationalExchangeItem
- SubjectOfOperationalStateMachine
- SubjectOfOperationalConstraint
- OperationalElement

8.1.1.1.4.2.2 InformationElementProperties

Predefined additional DoDAF properties for InformationElement

Measurements

The following are measurements (Attributes) for InformationElementProperties:

- «measurement» accuracy : String [*] – Accuracy
- «measurement» language : String [*] – Language

- «measurement» scope : String [*] – Scope
- «measurement» content: String [*] - Content

8.1.1.1.4.2.3 «MeasurementSet» ExchangeProperties

Predefined additional DoDAF properties for Exchanges

Measurements

The following are measurements (Attributes) for ExchangeProperties:

- «measurement» accountability: String [*] – Accountability
- «measurement» interoperabilityLevelAchievable: String [*] – Interoperability Level Achievable
- «measurement» classification: String [*] – Classification
- «measurement» classificationCaveat: String [*] – Classification Caveat
- «measurement» criticality: String [*] – Criticality
- «measurement» periodicity: String [*] – Periodicity
- «measurement» protectionDuration: String [*] – Protection Duration
- «measurement» protectionSuspenseCalendarDate: String [*] – Protection Suspense Calendar Date
- «measurement» protectionTypeName: String [*] – Protection Type Name
- «measurement» timeliness: String [*] – Timeliness
- «measurement» transactionType: String [*] – Transaction Type
- «measurement» protectionDurationCode: String [*] – Protection Duration Code
- «measurement» releasability: String [*] – Releasability
- «measurement» size: String [*] – Size
- «measurement» throughput: String [*] – Throughput

8.1.1.1.4.2.4 «MeasurementSet» InformationAssuranceProperties

Predefined additional DoDAF properties for Exchanges.

Measurements

The following are measurements (Attributes) for ExchangeProperties:

- «measurement» accessControl: String [*] – Access Control
- «measurement» availability: String [*] – Availability
- «measurement» confidentiality: String [*] – Confidentiality
- «measurement» disseminationControl: String [*] – Dissemination Control
- «measurement» integrity: String [*] – Integrity

- «measurement» nonRepudiationConsumer: String [*] – Non Repudiation Consumer
- «measurement» nonRepudiationProducer : String [*] – Non Repudiation PRODUCER

8.1.1.1.4.2.5 «MeasurementSet» OperationalActivityProperties

Predefined additional DoDAF properties for Operational Activities.

Measurements

The following are measurements (Attributes) for OperationalActivityProperties:

- « measurement » cost: float [1] – Cost

8.1.1.1.4.2.6 «MeasurementSet» CommunicationsLinkProperties

Predefined additional DoDAF properties for Communications Links.

Measurements

The following are measurements (Attributes) for CommunicationsLinkProperties:

- « measurement » cost: float [1] – Cost

8.1.1.1.4.3 UPDM L1 ::UPDM L0::Core::OperationalElements::Flows

Section of the OperationalElements profile that describes flows exists or are required between Nodes such as flows of information, people, material, or energy.

8.1.1.1.4.3.1 Commands

MODAF: Asserts that one OrganizationalResource (source) commands another (target).

DoDAF: NA

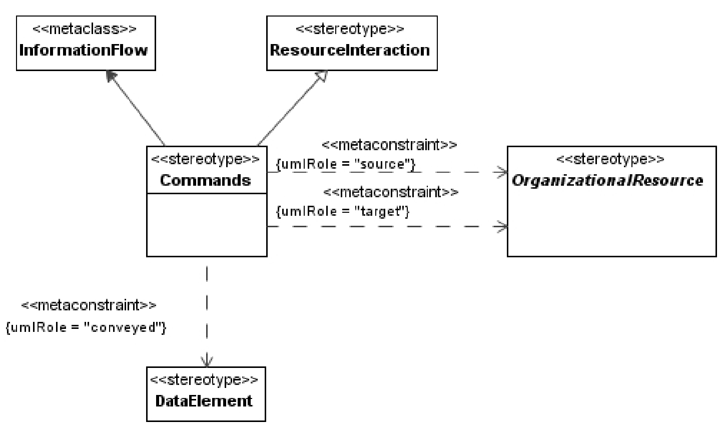


Figure 8.49 - Elements related to the Commands stereotype

Constraints

The following are constraints for Commands:

- Command.source - Value for the source property must be stereotyped «OrganizationalResource» or its specializations.
- Command.target - Value for the conveyed property must be stereotyped «DataElement» or its specializations.
- Command.conveyed - Value for the target property must be stereotyped «OrganizationalResource» or its specializations.

Extensions

The following are extensions for Commands:

- InformationFlow

Generalizations

The following are generalization relationships for Commands:

- ResourceInteraction

8.1.1.1.4.3.2 Information Exchange

MODAF: A relationship specifying the need to exchange information between nodes.

DoDAF: Interface: An overlap between Performers for the purpose of producing a Resource that is consumed by the other.

Note – additional information about the requirements for the InformationExchange may be provided by the requirementText attribute.

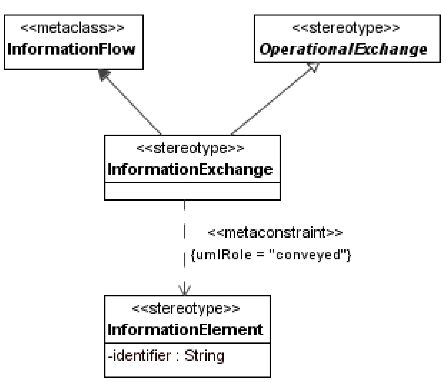


Figure 8.50 - Elements related to the InformationExchange stereotype

Constraints

The following are constraints for InformationExchange:

- InformationExchange.conveyed - Values for conveyed property must be stereotyped «InformationElement» or its specializations.

Extensions

The following are extensions for InformationExchange:

- InformationFlow

Generalizations

The following are generalization relationships for InformationExchange:

- OperationalExchange

8.1.1.1.4.3.3 OperationalExchange

UPDM Abstract element. An abstract utility element used as common ancestor for:

- InformationExchange
- OrganizationalExchange
- EnergyExchange
- MaterielExchange

An operational exchange is formed when an activity of one operational node consumes items produced by another activity of a different operational node.

An operational exchange describes the characteristics of the exchanged item, such as the content, format (voice, imagery, text and message format), throughput requirements, security or classification level, timeliness requirement, and the degree of interoperability.

MODAF: An OperationalExchange (MODAF::LogicalFlow) asserts that a flow exists or is required between Nodes (e.g., flows of information, people, material, or energy).

Note – OperationalExchange is abstract.

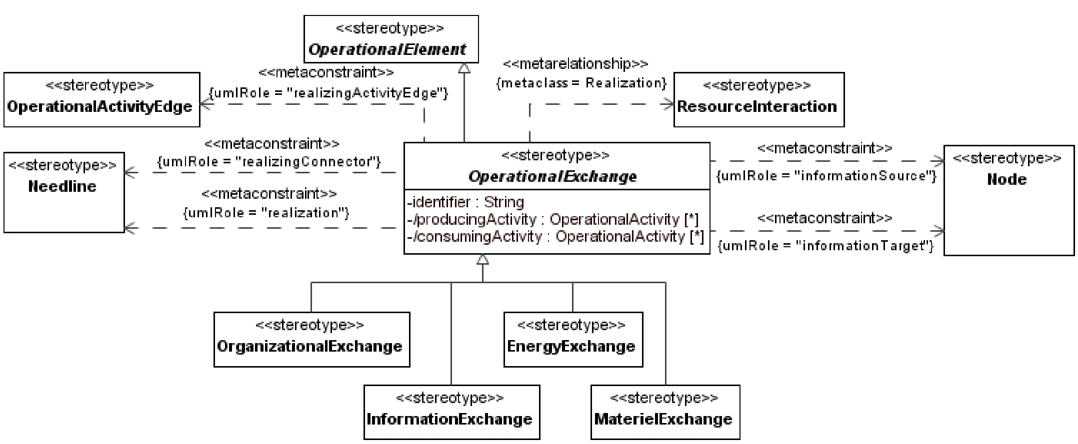


Figure 8.51 - Elements related to the abstract OperationalExchange stereotype

Attributes

The following are attributes for OperationalExchange:

- Identifier : String - Identifier.
- consumingActivity : OperationalActivity [*] – A list of OperationalActivites that consume an OperationalExchange via OperationalActivityEdge and their directly/indirectly connected OperationalActivityActions. This is derived as follows:
  - OperationalExchange to OperationalActivityEdge via the “realizingMessage” role.
  - OperationalActivityEdge to OperationActivityAction via the “source” role (if the source is a pin on an OperationalActivityAction, another jump is required from the pin to the action via an inverse navigation of the action’s “result” role).
  - OperationalActivityAction to OperationalActivity via the “behavior” role.
- producingActivity : OperationalActivity [*] – A list of OperationalActivites that produce an OperationalExchange via OperationalActivityEdges and their directly/indirectly connected OperationalActivityActions. This is derived as follows:
  - ·OperationalExchange to OperationalActivityEdge via the “realizingMessage” role.
  - ·OperationalActivityEdge to OperationalActivityAction via the “source” role (if the source is a pin on an OperationalActivityAction, another jump is required from the pin to the action via an inverse navigation of the action’s “result” role).
  - ·OperationalActivityAction to OperationalActivity via the “behavior” role.
- /producingActivity: OperationalActivity [*] - Producing Activities
- /consumingActivity: OperationalActivity [*] - Consuming Activities

Constraints

The following are constraints for OperationalExchangeNeedlineExchange:

- OperationalExchange.Realization - If there is a Realization going out of “NeedlineExchange” element, the realized element must be stereotyped «ResourceInteraction» or its specializations.
- OperationalExchange.realizingActivityEdge - Value for realizingActivityEdge property has to be stereotyped «OperationalActivityEdge» or its specializations.
- OperationalExchange.realization/realizingConnector - Value for realization or realizingConnector property has to be stereotyped «Needline» or its specializations.

Extensions

The following are extensions for NeedlineExchange:

- InformationFlow

Generalizations

The following are generalization relationships for OperationalExchange:

- UPDMElement

8.1.1.1.4.3.4 OperationalExchangeItem

An abstract utility element used as common ancestor for:

- InformationElement
- ResourceArtifact
- Energy
- OrganizationalResource

Note – OperationalExchangeItem is abstract.

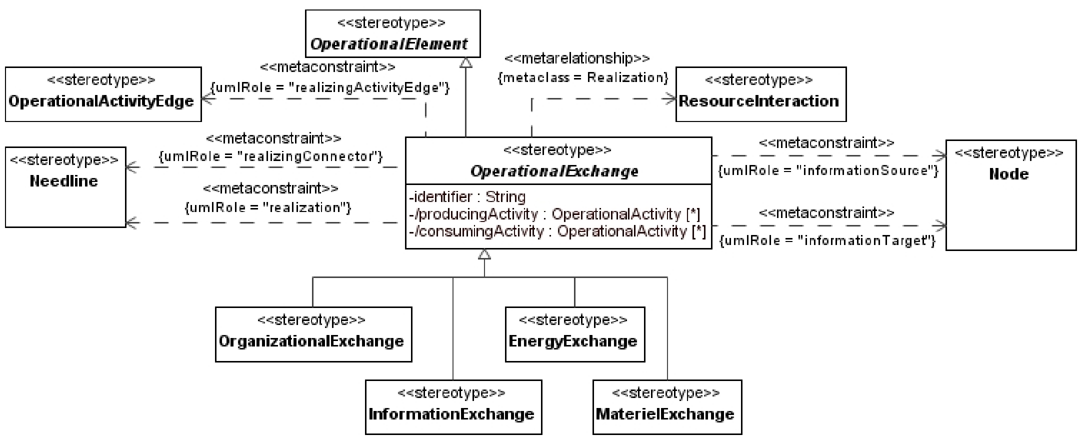


Figure 8.52 - Elements related to the abstract OperationalExchangeItem stereotype

Constraints

- OperationalExchange.Realization - If there is a Realization going out of «OperationalExchange» element, the realized element must be stereotyped «ResourceInteraction» or its specializations.
- OperationalExchange.realizingActivityEdge - Value for realizingActivityEdge property has to be stereotyped «OperationalActivityEdge» or its specializations.
- OperationalExchange.realization/realizingConnector - Value for realization or realizingConnector property has to be stereotyped «Needline» or its specializations.
- OperationalExchange.informationSource - Value for informationSource property has to be stereotyped «Node» or its specializations.
- OperationalExchange.informationTarget - Value for informationTarget property has to be stereotyped «Node» or its specializations.

Generalizations

The following are generalization relationships for OperationalExchangeItem:

- ActivitySubject

8.1.1.1.4.4 UPDM L1 ::UPDM L0::Core::OperationalElements::Structure

Section of the OperationalElements profile that describe stuctural concepts.

8.1.1.1.4.4.1 ArbitraryRelationship

UPDM: Represents a visual indication of a connection used in high level operational concept diagrams. The connections are purely visual and cannot be related to any architectural semantics.

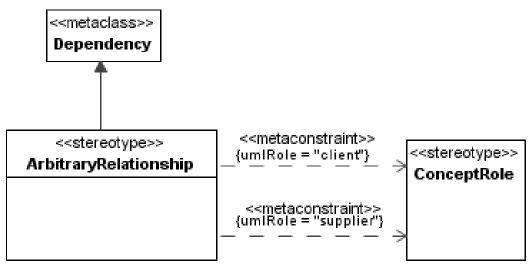


Figure 8.53 - Elements related to the ArbitraryRelationshipConnector stereotype

Constraints

The following are constraints for ArbitraryRelationship:

- ArbitraryRelationship.client - The value for client property has to be stereotyped «ConceptRole» or its specializations.
- ArbitraryRelationship.supplier - The value for supplier property has to be stereotyped «ConceptRole» or its specializations.

Extensions

The following are extensions for ArbitraryRelationship:

- ConnectorDependency

Generalizations

The following are generalization relationships for ArbitraryRelationship:

- UPDMElement

8.1.1.1.4.4.2 CompatibleWith

Relates a node to a location to assert that the operational node is required to be situated at that location.

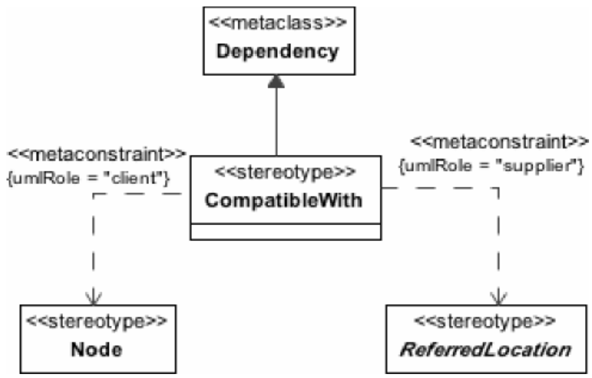


Figure 8.54 - Elements related to the CompatibleWith stereotype

Constraints

The following are constraints for CompatibleWith:

- CompatibleWith.supplier - Value for the client property must be stereotyped «Node» or its specializations.
- CompatibleWith.client - Value for the client property must be stereotyped «ReferredLocation» or its specializations.

Extensions

The following are extensions for CompatibleWith:

- Dependency

Generalizations

The following are generalization relationships for CompatibleWith:

- UPDMElement

8.1.1.1.4.4.3 Competence

MODAF: A specific set of abilities defined by knowledge, skills, and attitude.

DoDAF: (DoDAF::Skill): The ability, coming from one's knowledge, practice, aptitude to do something well.

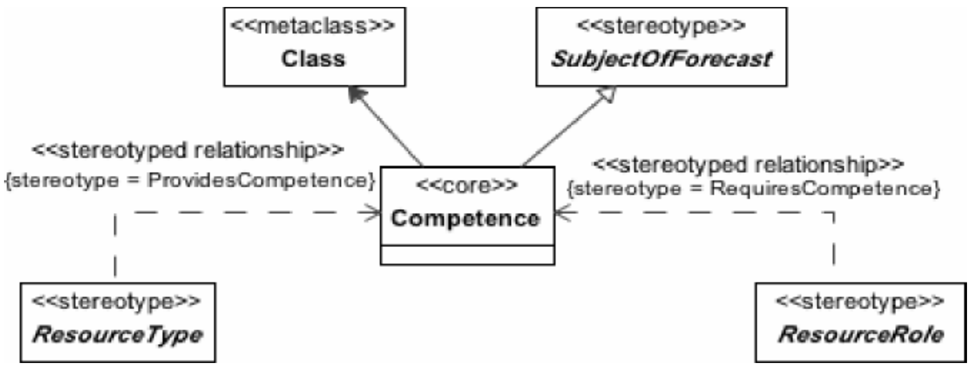


Figure 8.55 - Elements related to the Competence stereotype

Extensions

The following are extensions for Competence:

- Class

Generalizations

The following are generalization relationships for Competence:

- SubjectOfForecast

8.1.1.1.4.4.4 ConceptItem

UPDM: An item that may feature in a high level operational concept.

Note – ConceptItem is abstract.

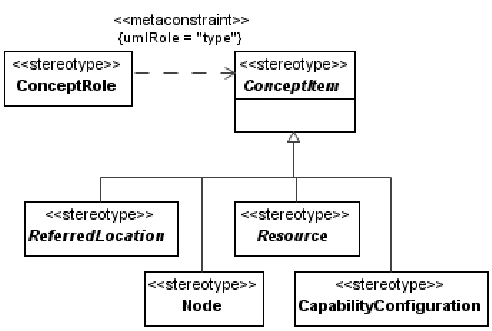


Figure 8.56 - Elements related to the abstract ConceptItem stereotype

Generalizations

The following are generalization relationships for ConceptItem:

- UPDMElement

8.1.1.1.4.4.5 HighLevelOperationalConcept

MODAF: A generalized model for operations.

DoDAF: NA

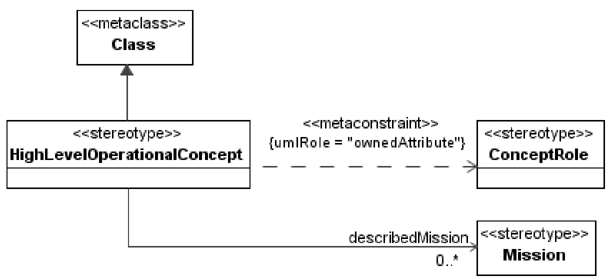


Figure 8.57 - Elements related to the HighLevelOperationalConcept stereotype

Constraints

The following are constraints for HighLevelOperationalConcept:

- HighLevelOperationalConcept.ownedAttribute - The values for the ownedAttribute properties must be stereotyped with specializations of the “ItemInConcept.”

Attributes

The following are attributes of HighLevelOperationalConcept:

- describedMission : Mission[0 ..*] - Mission that is described by this HighLevelOperationalConcept.

Extensions

The following are extensions for HighLevelOperationalConcept:

- Class

Generalizations

The following are generalization relationships for HighLevelOperationalConcept:

- UPDMElement

8.1.1.1.4.4.6 ConceptRole

UPDM: A relationship which asserts that a ConceptItem forms part of the high level operational concept.

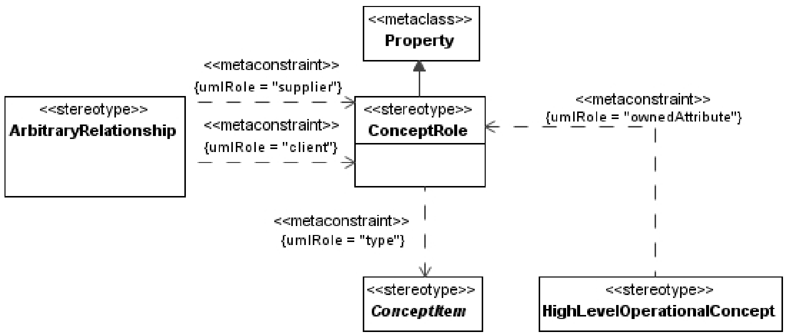


Figure 8.58 - Elements related to the abstract ConceptRole stereotype

Constraints

The following are constraints for ConceptRole:

- ConceptRole.type - Value for the type property must be steretotyped «ConceptItem» or its specializations.

Extensions

The following are extensions for ConceptRole:

- Property

Generalizations

The following are generalization relationships for ConceptRole:

- UPDMElement

8.1.1.1.4.4.7 KnownResource

MODAF: Asserts that a known Resource plays a part in the architecture.

DoDAF: NA – covered by the more general temporalWholePart element.

Note – An OV-2 is meant to show logical interactions between nodes. However, sometimes it is known.

In MODAF you should limit these to only being owned by LogicalArchitectures.

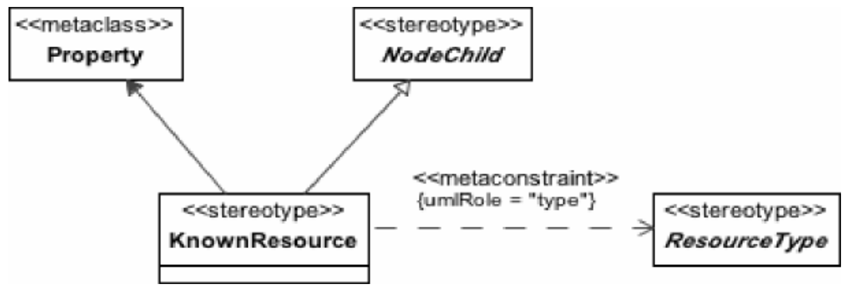


Figure 8.59 - Elements related to the KnownResource stereotype

Constraints

The following are constraints for KnownResource:

- KnownResource.type - Values for type property have to be stereotyped «ResourceType» or its specializations.

Extensions

The following are extensions for KnownResource:

- Property

Generalizations

The following are generalization relationships for KnownResource:

- NodeChild

8.1.1.1.4.4.8 LogicalArchitecture

MODAF: A CompositeStructureModel whose parts are either NodeRoles (MODAF::Node), ProblemDomains, or KnownResources.

DoDAF: NA

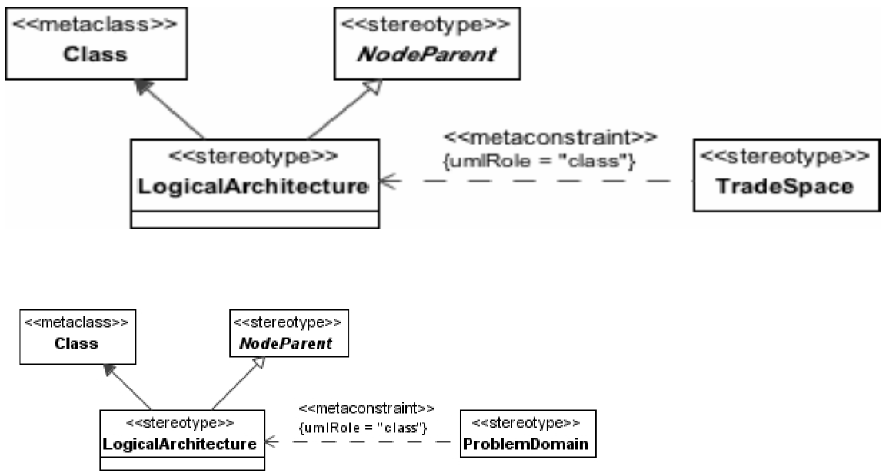


Figure 8.60 - Elements related to the LogicalArchitecture stereotype

Extensions

The following are extensions for LogicalArchitecture:

- Class

Generalizations

The following are generalization relationships for LogicalArchitecture:

- NodeParent

8.1.1.1.4.4.9 Needline

MODAF: NA

DoDAF: A needline documents the requirement to exchange information between nodes. The needline does not indicate how the information transfer is implemented.

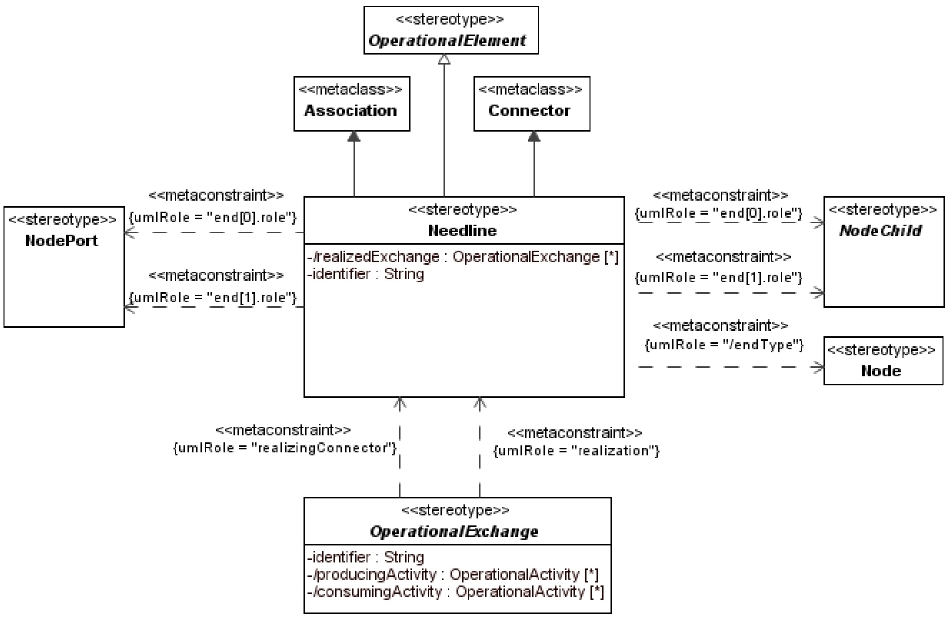


Figure 8.61 - Elements related to the Needline stereotype

Constraints

The following are constraints for Needline:

- ResourceInterface.end - In case of extending Association: the value for endType property has to be stereotyped «ResourceType» or its specializations. In case of extending Connector: the value for the role property for the owned ConnectorEnd must be stereotyped «ResourceRole» or its specializations.
- Needline.end - In case of extending Association: the value for endType property has to be stereotyped «Node» or its specializations. In case of extending Connector: the value for the role property for the owned ConnectorEnd must be stereotyped «NodeChild»/«NodePort» or its specializations.

Attributes

The following are attributes of Needline:

- realizedExchange : NeedlineExchange[*] - A list of NeedlineExchanges (or specializations) that are realized by the Needline. This is derived by navigating from the Needline to the NeedlineExchanges using the inverse of the realization/realizingConnector roles.
- / exchangedItem : NeedlineExchangeOperationalExchange[*] - A list of OperationalExchanges that are realized by the Needline. This is derived by doing the inverse of the UML “realizingConnector” and “realization” roles, that are owned by InformationFlow.
- / identifier : String - Identifier - This is a derived property that has to be filled in at the moment when this Needline is assigned as value realizingConnector (in case of Connector) or realization (in case of Association) property for Needline exchange.

Extensions

The following are extensions for Needline:

- Connector
- Association

Generalizations

The following are generalization relationships for Needline:

- UPDMElement

8.1.1.1.4.4.10 Node

MODAF: A Node (MODAF::NodeType) is a logical entity that performs operational activities.

Note – Nodes are specified independently of any physical realization.

DoDAF: A Node (DoDAF::OperationalNode) is an element of the operational architecture that produces, consumes, or processes information.

Note – This is also a specialization of Performer.

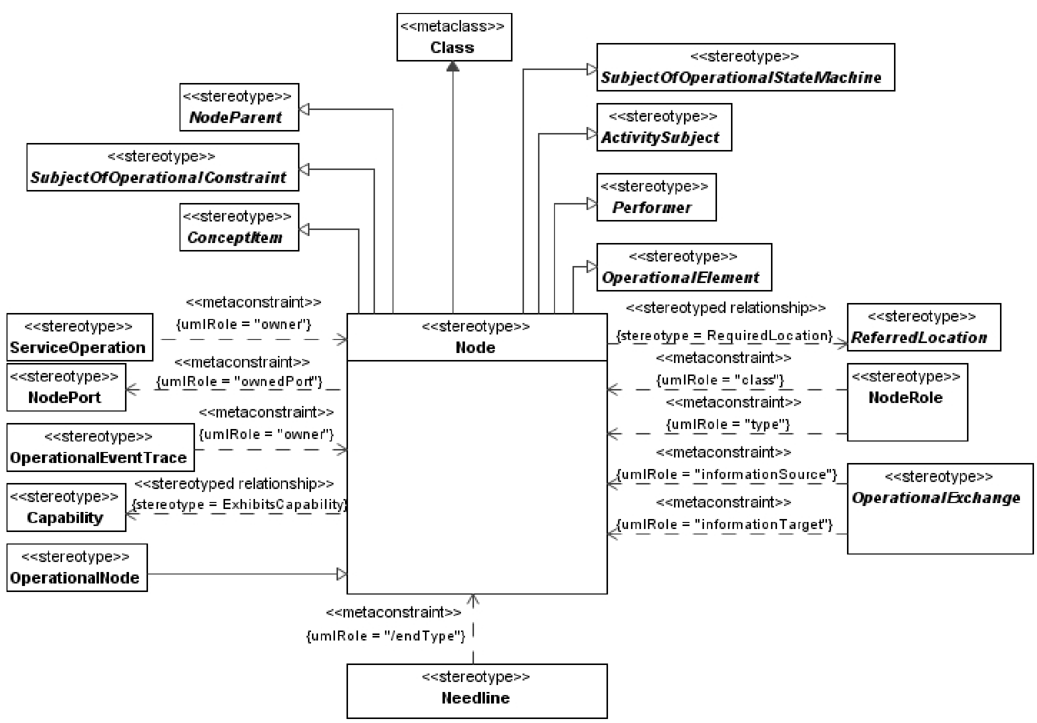


Figure 8.62 - Elements related to the Node stereotype

Constraints

The following are constraints for Node:

- Node.performs - Can perform only “OperationalActivity” elements or its specializations.
- Node.ownedPorts - Values for the ownedPort property must be stereotyped «NodePort» or its specializations.

Extensions

The following are extensions for Node:

- Class

Generalizations

The following are generalization relationships for Node:

- Performer
- ActivitySubject
- ConceptItem
- SubjectOfOperationalConstraint
- NodeParent
- SubjectOfOperationalStateMachine

8.1.1.1.4.4.11 NodeChild

UPDM Artifact: An abstract element used for supporting the composite structuring of operational elements such as Nodes and LogicalArchitectures.

Note – NodeChild is abstract.

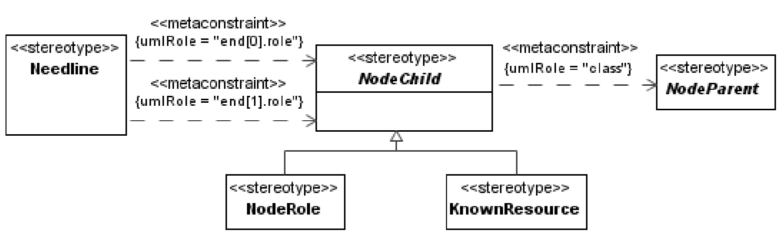


Figure 8.63 - Elements related to the abstract NodeChild stereotype

Constraints

The following are constraints for NodeChild:

- NodeChild.class - Class property value must be stereotyped «NodeParent» or its specializations.

Generalizations

The following are generalization relationships for NodeChild:

- UPDMElement

8.1.1.1.4.4.12 NodeParent

UPDM: An abstract element representing the owners/context of composite structure at the operational level.

Note – NodeParent is abstract.

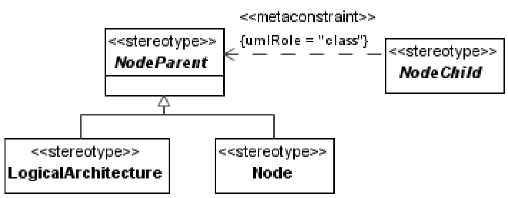


Figure 8.64 - Elements related to the abstract NodeParent stereotype

Generalizations

The following are generalization relationships for NodeParent:

- UPDMElement

8.1.1.1.4.4.13 NodePort

UPDM: A port is a property of a Node that specifies a distinct interaction point between the node and its environment or between the (behavior of the) node and its internal parts. It is the “entry/exit” point where resources (e.g., energy, information/data and people) flow in and out of a node.

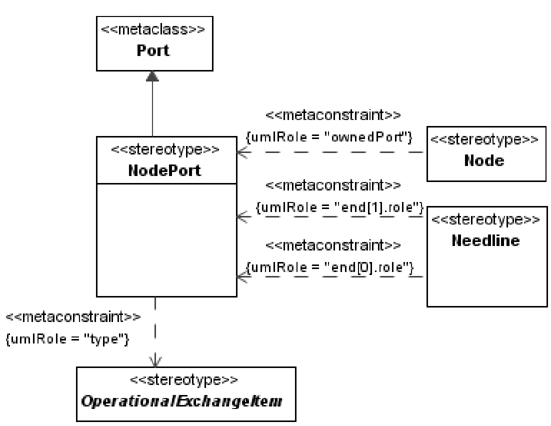


Figure 8.65 - Elements related to the NodePort stereotype

Constraints

The following are constraints for NodePort:

- NodePort.type - Value for the type property must be stereotyped specialization of “NeedlineExchangeItem.”

Extensions

The following are extensions for NodePort:

- Port

Generalizations

The following are generalization relationships for NodePort:

- UPDMElement

8.1.1.1.4.4.14 NodeRole

MODAF: A NodeRole (MODAF::Node) is used to link a parent Node to its sub-nodes.
DoDAF: NA

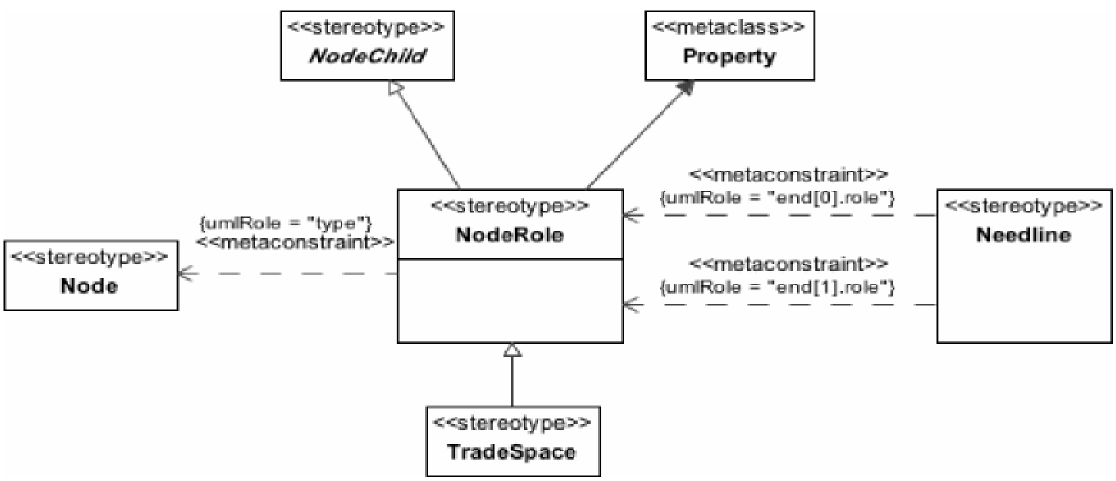


Figure 8.66 - Elements related to the NodeRole stereotype

Constraints

The following are constraints for NodeRole:

- NodeRole.class - Value for class metaproperty must be stereotyped «Node» or its specializations.
- NodeRole.type - Value for type metaproperty must be stereotyped «Node» or its specializations.

Extensions

The following are extensions for NodeRole:

- Property

Generalizations

The following are generalization relationships for NodeRole:

- UPDMElement
- NodeChild

8.1.1.1.4.4.15 OperationalConstraint

UPDM: An abstract Class that is extended by OperationalConstraint (A rule governing an operational behavior or property) and ResourceConstraint.

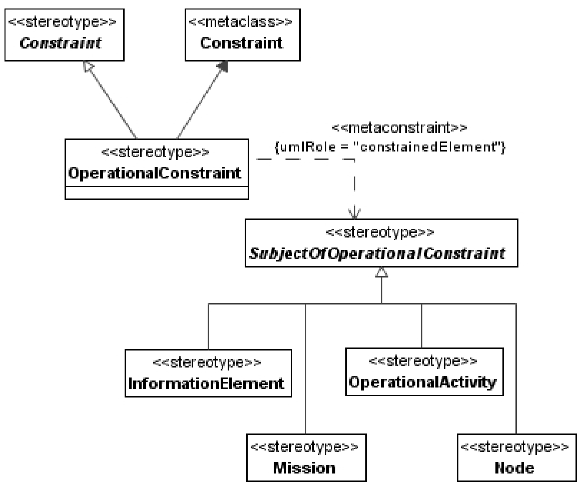


Figure 8.67 - Elements related to the OperationalConstraint stereotype

Constraints

The following are constraints for OperationalConstraint:

- OperationalConstraint.constrainedElement - Value for the constrainedElement property must be stereotyped by any specialization of “SubjectOfOperationalConstraint.”

Extensions

The following are extensions for OperationalConstraint:

- Constraint

Generalizations

The following are generalization relationships for ResourceConstraint:

- Constraint
- UPDMElement

8.1.1.1.4.4.16 SubjectOfOperationalConstraint

MODAF: Abstract. An element of the architecture that may be subject to an OperationalConstraint or OperationalStateDescription.

Note – SubjectOfOperationalConstraint is abstract.

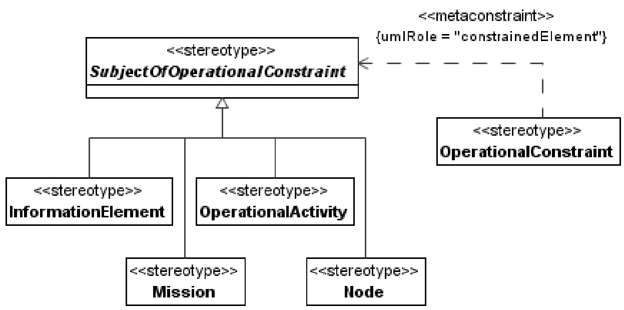


Figure 8.68 - Elements related to the abstract SubjectOfOperationalConstraint stereotype

Generalizations

The following are generalization relationships for SubjectOfOperationalConstraint:

- UPDMElement

8.1.1.1.4.4.17 UPDM L1: :UPDM L0: :Core: :OperationalElements: :Structure::Organizational

The organizational elements of the operational structure. Actual elements in the organizational part of the structural part of the Operational profile.

8.1.1.1.4.4.17.1 UPDM L1: :UPDM L0: :Core: :OperationalElements: :Structure::Organizational::Actual

Actual elements in the organizational part of the structural part of the Operational profile.

8.1.1.1.4.4.17.1.1 ActualOrganization

MODAF: An actual specific organization, an instance of an organization class (e.g., “The US Department of Defense”).

DoDAF: [DoDAF::Organization]: A specific real-world assemblage of people and other resources organized for an ongoing purpose.

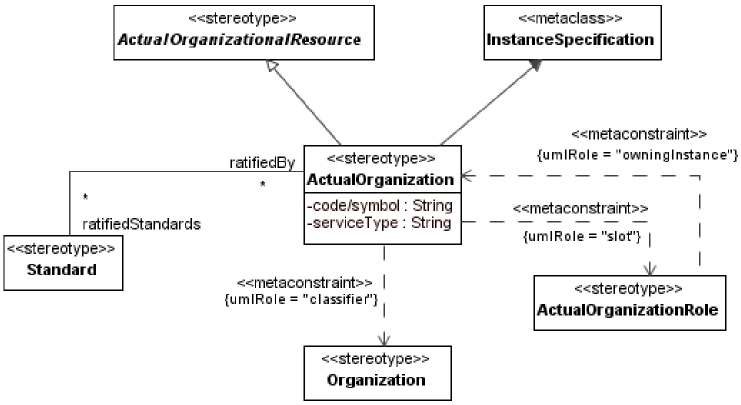


Figure 8.69 - Elements related to the ActualOrganization stereotype

Constraints

The following are constraints for ActualOrganization:

- ActualOrganization.classifier - Classifier property value must be stereotyped «Organization» or its specializations.
- ActualOrganization.classifier - Slot property value must be stereotyped «ActualOrganizationRole» or its specializations.
- code/symbol : String - Army, Navy, Air Force, Marine Corps, Joint
- serviceType: String - Service office code or symbol

Attributes

The following are attributes of ActualOrganization:

- ratifiedStandards : Standard[*] - Standards that were ratified by this ActualOrganization.

Extensions

The following are extensions for ActualOrganization:

- InstanceSpecification

Generalizations

The following are generalization relationships for ActualOrganization:

- ActualOrganizationalResource

8.1.1.1.4.4.17.1.2 ActualOrganizationalResource

UPDM: An ActualOrganization or an ActualPost.

MODAF: NA

DoDAF: NA

Note – ActualOrganizationalResource is abstract.

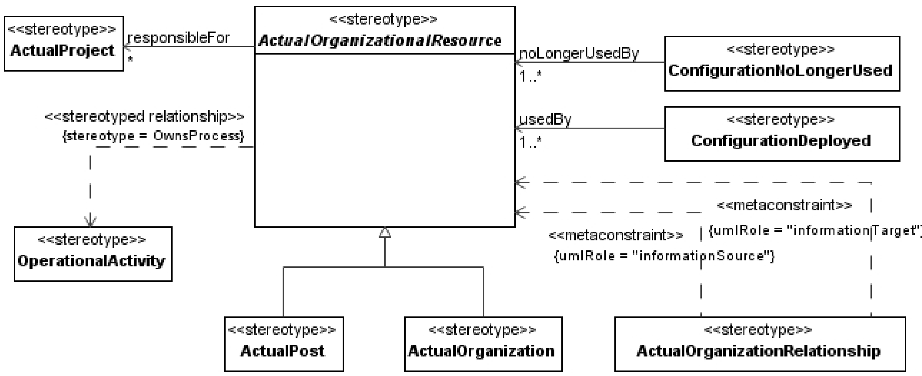


Figure 8.70 - Elements related to the abstract ActualOrganizationalResource stereotype

Attributes

The following are attributes of ActualOrganizationalResource:

- responsibleFor : Project[*] - Project that this ActualOrganizationalResource is responsible for.

Generalizations

The following are generalization relationships for ActualOrganizationalResource:

- UPDMElement

8.1.1.1.4.4.17.1.3 ActualOrganizationRole

UPDM: Relates an actual specific organization to an actual specific organizational resource that fulfills a role in that organization.

MODAF: NA

DoDAF: NA

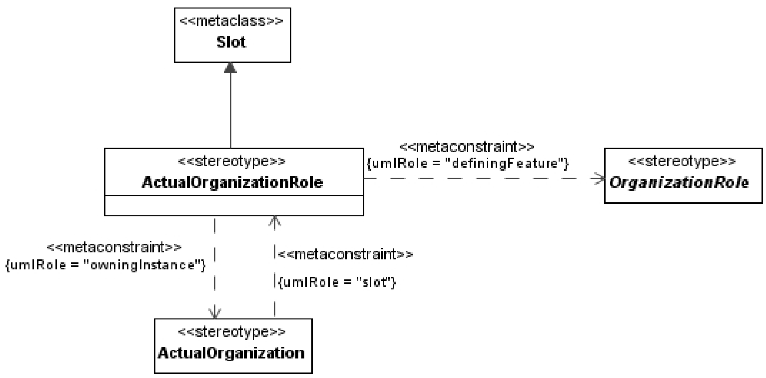


Figure 8.71 - Elements related to the ActualOrganizationRole stereotype

Constraints

The following are constraints for ActualOrganizationRole:

- ActualOrganizationRole.definingFeature - Value for owningInstance property has to be stereotyped «ActualOrganization» or its specializations.
- ActualOrganizationRole.owningInstance - Value for definingFeature property has to be stereotyped «OrganizationPart» or its specializations.

Extensions

The following are extensions for ActualOrganizationRole:

- Slot

Generalizations

The following are generalization relationships for ActualOrganizationRole:

- UPDMElement

8.1.1.1.4.4.17.1.4 ActualOrganizationRelationship

UPDM: A relationship between two ActualOrganizationResources.

MODAF: NA

DoDAF: NA

Note – The "TypicalOrganizationRelationship" that is realized by the "ActualOrganizationRelationship" is referred to via the typicalRelationship attribute.

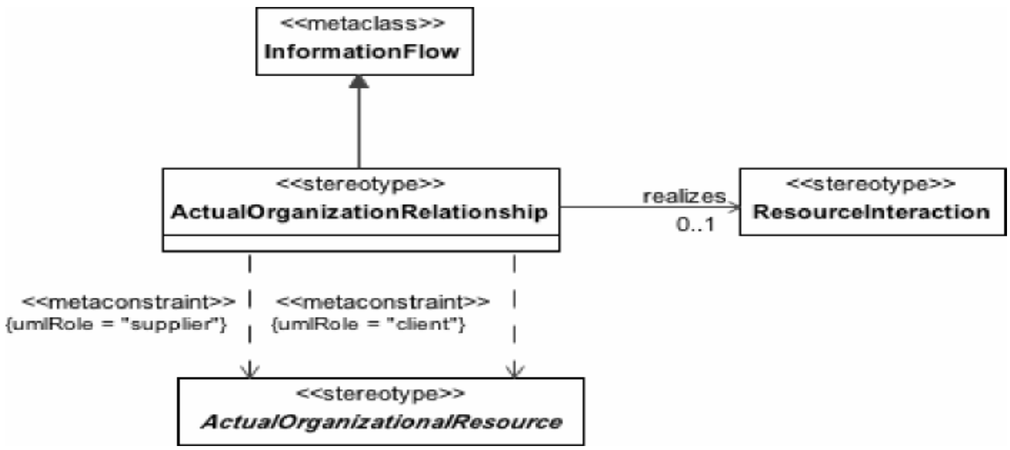


Figure 8.72 - Elements related to the ActualOrganizationRelationship stereotype

Constraints

The following are constraints for ActualOrganizationRelationship:

- ActualOrganizationRelationship.informationSource - Value for informationSource metaproperty must be stereotyped «ActualOrganizationalResource» or its specializations.
- ActualOrganizationRelationship.Target - Value for Target metaproperty must be stereotyped «ActualOrganizationalResource» or its specializations.

Extensions

The following are extensions for ActualOrganizationRelationship:

- InformationFlow

Generalizations

The following are generalization relationships for ActualOrganizationRelationship:

- UPDMElement

8.1.1.1.4.4.17.1.5 ActualPost

UPDM: An actual, specific post, an instance of a PostType class - e.g., “President of the United States of America.”

MODAF: NA

DoDAF: NA

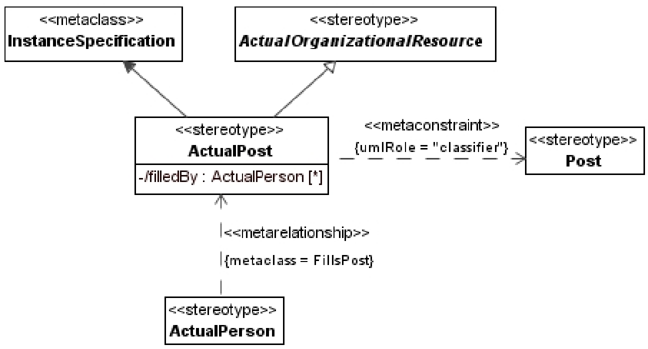


Figure 8.73 - Elements related to the ActualPost stereotype

Attributes

The following are attributes for ActualPost:

- /filledBy: ActualPerson[*] – Persons filling this ActualPost. This is derived as follows:
    - ActualPost to FillsPost via the “supplierDependency” role
    - FillsPost to ActualPerson via the “client” role

Constraints

The following are constraints for ActualPost:

- ActualPost.classifier - Classifier property value must be stereotyped «Post» or its specializations.

Extensions

The following are extensions for ActualPost:

- InstanceSpecification

Generalizations

The following are generalization relationships for ActualPost:

- ActualOrganizationalResource

8.1.1.1.4.4.17.1.6 OwnsProcess

MODAF: The OwnsProcess (MODAF::ProcessOwner) relationship asserts that an ActualOrganizationalResource has responsibility for an OperationalActivity. Note this does not imply the resource conducts the activity, merely that it has managerial responsibility for it.

DoDAF: NA

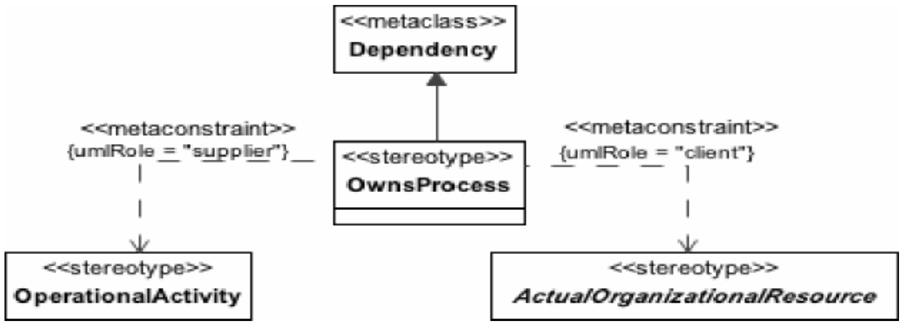


Figure 8.74 - Elements related to the OwnsProcess stereotype

Constraints

The following are constraints for OwnsProcess:

- OwnsProcess.supplier - Value for the supplier property has to be stereotyped «OperationalActivity» or its specializations.
- OwnsProcess.client - Value for the client property has to be stereotyped «ActualOrganizationalResource» or its specializations.

Extensions

The following are extensions for OwnsProcess:

- Dependency

Generalizations

The following are generalization relationships for OwnsProcess:

- UPDMElement

8.1.1.1.4.4.17.1.7 PostRole

MODAF: A PostRole (MODAF::Post) asserts that a post exists in an Organization (MODAF::OrganizationType) of the type specified by the related Post (MODAF::PostType).

DoDAF: NA – covered by the more general temporalWholePart element.

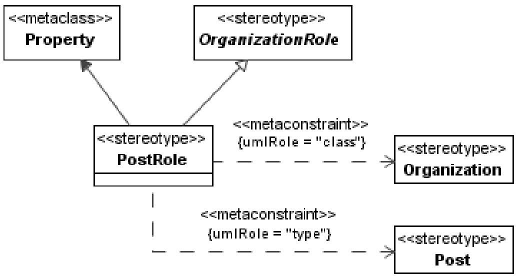


Figure 8.75 - Elements related to the PostRole stereotype

Constraints

The following are constraints for PostRole:

- PostRole.class - Value for the class property must be stereotyped «Organization» or its specializations.
- PostRole.type - Value for the type property must be stereotyped «Post» or its specializations.

Extensions

The following are extensions for PostRole:

- Property

Generalizations

The following are generalization relationships for PostRole:

- OrganizationPart

8.1.1.1.4.4.17.1.8 SubOrganization

MODAF: Asserts that one type of organization is typically the parent of another - e.g., a squadron may be part of a battalion.

DoDAF: NA

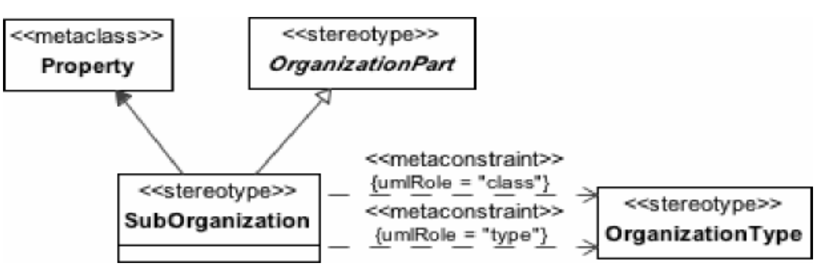


Figure 8.76 - Elements related to the SubOrganization stereotype

Constraints

The following are constraints for SubOrganization:

- SubOrganization.type - Value for the type property must be stereotyped «Organization» or its specializations.
- SubOrganization.class - Value for the class property must be stereotyped «Organization» or its specializations.

Extensions

The following are extensions for SubOrganization:

- Property

Generalizations

The following are generalization relationships for SubOrganization:

- OrganizationPart

Typical elements in the organizational part of the structural part of the Operational profile.

8.1.1.1.4.4.17.1.9 ActualPerson

UPDM: Named individual that fulfills an ActualPost. An individual human being (vs. Person which is a type), that is recognized by law as the subject of rights and duties.

MODAF: NA

DoDAF: NA

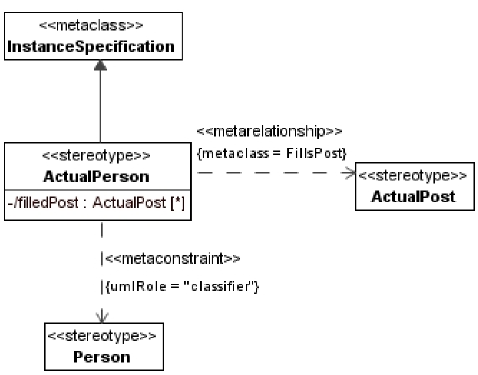


Figure 8.77 - Elements related to the ActualPerson stereotype

Attributes

The following are attributes for ActualPerson:

- /fillsPost: ActualPost[*] - ActualPosts filled by this person This is derived as follows:
  - ActualPerson to FillsPost via the “clientDependency” role.
  - FillsPost to ActualPost via the “supplier” role.

Constraints

The following are constraints for ActualPerson:

- ActualPerson.classifier - Value for the classifierproperty has to be stereotyped «Person» or its specializations.

Extensions

The following are extensions for ActualPerson:

- InstanceSpecification

8.1.1.1.4.4.17.1.10 Person

UPDM: A type of a human being that is recognized by law as the subject of rights and duties. This is used to define the characteristics that require capturing for ActualPersons (e.g., properties such as address, rank, telephone number).

MODAF: NA

DoDAF: NA

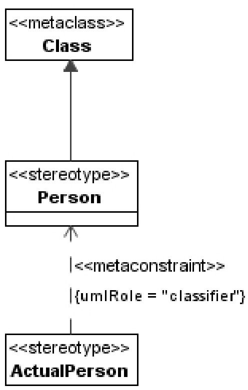


Figure 8.78 - Elements related to the Person stereotype

Extensions

The following are extensions for Person:

- Class

8.1.1.1.4.4.17.1.11 FillsPost

UPDM: Asserts that ActualPerson fills an ActualPost.

MODAF: NA

DoDAF: NA

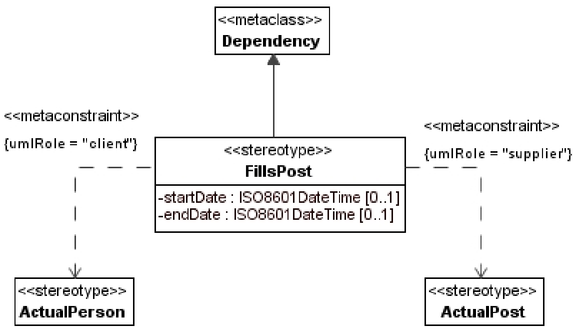


Figure 8.79 - Elements related to the FillsPost stereotype

Attributes

The following are attributes for FillsPost:

- startDate: ISO8601DateTime[0..1] - Start date
- endDate: ISO8601DateTime[0..1] - End date

Constraints

The following are constraints for FillsPost:

- FillsPost.client - Value for the client property must be stereotyped by «ActualPerson» or its specializations.
- FillsPost.supplier - Value for the supplier property must be stereotyped by «ActualPost» or its specializations.

Extensions

The following are extensions for FillsPost:

- Dependency

8.1.1.1.4.4.17.2 UPDM L1 ::UPDM L0: :Core: :OperationalElements::Structure::Organizational::Typical

Typical elements in the organizational part of the structural part of the Operational profile.

8.1.1.1.4.4.17.2.1 OrganizationalResource

UPDM An abstract element that represents Organizations and Posts.
MODAF: Either an organization, or a post.

Note – OrganizationalResource is abstract.

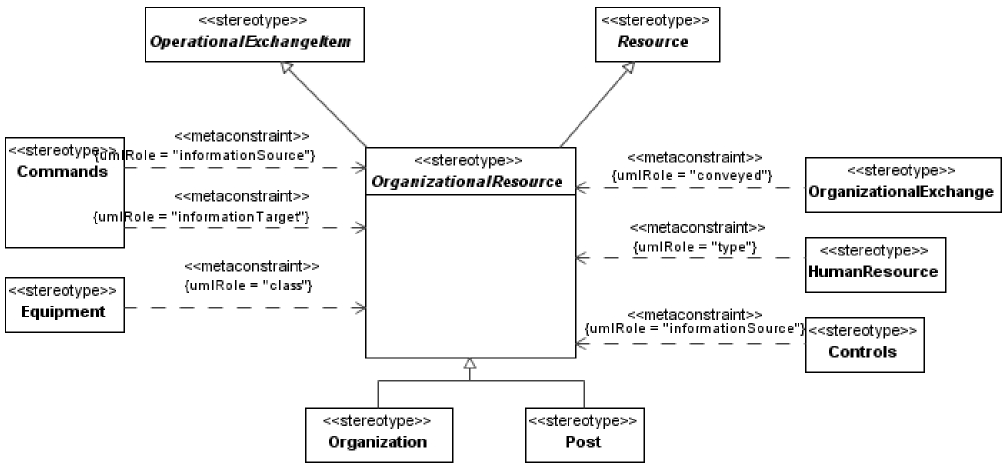


Figure 8.80 - Elements related to the abstract OrganizationalResource stereotype

Extensions

The following are extensions for OrganizationalResource:

- Class

Generalizations

The following are generalization relationships for OrganizationalResource:

- OperationalExchangeItem
- ResourceType

8.1.1.1.4.4.17.2.2 OrganizationRole

UPDM - An abstract element that's used to represent properties in an Organization that are typed by another Organization or a Post.

Note – OrganizationRole is abstract.

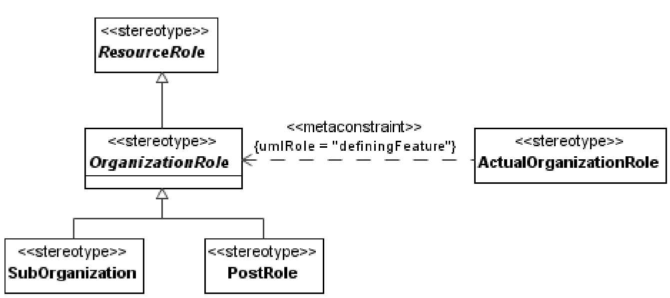


Figure 8.81 - Elements related to the abstract OrganizationRole stereotype

Generalizations

The following are generalization relationships for OrganizationPart:

- ResourceRole

8.1.1.1.4.4.17.2.3 Organization

MODAF: A group of persons, associated for a particular purpose.

DoDAF: A type of Organization.

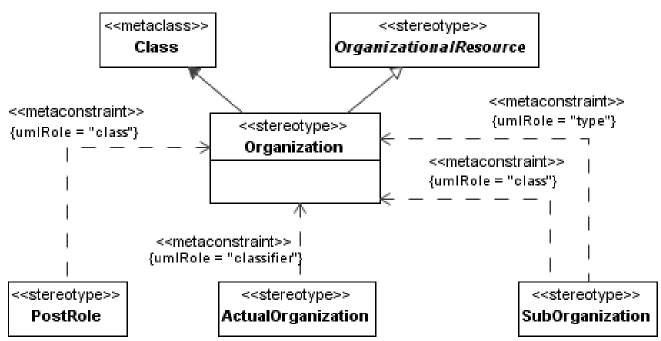


Figure 8.82 - Elements related to the Organization stereotype

Extensions

The following are extensions for Organization:

- Class

Generalizations

The following are generalization relationships for Organization:

- OrganizationalResource

8.1.1.1.4.4.17.2.4 Post

MODAF: A Post (MODAF::PostType) is a type of point of contact or responsible person. Note that this is the type of post (e.g., Desk Officer, Commander Land Component).

DoDAF: A Post (DoDAF:: PersonType) is a category of persons defined by the role or roles they share that are relevant to an architecture.

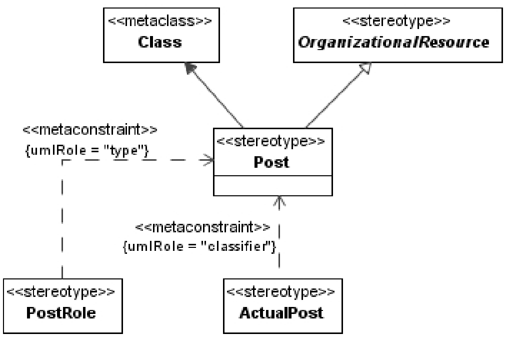


Figure 8.83 - Elements related to the Post stereotype

Extensions

The following are extensions for Post:

- Class

Generalizations

The following are generalization relationships for Post:

- OrganizationalResource

8.1.1.1.4.4.17.2.5 ProvidesCompetence

UPDM: Asserts that a Resource type provides a competence.

MODAF: Asserts that a Role requires a Competence (MODAF::CompetenceForRole).

DoDAF: An overlap between a Personnel Type and the Skills it entails (DoDAF:: skillPartOfPersonType).

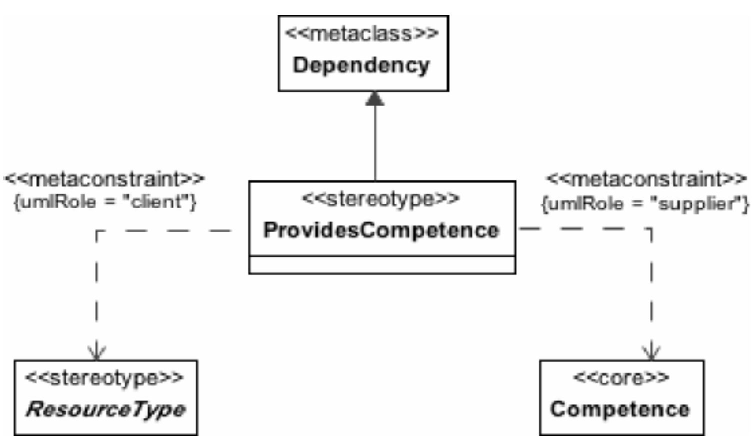


Figure 8.84 - Elements related to the ProvidesCompetence stereotype

Constraints

The following are constraints for ProvidesCompetence:

- ProvidesCompetence.client - Value for the client property must be stereotyped by a specialization of “ResourceType.”
- ProvidesCompetence.supplier - Value for the client property must be stereotyped «Competence» or its specializations.

Extensions

The following are extensions for ProvidesCompetence:

- Dependency

Generalizations

The following are generalization relationships for ProvidesCompetence:

- UPDMElement

8.1.1.1.4.4.17.2.6 RequiresCompetence

MODAF:: Asserts that a Role requires a Competence (MODAF::CompetenceForRole).
DoDAF: An overlap between a Personnel Type and the Skills it entails (DoDAF:: SkillPartOfPersonType).

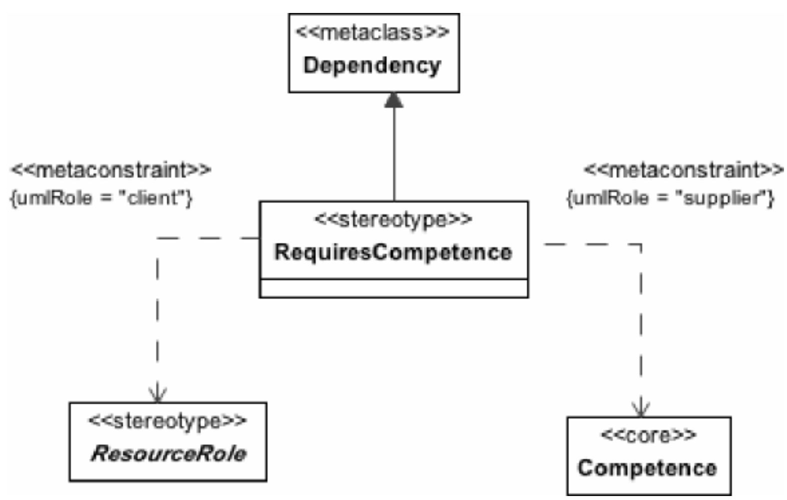


Figure 8.85 - Elements related to the RequiresCompetence stereotype

Constraints

The following are constraints for RequiresCompetence:

- RequiresCompetence.client - Value for the client property must be stereotyped a specialization of “ResourceRole.”
- RequiresCompetence.supplier - Value for the client property must be stereotyped «Competence» or its specializations.

Extensions

The following are extensions for RequiresCompetence:

- Dependency

Generalizations

The following are generalization relationships for RequiresCompetence:

- UPDMElement

8.1.1.1.4.4.18 Constraint

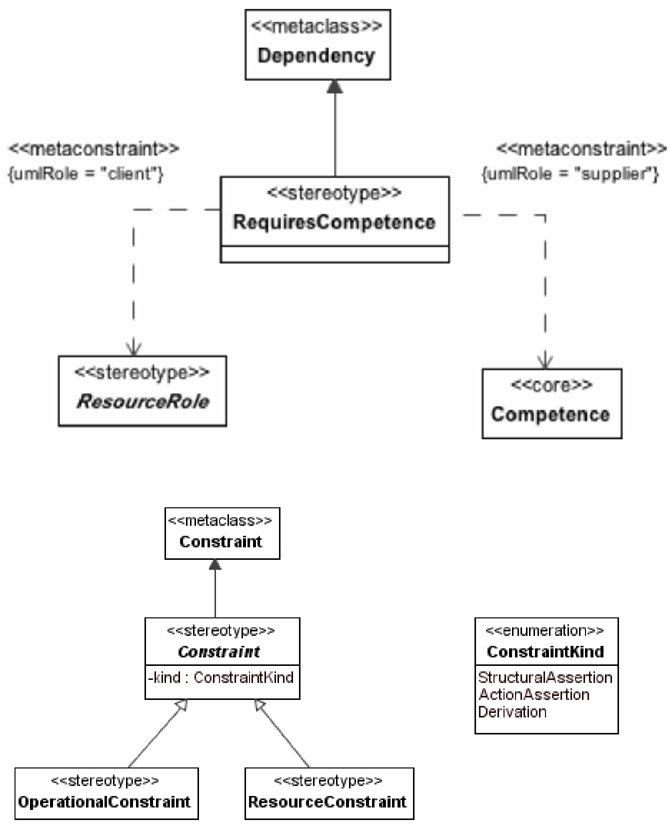


Figure 8.86 - Elements related to the abstract Constraint stereotype

Extensions

The following are extensions for Constraint:

- Constraint

8.1.1.1.4.4.19 Enumeration ConstraintKind

Possible kinds for constraints.

EnumerationLiterals

- StructuralAssertion
- ActionAssertion
- Derivation

8.1.1.1.5 UPDM L1 ::UPDM L0::Core::ServiceElements

The Service-Orientated View (SOV) is a description of services needed to directly support the operational domain as described in the Operational View. A service should be understood in its broadest sense, as a unit of work through which a provider provides a useful result to a consumer. This could be anything from web-based services to delivering an effect to transporting troops.

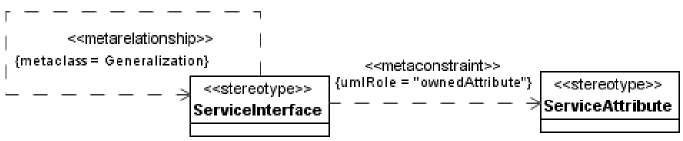


Figure 8.87 - Elements related to the SOV-1 product

MODAF: The Service Taxonomy View (SOV-1) specifies a hierarchy of services. The elements in the hierarchy are service specifications (rather than service implementations), and the relationships between the elements are specializations – i.e., one Service is a special type of another.

DoDAF: NA

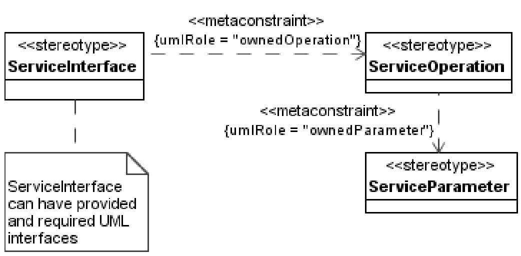


Figure 8.88 - Elements related to the SOV-2 product

MODAF: The Service Taxonomy View (SOV-1) specifies a hierarchy of services. The elements in the hierarchy are service specifications (rather than service implementations), and the relationships between the elements are specializations – i.e., one Service is a special type of another.

DoDAF: NA

The purpose of the Service Interface Specification View (SOV-2) is to define the interfaces presented by a service. A Service presents one or more interfaces to consumers (a "consumer" being any agent capable of using the service - a person, an organization, a system, or another service). In this case, the architect specifies provided interfaces. A service may also be capable of using interfaces exposed by other services, and the architect may specify these as used interfaces.

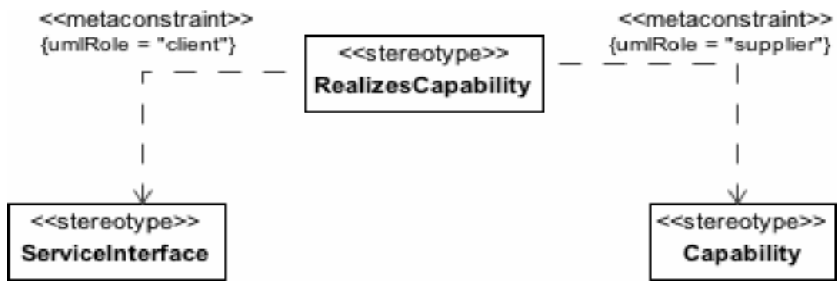


Figure 8.89 - Elements related to the SOV-3 stereotype product

MODAF: The Capability to Service Mapping View (SOV-3) depicts which services contribute to the achievement of a capability.

DoDAF: The Operational Activity to Services Function Traceability Matrix (SvcV-5) DoDAF-described View addresses the linkage between service functions described in SvcV-4 and Operational Activities specified in OV-5.

The Capability to Service Mapping View (SOV-3) depicts which services contribute to the achievement of a capability. It is in the form of a table generated from the database. If network enabled capability is to be delivered by the orchestration of loosely coupled services (i.e., a service-oriented architecture), it is important to know which services have the potential to support particular capabilities. This helps to prevent redundant services or capabilities (except where specifically required) and what is known as stovepipe development. An SOV-3 presents a simple mapping of services to capabilities, showing which services contribute to which capability.

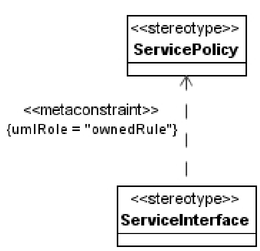


Figure 8.90 - Elements related to the SOV-4a product

MODAF: The purpose of the Service Constraints View (SOV-4a) is to specify constraints that apply to implementations of services.

DoDAF: The SvcV-10a DoDAF-described View describes constraints on the resources, functions, data, and ports that make up the Service View physical architecture. The constraints are specified in text and may be functional or structural (i.e., nonfunctional).

Figure 8.91 - Elements related to the SOV-4b product

MODAF: The purpose of the Service State Model View (SOV-4b) is to specify the possible states a service may have, and the possible transitions between those states.

DoDAF: The Services State Transition Description DoDAF-described View is a graphical method of describing a resource (or function) response to various events by changing its state. The diagram basically represents the sets of events to which the resources in the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

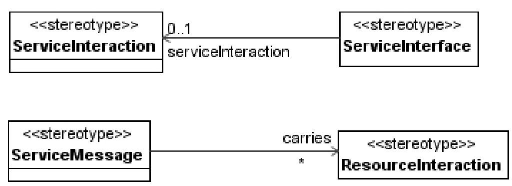


Figure 8.92 - Elements related to the SOV-4c product

MODAF: The purpose of the Service Interaction Specification View (SOV-4c) is to specify how a service interacts with external agents, and the sequence and dependencies of those interactions.

DoDAF: The Services Event-Trace Description DoDAF-described View provides a time-ordered examination of the interactions between services functional resources. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

The purpose of the Service Interaction Specification View (SOV-4c) is to specify how a service interacts with external agents, and the sequence and dependencies of those interactions. An SOV-4c product does not specify the sequencing of an orchestrated set of services (see OV-6c). Its purpose is to specify the general sequence of interactions that are possible for a given service.

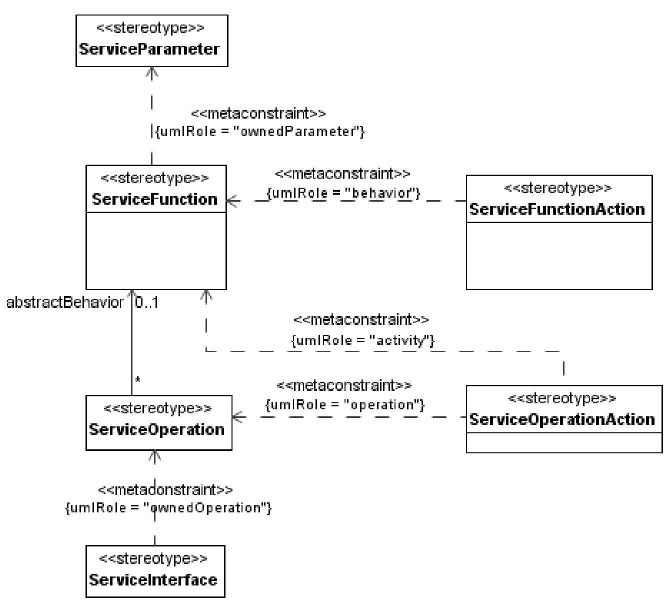


Figure 8.93 - Elements related to the SOV-5 product

MODAF: The Service Functionality View (SOV-5) defines the behavior of a service in terms of the functions it is expected to perform.

DoDAF: The Services Functionality Description provides detailed information regarding the Allocation of service functions to resources, and Flow of resources between service functions.

This view enables the description of a service by means of a state-diagram, an activity model, or by an event-trace description. In order to fully define the sequencing of messages and operations that form part of the service interface, event-trace descriptions are useful. In order to fully define the handling that takes place in a service due to possible different internal states, a state description is also useful. For all of these descriptions standard UML entities provide the best means of representation.

8.1.1.1.5.1 UPDM L1 ::UPDM L0::Core::ServiceElements::Behavior

Behavior elements of the service oriented view.

8.1.1.1.5.1.1 ServiceFunction

UPDM: A ServiceFunction describes the abstract behavior of ServiceOperations, regardless of the actual implementation.

MODAF: A type of activity describing the functionality of a service.

DoDAF: Information necessary to interact with the service in such terms as the service inputs, outputs, and associated semantics. The service description also conveys what is accomplished when the service is invoked and the conditions for using the service.

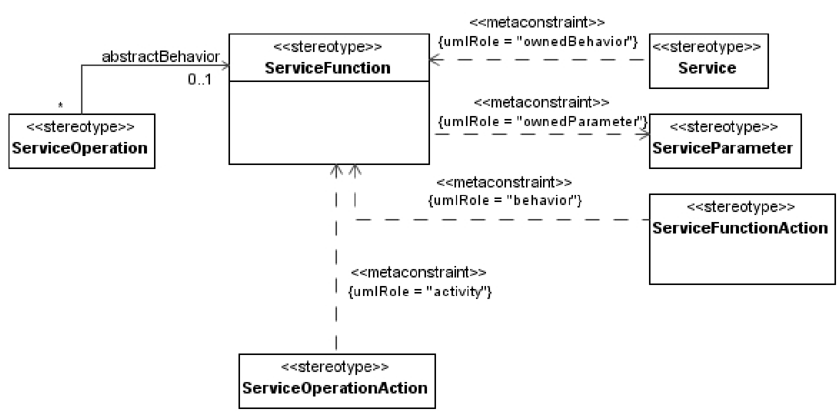


Figure 8.94 - Elements related to the ServiceFunction stereotype

Constraints

The following are constraints for ServiceFunction:

- ServiceFunction.ownedParameter - The values for the ownedParameter property must be stereotyped «ServiceParameter».

Extensions

The following are extensions for ServiceFunction:

- Activity

Generalizations

The following are generalization relationships for ServiceFunction:

- UPDMElement

8.1.1.1.5.1.2 ServiceFunctionAction

UPDM: A call behavior action that invokes the ServiceFunction that needs to be preformed. This concept is required for mapping the architecture with UML and does not have a DoDAF or MoDAF equivalent.

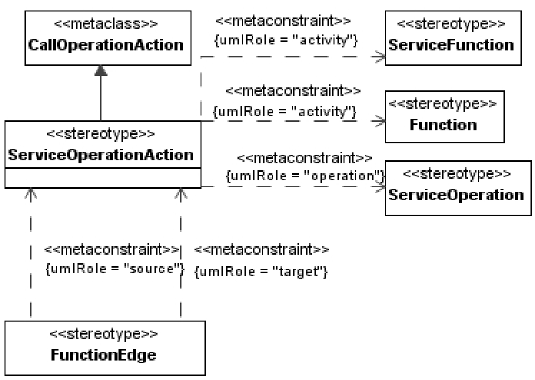


Figure 8.95 - Elements related to the ServiceFunctionAction stereotype

Constraints

The following are constraints for ServiceFunctionAction:

- ServiceFunctionAction.behavior - Value for the behavior property must be stereotyped «ServiceFunction» or its specializations.

Extensions

The following are extensions for ServiceFunctionAction:

- CallBehaviorAction

Generalizations

The following are generalization relationships for ServiceFunctionAction:

- UPDMElement

8.1.1.1.5.1.3 ServiceInteraction

UPDM: Interaction for a service interface.

MODAF: A model representing how a set of Service classes interacts with one another (MODAF::ServiceInteractionSpecification).

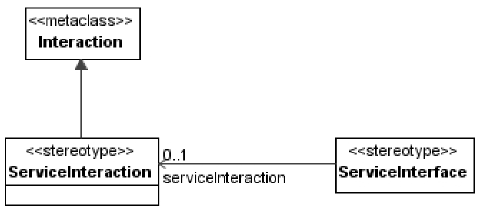


Figure 8.96 - Elements related to the ServiceInteraction stereotype

Extensions

The following are extensions for ServiceInteraction:

- Interaction

8.1.1.1.5.1.4 ServiceOperation

UPDM: A ServiceOperation provides the access point for invoking the behavior of a provided service. The ServiceOperations are defined on ServiceInterfaces and mirrored on the providing Resource to handle calls forwarded on by the interface.

MODAF: a function or procedure which enables programmatic communication with a Service via a ServiceInterface (MODAF:: ServiceInterfaceOperation).

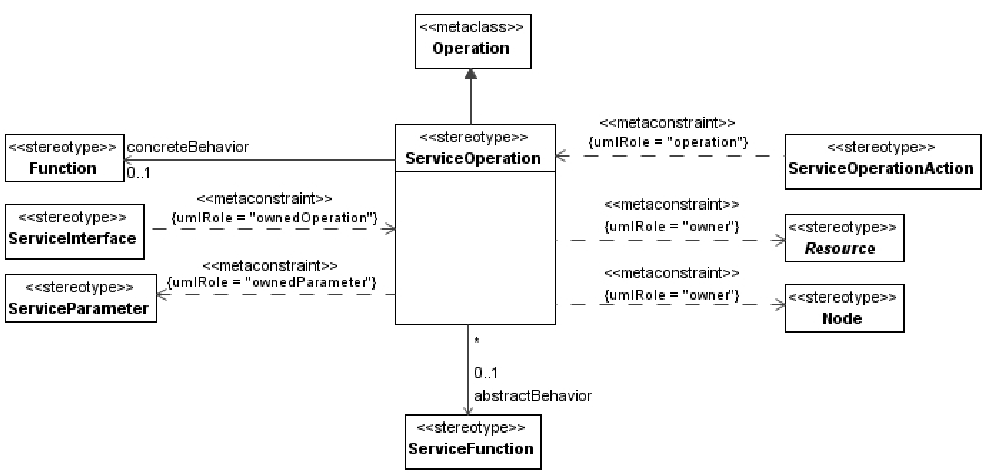


Figure 8.97 - Elements related to the ServiceOperation stereotype

Constraints

The following are constraints for ServiceOperation:

- ServiceOperation.ownedParameter - The values for the ownedParameter property must be stereotyped «ServiceParameter».
- ServiceOperation.owner - The values for the owner property must be stereotyped «Node»/«Resource» or its specializations.

Attributes

The following are attributes of ServiceOperation:

- abstractBehavior : ServiceFunction[0..1] - Links a ServiceOperation to the abstract description of its behavior, as provided by a ServiceFunction.
- concreteBehavior : Function[0..1] - Links a ServiceOperation to the concrete description of its behavior, as provided by a Function.

Extensions

The following are extensions for ServiceOperation:

- Operation

Generalizations

The following are generalization relationships for ServiceOperation:

- UPDMElement

8.1.1.1.5.1.5 ServiceOperationAction

UPDM Artifact: A call action that represents a Resource or Service Function invoking a ServiceOperation. This is used by a consuming Resource to model the call into the service. This concept is required for mapping the architecture with UML and does not have a DoDAF or MoDAF equivalent.

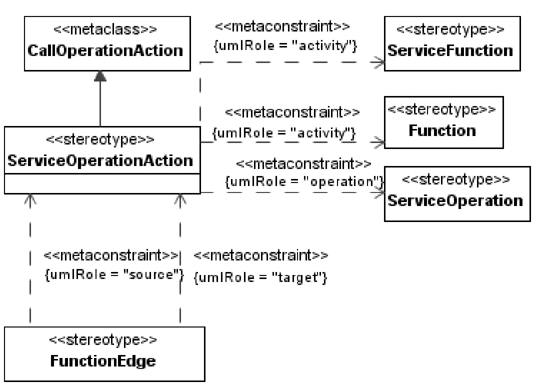


Figure 8.98 - Elements related to the ServiceOperationAction stereotype

Constraints

The following are constraints for ServiceOperationAction:

- ServiceOperationAction.operation - Values for the operation property must be stereotyped «ServiceOperation» or its specializations.
- ServiceOperationAction.operation - Values for the activity property must be stereotyped «ServiceFunction»/ «Function» or its specializations.

Extensions

The following are extensions for ServiceOperationAction:

- CallOperationAction

Generalizations

The following are generalization relationships for ServiceOperationAction:

- UPDMElement

8.1.1.1.5.1.6 ServiceParameter

UPDM: Represents inputs and outputs of Service. It is typed by ResourceInteractionItem.

MODAF: A constant or variable passed into or out of a ServiceInterface as part of the execution of a ServiceInterfaceOperation (MODAF:: ServiceInterfaceParameter).

DoDAF: NA

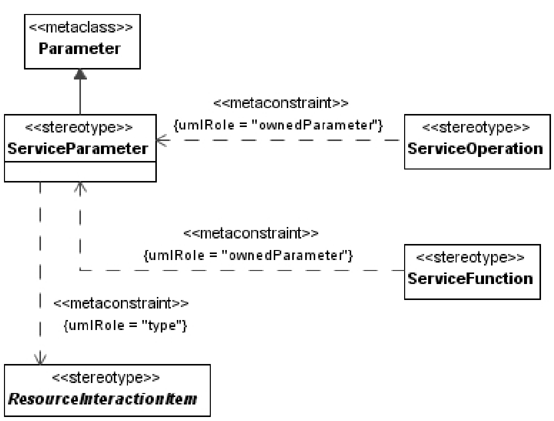


Figure 8.99 - Elements related to the ServiceParameter stereotype

Constraints

The following are constraints for ServiceParameter:

- Service Parameter.type – The values for the type property msut be stereotyped «ResourceInteractionItem» or its specializations.

Extensions

The following are extensions for ServiceParameter:

- Parameter

Generalizations

The following are generalization relationships for ServiceParameter:

- FunctionParameter

8.1.1.1.5.1.7 ServiceStateMachine

UPDM Artifact that extends a UML StateMachine.

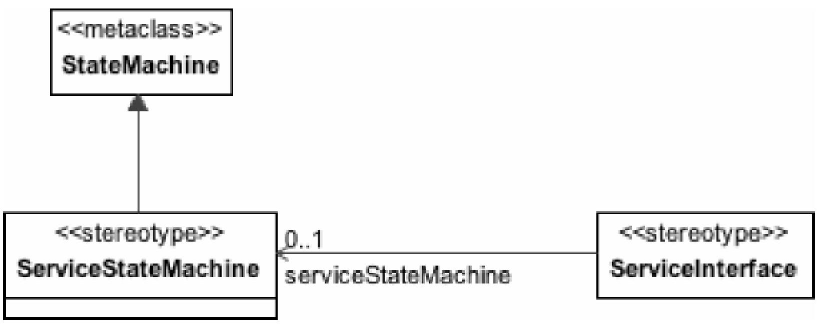


Figure 8.100 - Elements related to the ServiceStateMachine stereotype

Extensions

The following are extensions for ServiceStateMachine:

- StateMachine

8.1.1.1.5.2 UPDM L1 ::UPDM L0::Core::ServiceElements::Structure

Structure elements of the service oriented view.

8.1.1.1.5.2.1 RequestPoint

This is a SoaML element.

MODAF: “The mechanism by which a Service communicates.”

UPDM: “A Request port provides an interaction point for accessing a Resource’s capabilities, as defined by the ServiceInterface typing the port. It includes the specification of the work required from another, and the request to perform work by another.”

The Request is basically a conjugated version of a Service port and as such can use the same ServiceInterface. By adding the Request port to the consumer of the Service, the implementation of a service is kept transparent.

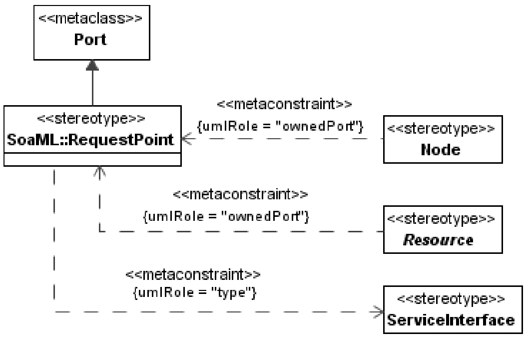


Figure 8.101 - Elements related to the RequestPoint stereotype

Constraints

The following are constraints for RequestPoint:

- RequestPoint.type - Values for type property must be stereotyped «ServiceInterface» or its specializations.

Extensions

The following are extensions for RequestPoint:

- Port

Generalizations

The following are generalization relationships for RequestPoint:

- UPDMElement

8.1.1.1.5.2.2 ServicePoint

This is a SoaML element.

MODAF: "The mechanism by which a Service communicates."

UPDM: "A Service port supplies a mechanism for making available the capabilities provided by the Resource. They represent the offer to perform work requested by a service consumer."

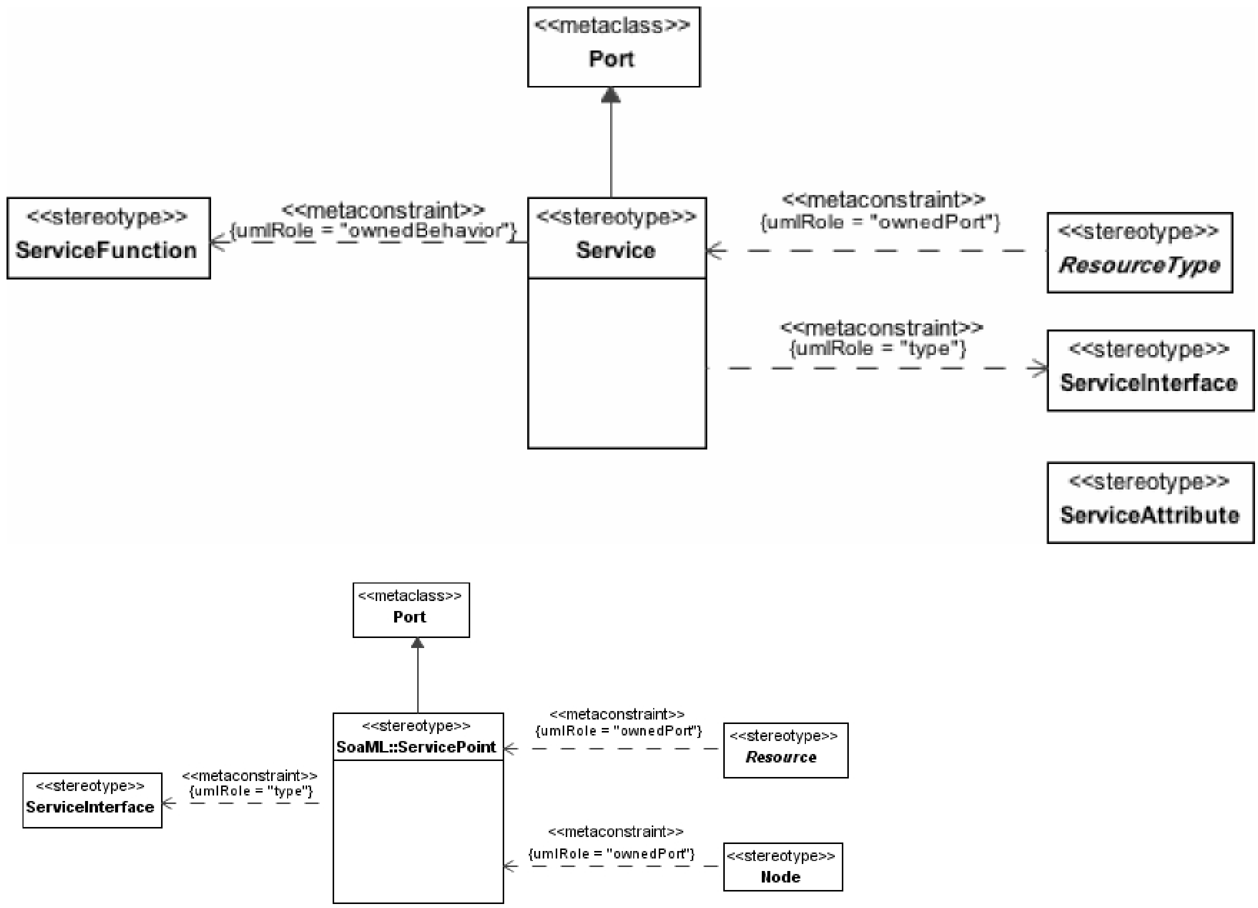


Figure 8.102 - Elements related to the ServicePoint stereotype

Constraints

The following are constraints for ServicePoint:

- ServicePoint.type - Values for type property must be stereotyped «ServiceInterface» or its specializations.
  - An element with the stereotype «Service Point » applied must have the «ServiceInterface» stereotype applied to the targets of its extended metaclass property "type."

Extensions

The following are extensions for ServicePoint:

- Port

Generalizations

The following are generalization relationships for ServicePoint:

- UPDMElement

8.1.1.1.5.2.3 ServiceAttribute

UPDM: “A property of a ServiceInterface that allows performance, reliability, and cost values to be captured. This allows a user to choose between different ServiceInterfaces providing the same Capabilities.”

MODAF: “A property of Service.”

DoDAF: NA

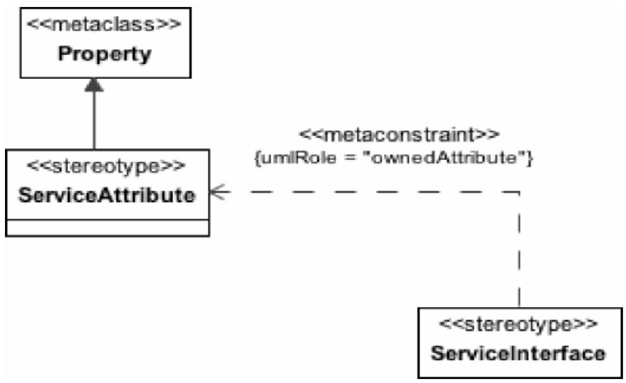


Figure 8.103 - Elements related to the ServiceAttribute stereotype

Extensions

The following are extensions for ServiceAttribute:

- Property

Generalizations

The following are generalization relationships for ServiceAttribute:

- UPDMElement

8.1.1.1.5.2.4 ServiceInterface

UPDM: A contractual agreement between two resources that implement protocols through which the source service interacts to the destination resource.

A physical connection between two resources that implements protocols through which the source resource can transmit items to the destination resource.

MODAF: The mechanism by which a Service communicates.

DoDAF: An overlap between Performers for the purpose of producing a Resource that is consumed by the other. (DoDAF::Interface)

SOAML: Defines the interface to a Service Point or Request Point and is the type of a role in a service contract.

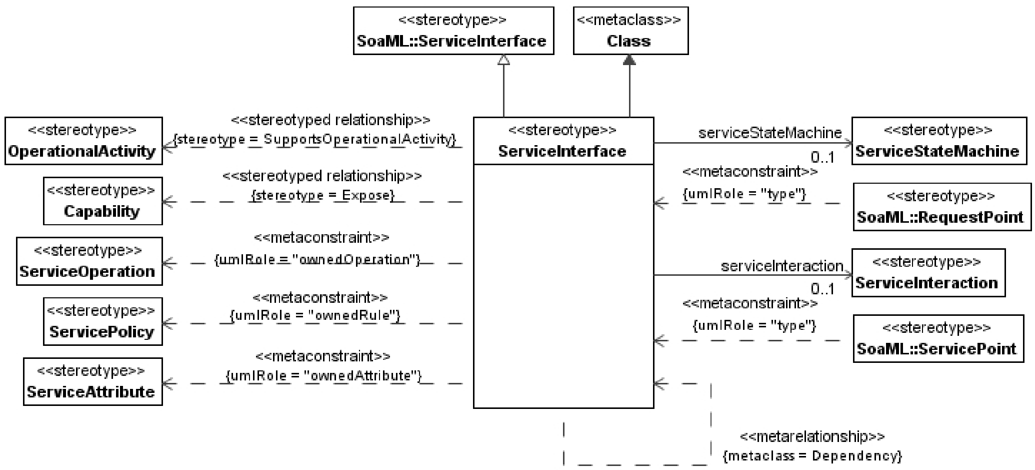


Figure 8.104 - Elements related to the ServiceInterface stereotype

Constraints

The following are constraints for ServiceInterface:

- ServiceInterface.ownedRule - Value for the ownedRule property must be stereotyped «ServiceOperation» or its specializations.
- Service.ownedAttribute - Values for ownedAttribute property must be stereotyped «ServiceAttribute» or its specializations.
- ServiceInterface.ownedOperation - Value for the ownedOperation property must be stereotyped «ServicePolicy» or its specializations.

Extensions

The following are attributes of ServiceInterface:

- Class

Attributes

The following are attributes of ServiceInterface:

- serviceInteraction : ServiceInteraction[0.. 1] - Service interaction.
- serviceStateMachine : ServiceStateMachine[0..1] - Service state machine.

Extensions

The following are extensions for ServiceInterface:

- Interface

Generalizations

The following are generalization relationships for ServiceInterface:

- SoaML::ServiceInterface
- UPDMElement

8.1.1.1.5.2.5 ServicePolicy

UPDM: A constraint governing the consumers and providers of services

MODAF: A constraint governing one or more Services

DoDAF: Agreement: A consent among parties regarding the terms and conditions of activities that said parties participate in.

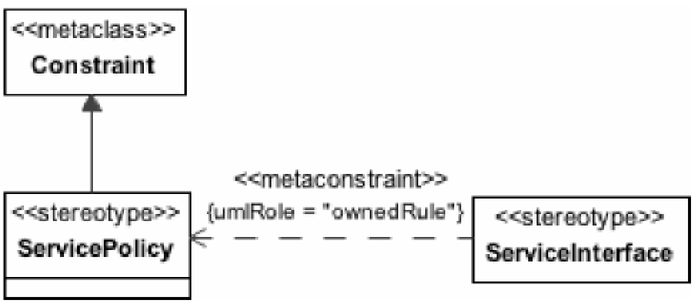


Figure 8.105 - Elements related to the ServicePolicy stereotype

Extensions

The following are extensions for ServicePolicy:

- Constraint

Generalizations

The following are generalization relationships for ServicePolicy:

- UPDMElement

8.1.1.1.5.2.6 SupportsOperationalActivity

MODAF: An assertion that a Service in some way contributes or assists in the execution of an OperationalActivity (MODAF::ServiceSupportsActivity).

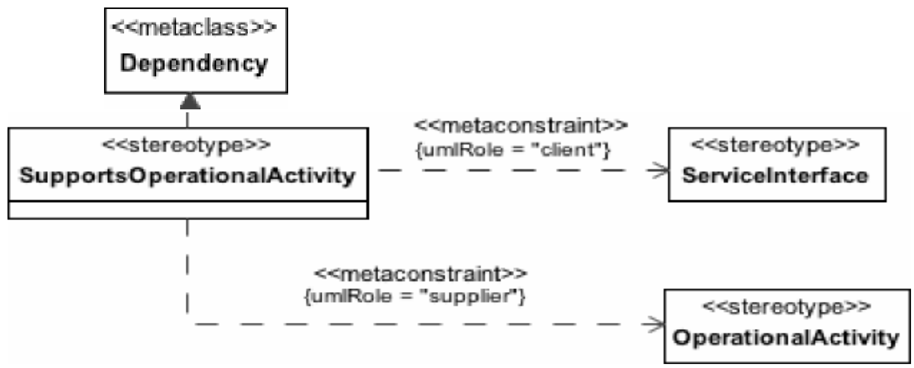


Figure 8.106 - Elements related to the SupportsOperationalActivity stereotype

Constraints

The following are constraints for SupportsOperationalActivity:

- SupportsOperationalActivity.client - Value for the client property must be stereotyped «ServiceInterface» or its specializations.
- SupportsOperationalActivity. supplier - Value for the client property must be stereotyped «OperationalActivity» or its specializations.

Extensions

The following are extensions for SupportsOperationalActivity:

- Dependency

Generalizations

The following are generalization relationships for SupportsOperationalActivity:

- UPDMElement

8.1.1.1.6 UPDM L1 ::UPDM L0::Core::StrategicElements

The Strategic Elements are used in the Strategic View that provides an overall Enterprise Architecture assessment of the Capabilities and their relationships facilitating Capability Management (e.g., capability introduction, integration, realignment, and removal). While an Enterprise will have a number of UPDM Architecture Descriptions that have the Operational, System, Technical Standards, and All Views, only one Strategic View will exist across a number of Architecture Descriptions.

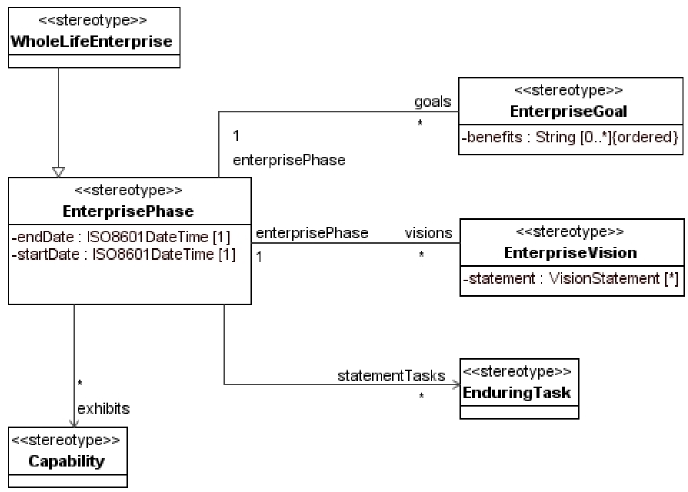


Figure 8.107 - Elements related to the StV-1 product

StV-1 addresses the enterprise concerns associated with the overall vision for transformational endeavors and thus defines the strategic context for a group of Enterprise capabilities. 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 that is more general than the scenario-based scope defined in an OV-1. The Views are high-level and describe capabilities using terminology that is easily understood by non-technical readers (though they may make extensive use of military terminology and acronyms that are clearly defined in the AV-2 View).

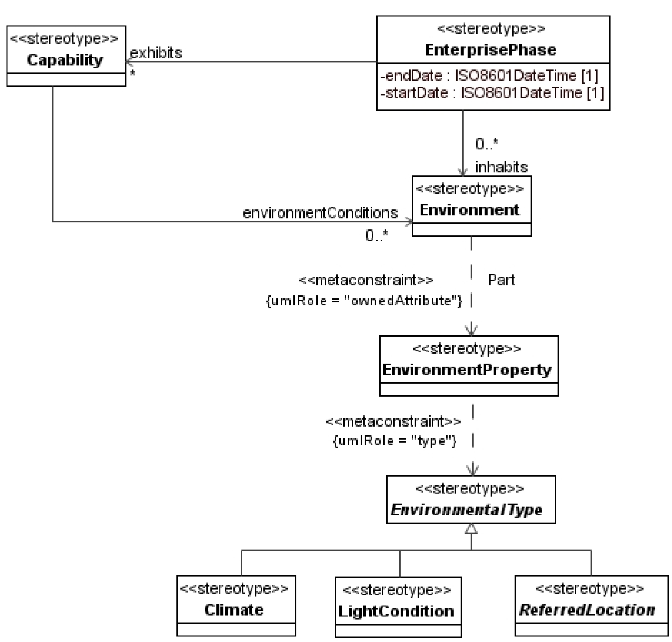


Figure 8.108 - Elements related to the StV-2 product

The view presents a hierarchy of capabilities. These capabilities may be presented in context of an Enterprise Phase (i.e., it can show the required capabilities for current and future enterprises). StV-2 specifies all the capabilities that are reference throughout one or more architectures. In addition it can be used as a source document for the development of high-level use cases and Key User Requirements (KUR). The StV-2 also provides metrics against each capability that may be used to measure successfully fielded capability.

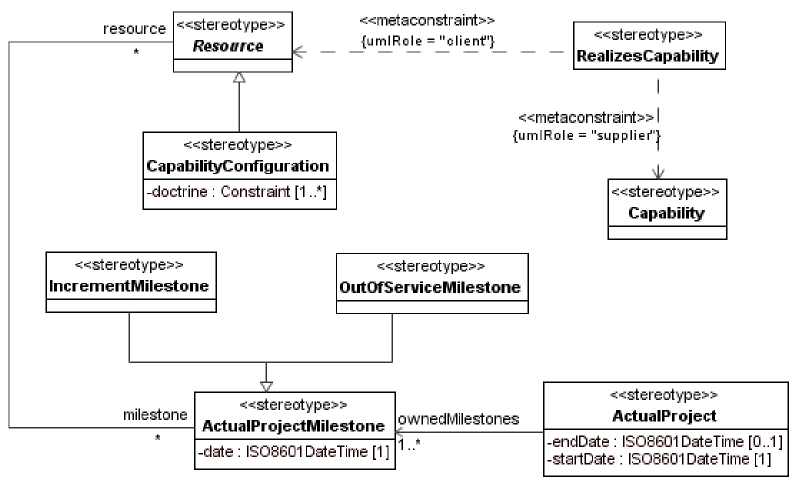


Figure 8.109 - Elements related to the StV-3 stereotype

StV-3 addresses the planned achievement of capability at different points in time or during specific periods of time (i.e., capability phasing). StV-3 Views support the Capability Audit process by providing a method to identify gaps or duplication in capability provision. The view indicates capability increments, which should be associated with delivery milestones within acquisition projects (when the increments are associated with capability deliveries).

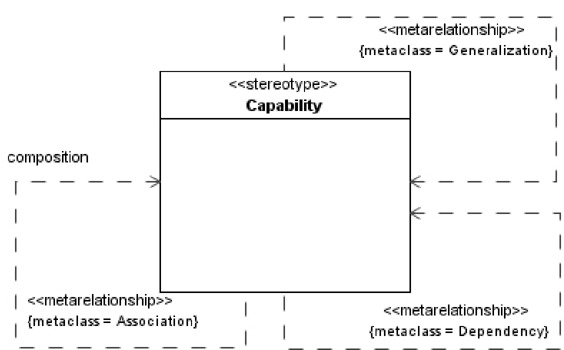


Figure 8.110 - Elements related to the StV-4 product

The StV-4 Product describes the dependencies between planned capabilities. It also defines logical groupings of capabilities. The StV-4 View provides a means of analyzing the dependencies between capabilities. The groupings of capabilities are logical, and the purpose of the groupings is to guide enterprise management.

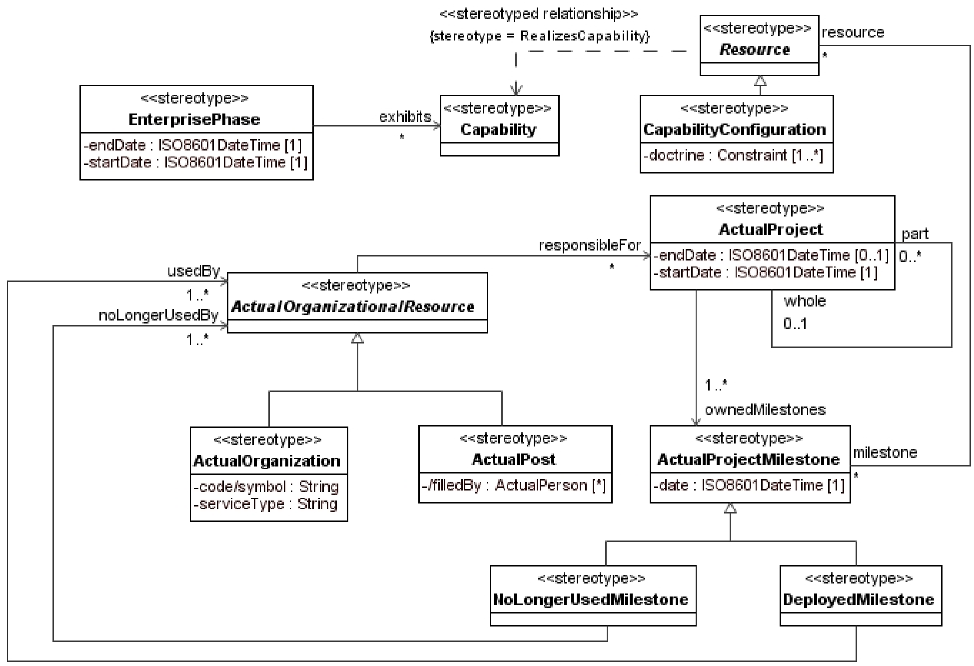


Figure 8.111 - Elements related to the StV-5 stereotype

The StV-5 defines Capability to Organization Deployment Mapping. It addresses the fulfillment of capability requirements, in particular by network enabled capabilities. This view shows the planned capability deployment and interconnection for a particular EnterprisePhase. This view will provide a more detailed dependency analysis than is possible using StV-3. The StV-5 View is used to support the capability management process and, in particular, assist the planning of fielding.

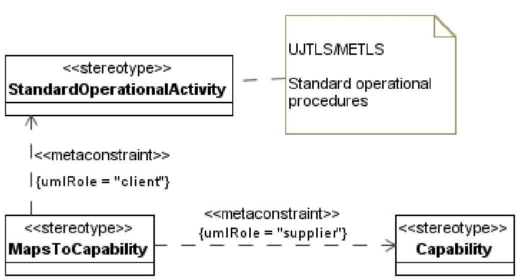


Figure 8.112 - Elements related to the StV-6 stereotype product

The StV-6 describes the mapping between the capabilities required by an Enterprise and the operational activities that those capabilities support.

8.1.1.1.6.1 UPDM L1 ::UPDM L0::Core::StrategicElements::Structure

Structural section of the StrategicElements profile.

8.1.1.1.6.1.1 Capability

A Capability is a high-level specification of an ability or capacity that achieves specific objectives. An Enterprise will consist of many Capabilities each used in various combinations with other Capabilities to achieve the declared vision, goals, and objectives of the Enterprise. A Capability may evolve over time with regard to how it is delivered but its specific objective typically remains unchanged.

MODAF: A high level specification of the enterprise's ability.

DoDAF: The ability to achieve a desired effect under specified [performance] standards and conditions through combinations of ways and means [activities and resources] to perform a set of activities.

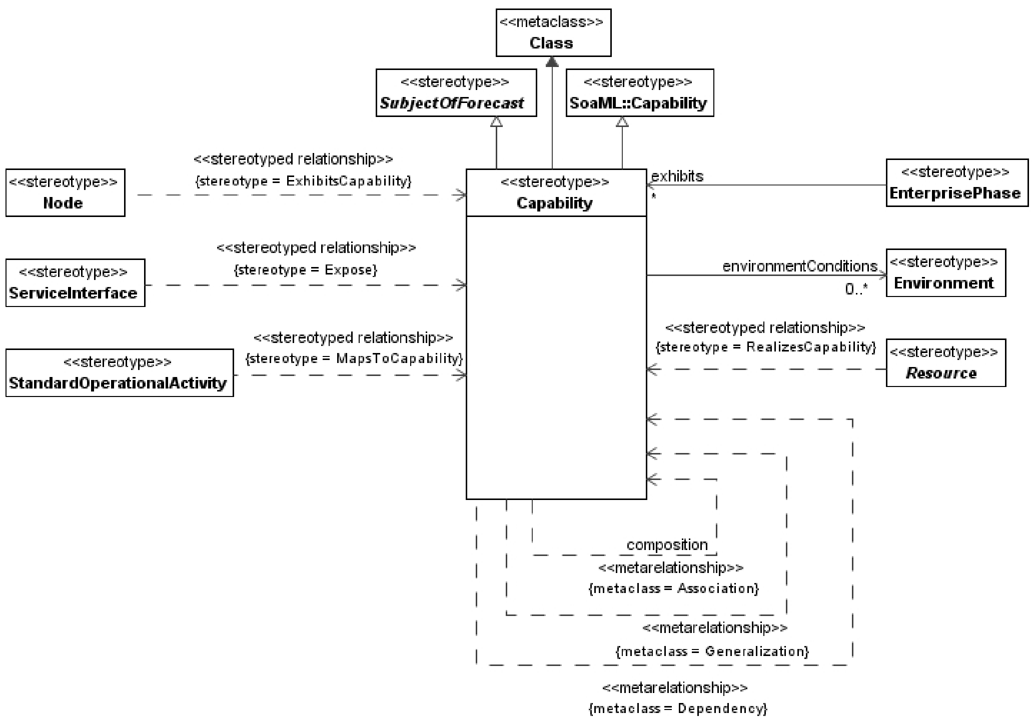


Figure 8.113 - Elements related to the Capability stereotype

Attributes

The following are attributes of Capability:

- environmentConditions : Environment[0..*] - The environmental conditions pertinent to this Capability typically including weather, threat, security, terrain.

Extensions

The following are extensions for Capability:

- Class

Generalizations

The following are generalization relationships for Capability:

- SubjectOfForecast
- SoaML::Capability

8.1.1.1.6.1.2 WholeLifeEnterprise

UPDM: A WholeLifeEnterprise is a purposeful endeavor of any size involving people, organizations, and supporting systems (including physical systems and/or processes).

MODAF: An EnterprisePhase that represents the whole existance of an enterprise.

DoDAF: NA

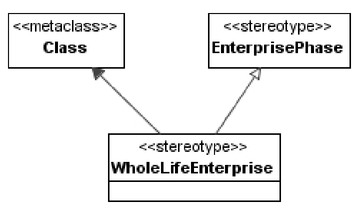


Figure 8.114 - Elements related to the WholeLifeEnterprise stereotype

Extensions

The following are extensions for WholeLifeEnterprise:

- Class

Generalizations

The following are generalization relationships for WholeLifeEnterprise:

- EnterprisePhase

8.1.1.1.6.1.3 EnterpriseGoal

MODAF: A specific, required objective of the enterprise that the architecture represents.

TBD DoDAF: (DoDAF::IndividualDesiredEffect): A desired change in the state as a result of some activity.

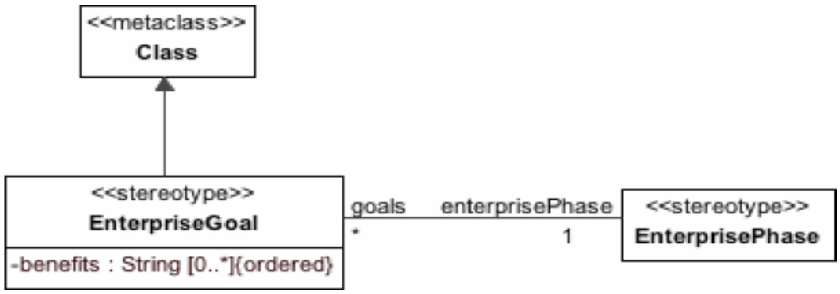


Figure 8.115 - Elements related to the EnterpriseGoal stereotype

Attributes

The following are attributes of EnterpriseGoal:

- benefits : String[0..*] - A description of the usefulness of the Goal in terms of why the state or condition of the Enterprise is worth attaining.
- enterprisePhase : EnterprisePhase[1] - Phase of the goal.

Extensions

The following are extensions for EnterpriseGoal:

- Class

Generalizations

The following are generalization relationships for EnterpriseGoal:

- UPDMElement

8.1.1.1.6.1.4 EnterprisePhase

MODAF: A specific, required objective of the enterprise that the architecture represents.

DoDAF: NA

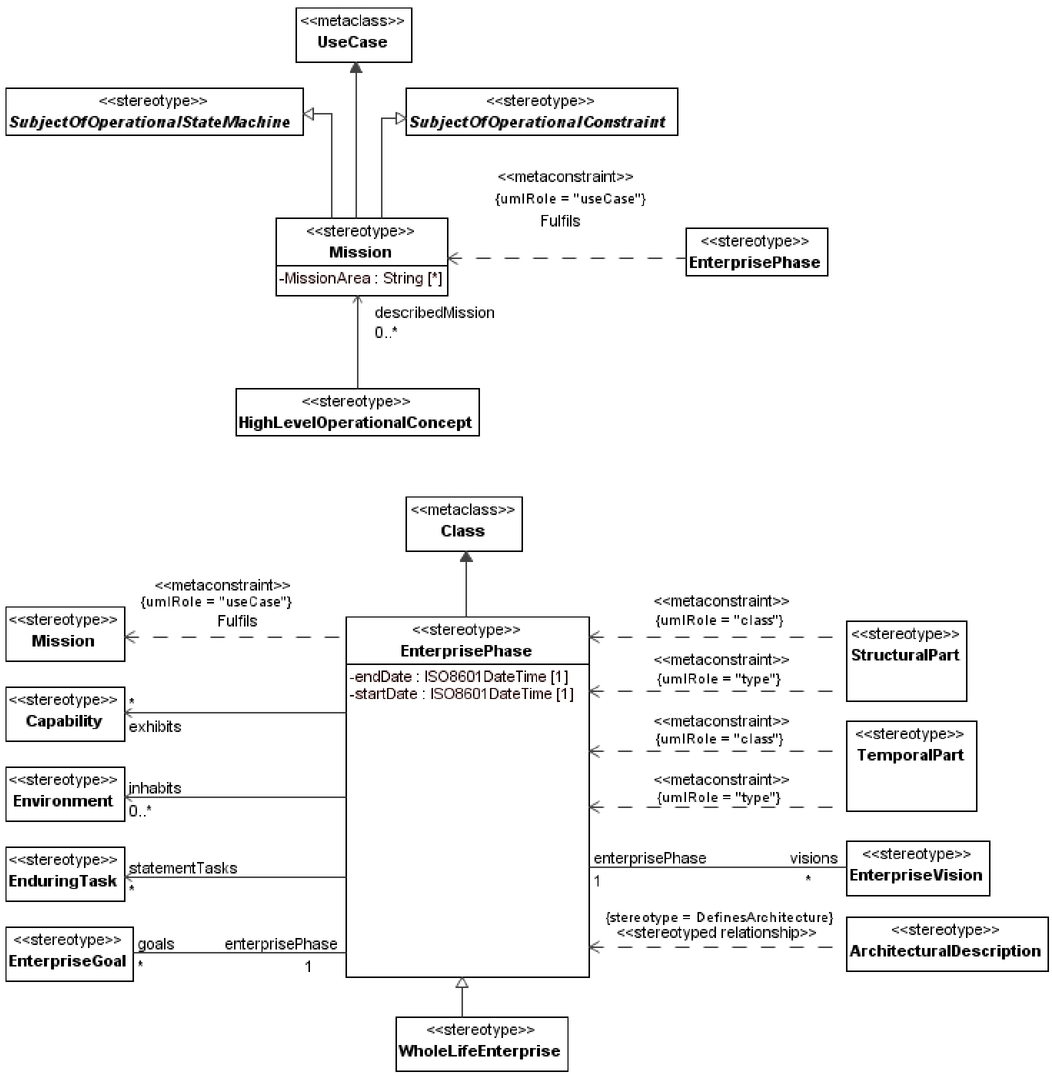


Figure 8.116 - Elements related to the EnterprisePhase stereotype

Constraints

The following are constraints for EnterprisePhase:

- Enterprise from/to - Must fall within the Enterprise to and from time, the complete lifecycle.
- EnterprisePhase.useCase - Values for the useCase property must be stereotyped «Mission».

Attributes

The following are attributes of EnterprisePhase:

- endDate : ISO8601DateTime[1] - The time and date at which the Phase ends.
- Enterprise from/to - Must fall within the Enterprise to and from time, the complete lifecycle.

- startDate : ISO8601DateTime[1] - The time and date at which the Phase starts.
- goals : EnterpriseGoal[*] - The Goal towards which this Phase is directed and is in support of.
- visions : EnterpriseVision[*] - The Vision towards which this Phase is directed and is in support of.
- statementTasks : EnduringTask[*] - Collection of statement tasks.
- inhabits : Environment[0..*] - Environment supported by this Capability.
- exhibits : Capability[*] - Exhibited Capabilities.

Extensions

The following are extensions for EnterprisePhase:

- Class

Generalizations

The following are generalization relationships for EnterprisePhase:

- UPDMElement

8.1.1.1.6.1.5 EnterpriseVision

MODAF: The overall aims of an enterprise over a given period of time.

DoDAF: (DoDAF::Vision): An end that describes the future state of the enterprise, without regard to how it is to be achieved; a mental image of what the future will or could be like.

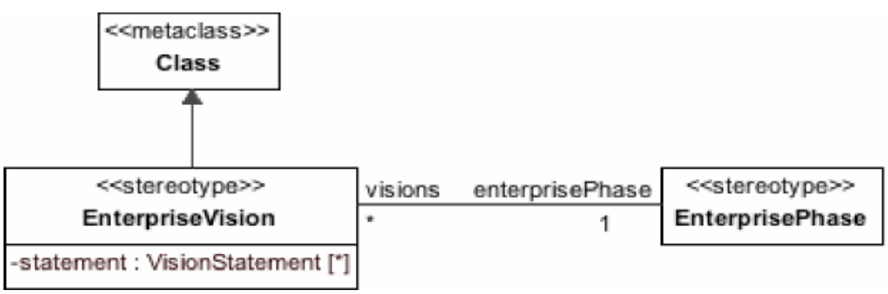


Figure 8.117 - Elements related to the EnterpriseVision stereotype

Attributes

The following are attributes of EnterpriseVision:

- statement : VisionStatement[*] - A description of the Vision.
- enterprisePhase : EnterprisePhase[1] - The phase which temporally locates the Vision.

Extensions

The following are extensions for EnterpriseVision:

- Class

Generalizations

The following are generalization relationships for EnterpriseVision:

- UPDMElement

8.1.1.1.6.1.6 MapsToCapability

MODAF: Asserts that a StandardOperationalActivity is in some way part of a capability.

DoDAF: MapsToCapability (DoDAF::ActivityPartOfCapability) is a disposition to manifest an Activity. An Activity to be performed to achieve a desired effect under specified [performance] standards and conditions through combinations of ways and means.

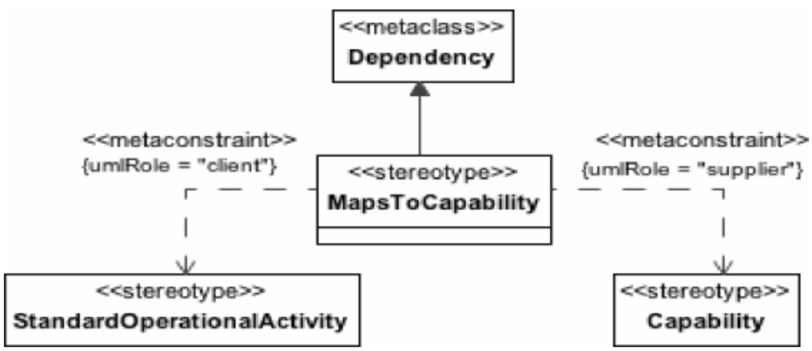


Figure 8.118 - Elements related to the MapsToCapability stereotype

Constraints

The following are constraints for MapsToCapability:

- MapsToCapability.supplier - Value for the supplier property must be stereotyped «Capability».
- MapsToCapability.client - Value for the client property must be stereotyped «StandardOperationalActivity».

Extensions

The following are extensions for MapsToCapability:

- Dependency

Generalizations

The following are generalization relationships for MapsToCapability:

- UPDMElement

8.1.1.1.6.1.7 RealizesCapability

UPDM: Asserts that a Resource type provides a Capability.

DoDAF: A couple that represents the capability that a performer manifests (DoDAF ::CapabilityPerformerManifestation).

MODAF: Asserts that a CapabilityConfiguration is capable of achieving a Capability (MODAF::CapabilityRealization).

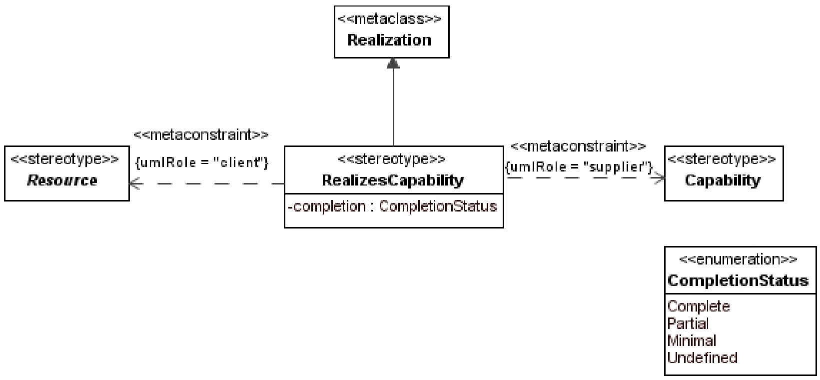


Figure 8.119 - Elements related to the RealizesCapability stereotype

Constraints

The following are constraints for RealizesCapability:

- RealizesCapability.supplier - Values for the supplier property must be stereotyped «Capability» or its specializations.
- RealizesCapability.client - Values for the client property must be stereotyped «ResourceType» or its specializations.
- RealizesCapability.measurements - For every Capability related “required” ActualMeasurementSet, there must be a matching Resource related “estimate”/”result” ActualMeasurementSet. The term “matching” is used to mean the classifier for the ActualMeasurementSets must be the same or a specialization.

Attributes

The following are extensions for RealizesCapability:

- completion : CompletionStatus – Completion status

Extensions

The following are extensions for RealizesCapability:

- Dependency

Generalizations

The following are generalization relationships for RealizesCapability:

- UPDMElement

8.1.1.1.6.1.8 ExhibitsCapability

UPDM: Relationship between a Node and a capability the node provides.

MODAF: (MODAF::CapabilityForNode): An assertion that a Node is required to have a Capability.

DoDAF: A couple that represents the capability that a performer manifests.

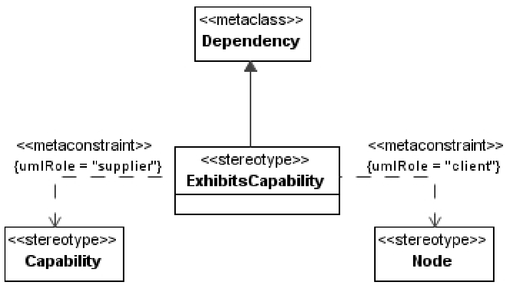


Figure 8.120 - Elements related to the ExhibitsCapability stereotype

Constraints

The following are constraints for ExhibitsCapability:

- ExhibitsCapability.supplier - Value for the supplier property must be stereotyped «Capability».
- ExhibitsCapability.client - Value for the client property must be stereotyped «Node» or its specializations.
- ExhibitsCapability.measurements - For every Capability related “required” ActualMeasurementSet, there must be a matching Node related “estimate”/”result” ActualMeasurementSet. The term “matching” is used to mean the classifier for the ActualMeasurementSets must be the same or a specialization.

Extensions

The following are extensions for ExhibitsCapability:

- Dependency

Generalizations

The following are generalization relationships for ExhibitsCapability:

- UPDMElement

8.1.1.1.6.1.9 StructuralPart

UPDM: An EnterprisePhase can be sub-divided into structural and temporal parts. StructuralPart describes the EnterprisePhase elements that describe the structure.

MODAF: Asserts that one EnterprisePhase is a spatial part of another, (MODAF::EnterpriseStructure).

Note – This is a topological structuring relationship, hence the EnterprisePhase may be physically disjoint.

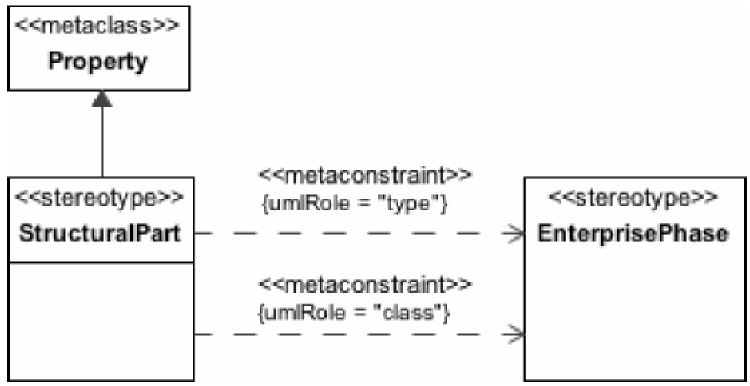


Figure 8.121 - Elements related to the StructuralPart stereotype

Constraints

The following are constraints for StructuralPart:

- StructuralPart.class - Value for class metaproperty must be stereotyped «EnterprisePhase» or its specializations.
- StructuralPart.type - Value for type metaproperty must be stereotyped «EnterprisePhase» or its specializations.

Extensions

The following are extensions for StructuralPart:

- Property

Generalizations

The following are generalization relationships for StructuralPart:

- UPDMElement

8.1.1.1.6.1.10 TemporalPart

UPDM Artifact: An EnterprisePhase can be sub-divided into structural and temporal parts. TemporalPart describes the EnterprisePhase elements that have a time based nature.

MODAF: Asserts that one EnterprisePhase is a temporal part of another.

Note – This means that both EnterprisePhases have the same spatial extent - i.e., this is only a temporal structure (MODAF::EnterpriseTemporalPart).

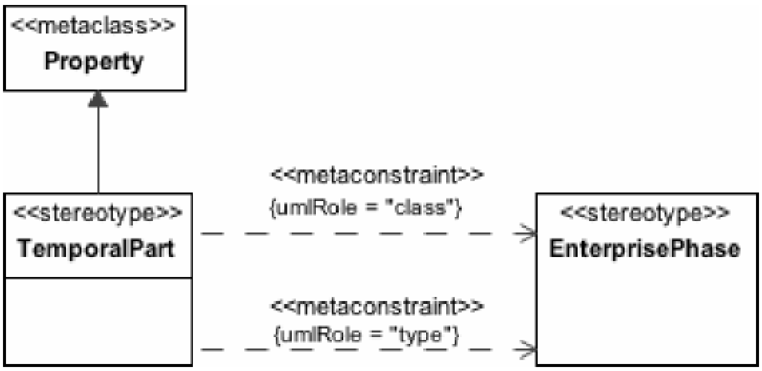


Figure 8.122 - Elements related to the TemporalPart stereotype

Constraints

The following are constraints for TemporalPart:

- TemporalType.type - Values for class property must be stereotyped «EnterprisePhase» or its specializations.
- TemporalType.class - Values for type property must be stereotyped «EnterprisePhase» or its specializations.

Extensions

The following are extensions for TemporalPart:

- Property

Generalizations

The following are generalization relationships for TemporalPart:

- UPDMElement

8.1.1.1.6.1.11 VisionStatement

MODAF: A high-level textual description of an EnterpriseVision.

DoDAF: An end that describes the future state of the enterprise, without regard to how it is to be achieved; a mental image of what the future will or could be like (DODAF::Vision).

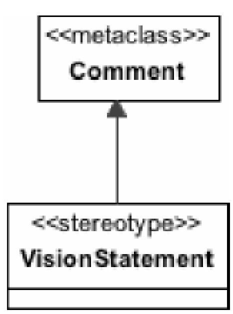


Figure 8.123 - Elements related to the VisionStatement stereotype

Extensions

The following are extensions for VisionStatement:

- Comment

Generalizations

The following are generalization relationships for VisionStatement:

- UPDMElement

8.1.1.1.6.1.12 Enumeration CompletionStatus

Possible completion statuses for capability realization.

EnumerationLiterals

- Complete
- Partial
- Minimal
- Undefined

8.1.1.1.6.1.13 Expose

SoaML:: A dependency between a service interface and a capability. The service interface exposes the capability.

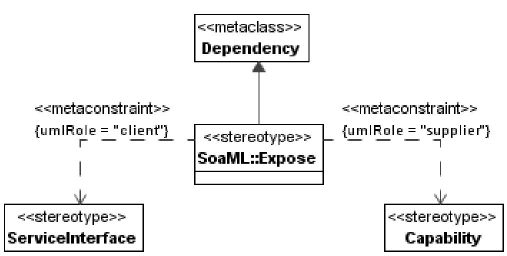


Figure 8.124 - Elements related to the Expose stereotype

Constraints

The following are constraints for Expose:

- Expose.client - Value for the client property must be stereotyped «ServiceInterface» or its specializations.
- Expose.supplier- Value for the client property must be stereotyped «Capability» or its specializations.

Extensions

The following are extensions for Expose:

- Dependency

8.1.1.1.7 UPDM L1 ::UPDM L0::Core::SystemsElements

Models in the System Viewpoint represent alternate realizations in terms of equipment capability of the operational capabilities expressed through models in the Operational Viewpoint and in the User Requirements. The System Viewpoint primarily addresses the specification of the system capability needed (rather than implementation details). Significant changes originally made in MODAF improved the ability for modelers to represent configuration of capability that include people as well as systems and platforms.

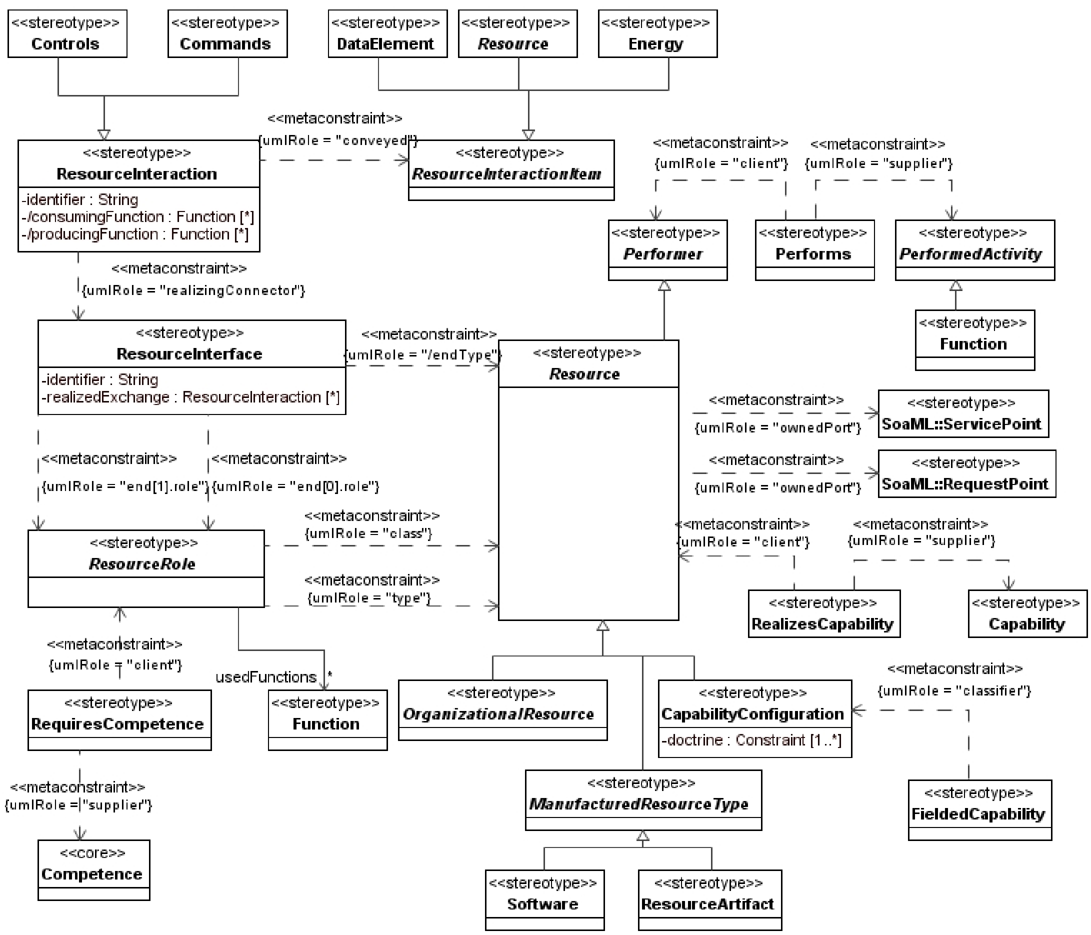


Figure 8.125 - Elements related to the SV-1 product

MODAF: Resource Interaction Specification (SV-1) address the composition and interaction of resources. From MODAF v1.1, SV-1 incorporates the human elements – Posts, Organizations, and Roles.

DoDAF: The Systems Interface Description (SV-1) DoDAF-described View addresses the composition and interaction of Systems. For DoDAF v2.0, the SV-1 incorporates the human elements as types of Performers - Organizations and Personnel Types.

The Resource Interaction Specification (SV-1) addresses the composition and interaction of resources. SV-1 now incorporates the human elements – Posts and Organizations. This view was previously known as the System Interface Description; the name change reflects the expanded scope of modeling in the solution space. The Resource Interaction

Specification (SV-1) links together the operational and systems architecture views by depicting how resources are structured and interact in order to realize the logical architecture specified in an OV-2. An SV-1 may represent the realization of a requirement specified in an OV-2 (i.e., in a to-be architecture), and so there may be many alternative SV configurations that could realize the operational requirement. Alternatively, in an as-is architecture, the OV-2 may simply be a simplified, logical representation of the SV-1 to allow communication of key information flows to non-technical stakeholders. A resource interaction is a simplified representation of a pathway or network, usually depicted graphically as a connector (i.e., a line with possible amplifying information).

The SV-1 depicts all interactions between resources that are of interest to the architect. Note that interactions between systems may be further specified in detail in SV-2 and SV-6. Sub-resource assemblies may be identified in SV-1 to any level (i.e., depth) of decomposition the architect sees fit. SV-1 may also identify the Physical Assets (e.g., Platforms) at which resources are deployed, and optionally overlay Operational Nodes that utilize those resources. In many cases, an operational node depicted in an OV-2 product may well be the logical representation of the resource that is shown in SV-1.

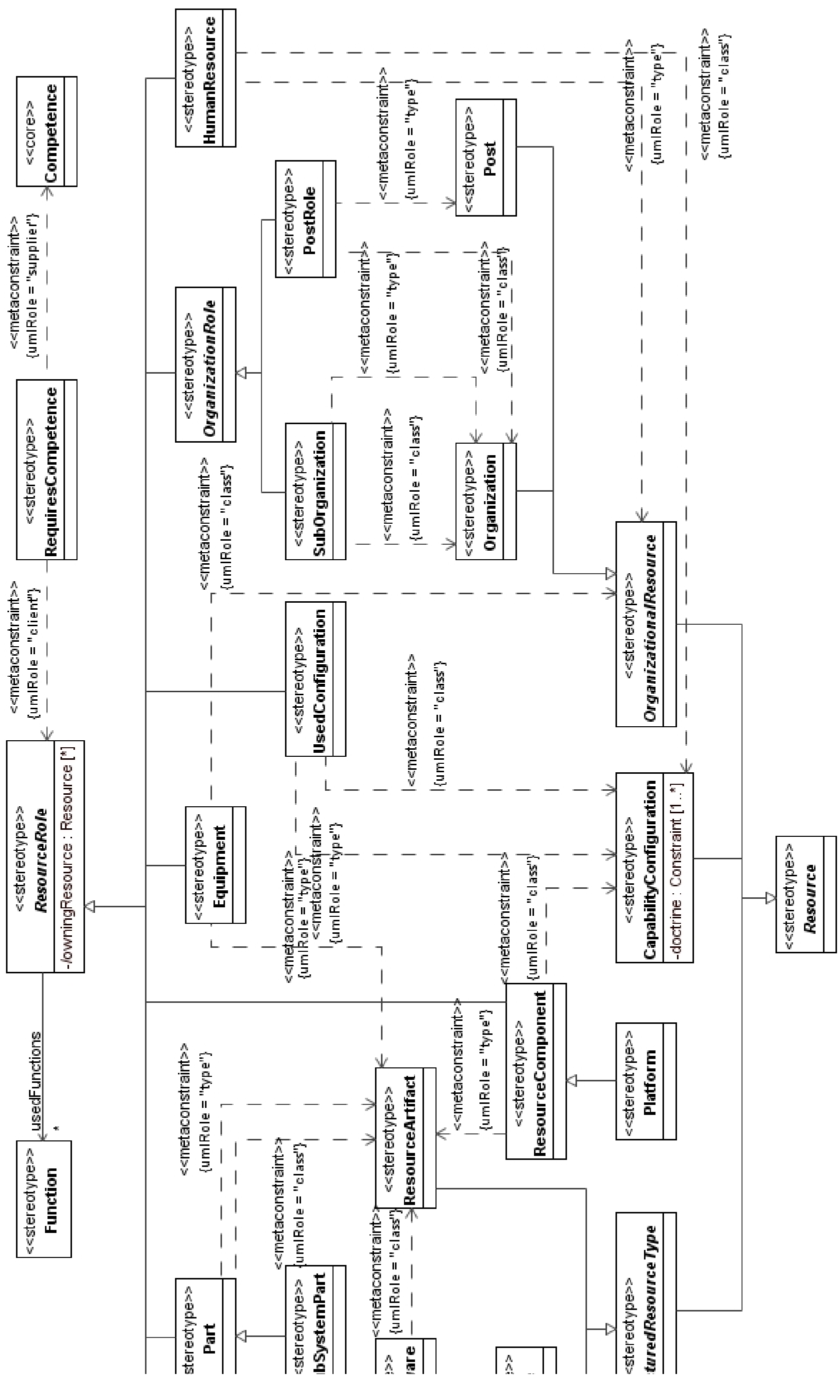


Figure 8.126 - Elements related to the SV-1 (Resources and ResourceRoles) product

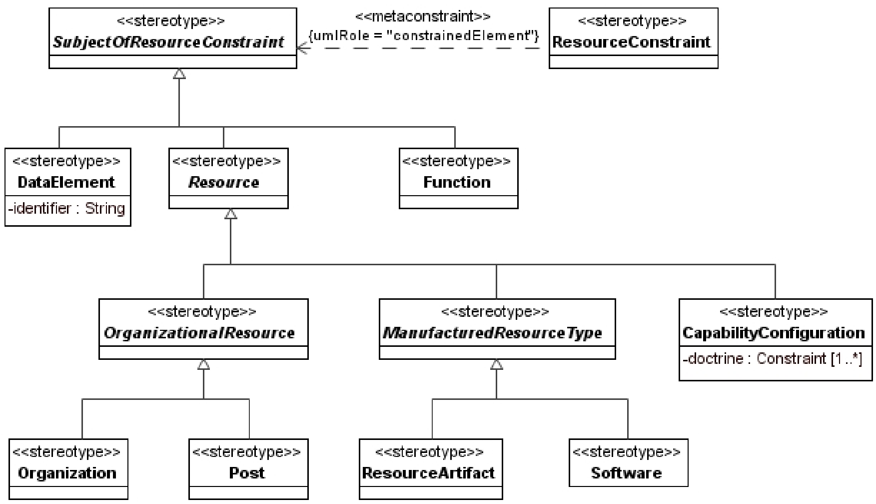


Figure 8.127 - Elements related to the SV-10a product

MODAF: The purpose of this Product is to specify functional and non-functional constraints on the implementation aspects of the architecture (i.e., the structural and behavioral elements of the SV viewpoint).

DoDAF: The SV-10a Systems Rules Model DoDAF-described View 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).

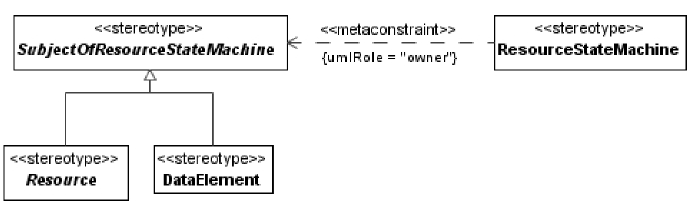


Figure 8.128 - Elements related to the SV-10b product

MODAF: The Resource State Transition Description is a graphical method of describing a resource (or function) response to various events by changing its state. The diagram basically represents the sets of events to which the Resources in the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

DoDAF: The Systems State Transition Description DoDAF-described View is a graphical method of describing a resource (or system function) response to various events by changing its state. The diagram basically represents the sets of events to which the resources in the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

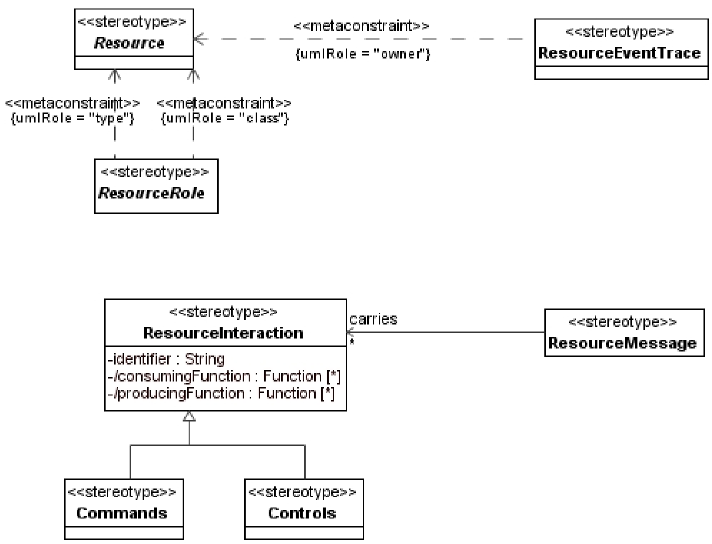


Figure 8.129 - Elements related to the SV-10c product

MODAF: The Resource Event-Trace Description 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.

DoDAF: The Systems Event-Trace Description provides a time-ordered examination of the interactions between functional resources. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

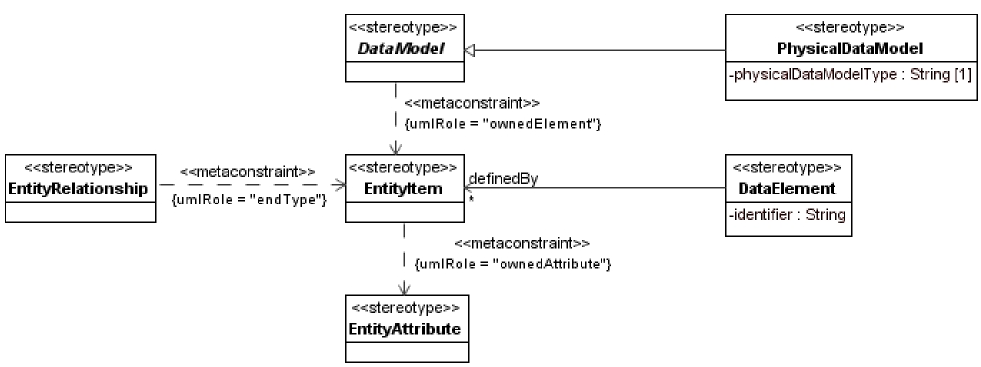


Figure 8.130 - Elements related to the SV-11 product

SV-11

MODAF: The SV-11 View defines the structure of the various kinds of system data that are utilized by the systems in the Architecture.

DoDAF: The DIV-3 Physical Data Model DoDAF-described view defines the structure of the various kinds of system or service data that are utilized by the systems or services in the Architecture.

The SV-11 View defines the structure of the various kinds of system data that are utilized by the systems in the Architecture. The Physical Schema is one of the Architectural Products closest to actual system design in the Framework. SV-11 is used to describe how the information represented in the Information Model (OV-7) is actually implemented. While the mapping between the logical and physical data models is relatively straightforward, the relationship between the components of each model (e.g., entity types in the logical model versus relational tables in the physical model) is frequently one-to-many or many-to-many.

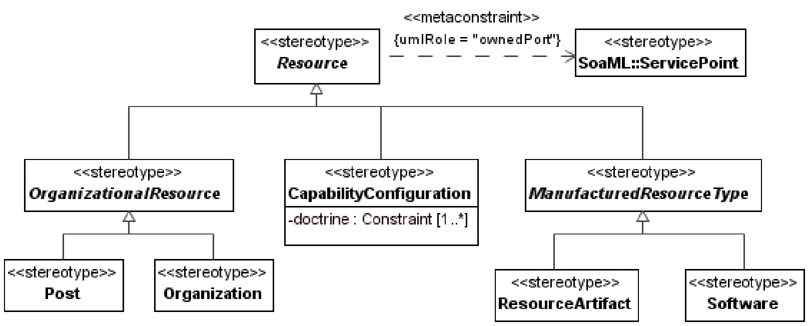


Figure 8.131 - Elements related to the SV-12 product

MODAF: The Service Provision View (SV-12) specifies configurations of resources that can deliver a service, and the levels of service those resources can deliver in different environments.

DoDAF: NA

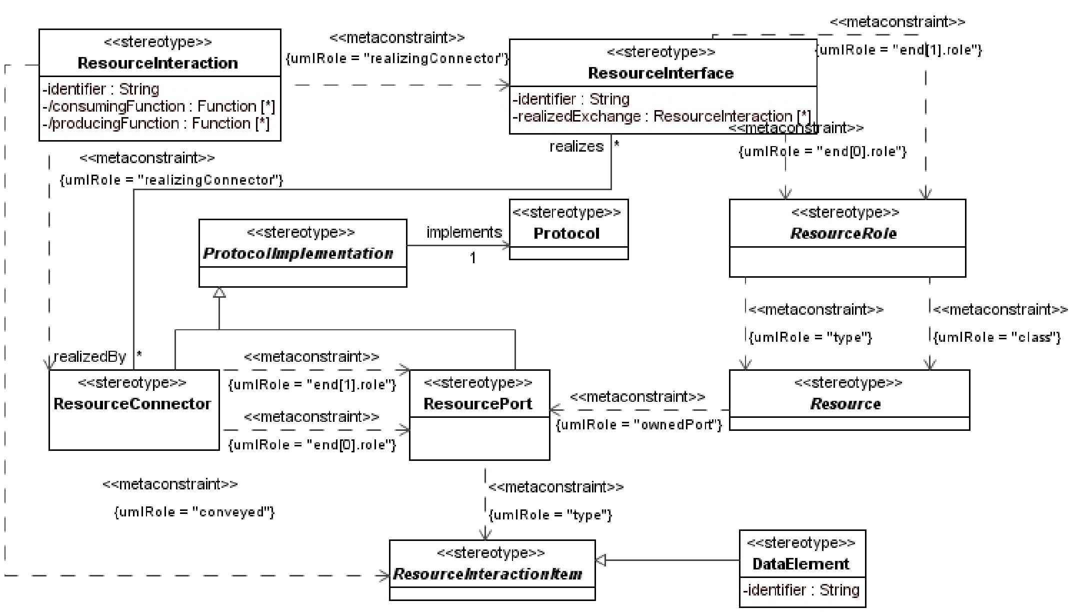


Figure 8.132 - Elements related to the SV-2 product

MODAF: The Systems Communications Description (SV-2a/2b/2c) series of views is intended for the representation of communications networks and pathways that link communications systems, and provides details regarding their configuration.

DoDAF: A Systems Resource Flow Description (SV-2) DoDAF-described View specifies the resource flows between Systems and may also list the protocol stacks used in connections.

The Systems Communications Description series of views specifies the communications networks and pathways that link systems, and provides details regarding their configuration. The networks and pathways documented through these views represent the physical implementation of the information needlines identified in an Operational Node Connectivity Description (OV-2). The SV-2 series focuses on the physical characteristics of each link, to include specification of such attributes as the geographic location of network components (e.g., routers, switches, amplifiers, and repeaters). Attributes such as capacities (e.g., bandwidth, throughput), frequencies used, security encryption methods used, and other descriptive information are usually presented in a corresponding SV-6 product (though most architecture tools would prompt the architect to enter such data as the SV-2 products are being developed).

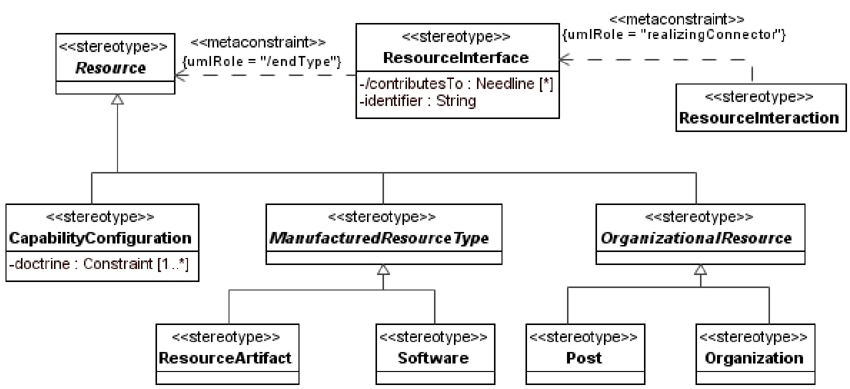


Figure 8.133 - Elements related to the SV-3 product

MODAF: The Resource Interaction Matrix provides a tabular summary of the resource interactions specified in the SV-1 for the Architecture.

DoDAF: The Systems – Systems Matrix (SV-3) DoDAF-described View provides a tabular summary of the system interactions specified in the SV-1 for the Architecture.

The Resource Interaction Matrix provides a tabular summary of the resource interactions specified in the SV- 1 for the Architecture. An SV-3 Product allows a quick overview of all the resource interactions specified in one or more SV-1 diagrams. The SV-3 can be organized in a number of ways to emphasize the association of groups of system pairs in context with the architecture’s purpose.

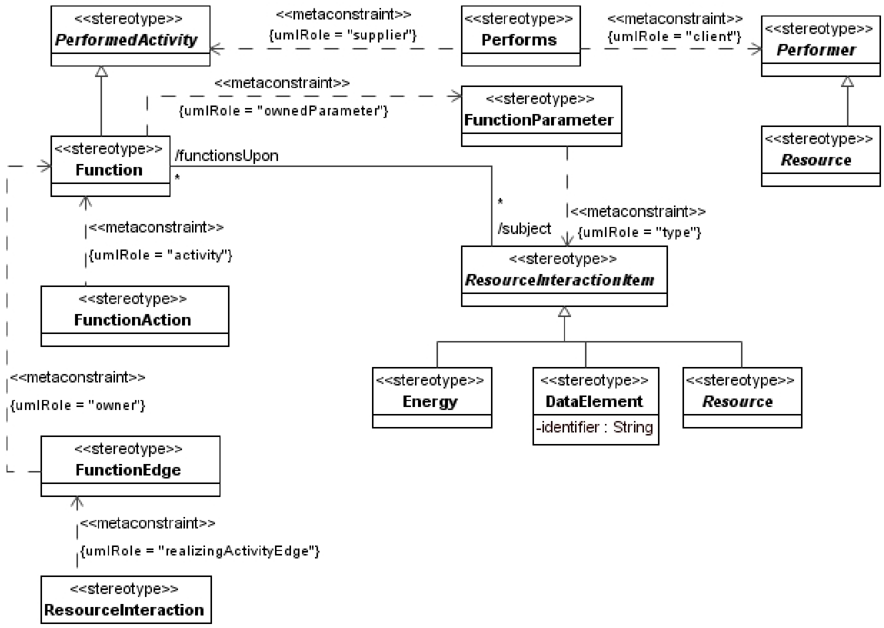


Figure 8.134 - Elements related to the SV-4 product

MODAF: Functionality Descriptions (SV-4) address human and system functionality.

DoDAF: The Systems Functionality Description (SV-4) DoDAF-described View addresses human and system functionality.

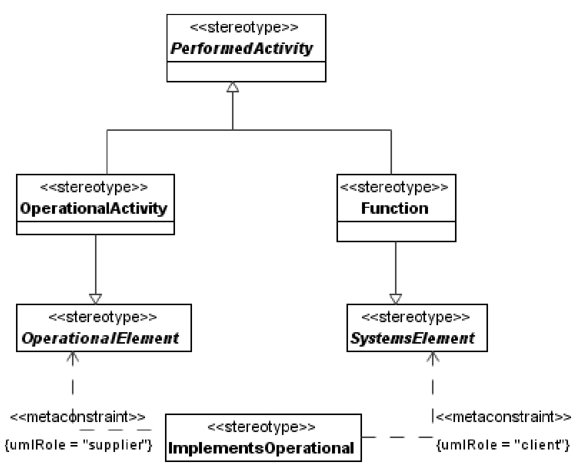


Figure 8.135 - Elements related to the SV-5 product

MODAF: This view has been expanded for the Service Orientated community by allowing for Service Functions as well as Operational Activities.

DoDAF: The Operational Activity to Systems Function Traceability Matrix (SV-5a) DoDAF-described View depicts the mapping of system functions (and, optionally, the capabilities and performers that provide them) to operational activities and thus identifies the transformation of an operational need into a purposeful action performed by a system or solution.

The Operational Activity to Systems Traceability Matrix (SV-5b) DoDAF-described View depicts the mapping of systems (and, optionally, the capabilities and performers that provide them) to operational activities and thus identifies the transformation of an operational need into a purposeful action performed by a system or solution.

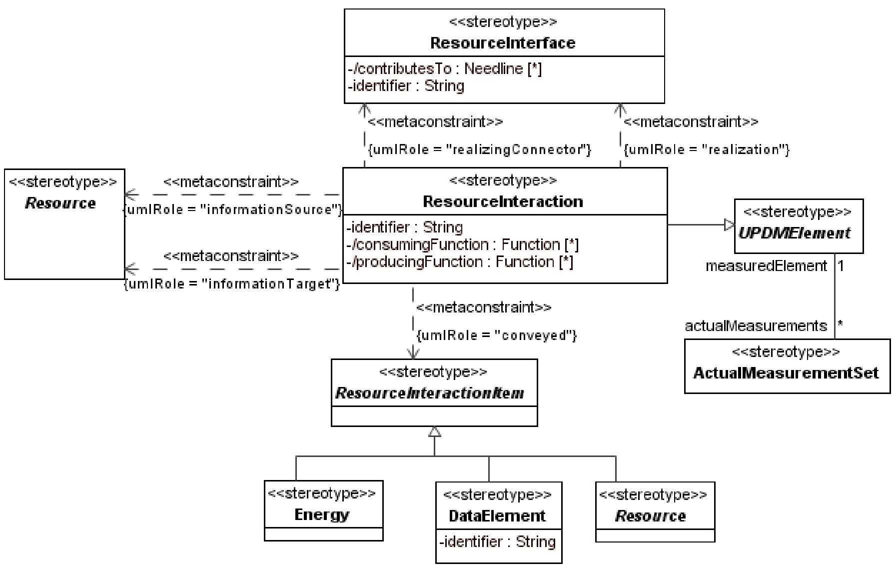


Figure 8.136 - Elements related to the SV-6 stereotype

MODAF: The Systems Data Exchange Matrix specifies the characteristics of the system data exchanged between systems. The focus is on data crossing the system boundary.

DoDAF: The Systems Resource Flow Exchange Matrix DoDAF-described View specifies the characteristics of the system resource flows exchanged between systems. The focus is on resource crossing the system boundary.

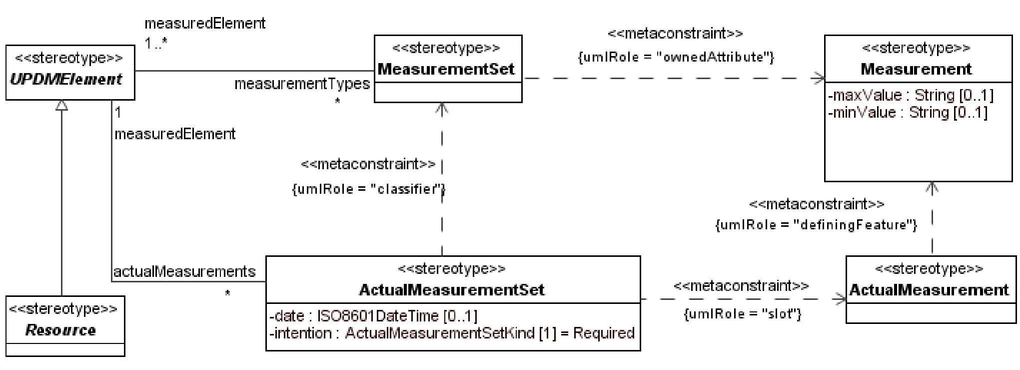


Figure 8.137 - Elements related to the SV-7 product

MODAF: The SV-7 is the Resource Performance Parameters Matrix and depicts the performance characteristics of a Resource (e.g., system, role, or capability configuration).

DoDAF: The SV-7 DoDAF-described View is the Systems Measures Matrix and depicts the measures (metrics) of resources.

The SV-7 is the Resource Performance Parameters Matrix and depicts the performance characteristics of a Functional Resource (system, role, or capability configuration). The Resource Performance Parameters Matrix expands on the information presented in an SV-1 by depicting the characteristics of the Functional Resources shown in the SV-1. The Resource Performance Parameters Matrix View specifies qualitative & quantitative characteristics of functional resources. It specifies the performance parameters of each resource. The performance parameters are selected by the architect and end user community.

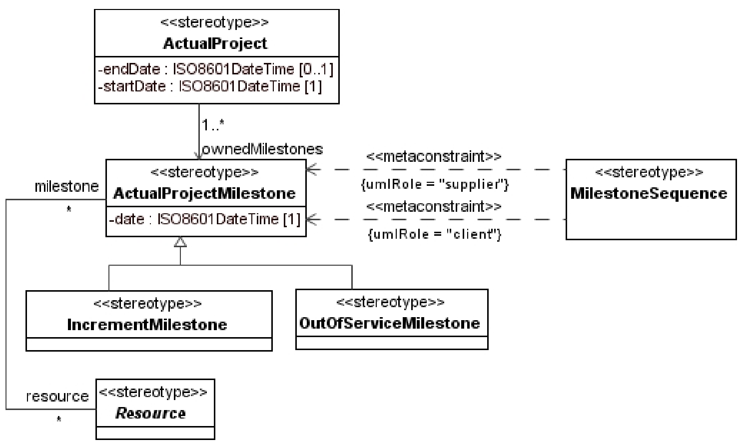


Figure 8.138 - Elements related to the SV-8 product

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

DoDAF: The Systems Evolution Description DoDAF-described View presents a whole lifecycle view of resources (systems), describing how it changes over time. It shows the structure of several resources mapped against a timeline.

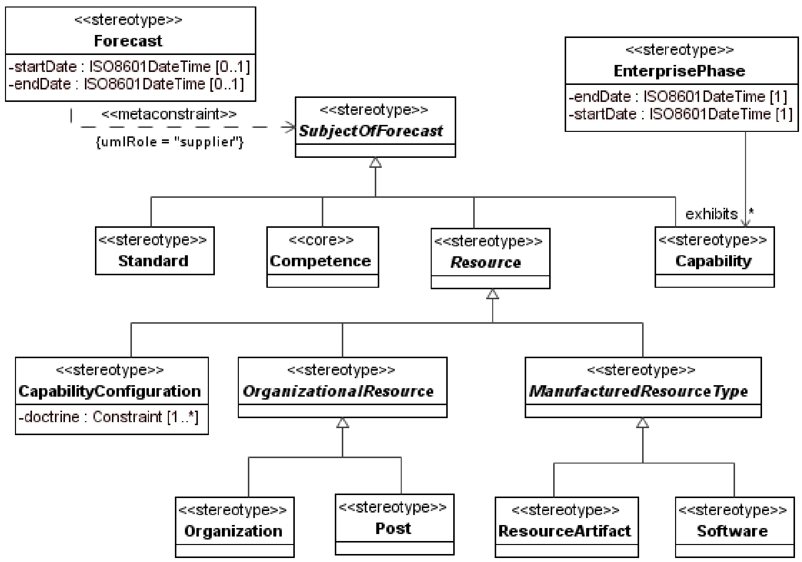


Figure 8.139 - Elements related to the SV-9 product

MODAF: The Technology & Skills Forecast defines the underlying current and expected supporting technologies and skills. Expected supporting technologies and skills are those that can be reasonably forecast given the current state of technology and skills, and expected improvements / trends. New technologies and skills will be tied to specific time periods, which can correlate against the time periods used in SV-8 milestones and linked to Enterprise Phases.

DoDAF: The Technology & Skills Forecast defines the underlying current and expected supporting technologies and skills. Expected supporting technologies and skills are those that can be reasonably forecast given the current state of technology and skills, and expected improvements / trends. New technologies and skills will be tied to specific time periods, which can correlate against the time periods used in SV-8 milestones and linked to Enterprise Phases.

The Technology & Skills Forecast defines the underlying current and expected supporting technologies and skills. Expected supporting technologies and skills are those that can be reasonably forecast given the current state of technology and skills, and expected improvements / trends. New technologies and skills will be tied to specific time periods, which can correlate against the time periods used in SV-8 milestones and linked to EnterprisePhases. SV-9 provides a summary of emerging technologies and skills that impact the Resources that constitute the Architecture. The SV-9 provides descriptions of relevant: emerging capabilities, industry trends, predictions of the availability and readiness of specific hardware and software products, and current and possible future skills 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 architecture.

8.1.1.1.7.1 UPDM L1 ::UPDM L0::Core::SystemsElements::Behavior

The Behavior section of the SystemsElements profile.

8.1.1.1.7.1.1 Function

Defines the behavior of a System or OperationalActivityRealization as an activity in terms of tasks that is performed by the resource.

MODAF:An activity which is specified in context of the resource (human or machine) that performs it.

Note – Contrast with “OperationalActivity,” where the actor performing the activity is not known (i.e., it is just a logical node). A “Function” is implementation-specific.

Note – Should the “Function” be specific to one usage of a type of system, then the usageContext is specified by a reference to the composite structure property “ResourceComposition” typed by the system.

DoDAF: Activity: Work, not specific to a single organization, weapon system, or individual that transforms inputs (Resources) into outputs (Resources) or changes their state.

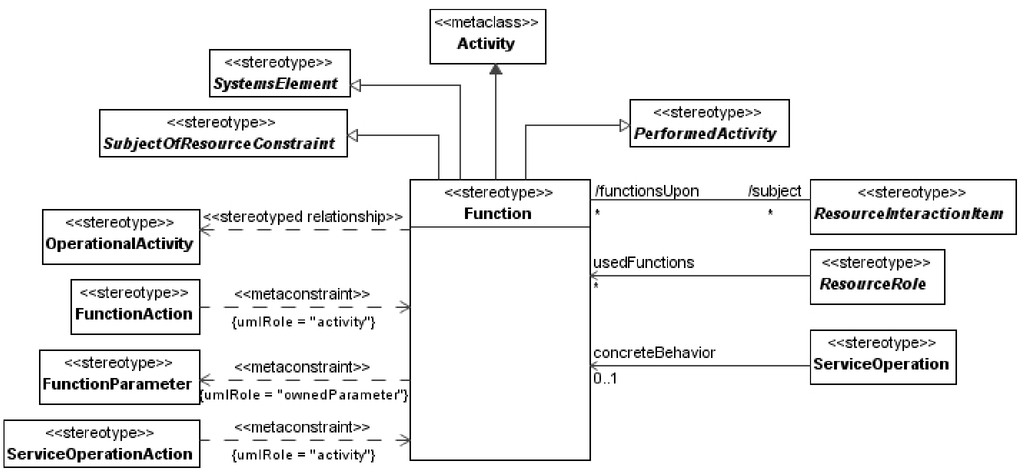


Figure 8.140 - Elements related to the Function stereotype

Constraints

The following are constraints for Function:

- Function.ownedParameter - The values for the ownedParameter property must be stereotyped «FunctionParameter».

Attributes

The following are attributes of Function:

- / subject : ResourceInteractionItem[*] - A list of all the ActivitySubjects that are acted /f unctioned upon by the OperationalActivity/Function. The list is derived by looking at all the ObjectNodes scoped within the OperationalActivity (directly or indirectly via StructuredActivityNodes) and getting a unique list of their types.

Extensions

The following are extensions for Function:

- Activity

Generalizations

The following are generalization relationships for Function:

- PerformedActivity

- SubjectOfResourceConstraint
- SubjectOfResourceStateMachine

8.1.1.1.7.1.2 FunctionAction

UPDM Artifact: The FunctionAction is defined as a call behavior action that invokes the function that needs to be performed. This concept is required for mapping the architecture with UML and does not have a DoDAF or MoDAF equivalent.

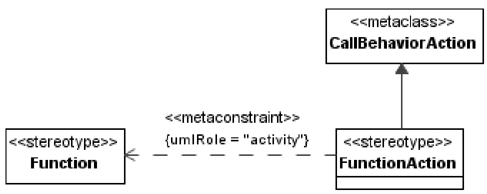


Figure 8.141 - Elements related to the FunctionAction stereotype

Extensions

The following are extensions for FunctionAction:

- CallBehaviorAction

Generalizations

The following are generalization relationships for FunctionAction:

- UPDMElement

8.1.1.1.7.1.3 FunctionEdge

The FunctionEdge is represented as the ActivityEdge that is associated with the Function Activity and also specifies the resource interaction Items that are carried by the activity edge.

UPDM: An extension of «ActivityEdge» that is used to model the flow of control/objects through a Function.

MODAF: A FunctionEdge (MODAF::FunctionFlow) is a UML::ObjectFlow between Functions.

Note – This has been extended in UPDM to include UML::ControlFlows.

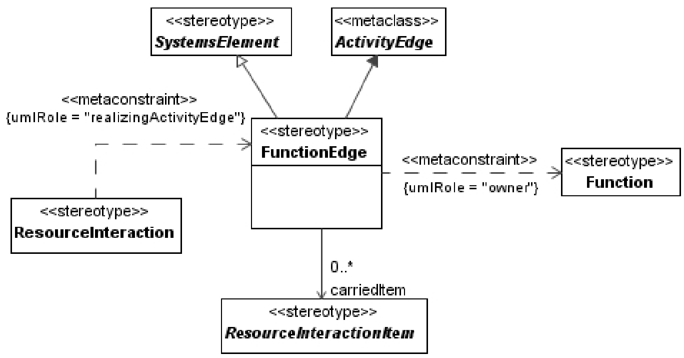


Figure 8.142 - Elements related to the FunctionEdge stereotype

Constraints

The following are constraints for FunctionEdge:

- FunctionEdge.owner - «FunctionEdge» must be owned directly or indirectly by «Function».

Attributes

The following are attributes of FunctionEdge:

- carriedItem : ResourceInteractionItem[0. .*] - The ResourceInteractionItem that is conveyed.

Extensions

The following are extensions for FunctionEdge:

- ActivityEdge

Generalizations

The following are generalization relationships for FunctionEdge:

- UPDMElement

8.1.1.1.7.1.4 FunctionParameter

UPDM: FunctionParameter represents inputs and outputs of Function. It is typed by ResourceInteractionItem.

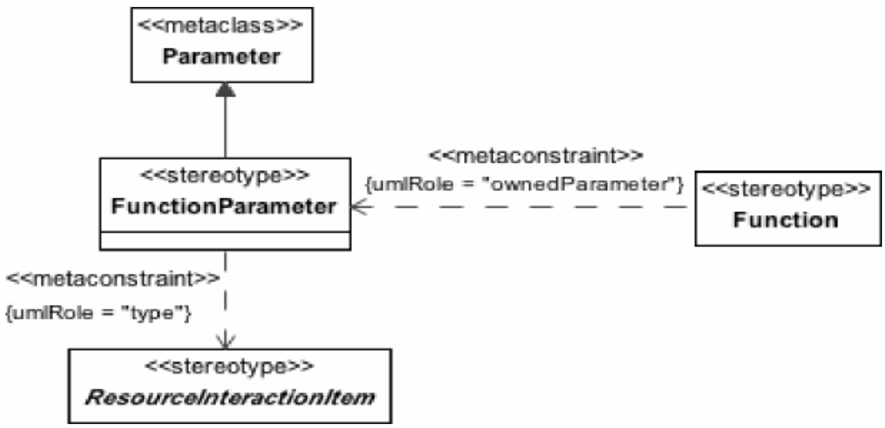


Figure 8.143 - Elements related to the FunctionParameter stereotype

Constraints

The following are constraints for FunctionParameter:

- FunctionParameter.type - Value for the type property must be stereotyped with specialization of “ResourceInteractionItem.”

Extensions

The following are extensions for FunctionParameter:

- Parameter

Generalizations

The following are generalization relationships for FunctionParameter:

- UPDMElement

8.1.1.1.7.1.5 ResourceStateMachine

UPDM Artifact that extends a UML StateMachine applied to Resources.

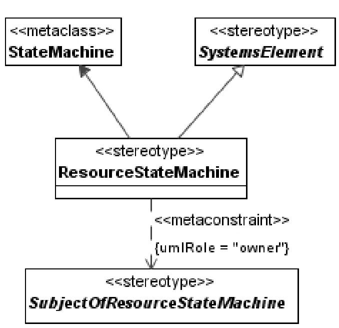


Figure 8.144 - Elements related to the ResourceStateMachine stereotype

Extensions

The following are extensions for ResourceStateMachine:

- StateMachine

Generalizations

The following are generalization relationships for ResourceStateMachine:

- UPDMElement

8.1.1.1.7.1.6 SubjectOfResourceStateMachine

UPDM Abstract Element: The element being described by the state machine.

Note – SubjectOfResourceStateMachine is abstract.

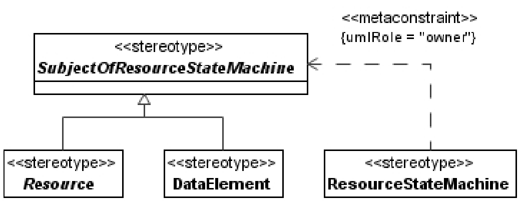


Figure 8.145 - Elements related to the abstract SubjectOfResourceStateMachine stereotype

Constraints

The following are constraints for SubjectOfResourceStateMachine:

- SubjectOfResourceStateMachine.owned state machines - Owned state machines must be stereotyped «ResourceStateMachine» or its specializations.

Generalizations

The following are generalization relationships for SubjectOfResourceStateMachine:

- UPDMElement

8.1.1.1.7.2 UPDM L1 ::UPDM L0::Core::SystemsElements::Data

The Data section of the SystemsElements profile.

8.1.1.1.7.2.1 DataElement

MODAF: A formalized representation of data that is managed by or exchanged between systems.

DoDAF: (DoDAF::Data): Representation of information in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means.

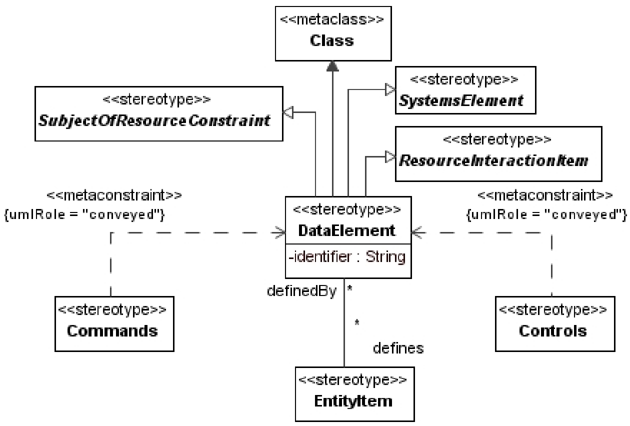


Figure 8.146 - Elements related to the DataElement stereotype

Attributes

The following are attributes of DataElement:

- defines : EntityItem[*] - The list of EntityItems that are defined by the DataElement.
- Identifier : String - Identifier

Extensions

The following are extensions for DataElement:

- Class

Generalizations

The following are generalization relationships for DataElement:

- SubjectOfResourceConstraint
- ResourceInteractionItem
- SubjectOfResourceStateMachine

8.1.1.1.7.2.2 DataModel

MODAF: A structural specification of data, showing classifications of data elements and relationships between them.

DoDAF: NA

Note – DataModel is abstract.

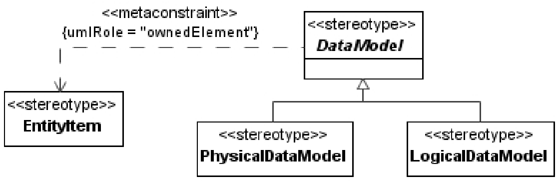


Figure 8.147 - Elements related to the abstract DataModel stereotype

Constraints

The following are constraints for DataModel:

- DataModel.ownedElement - All classifiers owned by DataModel must be stereotyped «EntityItem».

Generalizations

The following are generalization relationships for DataModel:

- UPDMElement

8.1.1.1.7.2.3 PhysicalDataModel

MODAF: A PhysicalDataModel is an implementable specification of a data structure. A PhysicalDataModel realizes a LogicalDataModel, taking into account implementation restrictions and performance issues while still enforcing the constraints, relationships, and typing of the logical model.

DoDAF: A Physical Data Model defines the structure of the various kinds of system or service data that are utilized by the systems or services in the Architecture.

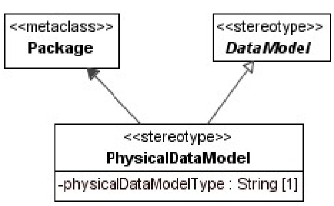


Figure 8.148 - Elements related to the PhysicalDataModel stereotype

Attributes

The following are attributes of PhysicalDataModel:

- physicalDataModelType : String[1] - Uniquely identifies the type of the internal data model.

Extensions

The following are extensions for PhysicalDataModel:

- Package

Generalizations

The following are generalization relationships for PhysicalDataModel:

- DataModel

8.1.1.1.7.2.4 InternalDataModel

DoDAF alias for PhysicalDataModel

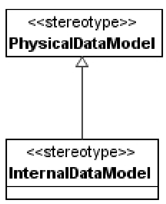


Figure 8.149 - Elements related to the InternalDataModel stereotype

Extensions

The following are extensions for InternalDataModel:

- Package

Generalizations

The following are generalization relationships for InternalDataModel:

- PhysicalDataModel

8.1.1.1.7.3 UPDM L1 ::UPDM L0::Core::SystemsElements::Flows

The Flows section of the SystemsElements profile.

8.1.1.1.7.3.1 ResourceInteraction

UPDM: ResourceInteraction represents data that is exchanged between the resources.

MODAF: An assertion that two FunctionalResources interact. Examples: data exchange between systems, conversations between people, people using systems.

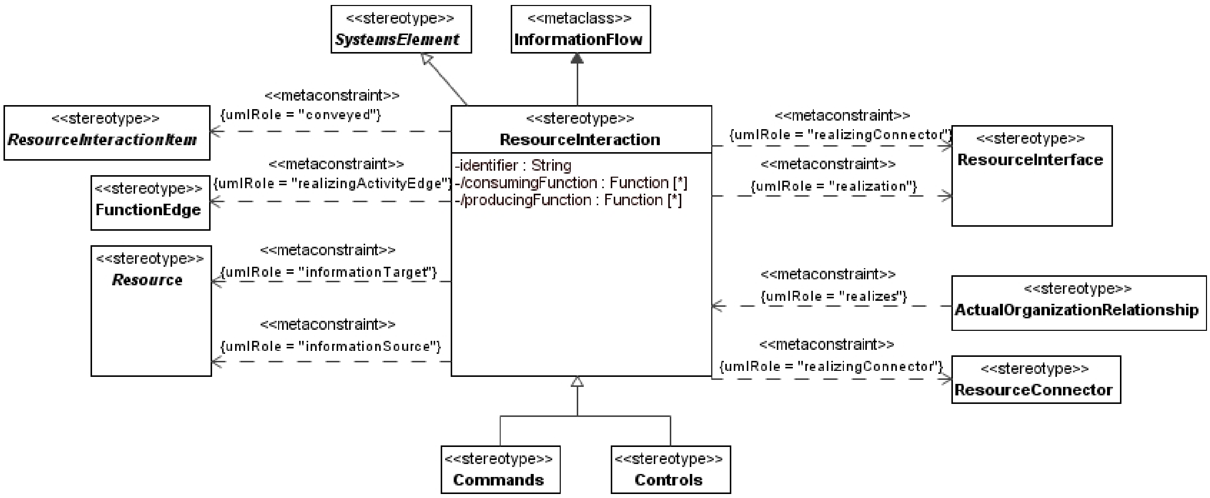


Figure 8.150 - Elements related to the ResourceInteraction stereotype

Attributes

The following are attributes for ResourceInteraction:

- identifier : String – Identifier
- /consumingFunction : Function [*] – A list of Functions that consume a ResourceInteraction via FunctionEdges and their directly/indirectly connected FunctionActions. This is derived as follows:
  - ResourceInteraction to FunctionEdge via the “realizingMessage” role.
  - FunctionEdge to FunctionAction via the “target” role (if the target is a pin on a FunctionAction, another jump is required from the pin to the action via an inverse navigation of the action’s “argument” role).
  - FunctionAction to Function via the “behavior” role.
- /producingFunction : Function [*] – A list of Functions that produce a ResourceInteraction via FunctionEdges and their directly/indirectly connected FunctionActions. This is derived as follows:
  - ResourceInteraction to FunctionEdge via the “realizingMessage” role.
  - FunctionEdge to FunctionAction via the “source” role (if the source is a pin on a FunctionAction, another jump is required from the pin to the action via an inverse navigation of the action’s “result” role).
  - FunctionAction to Function via the “behavior” role.
- /producingFunction: Function [*] - Producing Functions
- /consumingFunction: Function [*] - Consuming Functions

Constraints

The following are constraints for ResourceInteraction:

- ResourceInteraction.realizingConnector - Value for the realizingConnector property must be stereotyped «ResourceInterface» or its specializations.

- ResourceInteraction.realization - Value for the realization property must be stereotyped «ResourceInterface» or its specializations.
- ResourceInteraction.conveyedElement - Value for the conveyedElement property must be stereotyped «ResourceInteractionItem» or its specializations.
- ResourceInteraction.realizingActivityEdge - Value for the realizingActivityEdge property must be stereotyped «FunctionEdge» or its specializations.
- ResourceInteraction.informationSource - Value for the informationSource property must be stereotyped «Resource» or its specializations.
- ResourceInteraction.informationTarget - Value for the informationTarget property must be stereotyped «Resource» or its specializations.

Extensions

The following are extensions for ResourceInteraction:

- InformationFlow

Generalizations

The following are generalization relationships for ResourceInteraction:

- UPDMElement

8.1.1.1.7.3.2 ResourceInteractionItem

UPDM Abstract: Represents the item(s) exchanged between the resources through a ResourceInteraction.

MODAF: Formalized representation of data that is managed by or exchanged between systems (MODAF::DataElement).

DoDAF: Representation of information in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means (DoDAF::Data).

Note – ResourceInteractionItem is abstract.

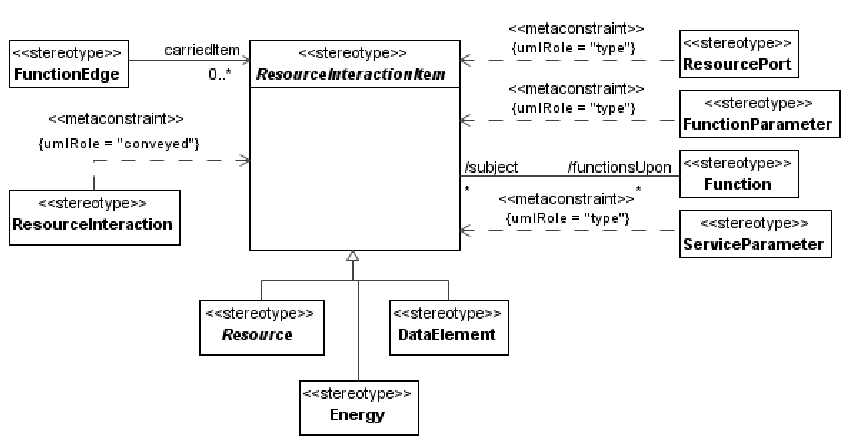


Figure 8.151 - Elements related to the abstract ResourceInteractionItem stereotype

Attributes

The following are attributes of ResourceInteractionItem:

- functionsUpon : Function[*] – The Functions affected by the ResourceInteractionItem.
- functions Upon : Function[*] – A list of all the Functions that act upon the ActivitySubject. This is the inverse of the Function “subject” role.

Generalizations

The following are generalization relationships for ResourceInteractionItem:

- UPDMElement

8.1.1.1.7.3.3 OperationalMessage

UPDM: Message for use in an Operational Event-Trace which carries any of the subtypes of OperationalExchange. This is used to provide additional information about OperationalMessages for display on an OV-6c.

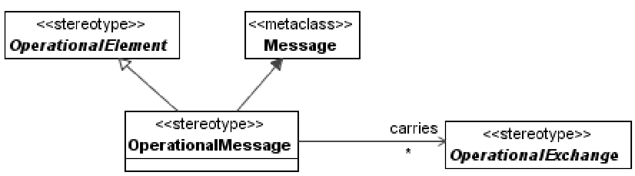


Figure 8.152 - Elements related to the OperationalMessage stereotype

Attributes

The following are attributes of OperationalMessage:

- carries : OperationalExchange[*] - A list of realized OperationalExchanges/ResourceInteractions that are realized by the message. This is derived by doing the inverse of the UML “realizingMessage” role, that are owned by InformationFlow.

Extensions

The following are extensions for OperationalMessage:

- Message

Generalizations

The following are generalization relationships for OperationalMessage:

- OperationalElement

8.1.1.1.7.3.4 ServiceMessage

UPDM: Message for use in a Service Interaction Specification, implements a resourceInteraction or any of the subtypes.

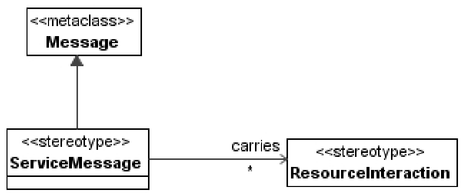


Figure 8.153 - Elements related to the ServicelMessage stereotype

Attributes

The following are attributes of ServiceMessage:

- carries : OperationalExchange[*] - A list of realized OperationalExchanges/ResourceInteractions that are realized by the message. This is derived by doing the inverse of the UML “realizingMessage” role, that are owned by InformationFlow.

Extensions

The following are extensions for ServiceMessage:

- Message

8.1.1.1.7.3.5 ResourceMessage

UPDM: Message for use in a Resource Event-Trace, implements a ResourceInteraction.

MODAF: A specification of the interactions between aspects of a Resources architecture (MODAF::ResourceInteractionSpecification).

DoDAF: An overlap of an Activity with a Resource, in particular a consuming or producing Activity that expresses an input, output, consumption, or production Activity of the Resource (DoDAF:: activityResourceOverlap).

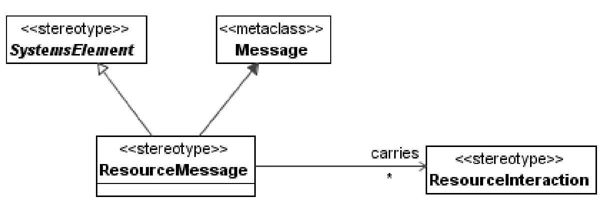


Figure 8.154 - Elements related to the ResourceMessage stereotype

Attributes

The following are attributes of ResourceMessage:

- carries : OperationalExchange[*] - A list of realized OperationalExchanges/ResourceInteractions that are realized by the message. This is derived by doing the inverse of the UML “realizingMessage” role, that are owned by InformationFlow.

Extensions

The following are extensions for ResourceMessage:

- Message

Generalizations

The following are generalization relationships for ResourceMessage:

- SystemsElement

8.1.1.1.7.3.6 ResourceConnector

UPDM: A physical connection between two resources that implements protocols through which the source resource can transmit items to the destination resource.

MODAF: (SystemPortConnector): Asserts that a connection exists between two ports belonging to parts in a system composite structure model.

DoDAF: NA

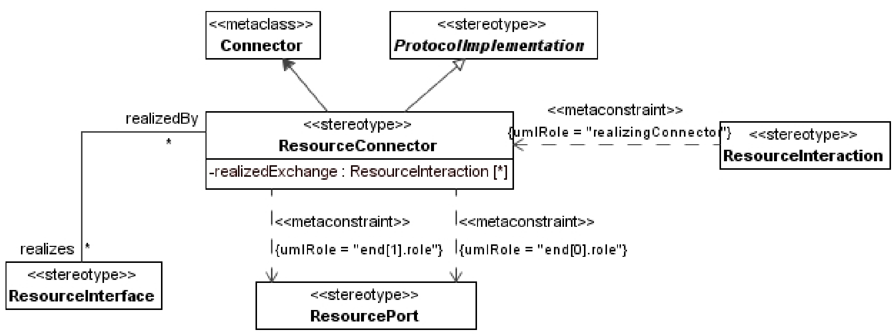


Figure 8.155 - Elements related to the ResourceConnector stereotype

Constraints

The following are constraints for ResourceConnector:

- ResourceConnector.end - The value for the role property for the owned ConnectorEnd must be stereotype «ResourcePort» or its specializations.

Attributes

The following are attributes for ResourceConnector:

- realizes: ResourceInterface[*] – Realized ResourceInterfaces
- realizedExchange: ResourceInteraction[*] - A list of ResourceInteractions (or specializations) that realized by the ResourceInterface/ResourceConnector. This is derived by navigating from the ResourceInteraction to the ResourceInterfaces/ResourceConnectors using the inverse of the realization/realizingConnector roles.

Extensions

The following are extensions for ResourceConnector:

- Connector

Generalizations

The following are generalization relationships for ResourceConnector:

- ProtocolImplementation

8.1.1.1.7.4 UPDM L1 ::UPDM L0::Core::SystemsElements::Structure

The Structure section of the SystemsElements profile.

8.1.1.1.7.4.1 ResourceArtifact

UPDM: A combination of physical element, energy, and data that are combined used to accomplish a task or function.

MODAF: A type of man-made object. Examples are “car,” “radio,” “fuel.” (MODAF:: Artefact)

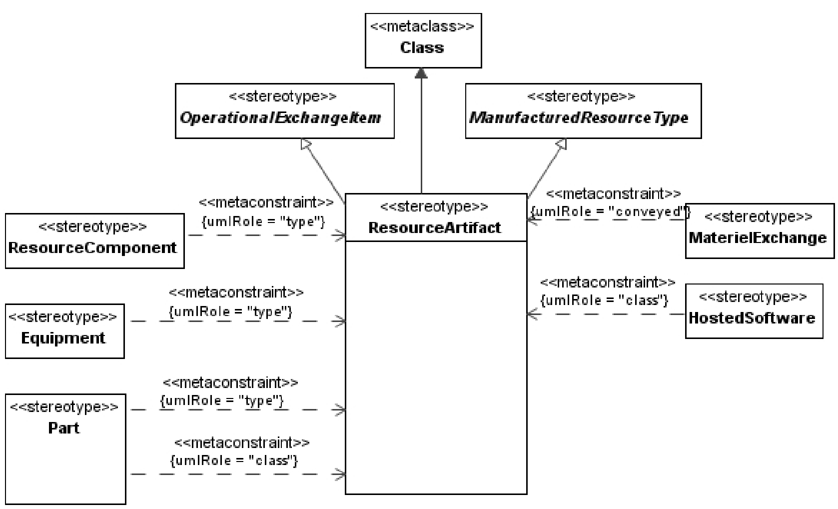


Figure 8.156 - Elements related to the ResourceArtifact stereotype

Extensions

The following are extensions for ResourceArtifact:

- Class

Generalizations

The following are generalization relationships for ResourceArtifact:

- OperationalExchangeItem
- ManufacturedResourceType

8.1.1.1.7.4.2 CapabilityConfiguration

MODAF: A composite structure representing the physical and human resources (and their interactions) in an enterprise. A CapabilityConfiguration is a set of artifacts or an organization configured to provide a capability, and should be guided by [doctrine] that may take the form of Standard or OperationalConstraint stereotypes.

DoDAF: NA

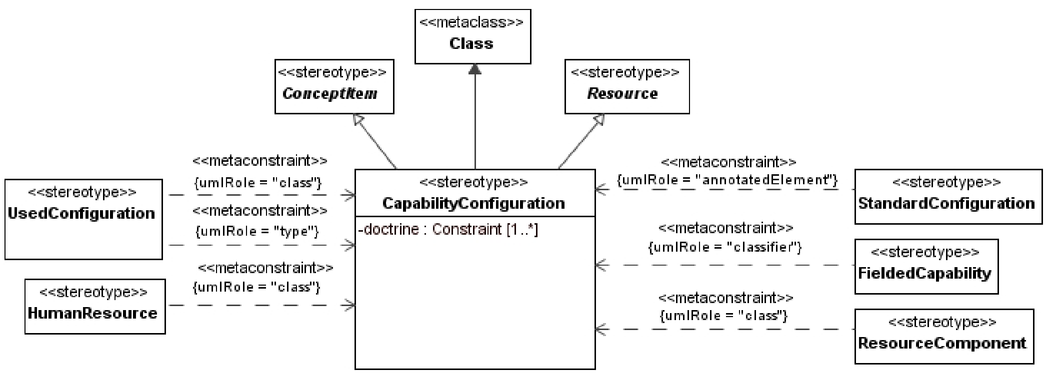


Figure 8.157 - Elements related to the CapabilityConfiguration stereotype

Extensions

The following are extensions for CapabilityConfiguration:

- Class

Generalizations

The following are generalization relationships for CapabilityConfiguration:

- ResourceType
- ConceptItem

8.1.1.1.7.4.3 ResourceComponent

UPDM: A well defined resource that is used by a CapabilityConfiguration to accomplish a capability.

MODAF: Usage of an Artifact as a component of a ResourceConfiguration (MODAF::PhysicalAsset).

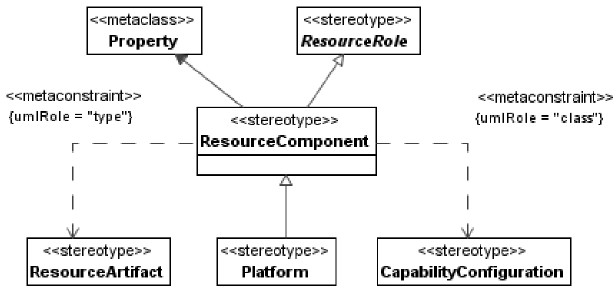


Figure 8.158 - Elements related to the ResourceComponent stereotype

Constraints

The following are constraints for ResourceComponent:

- ResourceComponent.class - Value for the class property must be stereotyped «CapabilityConfiguration» or its specializations.
- ResourceComponent.type - Value for the type property must be stereotyped «Artifact» or its specializations.

Extensions

The following are extensions for ResourceComponent:

- Property

Generalizations

The following are generalization relationships for ResourceComponent:

- ResourceRole

8.1.1.1.7.4.4 Equipment

UPDM: Equipment is a physical resource that is used to accomplish a task or function in a system or an environment.

MODAF: (MODAF::PhysicalAsset): Usage of a ResourceArtifact (MODAF::Artefact) as a component of a ResourceConfiguration.

DoDAF: NA

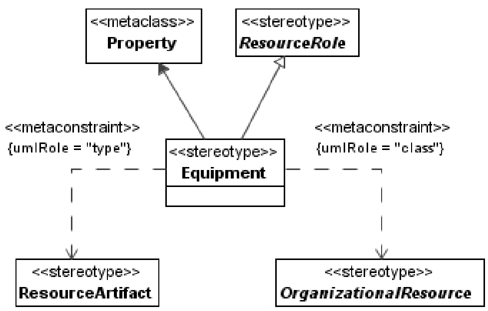


Figure 8.159 - Elements related to the Equipment stereotype

Constraints

The following are constraints for Equipment:

- Equipment.type - Value for the type property must be stereotyped «ResourceArtifact» or its specializations.
- Equipment.class - Value for the class property must be stereotyped «OrganizationalResource» or its specializations.

Extensions

The following are extensions for Equipment:

- Property

Generalizations

The following are generalization relationships for Equipment:

- ResourceRole

8.1.1.1.7.4.5 FieldedCapability

FieldCapability is an instance of a configured capability that actually represents the capability that is deployed and fully realized. FieldCapability extends InstanceSpecification to denote that it is an instance of the configured capability it is associated to.

MODAF: An actual, fully-realized capability. A “FieldedCapability” must indicate its configuration “CapabilityConfiguration.”

Example: “HMS Iron Duke, configured and crewed, operating under the appropriate doctrine.”

Note – the “CapabilityConfiguration” that this realizes would specify a Type 23 Frigate, the crew, the weapons systems.

DODAF: N/A

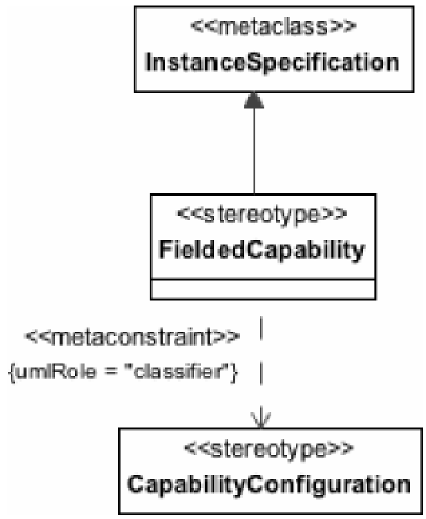


Figure 8.160 - Elements related to the FieldedCapability stereotype

Constraints

The following are constraints for FieldedCapability:

- FieldedCapability.classifier - Value for the classifier property must be stereotyped «CapabilityConfiguration» or its specializations.

Extensions

The following are extensions for FieldedCapability:

- InstanceSpecification

Generalizations

The following are generalization relationships for FieldedCapability:

- UPDMElement

8.1.1.1.7.4.6 Forecast

A Forecast describes the actual or predicted status of a System at a Project Milestone - i.e., a point in the lifecycle of the system. It can be a statement about the future state of one or more types of System or TechnicalStandardsForecast. The Forecast is effective for a given timePeriod.

MODAF: A statement about the future state of one or more types of system or standard.

Note – This is an "EffectivityConstrainedItem" (i.e., the forecast is effective for a given period).

DoDAF: A description of merging technologies and software/hardware products that are expected to be available in a given set of time frames and that will affect future development of the architecture.

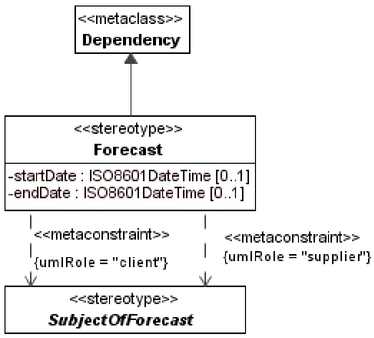


Figure 8.161 - Elements related to the Forecast stereotype

Constraints

The following are constraints for Forecast:

- Forecast.client - Value for the client property must be stereotyped «SubjectOfForecast » or its specializations.
- Forecast.supplier- Value for the supplier property must be stereotyped «SubjectOfForecast» or its specializations.
- Forecast.pair - The client and supplier must be stereotyped by the same specialization of «SubjectOfForecast» (e.g., «Software» to «Software», «Standard» to «Standard»).

Attributes

The following are attributes of Forecast:

- startDate : ISO8601DateTime[0..1] - Start date of the forecast.
- endDate : ISO8601 DateTime[0..1] - End date of the forecast.

Extensions

The following are extensions for Forecast:

- Dependency

8.1.1.1.7.4.7 HostedSoftware

MODAF: Asserts that Software is hosted on a ResourceArtifact (MODAF::Artefact), which means the artifact is some kind of computer system.

DoDAF: NA – covered by the more general temporalWholePart element.

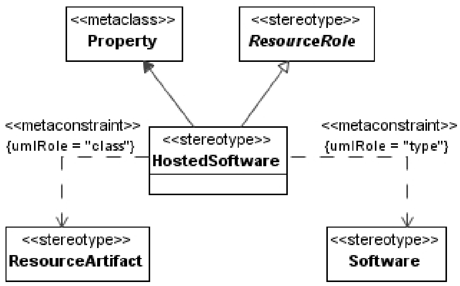


Figure 8.162 - Elements related to the HostedSoftware stereotype

Constraints

The following are constraints for HostedSoftware:

- HostedSoftware.type - Value for the type property must be stereotyped «Software» or its specializations.
- HostedSoftware.class - Value for the class property must be stereotyped «ResourceArtifact» or its specializations.

Extensions

The following are extensions for HostedSoftware:

- Property

Generalizations

The following are generalization relationships for HostedSoftware:

- ResourceRole

8.1.1.1.7.4.8 HumanResource

MODAF: The role of a Post (MODAF::PostType) or Organization (MODAF::OrganizationType) in a CapabilityConfiguration.

DoDAF: NA – covered by the more general temporalWholePart element.

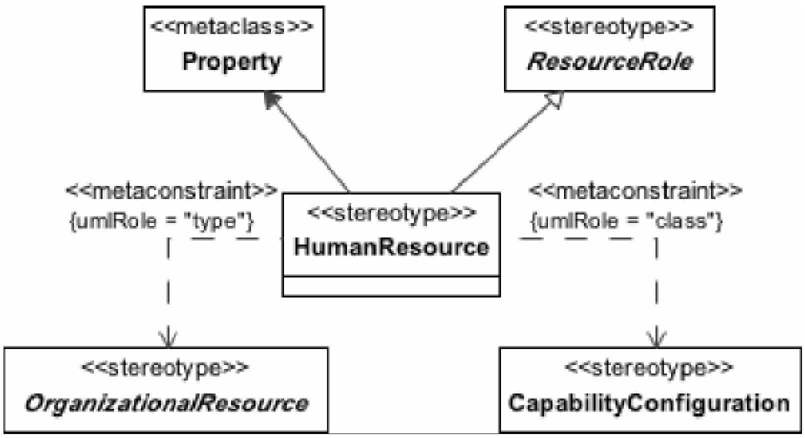


Figure 8.163 - Elements related to the HumanResource stereotype

Constraints

The following are constraints for HumanResource:

- HumanResource.class - Value for the class property must be stereotyped «CapabilityConfiguration» or its specializations.
- HumanResource.type - Value for the type property must be stereotyped «OrganizationalResource» or its specializations.

Extensions

The following are extensions for HumanResource:

- Property

Generalizations

The following are generalization relationships for HumanResource:

- ResourceRole

8.1.1.1.7.4.9 ManufacturedResourceType

UPDM: A ResourceArtifact or Software.

DoDAF: NA – this is an abstract specialization of Resource.

Note – ManufacturedResourceType is abstract.

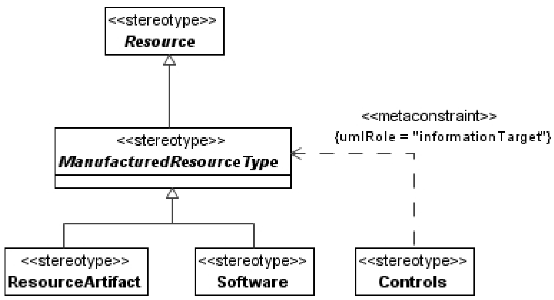


Figure 8.164 - Elements related to the abstract ManufacturedResourceType stereotype

Generalizations

The following are generalization relationships for ManufacturedResourceType:

- Resource

8.1.1.1.7.4.10 Part

MODAF: Usage of a ResourceArtifact (UPDM::Artefact) as a part of another ResourceArtifact.

DoDAF: NA – covered by the more general temporalWholePart element.

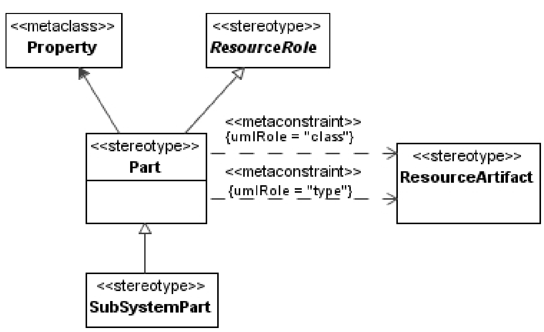


Figure 8.165 - Elements related to the Part stereotype

Constraints

The following are constraints for Part:

- Part.type - Value for the type property must be stereotyped «ResourceArtifact» or its specializations.
- Part.class - Value for the class property must be stereotyped «ResourceArtifact» or its specializations.

Extensions

The following are extensions for Part:

- Property

Generalizations

The following are generalization relationships for Part:

- ResourceRole

8.1.1.1.7.4.11 Platform

MODAF: Usage of an Artifact as a platform (e.g., vessel, aircraft) in a particular ResourceConfiguration.

DoDAF: NA – covered by the more general temporalWholePart element.

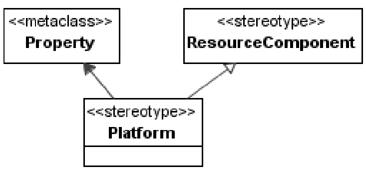


Figure 8.166 - Elements related to the Platform stereotype

Extensions

The following are extensions for Platform:

- Property

Generalizations

The following are generalization relationships for Platform:

- ResourceComponent

8.1.1.1.7.4.12 ResourceConstraint

ResourceConstraint specifies the set of rules that govern the structural or functional functionality of the system. ResourceConstraint extends UML2 constraint to specify the rules of the system.

MODAF: A rule governing the structural or functional aspects of a system.

DoDAF: Systems rules are constraints on an architecture, on a system(s), or system hardware/software item(s), and/or on a system function(s).

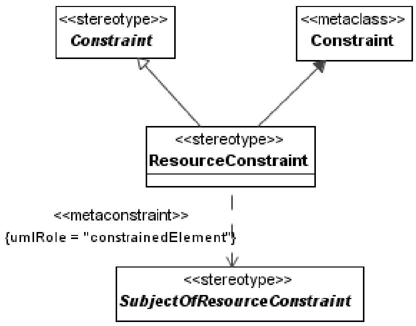


Figure 8.167 - Elements related to the ResourceConstraint stereotype

Constraints

The following are constraints for ResourceConstraint:

- ResourceConstraint.constrainedElement - Value for the constrainedElement property must be stereotyped «SubjectOfResourceConstraint» or its specializations.

Extensions

The following are extensions for ResourceConstraint:

- Constraint

Generalizations

The following are generalization relationships for ResourceConstraint:

- Constraint
- UPDMElement

8.1.1.1.7.4.13 ResourceInterface

UPDM: ResourceInterface is a contractual agreement between two resources that implement protocols through which the source resource to the destination resource.

MODAF: NA

DoDAF: An overlap between Performers for the purpose of producing a Resource that is consumed by the other (DoDAF::Interface).

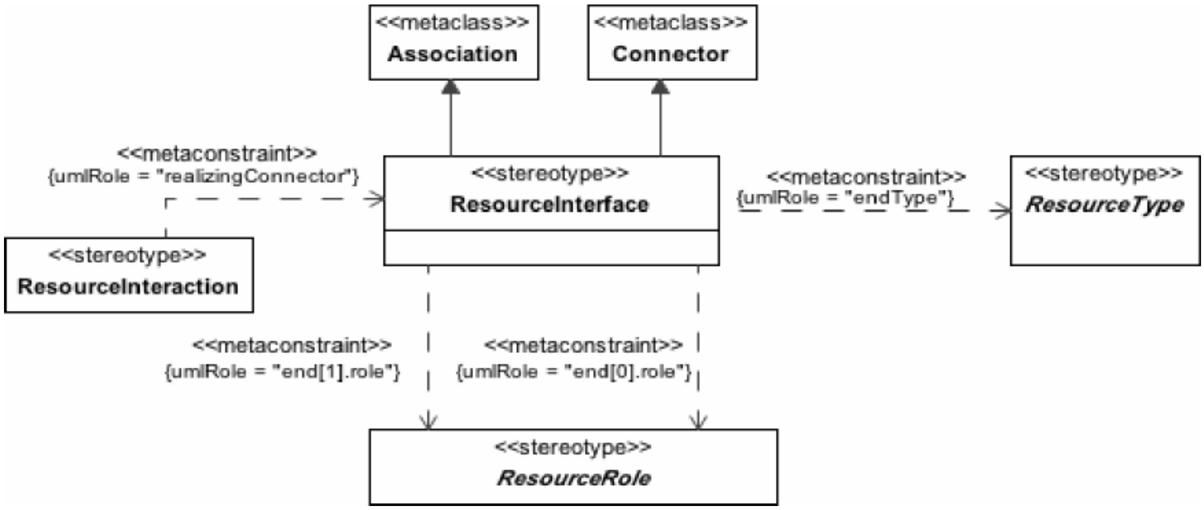


Figure 8.168 - Elements related to the ResourceInterface stereotype

Constraints

The following are constraints for ResourceInterface:

- ResourceInterface.end - In case of extending Association: the value for endType property has to be stereotyped «ResourceType» or its specializations.
- In case of extending Connector: the value for the role property for the owned ConnectorEnd must be stereotype «ResourceRole» or its specializations.

Attributes

The following are attributes for ResourceInterface:

- realizedBy: ResourceConnector[*] – Realizing ResourceConnectors
- realizedExchange: ResourceInteraction[*] - A list of ResourceInteractions (or specializations) that are realized by the ResourceInterface/ResourceConnector. This is derived by navigating from the ResourceInteraction to the ResourceInterfaces/ResourceConnectors using the inverse of the realization/realizingConnector roles.

Extensions

The following are extensions for ResourceInterface:

- Association
- Connector

Generalizations

The following are generalization relationships for ResourceInterface:

- ProtocolImplementation

8.1.1.1.7.4.14 ResourcePort

UPDM: Port is an interaction point for a resource through which it can interact with the outside environment.

MODAF: An interface (logical or physical) provided by a System. A SystemPort may implement a PortType though there is no requirement for SystemPorts to be typed (MODAF:: SystemPort).

DoDAF: An interface (logical or physical) provided by a System (DoDAF::Port).

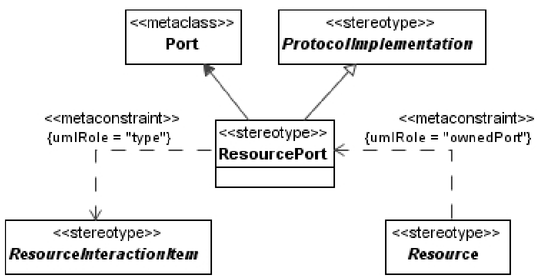


Figure 8.169 - Elements related to the ResourcePort stereotype

Constraints

The following are constraints for ResourcePort:

- ResourcePort.type - Value for the type property must be stereotyped «ResourceInteractionItem» or its specializations.

Extensions

The following are extensions for ResourcePort:

- Port

Generalizations

The following are generalization relationships for ResourcePort:

- ProtocolImplementation

8.1.1.1.7.4.15 ResourceRole

ResourceUsage defines the usage of any resource in the system. Each resourceUsage is owned by a resource.
UPDM: abstract element.

Note – ResourceRole is abstract.

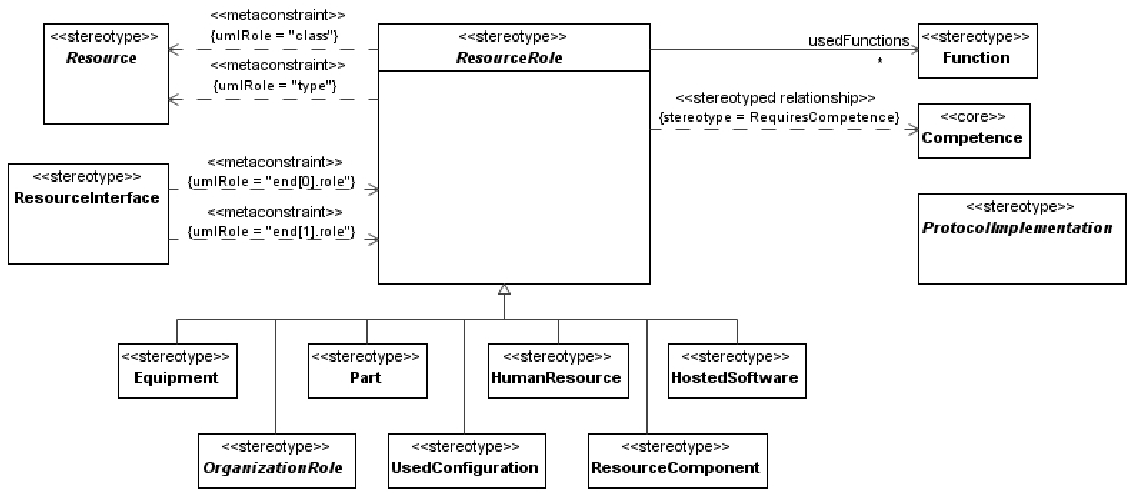


Figure 8.170 - Elements related to the abstract ResourceRole stereotype

Constraints

The following are constraints for ResourceRole:

- ResouceRole.type - An element with the stereotype “ResourceRole” applied must have the “ResourceType” stereotype (or its specializations) applied to the targets of its extended metaclass property “type.”
- ResourceRole.class - Value for the class property must be stereotyped «ResourceType» or its specializations.

Attributes

The following are attributes of ResourceRole:

- usedFunctions : Function[*] - Functions used by the ResourceRole.

Generalizations

The following are generalization relationships for ResourceRole:

- UPDMElement

8.1.1.1.7.4.16 Resource

UPDM: Abstract supertype for physical resources such as OrganizationalResource.

MODAF: A PhysicalAsset, OrganizationalResource, or FunctionalResource that can contribute towards fulfilling a capability (MODAF::ResourceType).

Note – Resource is abstract.

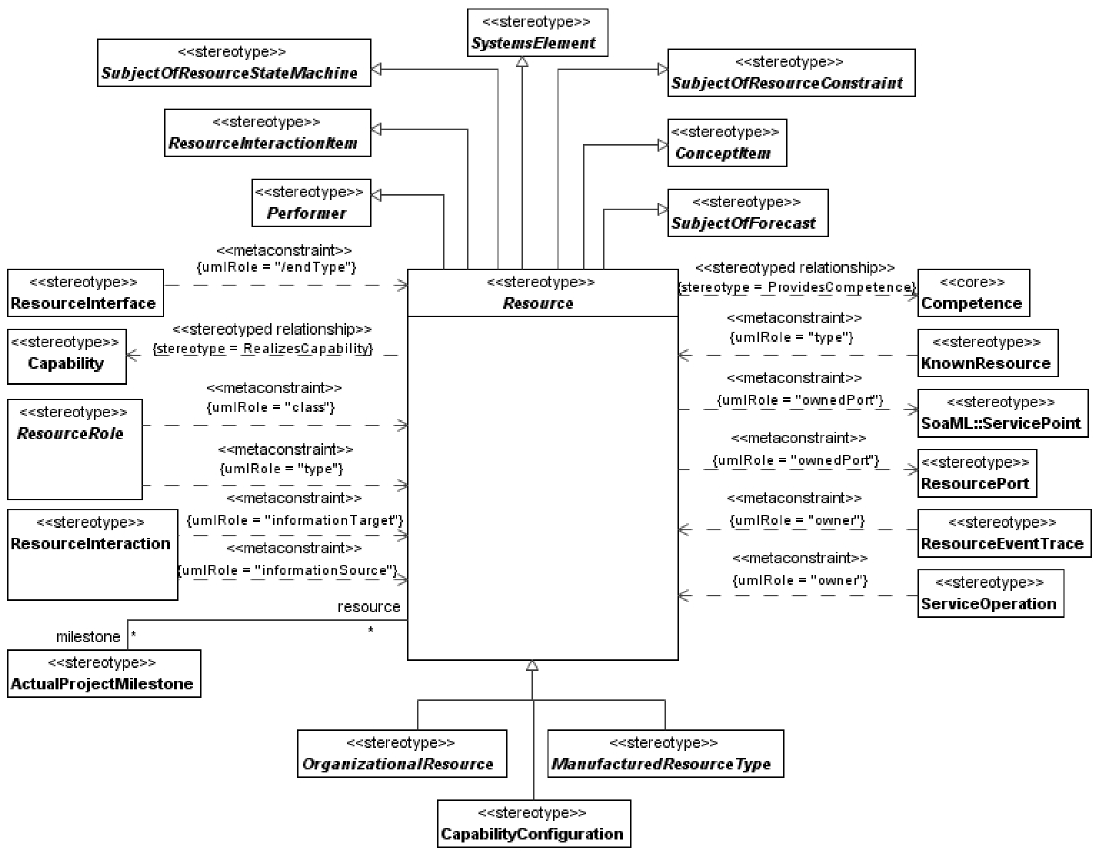


Figure 8.171 - Elements related to the abstract Resource stereotype

Attributes

The following are attributes of Resource:

- milestone: ActualProjectMilestone [*] - Linked milestone.

Constraints

The following are constraints for Resource:

- Resource.ownedPort - Values for the ownedPort property must be stereotyped with «ResourcePort», «Service / Request», or its specializations.
- Resource.performs - Can perform only “Functions.”

Generalizations

The following are generalization relationships for Resource:

- SubjectOfResourceConstraint
- SubjectOfForecast

- ResourceInteractionItem
- Performer
- ConceptItem
- SubjectOfResourceStateMachine
- UPDMElement

8.1.1.1.7.4.17 Software

Software needed for the functioning of the system that is typically included as expected background functionality, such as an operating system. The Tasks and Functions of the Software do not appear in the links to operational activities, typically, and represent functionality that is not the focal point of the architecture description.

MODAF: An executable computer program.

DoDAF: Material: Equipment, apparatus or supplies that are of interest, without distinction as to its application for administrative or combat purposes.

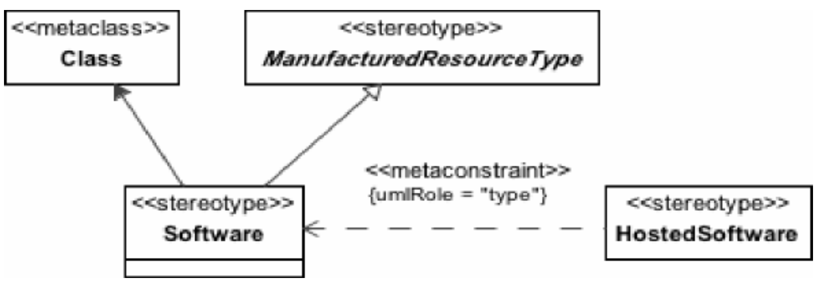


Figure 8.172 - Elements related to the Software stereotype

Extensions

The following are extensions for Software:

- Class

Generalizations

The following are generalization relationships for Software:

- ManufacturedResourceType

8.1.1.1.7.4.18 SubjectOfForecast

MODAF: Abstract Any element that may be subject to a Forecast.

Note – SubjectOfForecast is abstract.

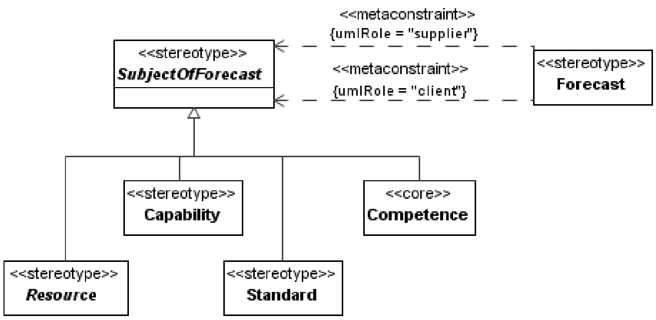


Figure 8.173 - Elements related to the abstract SubjectOfForecast stereotype

Generalizations

The following are generalization relationships for SubjectOfForecast:

- UPDMElement

8.1.1.1.7.4.19 SubjectOfResourceConstraint

MODAF: Abstract. Anything that may be constrained by a ResourceConstraint.

Note – SubjectOfResourceConstraint is abstract.

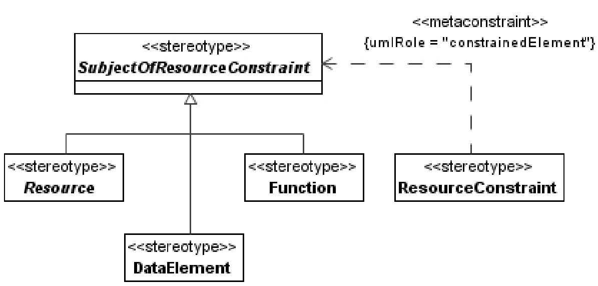


Figure 8.174 - Elements related to the abstract SubjectOfResourceConstraint stereotype

Generalizations

The following are generalization relationships for SubjectOfResourceConstraint:

- UPDMElement

8.1.1.1.7.4.20 SubSystemPart

UPDM: Indicates that a (sub)system is part of another system.

MODAF: Usage of an Artifact (UPDM::ResourceArtifact) as a part of another Artifact (UPDM::ResourceArtifact), equates to a MODAF::Part.

DoDAF: NA

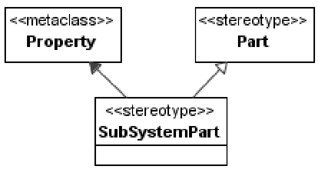


Figure 8.175 - Elements related to the SubSystemPart stereotype

Extensions

The following are extensions for SubSystemPart:

- Property

Generalizations

The following are generalization relationships for SubSystemPart:

- Part

8.1.1.1.7.4.21 SystemsNode

UPDM: DoDAF v1.5 alias for CapabilityConfiguration. Required for backward compatibility with DoDAF V1.5.

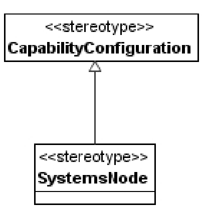


Figure 8.176 - Elements related to the SystemsNode stereotype

Extensions

The following are extensions for SystemsNode:

- Class

Generalizations

The following are generalization relationships for SystemsNode:

- CapabilityConfiguration

8.1.1.1.7.4.22 UsedConfiguration

MODAF: The usage of a CapabilityConfiguration in another CapabilityConfiguration.

DoDAF: NA

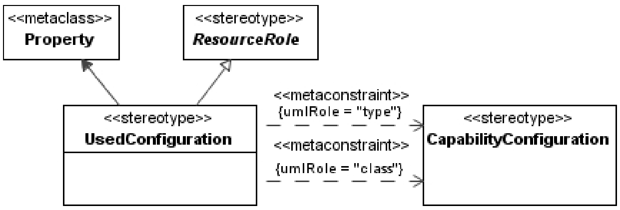


Figure 8.177 - Elements related to the UsedConfiguration stereotype

Constraints

The following are constraints for UsedConfiguration:

- UsedConfiguration.type - Value for the type property must be stereotyped «CapabilityConfiguration» or its specializations.
- UsedConfiguration.class - Value for the class property must be stereotyped «CapabilityConfiguration» or its specializations.

Extensions

The following are extensions for UsedConfiguration:

- Property

Generalizations

The following are generalization relationships for UsedConfiguration:

- ResourceRole

8.1.1.1.8 UPDM L1 : : UPDM L0: :Core: :TechnicalStandardsElements

Section 1.4.4 of the DoDAF version 1.5 Definitions and Guidelines (Volume I) Define the purpose of the Technical View as follows:

“The TV is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements. Its purpose is to ensure that a system satisfies a specified set of operational requirements. The TV provides the technical systems implementation guidelines upon which engineering specifications are based, common building blocks are established, and product lines are developed. It includes a collection of the technical standards, implementation conventions, standards options, rules, and criteria that can be organized into profile(s) that govern systems and system or service elements for a given architecture.”

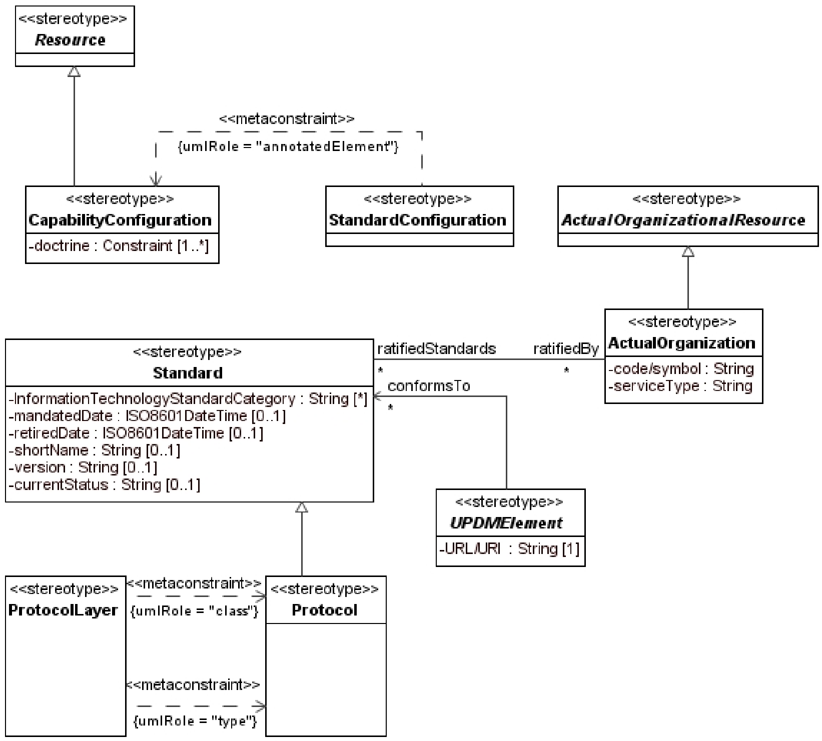


Figure 8.178 - Elements related to the TV-1&2&3 products

MODAF: Standards Profile (TV-1) defines the technical and non-technical standards, guidance, and policy applicable to the architecture.

The Standards Forecast (TV-2) contains expected changes in technology-related standards and conventions, which are documented in the TV-1 Product.

DoDAF: The Standards Profile StdV-1 DoDAF-described View defines the technical, operational, and business standards, guidance and policy applicable to the architecture.

The StdV-2 Standards Forecast DoDAF-described View contains expected changes in technology related standards, operational standards, or business standards and conventions, which are documented in the StdV-1 view.

The Technical Standards Profile (TV-1) is an exhaustive compilation of current standards that apply to Systems and Services View elements of the federated architecture. Standards identified in the Technical Standards Profile (TV-1) should be the same as those identified in the Systems Interface Description (SV-1). The TV-1 collects the various systems standards rules that implement and sometimes constrain the choices that can be made in the design and implementation of an architecture.

Standards Profiles for a particular architecture must maintain full compatibility with the root standards they have been derived from. In addition, the TV-1 View may state a particular method of implementation for a standard, as compliance with a standard does not ensure interoperability. The standards cited are referenced as relationships to the systems, system functions, system data, hardware/software items, or communication protocols in SV- 1, SV-2, SV-4, SV-6, OV-7, and SV11 Products, where applicable. That is, each standard listed in the profile will be associated with the SV elements that implement or use the standard (e.g., SV-1, SV-2, SV-4, SV-6, OV-7, and SV-1 1 element standards, where applicable).

The Protocols referred to such as 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.

The Technical Standards Forecast (TV-2) provides a description of emerging standards and their potential impact on current Systems and Services View elements, within a set of time frames. The forecast for evolutionary changes in the standards should be correlated against the time periods as mentioned in the SV-8 and SV-9 products. A TV-2 complements and expands on the Standards Profile (TV-1) product and should be used when more than one emerging standard time-period is applicable to the architecture.

The specific time periods selected (e.g., 6-month, 12-month, and 18-month intervals) and the standards being tracked will be coordinated with architecture transition plans (see SV-8). That is, insertion of new capabilities and upgrading of existing systems may depend on, or be driven by, the availability of new standards and products incorporating those standards. The forecast specifies potential standards and thus impacts current architectures and influences the development of transition and objective (i.e., target) architectures. The forecast will be tailored to focus on standards areas that are related to the purpose for which a given architecture is being described and should identify potential standards affecting that architecture. If interface standards are an integral part of the technologies important to the evolution of a given architecture, then it may be convenient to combine TV-2 with SV-9. For other projects, it may be convenient to combine all the standards information into one document, combining TV-2 with TV-1.

TV-2 delineates the standards that will potentially impact the relevant system elements (from SV-1, SV-2, SV-4, SV-6, and OV-7) and relates them to the time periods that are listed in the SV-8 and SV-9. A system's evolution, specified in SV-8, may be tied to a future standard listed in TV-2. A timed technology forecast from SV-9 is related to a TV-2 standards forecast in the following manner: a certain technology may be dependent on a TV-2 standard (i.e., a standard listed in TV-2 may not be adopted until a certain technology becomes available). This is how a prediction on the adoption of a future standard, as applicable to systems elements from SV-1, SV-2, SV-4, SV-6, and OV-7, may be related to standards listed in TV-1 through the SV-9.

8.1.1.1.8.1 Protocol

MODAF: A Standard for communication. Protocols may be composite (i.e., a stack).

DoDAF: NA, See TechnicalStandard.

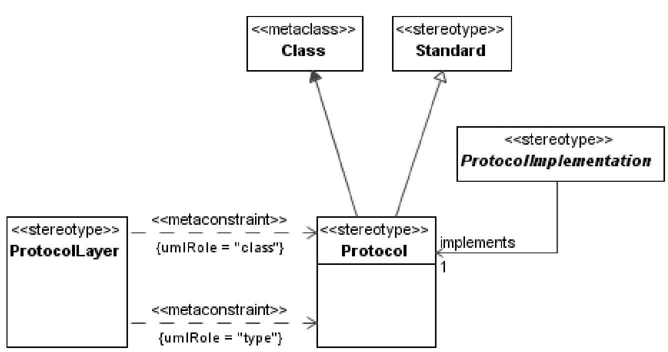


Figure 8.179 - Elements related to the Protocol stereotype

Extensions

The following are extensions for Protocol:

- Class

Generalizations

The following are generalization relationships for Protocol:

- Standard

8.1.1.1.8.2 ProtocolImplementation

UPDM: Abstract element: A connector that implements a specific Protocol.

MODAF: An element that can implement a Protocol.

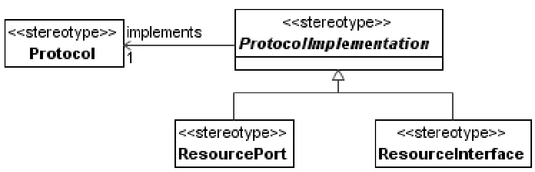


Figure 8.180 - Elements related to the abstract ProtocolImplementation stereotype

Attributes

The following are attributes of ProtocolImplementation:

- implements : Protocol[1] - The “Protocol” that can be implemented by the Connector targets.

Generalizations

The following are generalization relationships for ProtocolImplementation:

- UPDMElement

8.1.1.1.8.3 Standard

A ratified set of rules that are used to guide and/or constrain any UPDM element. Its purpose is to ensure that a system or architecture conforms to a specified set of operational requirements.

MODAF: A ratified and peer-reviewed specification that is used to guide or constrain the architecture. A Standard may be applied to any element in the architecture via the [constrainedItem] property of UML::Constraint.

DoDAF: A formal agreement documenting generally accepted specifications or criteria for products, processes, procedures, policies, systems, and/or personnel.

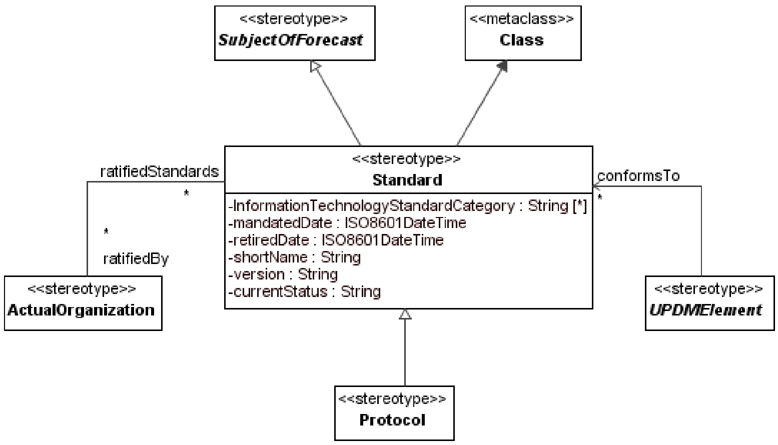


Figure 8.181 - Elements related to the Standard stereotype

Attributes

The following are attributes of Standard:

- InformationTechnologyStandardCategory : String[*] - The information technology standard category that the “Standard” belongs to.
- ratifiedBy : ActualOrganization[*] - Organization that ratified this Standard.
- mandatedDate: ISO8601 DateTime [0..1] – The date when this version of the Standard was published.
- retiredDate: ISO8601 DateTime [0..1] – The date when this version of the Standard was retired.
- shortName: String [0 ..1] – Short name of the Standard.
- version: String [0 ..1] – Represents the revision number of the Standard - e.g., “1.2.1,” “v2,” “:2004.”
- currentStatus: String [0 ..1] – Current status of the Standard.

Extensions

The following are extensions for Standard:

- Class

Generalizations

The following are generalization relationships for Standard:

- SubjectOfForecast

8.1.1.1.8.4 StandardConfiguration

UPDM: A Comment, attached to a “CapabilityConfiguration” which indicates that the annotated “CapabilityConfiguration” is a standard pattern for re-use in the architecture.

MODAF: A UML::Comment that when attached to a CapabilityConfiguration indicates that it is a standard pattern for re-use in the architecture.

DoDAF: NA

Constraints

The following are constraints for StandardConfiguration:

- StandardConfiguration.annotatedElement - Value for the annotatedElement property must be stereotyped «CapabilityConfiguration».

Extensions

The following are extensions for StandardConfiguration:

- Comment

Generalizations

The following are generalization relationships for StandardConfiguration:

- UPDMElement

8.1.1.1.8.5 EntityAttribute

MODAF: A defined property of an EntityItem.

DoDAF: NA

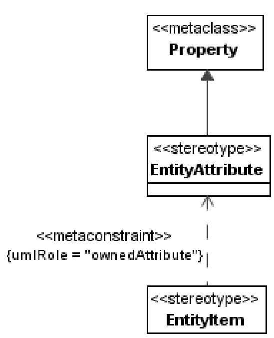


Figure 8.182 - Elements related to the Attribute stereotype

Constraints

The following are constraints for EntityAttribute:

- EntityRelationship.endType - Values for the end Type property must be stereotyped «Entity Attribute» or its specializations.
- EntityRelationship.canBeAppliedTo - «EntityAttribute» stereotype can be applied to Properties that are owned only by «EntityItem».

Extensions

The following are extensions for EntityAttribute:

- Property

Generalizations

The following are generalization relationships for EntityAttribute:

- UPDMElement

8.1.1.1.8.6 EntityRelationship

MODAF: Asserts that there is a relationship between two EntityItems.

DoDAF: (DoDAF::DataAssociation): A relationship or association between two elements of proceduralized information.

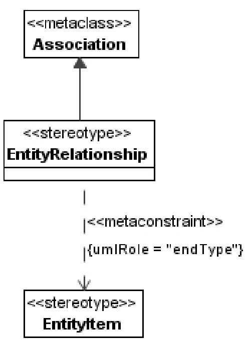


Figure 8.183 - Elements related to the EntityRelationship stereotype

Constraints

The following are constraints for EntityRelationship:

- EntityRelationship.endType - Values for the endType property must be stereotyped «Entity» or its specializations.

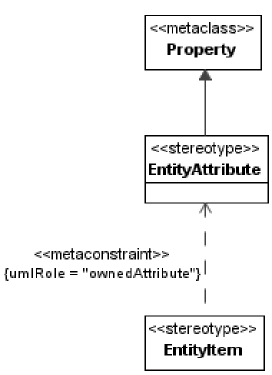


Figure 8.184 - Elements related to the EntityAttribute stereotype

Extensions

The following are extensions for EntityRelationship:

- Association

Generalizations

The following are generalization relationships for EntityRelationship:

- UPDMElement

8.1.1.1.8.7 ProtocolLayer

MODAF: Asserts that a Protocol (upperLayer) uses another Protocol (lowerLayer) (MODAF:: ProtocolStack).

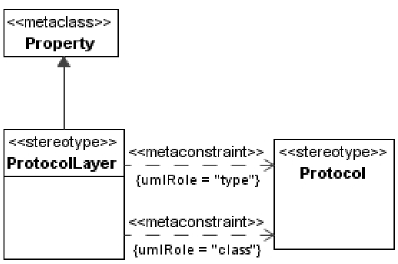


Figure 8.185 - Elements related to the ProtocolLayer stereotype

Constraints

The following are constraints for ProtocolImplementation:

- ProtocolLayer.class - The value for class property has to be stereotyped «Protocol» or its specializations.
- ProtocolLayer.type - The value for type property has to be stereotyped «Protocol» or its specializations.

Extensions

The following are extensions for ProtocolLayer:

- Property

Generalizations

The following are generalization relationships for ProtocolLayer:

- UPDMElement

8.1.1.2 UPDM L1::UPDM L0::DoDAF

Elements that are not considered part of the Core architectural model, but necessary for DoDAF.

8.1.1.2.1 UPDM L1::UPDM L0::DoDAF::AllElements

The All View elements for DoDAF specific models. The All View elements provide information about the entire Architecture. They are used for support rather than architectural models.

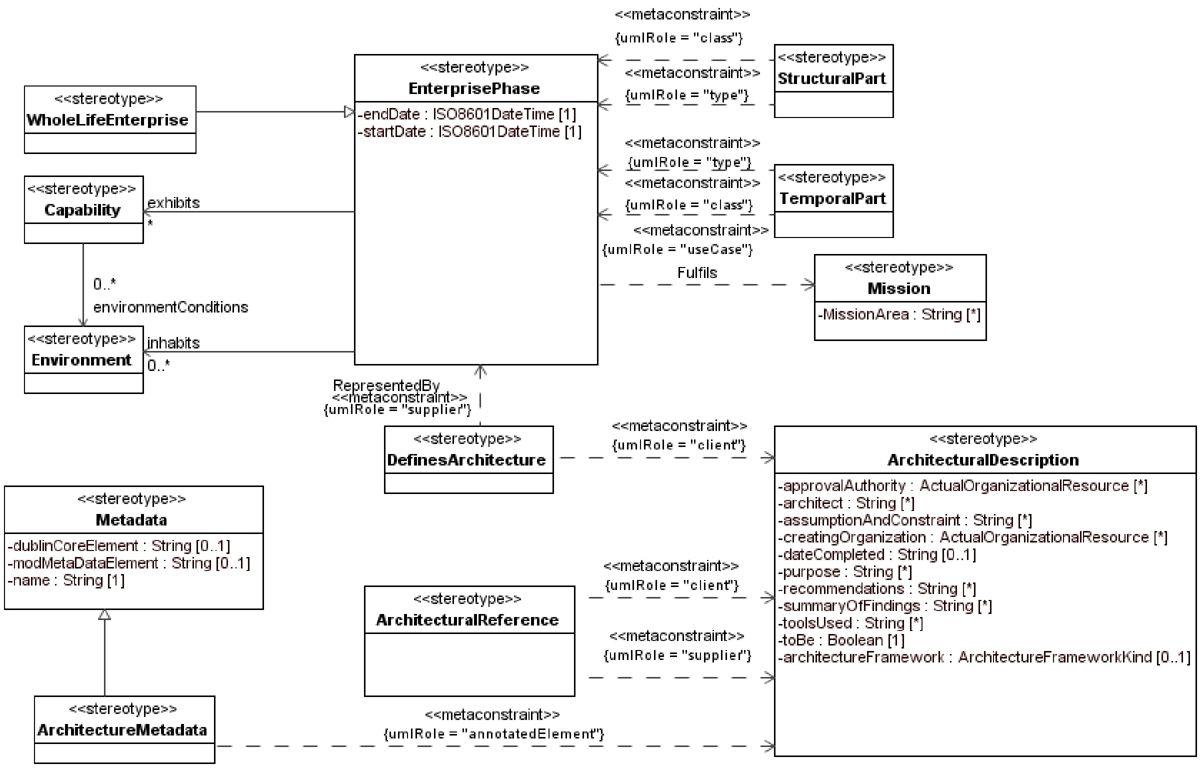


Figure 8.186 - Elements related to the AV-1 stereotype

MODAF: The overview and summary information contained within the AV-1 product provides executive-level summary information in a consistent form that allows quick reference and comparison between architectural descriptions. AV-1 includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.

DoDAF: The overview and summary information contained within the AV-1 DoDAF-described View provides executive-level summary information in a consistent form that allows quick reference and comparison between architectural descriptions. The AV-1 includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.

The Overview & Summary Information identifies the architecture goals, viewpoint, findings, and recommendations. The AV-1 contains sufficient textual information to enable a reader to select one architecture from among many to read in more detail. AV-1 serves two additional purposes:

1. In the initial phases of architecture development, it serves as a planning guide.
2. Upon completion of an architecture, the AV-1 provides summary textual information concerning that architecture.

The AV- 1 comprises a textual executive summary of a given architecture and documents the following descriptions:

Architecture Project Identification - identifies the architecture project name, the architect, and the organization developing the architecture. It also includes assumptions and constraints, identifies the approving authority and the completion date, and records the level of effort and costs (projected and actual) required to develop the architecture.

Scope - identifies 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, and so forth. Scope also identifies the organizations and Communities Of Interest (COI) that fall within the scope of the architecture. The scope also includes the COI that are related to the architecture.

Purpose and Viewpoint - explains the need for the architecture, what it should demonstrate, the types of analyses (e.g., Activity-Based Costing) that will be applied to it, who is expected to perform the analyses, what decisions are expected to be made on the basis of an analysis, who is expected to make those decisions, and what actions are expected to result. The viewpoint from which the architecture is developed is identified (e.g., planner or decision maker).

Context - describes the setting in which the architecture exists. It includes such things as mission, doctrine, relevant goals and vision statements, concepts of operation, scenarios, information assurance context (e.g., types of system data to be protected, such as classified or sensitive but unclassified, and expected information threat environment), other threats and environmental conditions, and geographical areas addressed, where applicable. Context also identifies authoritative sources for the rules, criteria, and conventions that were followed. The tasking for the architecture project and known or anticipated linkages to other architectures are identified.

Tools and File Formats Used - identifies the tool suite used to develop the architecture and file names and formats for the architecture and each product.

Findings - states the findings and recommendations that have been developed based on the architecture effort. Examples of findings include identification of shortfalls, recommended system implementations, and opportunities for technology insertion.

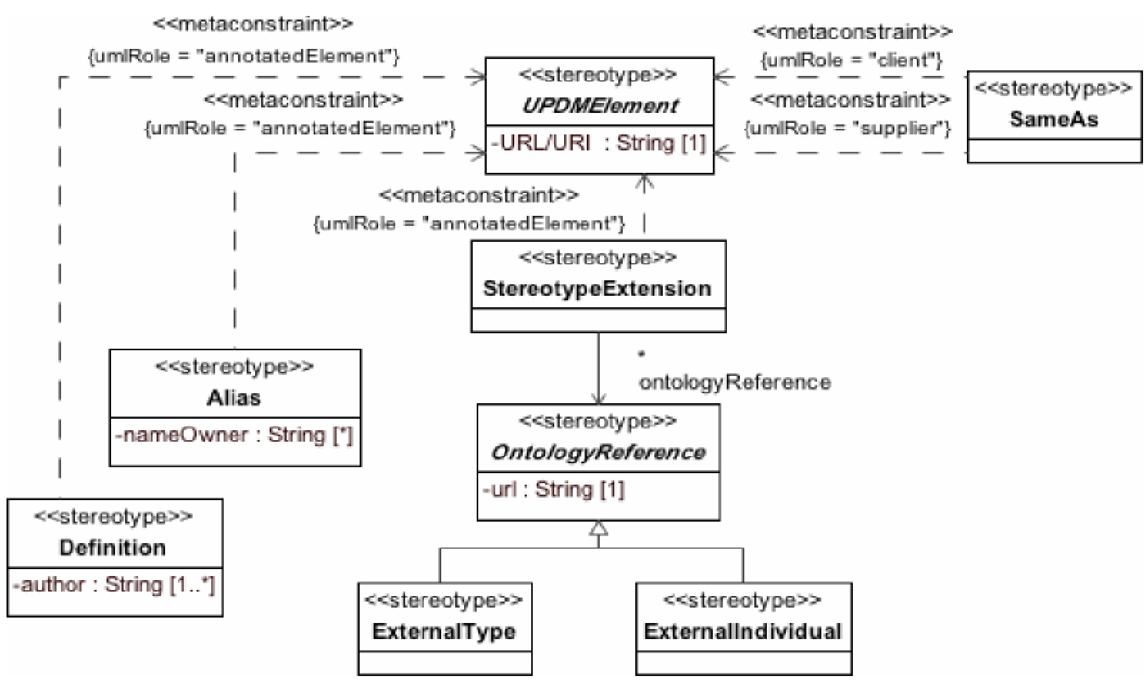


Figure 8.187 - Elements related to the AV-2 stereotype

MODAF: AV-2 presents all the Elements used in an architecture as a stand alone structure. An AV-2 presents all the Elements as a specialization hierarchy, provides a text definition for each one and references the source of the element (e.g., MODAF Ontology, IDEAS Model, local). 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.

DoDAF: The AV-2 presents all the metadata used in an architecture as a standalone structure. An AV-2 presents all the metadata as a specialization hierarchy, provides a text definition for each one and references the source of the element (e.g., DoDAF Meta-model, IDEAS, a published document or policy). An AV-2 shows elements from the DoDAF Meta-model that have been used in the architecture and new elements (i.e., not in the DoDAF Meta-model) that have been introduced by the architecture.

The AV-2 contains definitions of terms used in the given architecture. It consists of textual definitions in the form of a glossary, a repository of architecture data, their taxonomies, and their metadata (i.e., data about architecture data), including metadata for tailored products, associated with the architecture products developed. Metadata are the architecture data types, possibly expressed in the form of a physical schema. In this document, architecture data types are referred to as architecture data elements.

AV-2 provides a central repository for a given architecture's data and metadata. AV-2 enables the set of architecture products to stand alone, allowing them to be read and understood with minimal reference to outside resources. AV-2 is an accompanying reference to other products, and its value lies in unambiguous definitions. The key to long-term interoperability can reside in the accuracy and clarity of these definitions.

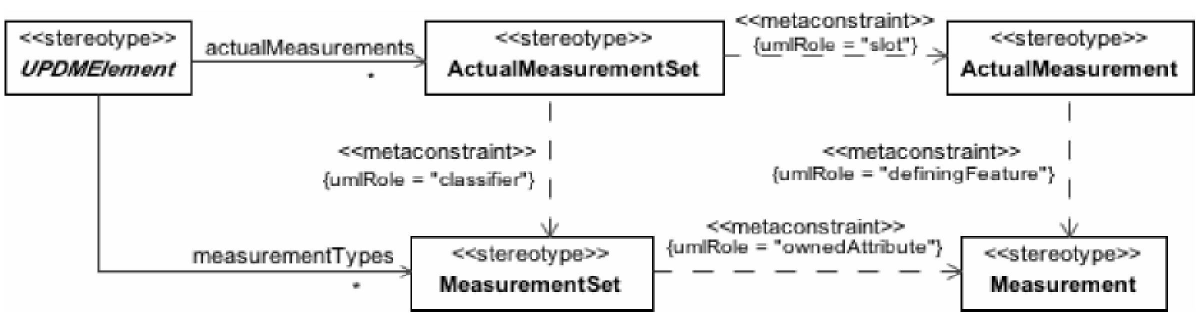


Figure 8.188 - Elements related to the AV-3 stereotype

The AV-3 defines the measureable properties of something in the physical world, expressed in amounts of a unit of measure that can be associated with a UPDMElement.

8.1.1.2.1.1 UPDM L1 ::UPDM L0::DoDAF::AllElements::Measurements

Defines performance parameters in terms of quantifiable properties.

8.1.1.2.1.1.1 MeasureOfPerformance

A DoDAF alias for ActualMeasurement.

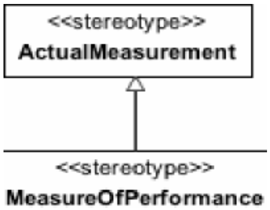


Figure 8.189 - Elements related to the MeasureOfPerformance stereotype

Extensions

The following are extensions for MeasureOfPerformance:

- Slot

Generalizations

The following are generalization relationships for MeasureOfPerformance:

- ActualMeasurement

8.1.1.2.1.1.2 PerformanceParameter

UPDM A DoDAF alias for Measurement.

DoDAF: A category of quality measures that address how well a Performer meets Capability needs.

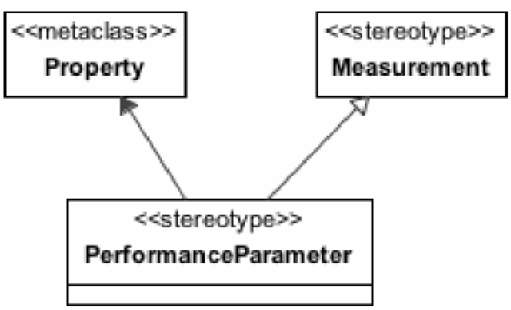


Figure 8.190 - Elements related to the PerformanceParameter stereotype

Extensions

The following are extensions for PerformanceParameter:

- Property

Generalizations

The following are generalization relationships for PerformanceParameter:

- Measurement

8.1.1.2.2 UPDM L1 ::UPDM L0::DoDAF::OperationalElements

The Operational View elements for DoDAF specific models.

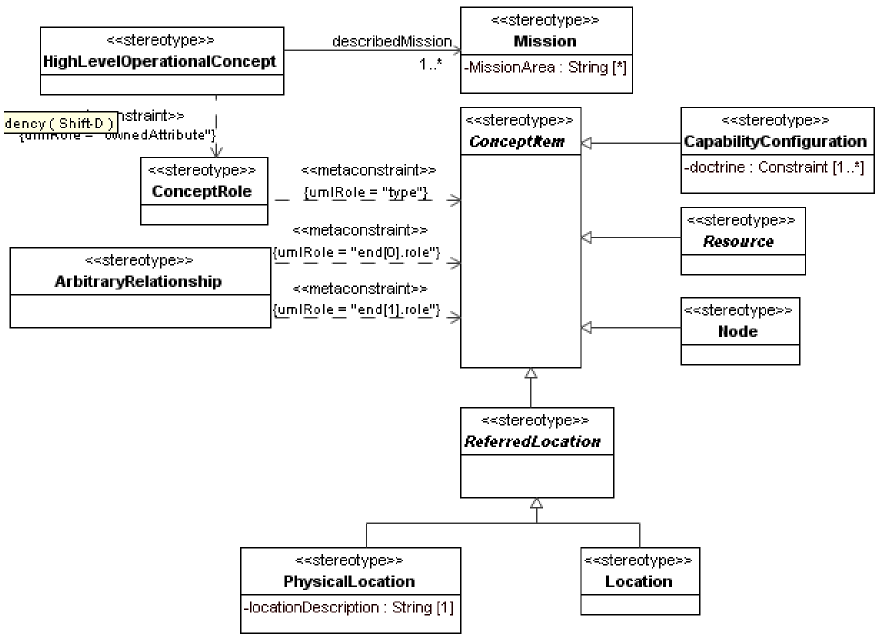


Figure 8.191 - Elements related to the OV-1 stereotype

The purpose of OV-1 is to provide a quick, high-level description of what the architecture is supposed to do, and how it is supposed to do it.

The purpose of High-level Operational Concept Graphic is to provide a high-level graphical and textual description of operational concept (high level organizations, missions, geographic configuration, connectivity, etc.) of what the architecture is supposed to do, and how it is supposed to do it. The OV-1, along with the corresponding AV-1 product, is intended to serve as an executive summary of the architecture.

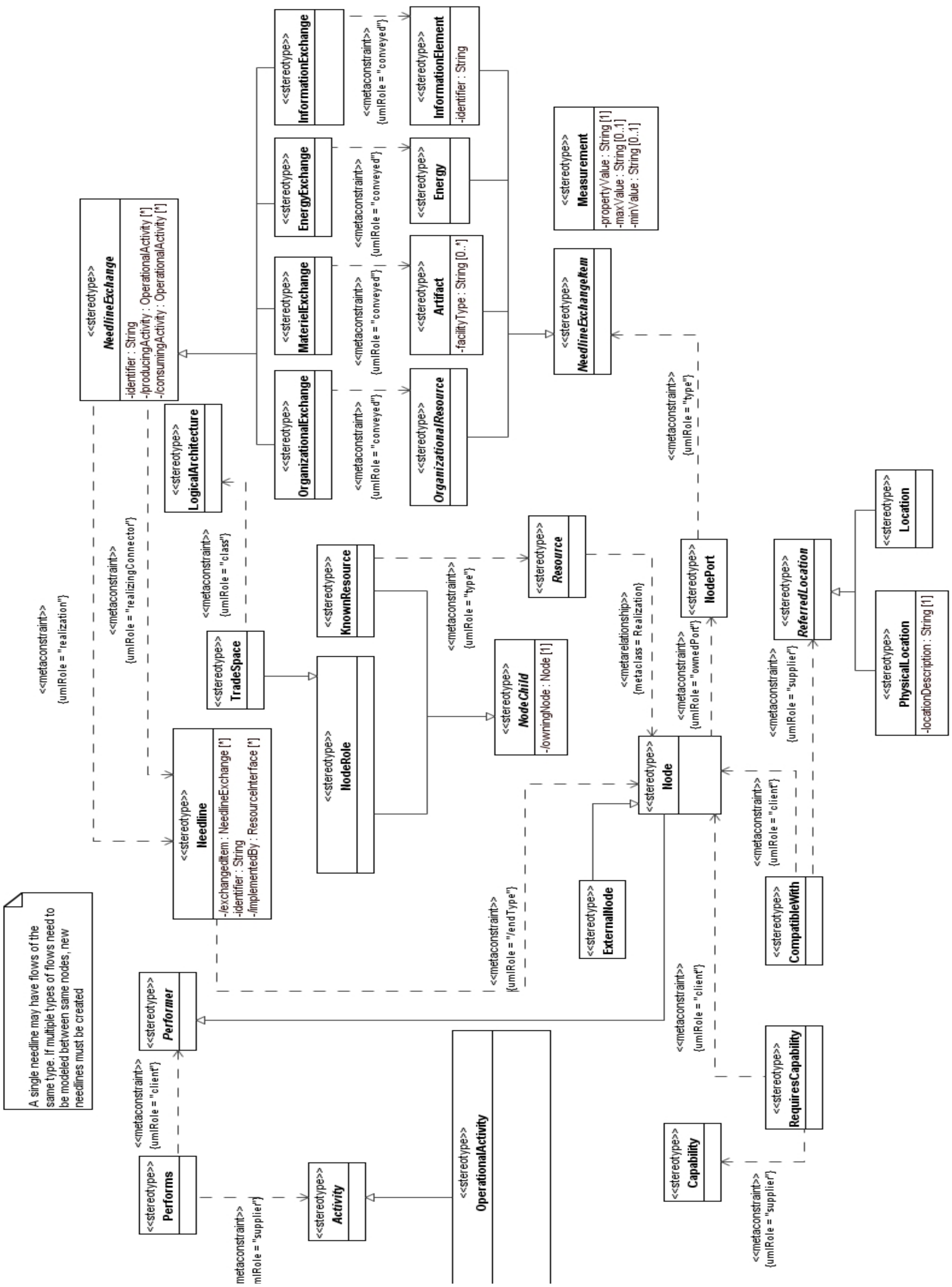


Figure 8.192 - Elements related to the OV-2 stereotype

MODAF: The Operational Node Relationships Description (OV-2) addresses localization of operational capability.

DoDAF: The Operational Resource Description (OV-2) DoDAF-described View applies the context of the operational capability to a community of anticipated users.

The Operational Node Connectivity Description is intended to track the need to exchange information from specific operational nodes (that play a key role in the architecture) to others. OV-2 does not depict the connectivity between the nodes. MoDAF modifies the OV-2 is in two ways. First of all, it recommends that an OV-2 diagram (now OV-2a) shows the platforms or geographic locations at which operational nodes are deployed. Secondly it provides additional information (OV-2b) about each needline in the form of a requirements specification. There are now four types of needlines identified as shown below:

1. InformationExchange
2. EnergyFlow
3. MaterielFlow
4. MovementOfPeople

In addition, MoDAF permits service-oriented architectures. Instead of needlines between nodes, it is possible simply to show which services the nodes provide and consume. Finally, MoDAF again permits known resources to be shown in an OV-2. However, this must be clearly shown as a KnownResource in an OV-2 model. The concept of a LogicalArchitecture is introduced - this is the container class for all the nodes and KnownResources.

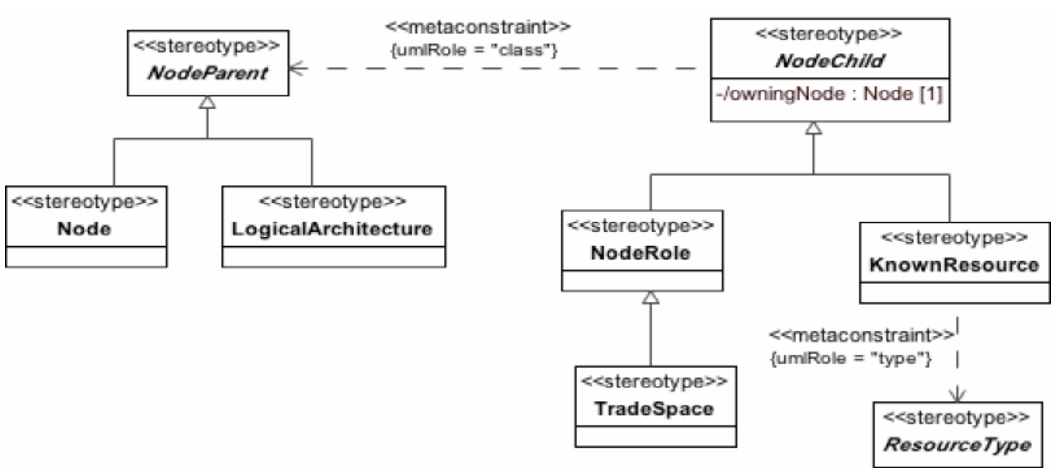


Figure 8.193 - Elements related to the OV-2 Nodes stereotype

Nodes related to OV-2 View.

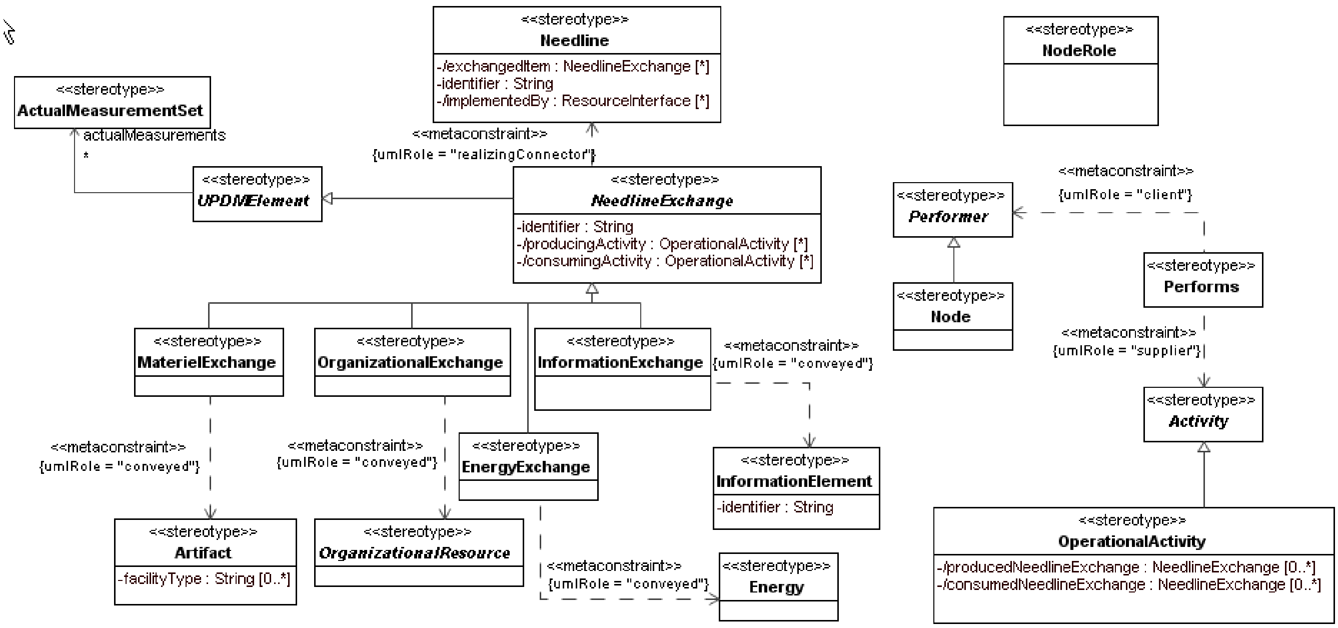


Figure 8.194 - Elements related to the OV-3 stereotype

Information exchanges express the relationship across the three basic architecture data elements of an OV (operational activities, operational nodes, and information flow) with a focus on the specific aspects of the information flow and the information content.

The Information Exchanges of the OV-3 should remain at a high level of aggregation to represent actual information workflow products that are used at the operational nodes shown in OV-2 (and not their subordinate operational nodes).

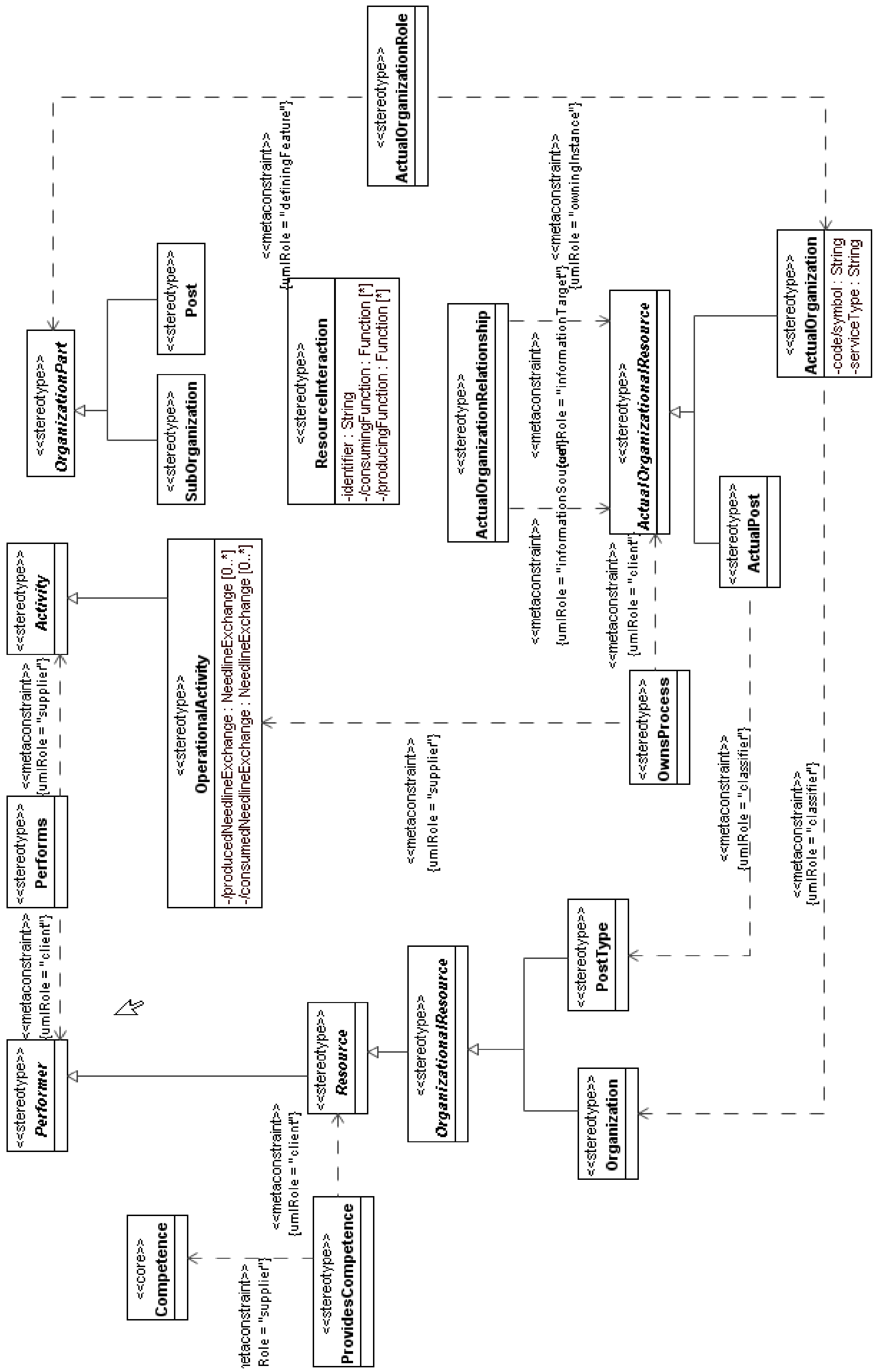


Figure 8.195 - Elements related to the OV-4 Actual stereotype

This is the OV-4 Actual View. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. MoDAF divides the OV-4 two views, an OV-4 Typical, and an OV-4 Actual. The former is exactly as the DoDAF OV-4, while the latter is a special form of the SV-1; where the resources are restricted to being organizational.

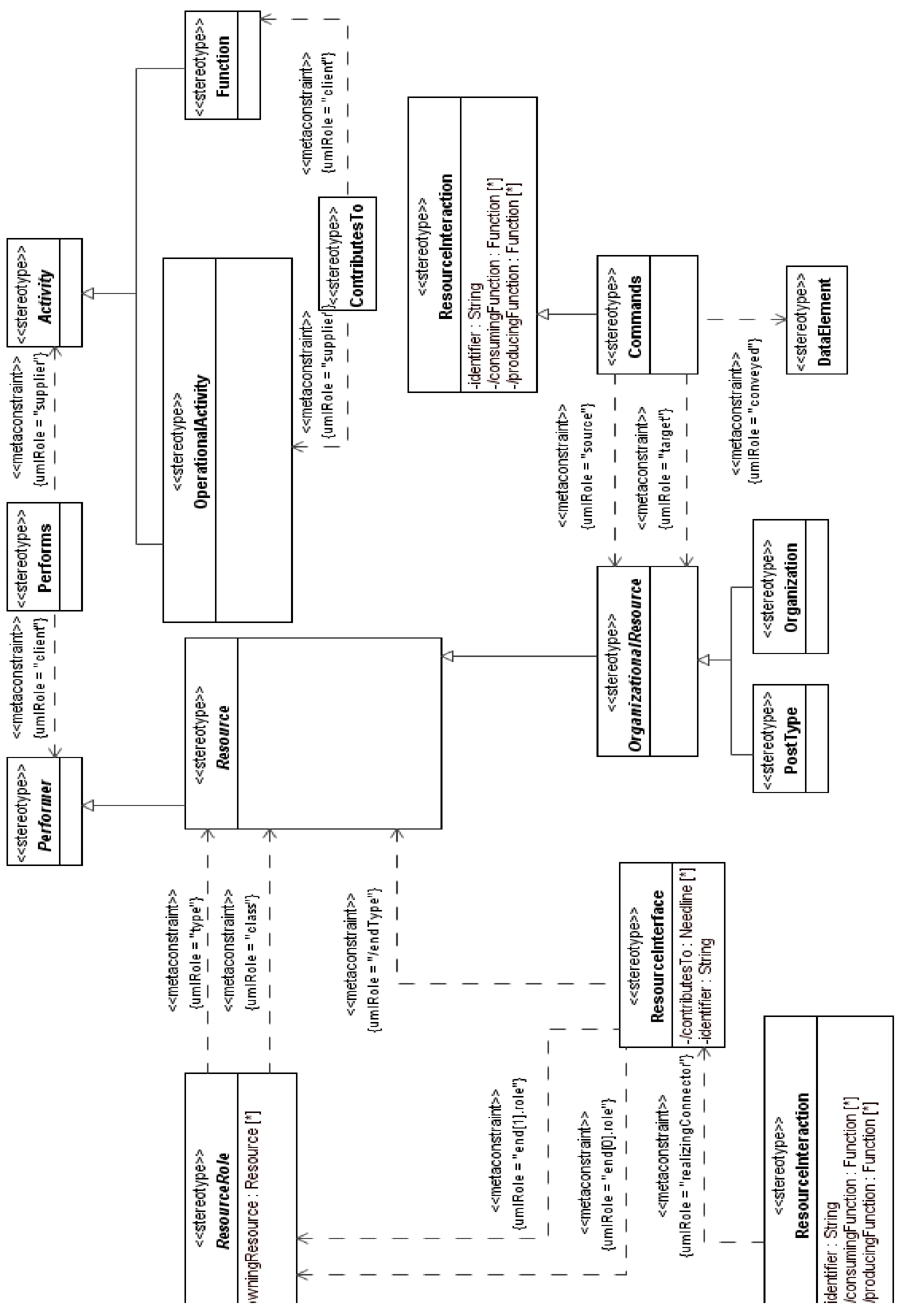


Figure 8.196 - Elements related to the OV-4 Typical stereotype

MODAF: The OV-4 shows organizational structures and interactions. The organizations shown may be civil or military. A typical OV-4 shows the possible relationships between organizational resources (organizations and posts).

DoDAF: DoDAF: The OV-4 DoDAF-described View shows organizational structures and interactions. The organizations shown may be civil or military. A typical OV-4 shows the possible relationships between organizational resources.

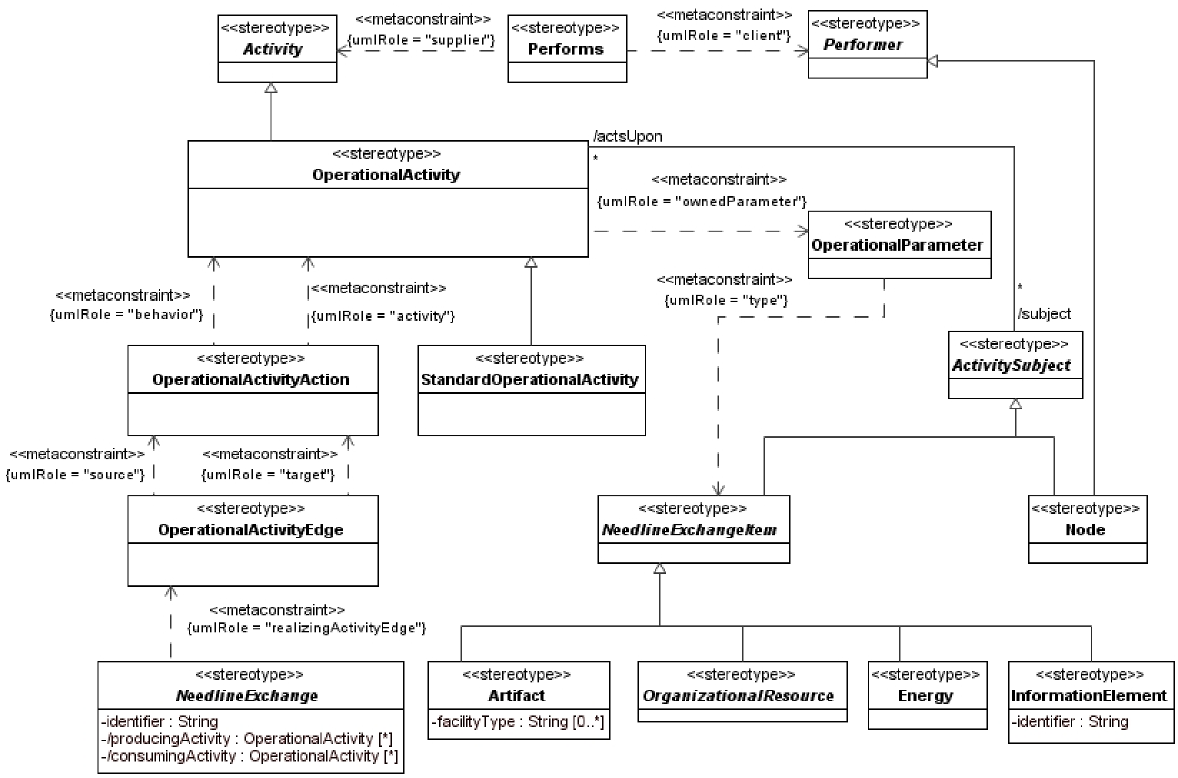


Figure 8.197 - Elements related to the OV-5 stereotype

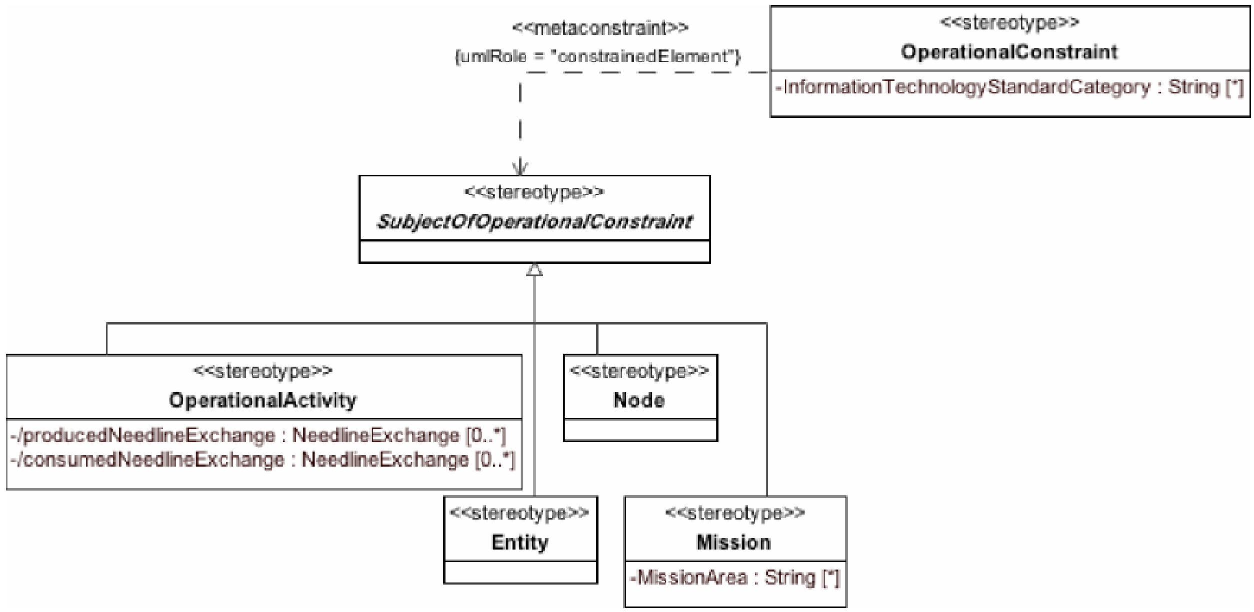


Figure 8.198 - Elements related to the OV-6a stereotype

The Operational Activity Model describes the operations that are normally conducted in the course of achieving a mission or a business goal, from a net-centric perspective. It describes capabilities, operational activities (or tasks), input and output (I/O) flows between activities, and I/O flows to/from activities that are outside the scope of the architecture. It is imperative that the levels-of-detail between the OV-2, OV-3, and OV-5 remain cohesive.

For example, if one diagram of OV-2 operational nodes is developed that shows aggregated organizations only, then it is imperative that the corresponding OV-5 product be developed to show only those operational activities that are meaningful with respect to these operational nodes. Similarly, the information exchanges of OV-3 should remain at a high level of aggregation to represent actual information workflow products that are used at the operational nodes depicted in OV-2 (and not their subordinate operational nodes). The net-centric OV-5 may be used in the following ways:

- Delineate lines of dependency on external activities when coupled with OV-2.
- Highlight information flows to depict the status of the information's refinement (raw, pre-processed, fused).
- Provide the critical foundation for depicting Task, Post, Process, and Use (TPPU) activity sequencing and timing in the OV-6a, OV-6b, and OV-6c.
- Identify critical mission threads and operational information exchanges by annotating which activities are critical, i.e., identify the activities in the model that are critical.

MODAF: An Operational Rules Model (OV-6a) specifies operational or business rules that are constraints on the way that business is done in the enterprise.

DoDAF: An Operational Rules Model (OV-6a) DoDAF-described View specifies operational or business rules that are constraints on the way that business is done in the enterprise.

The Operational Rules Model specifies operational or business rules that are constraints on an enterprise, a mission, operation, business, or an architecture. While other OV products (e.g., OV- 1, OV-2, and OV-5) describe the structure of a business-what the business can do-for the most part, they do not describe what the business must do, or what it cannot do. At the mission level, OV-6a may consist of doctrine, guidance, rules of engagement, and so forth. At the operation level, rules may include such things as a military Operational Plan (OPLAN). 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).” At a top level, rules should at least embody the concepts of operations defined in OV- 1, and should provide guidelines for the development and definition of more detailed rules and behavioral definitions that will occur later in the architecture definition process.

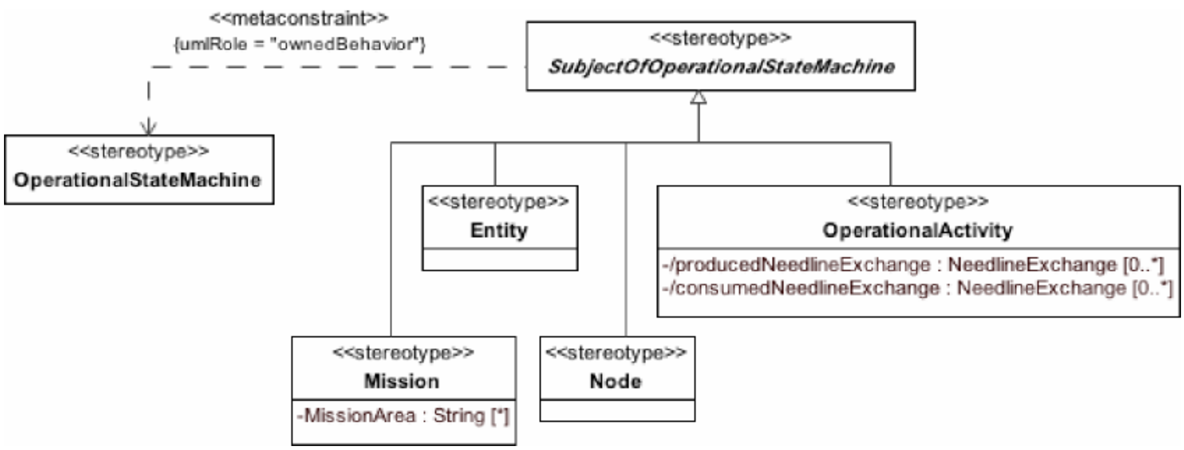


Figure 8.199 - Elements related to the OV-6b stereotype

The Operational State Transition Description is a graphical method of describing how an operational node or activity responds to various events by changing its state. The diagram represents the sets of events to which the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action. The explicit sequencing of activities in response to external and internal events is not fully expressed in OV-5. An OV-6b can be used to describe the explicit sequencing of the operational activities.

Alternatively, OV-6b can be used to reflect the explicit sequencing of actions internal to a single operational activity or the sequencing of operational activities with respect to a specific operational node. In a net-centric architecture, the OV6b is used to describe the set of state transitions for providers and consumers in the Net-Centric Environment (NCE) in response to the posting of information to the NCE or retrieving of information from the NCE.

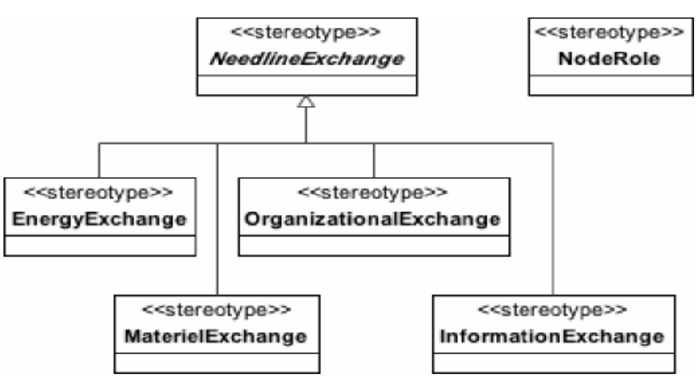


Figure 8.200 - Elements related to the OV-6c stereotype

The Operational Event-Trace Description provides a time-ordered examination of the information exchanges between participating operational nodes as a result of a particular operational thread or scenario. Each eventtrace diagram should have an accompanying description that defines the particular scenario or situation and represent a specific capability. The OV-6c is also used in conjunction with an OV-5 to depict process flow (such as an IDEF3 model). A process flow model captures precedence and causality relations between situations and events by providing a structured method for expressing knowledge about how a process or organization works. A process flow model should be annotated with the names of the operational nodes responsible for conducting those activities.

The net-centric OV-6c describes the business and mission processes that need to be executed to achieve Net-Centric Operations (NCO). The ability to discover, access, and understand information and capabilities from the NCE, where and when they are needed, is supported by the OV-6c and can be decomposed to the level of specificity required for the subject architecture. In the NCE, the OV-6c may depict the following:

- Exchanges between the Service Functionality Providers and Service Consumers, the Service Consumers and external Service Functionality Providers, and between the Service Functionality Providers, and Unanticipated Users.
- Sequences that describe the timeline for the availability of information for any of its refinement states (raw, preprocessed, fused).
- Handling, methodologies, and the Enterprise Information Environment (EIE) infrastructure components that support the operational concepts of post before processing.
- Illustration of one-to-many, many-to-one, and many-to-many exchanges between Service Functionality Providers and Service Consumers found in the net-centric OV-3.

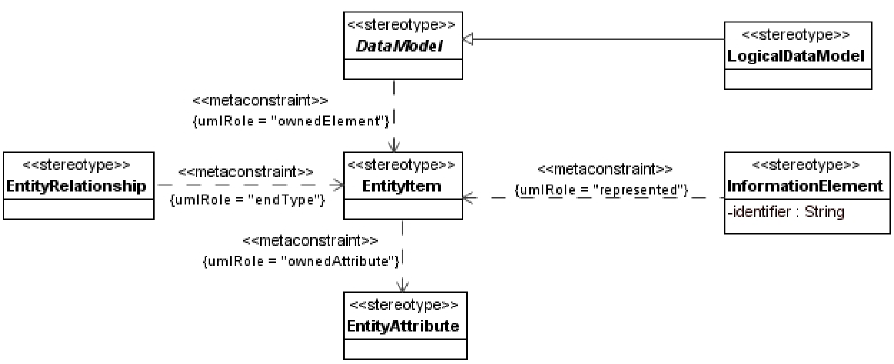


Figure 8.201 - Elements related to the OV-7 stereotype

The Logical Data Model describes the structure of an architecture domain’s system data types and the structural business process rules (defined in the architecture's Operational View) that govern the system data. It provides a definition of architecture domain data types, their attributes or characteristics, and their interrelationships.

OV-7, including the domain's system data types or entity definitions, is a key element in supporting interoperability between architectures, since these definitions may be used by other organizations to determine system data compatibility. Often, different organizations may use the same entity name to mean very different kinds of system data with different internal structure. This situation will pose significant interoperability risks, as the system data models may appear to be compatible, each having a Target Track data entity but having different and incompatible interpretations of what Target Track means.

In the NCE, the OV-7 describes the structure of data types (information elements) for information being made available or being consumed by the OV-5 activities and provides the organization and composition of metadata that can be used to characterize the information exchanged in the NCE.

8.1.1.2.2.1 UPDM L1: :UPDM L0: :DoDAF: :OperationalElements: :Structure

8.1.1.2.2.1.1 OperationalNode

An alias for Node in the DoDAF environment.

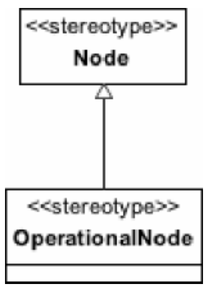


Figure 8.202 - Elements related to the OperationalNode stereotype

Extensions

The following are extensions for OperationalNode:

- Class

Generalizations

The following are generalization relationships for OperationalNode:

- Node

8.1.1.2.2.1.2 OperationalRule

UPDM: A DoDAF v1.5 alias for OperationalConstraint. Required for backward compatibility with DoDAF V1.5

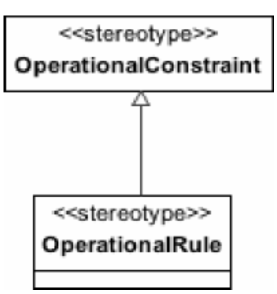


Figure 8.203 - Elements related to the OperationalRule stereotype

Extensions

The following are extensions for OperationalRule:

- Constraint

Generalizations

The following are generalization relationships for OperationalRule:

- OperationalConstraint

8.1.1.2.2.1.3 ExternalNode

UPDM: The OV-2 graphic includes internal operational nodes (internal to the architecture) as well as external nodes (external to the architecture). External Nodes are not within the functional scope of the architecture but the interface to the External Nodes must be considered as part of the operational and systems analysis.

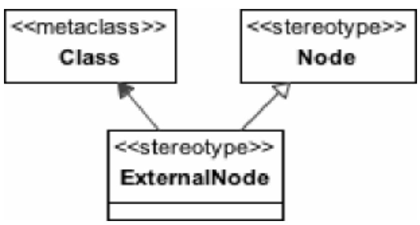


Figure 8.204 - Elements related to the ExternalNode stereotype

The following are extensions for ExternalNode:

- Class

Generalizations

The following are generalization relationships for ExternalNode:

- Node

8.1.1.2.2.1.4 Mission

A Mission summarizes the goals and objectives of the mission. It describes the purpose along with the intended action.

MODAF: A purpose to which a person, organization, or autonomous system is tasked.

DoDAF: The task, together with the purpose, that clearly indicates the action to be taken.

CADM: (1/3) (A) The task, together with the purpose, that clearly indicates the action to be taken.

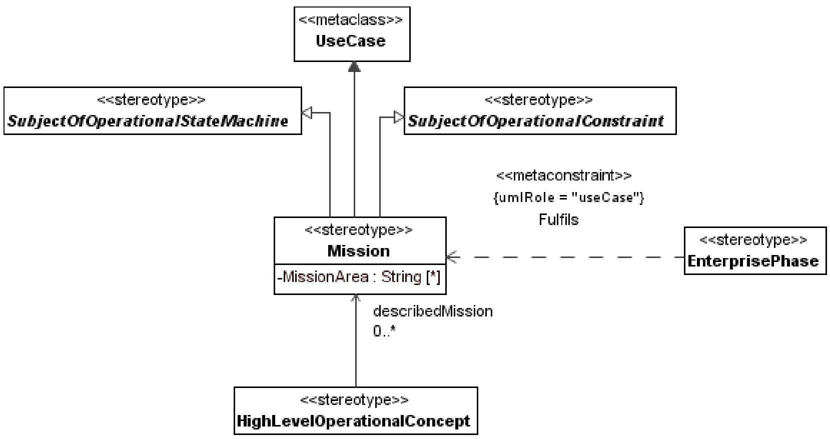


Figure 8.205 - Elements related to the Mission stereotype

Attributes

The following are attributes of Mission:

- MissionArea : String[*] - The area in which the Mission will take place.

Extensions

The following are extensions for Mission:

- UseCase

Generalizations

The following are generalization relationships for Mission:

- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine

8.1.1.2.3 UPDM L1 ::UPDM L0::DoDAF::SystemElements

The System View elements for DoDAF specific models.

8.1.1.2.3.1 UPDM L1: :UPDM L0: :DoDAF: :SystemElements: :Behavior

Defines the behavioral parts of the system elements.

8.1.1.2.3.1.1 SystemFunction

A DoDAF alias for Function.

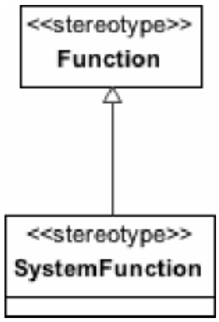


Figure 8.206 - Elements related to the SystemFunction stereotype

Extensions

The following are extensions for SystemFunction:

- Activity

Generalizations

The following are generalization relationships for SystemFunction:

- Function

8.1.1.2.3.1.2 SystemFunctionAction

A DoDAF alias for FunctionAction.

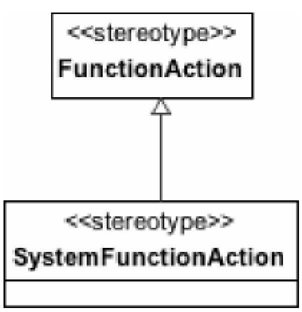


Figure 8.207 - Elements related to the SystemFunctionAction stereotype

Extensions

The following are extensions for SystemFunctionAction:

- CallBehaviorAction

Generalizations

The following are generalization relationships for SystemFunctionAction:

- FunctionAction

8.1.1.2.3.1.3 SystemFunctionEdge

A DoDAF alias for FunctionEdge.

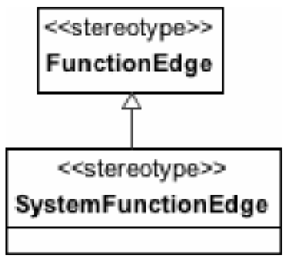


Figure 8.208 - Elements related to the SystemFunctionEdge stereotype

Extensions

The following are extensions for SystemFunctionEdge:

- ActivityEdge

Generalizations

The following are generalization relationships for SystemFunctionEdge:

- FunctionEdge

8.1.1.2.3.2 UPDM L1: :UPDM L0: :DoDAF: :SystemElements::Flows

Defines the flows parts of the system elements.

8.1.1.2.3.2.1 DataExchange

A DoDAF alias for ResourceInteraction.

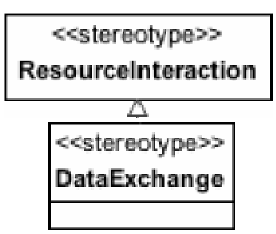


Figure 8.209 - Elements related to the DataExchange stereotype

Extensions

The following are extensions for DataExchange:

- InformationFlow

Generalizations

The following are generalization relationships for DataExchange:

- ResourceInteraction

8.1.1.2.3.3 UPDM L1: :UPDM L0: :DoDAF: :SystemElements: :Structure

Defines the structure parts of the system elements.

8.1.1.2.3.3.1 System

A DoDAF alias for ResourceArtifact.

A System is any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions. Systems consist of family of systems (FoS), System of systems (SoS), Networks of systems, individual Systems, and items (e.g., equipment SystemHardware and SystemSoftware).

A System provides the core construct of the SystemView. A System can be expressed in UML using a variety of constructs, for example, a component or a class. Systems will have a number of operations and functions that they will deliver. The subset of information that relates to SystemFunctions will typically reflect only those functions of interest to link to OperationalActivities.

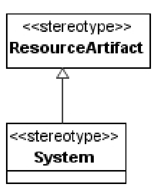


Figure 8.210 - Elements related to the System stereotype

Extensions

The following are extensions for System:

- Class

Generalizations

The following are generalization relationships for System:

- ResourceArtifact

8.1.1.2.3.3.2 SystemConnector

UPDM: A link between two systems.

MODAF: Asserts that a connection exists between two ports belonging to parts in a system composite structure model (MODAF:: SystemPortConnector).

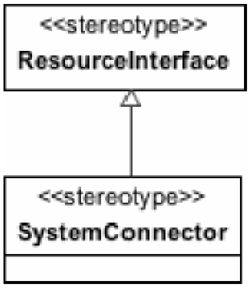


Figure 8.211 - Elements related to the SystemConnector stereotype

Extensions

The following are extensions for SystemConnector:

- Association
- Connector

Generalizations

The following are generalization relationships for SystemConnector:

- ResourceInterface

8.1.1.2.3.3.3 TechnologyForecast

UPDM: DoDAF v1.5 Element: A statement about the future state of one or more types of standard.

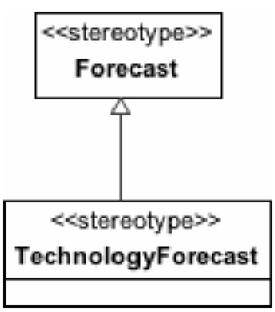


Figure 8.212 - Elements related to the TechnologyForecast stereotype

Extensions

The following are extensions for TechnologyForecast:

- Comment

Generalizations

The following are generalization relationships for TechnologyForecast:

- Forecast

8.1.1.3 UPDM L1::UPDM L0::MODAF

Elements that are not considered part of the Core architectural model, but necessary for MoDAF.

8.1.1.3.1 UPDM L1 ::UPDM L0::MODAF::AcquisitionElements

The Acquisition View elements for MoDAF specific models.

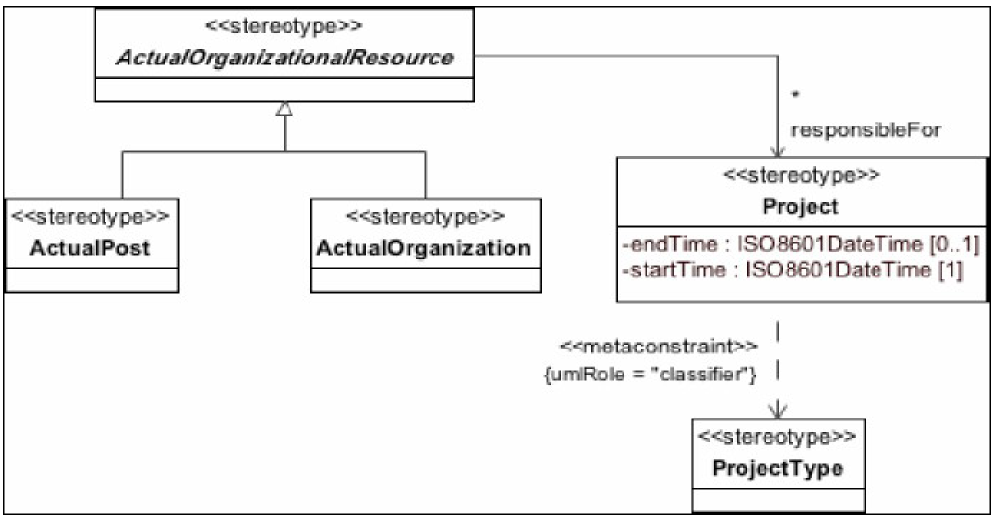


Figure 8.213 - Elements related to the AcV-1 stereotype

MODAF: AcV-1 view products represent an organizational perspective on projects.

DoDAF: AcV-1 view [DoDAF::Project Portfolio Relationships (PV-1) DoDAF-described View] represents an organizational perspective on programs, projects, or a portfolio of projects.

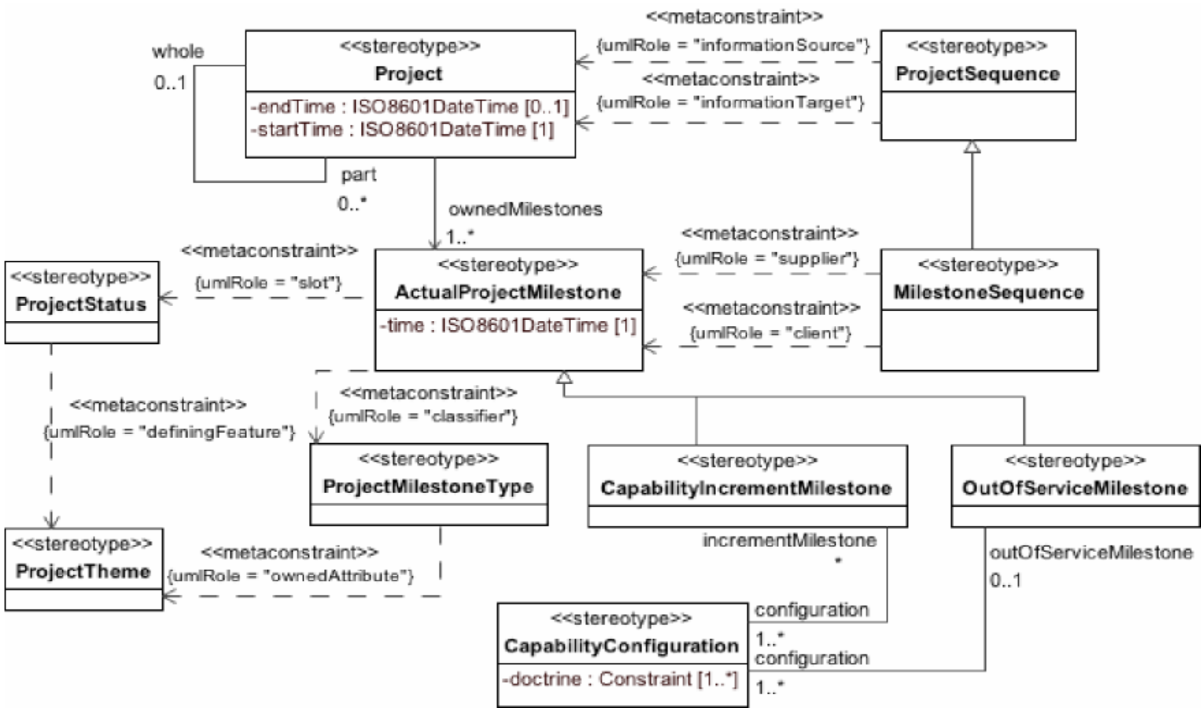


Figure 8.214 - Elements related to the AcV-2 stereotype

MODAF: AcV-2 view products provide a timeline perspective on projects.

DoDAF: AcV-2 (DoDAF::PV-2: Project Timelines DoDAF-described View) provides a timeline perspective on programs or projects.

8.1.1.3.1.1 UPDM L1 ::UPDM L0::MODAF::AcquisitionElements::Milestones

Milestones are an event in a Project by which progress is measured.

8.1.1.3.1.1.1 IncrementMilestone

MODAF: (MODAF::CapabilityIncrement): An ActualProjectMilestone (MODAF::ProjectMilestone) that indicates the point in time at which a project is predicted to deliver or has delivered a Capability.

DoDAF: NA

Example: When a project reaches Initial Operating Capability (IOC) it may deliver a CapabilityRequirement with a given set of metrics then deliver a second CapabilityRequirement corresponding to the same Capability when it reaches Full Operational Capability (FOC). Both the IOC and FOC milestones would be instances of CapabilityIncrement.

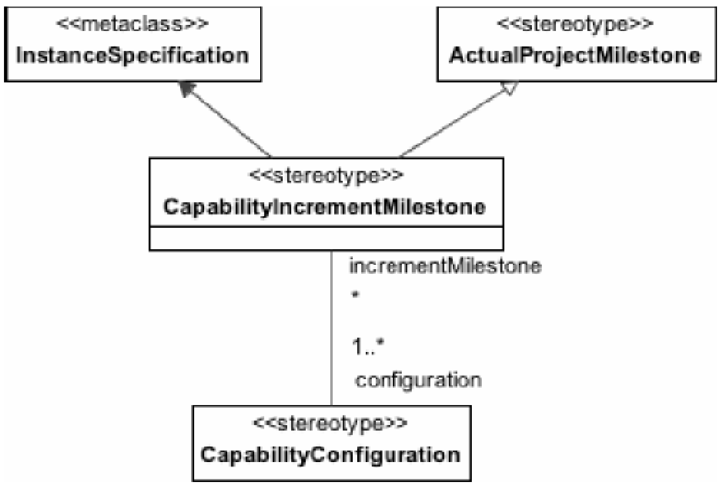


Figure 8.215 - Elements related to the CapabilityIncrementMilestone stereotype

Attribute

The following are attributes for CapabilityIncrementMilestone:

- configuration : CapabilityConfiguration[1..*] - CapabilityConfiguration that are added at this milestone.

Extensions

The following are extensions for CapabilityIncrementMilestone:

- InstanceSpecification

Generalizations

The following are generalization relationships for CapabilityIncrementMilestone:

- ActualProjectMilestone

8.1.1.3.1.1.2 OutOfServiceMilestone

MODAF: An OutOfServiceMilestone (MODAF::OutOfService) is a ProjectMilestone that indicates a project's deliverable is to go out of service.

DoDAF: NA

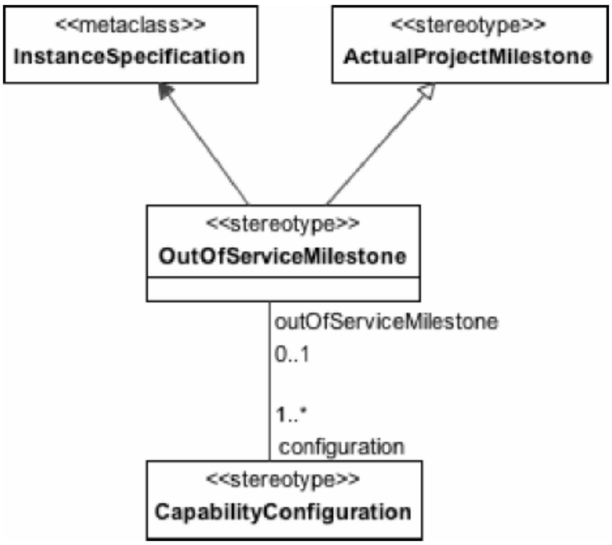


Figure 8.216 - Elements related to the OutOfServiceMilestone stereotype

Attribute

The following are attributes for OutOfServiceMilestone:

- configuration : CapabilityConfiguration[1 ..*] - CapabilityConfiguration that goes out of service at this OutOfServiceMilestone.

Extensions

The following are extensions for OutOfServiceMilestone:

- InstanceSpecification

Generalizations

The following are generalization relationships for OutOfServiceMilestone:

- ActualProjectMilestone

8.1.1.3.1.1.3 ProjectMilestoneType

A type of ProjectMilestone.

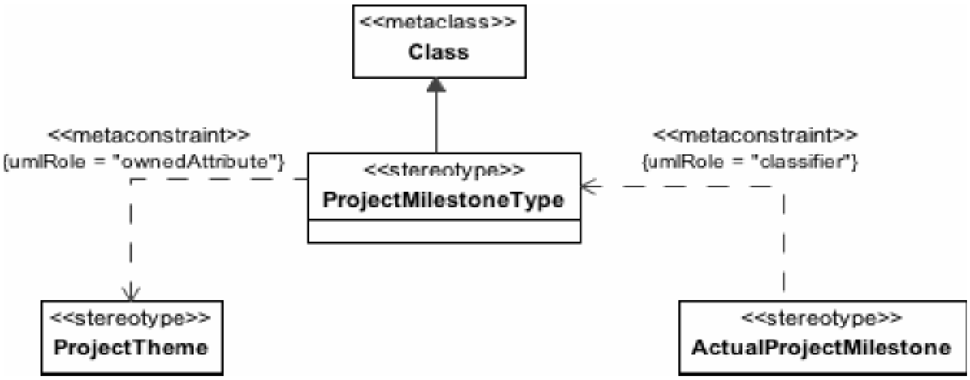


Figure 8.217 - Elements related to the ProjectMilestoneType stereotype

Constraints

The following are constraints for ProjectMilestoneType:

- ProjectMilestoneType.ownedAttributes - Owned attributes have to be stereotyped «ProjectTheme».

Extensions

The following are extensions for ProjectMilestoneType:

- Class

Generalizations

The following are generalization relationships for ProjectMilestoneType:

- UPDMElement

8.1.1.3.1.1.4 ProjectSequence

MODAF: Asserts that one ActualProject (MODAF::Project) follows from another - i.e., the target ActualProject cannot start until the source ActualProject has ended.

DoDAF: NA

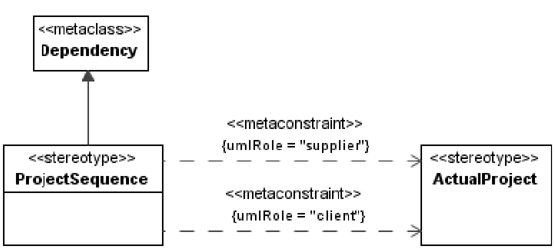


Figure 8.218 - Elements related to the ProjectSequence stereotype

Constraints

The following are constraints for ProjectSequence:

- ProjectSequence .client ProjectSequence.informationSource - Client property value must be stereotyped «ActualProject» or its specializations.
- ProjectSequence.supplier - Supplier property value must be stereotyped «ActualProject» or its specializations.

Extensions

The following are extensions for ProjectSequence:

- Dependency

Generalizations

The following are generalization relationships for ProjectSequence:

- UPDMElement

8.1.1.3.1.2 UPDM L1 ::UPDM L0::MODAF::AcquisitionElements::Structure

Structure for Acquisition View elements for MoDAF specific models.

8.1.1.3.1.2.1 ActualProject

MODAF: (MODAF::Project): A time-limited endeavor to create a specific set of products or services.

DoDAF: (DoDAF::Project): A temporary endeavor undertaken to create Resources or Desired Effects.

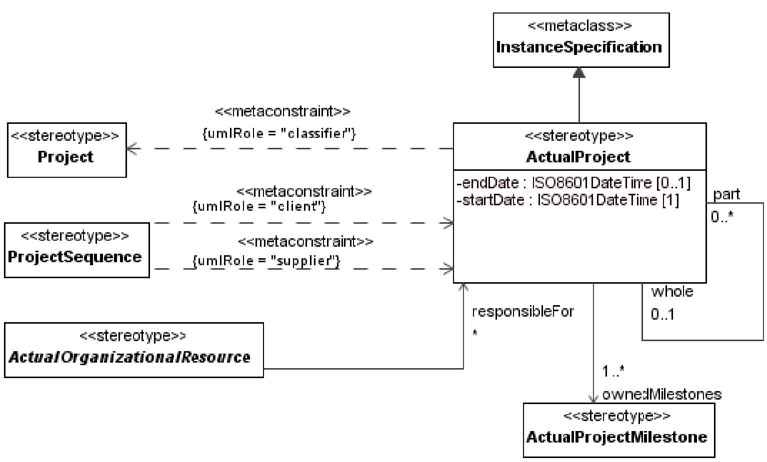


Figure 8.219 - Elements related to the ActualProject stereotype

Constraints

The following are constraints for ActualProject:

- Actual Project.classifier - Classifier property value must be stereotyped «Project» or its specializations.

Attribute

The following are attributes for ActualProject:

- endTime : ISO8601DateTime[0.. 1] - End time for this ActualProject.
- ownedMilestones : ActualProjectMilestone[1..*] - Milestones associates with this ActualProject.
- part : ActualProject[0..*] - Sub-projects.
- startTime : ISO8601DateTime[1] - Start time for this Project.
- whole : ActualProject[0..1] - Parent project.

Extensions

The following are extensions for ActualProject:

- InstanceSpecification

Generalizations

The following are generalization relationships for ActualProject:

- UPDMElement

8.1.1.3.1.2.2 ProjectStatus

MODAF: A ProjectStatus (MODAF::StatusAtMilestone) is a relationship between a Status and a milestone that asserts the status (i.e., level of progress) of a ProjectTheme for the project at the time of the ActualProjectMilestone (MODAF::Milestone).

DoDAF: NA

For example, a procurement project may have workstreams corresponding to lines of development. The status of each workstream is summarized on the milestone in the AcV-2.

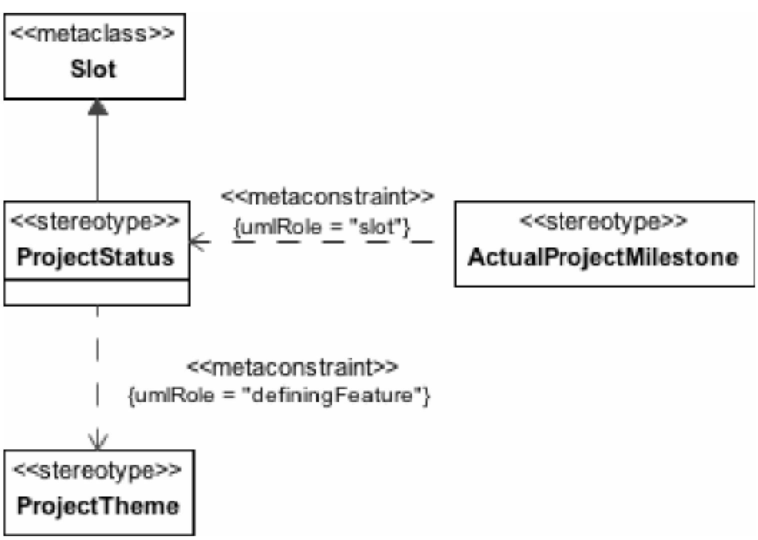


Figure 8.220 - Elements related to the ProjectStatus stereotype

Constraints

The following are constraints for ProjectStatus:

- ProjectStatus.definingFeature - DefiningFeature value must be stereotyped «ProjectTheme» or its specializations.

Extensions

The following are extensions for ProjectStatus:

- Slot

Generalizations

The following are generalization relationships for ProjectStatus:

- UPDMElement

8.1.1.3.1.2.3 ProjectTheme

MODAF:An aspect by which the progress of various Projects may be measured. In UK MOD, this could be one of the defense lines of development (DLOD), or DOTMLPF in the US.

DoDAF: NA

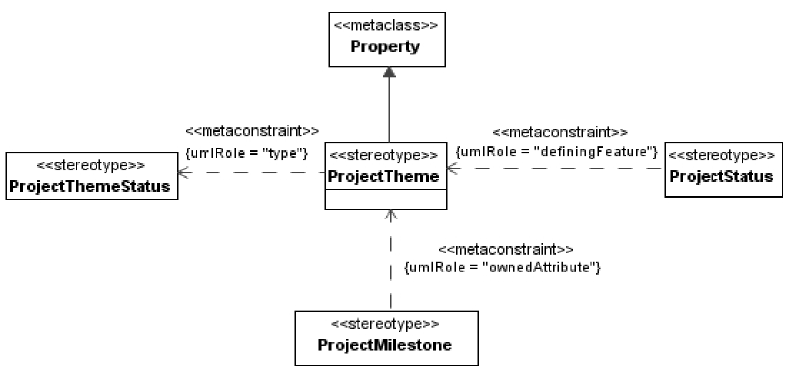


Figure 8.221 - Elements related to the ProjectTheme stereotype

Extensions

The following are extensions for ProjectTheme:

- Property

Generalizations

The following are generalization relationships for ProjectTheme:

- UPDMElement

8.1.1.3.1.2.4 Project

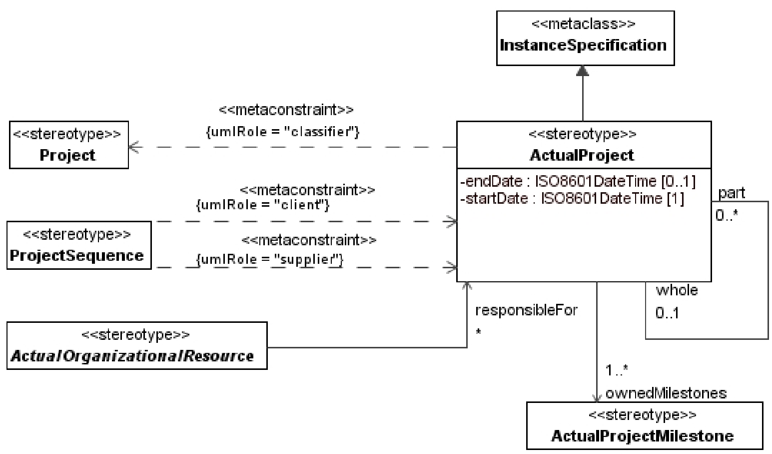
MODAF: A Project (MODAF::ProjectType) is used to define a category of project (for example, “Program,” “Acquisition Project,” or “Training Program”).

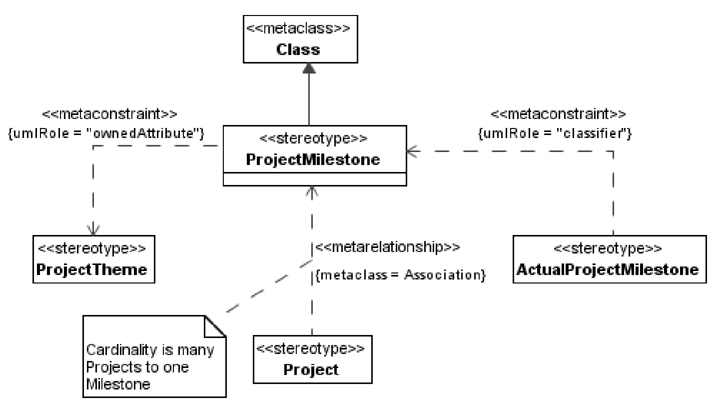
DoDAF: NA (only Individual Project in DoDAF)

Example: “Program”

Example: “Acquisition Project”

Example: “Training Program”





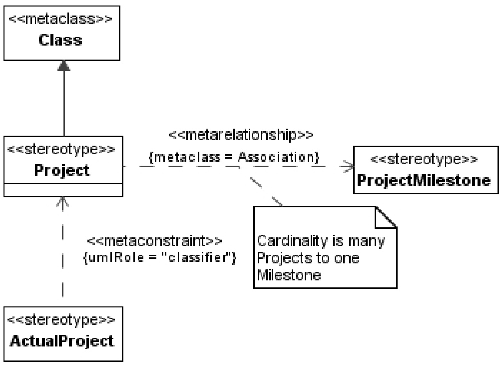


Figure 8.222 - Elements related to the Project stereotype

Attributes

The following are attributes for Project:

- endDate : ISODateTime[0 ..1] – End time for this project.
- startDate { ISODateTime[1] – Start time for this project.

Extensions

The following are extensions for Project:

- Class

Generalizations

The following are generalization relationships for Project:

- UPDMElement

8.1.1.3.1.2.5 ProjectThemeStatus

UPDM: Specifies a status for a ProjectTheme (such as training status).

MODAF: An enumeration of the possible statuses (MODAF::StatusIndicator) for one of more ProjectThemes.

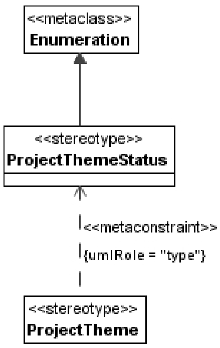


Figure 8.223 - Elements related to the ProjectThemeStatus stereotype

Extensions

The following are extensions for ProjectThemeStatus:

- Enumeration

Generalizations

The following are generalization relationships for ProjectThemeStatus:

- UPDMElement

8.1.1.3.2 UPDM L1 ::UPDM L0::MODAF::AllElements

The All View elements for MoDAF specific models.

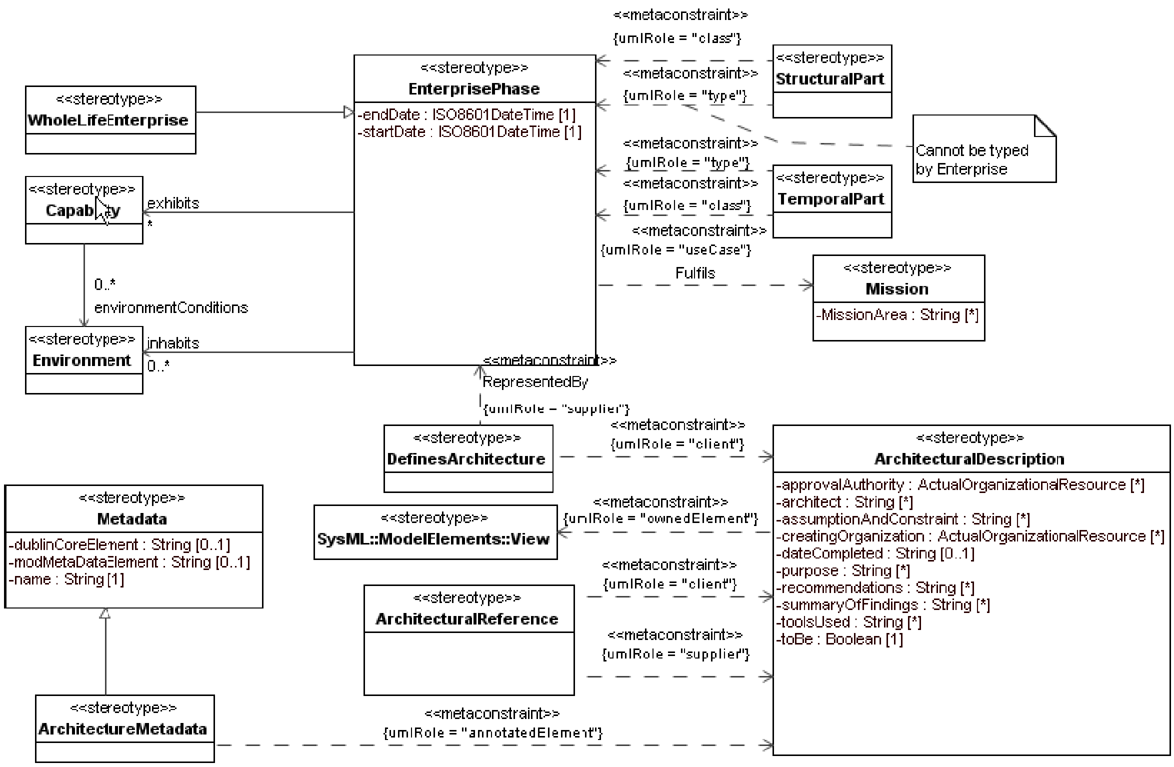


Figure 8.224 - Elements related to the AV-1 stereotype

The Overview & Summary Information identifies the architecture goals, viewpoint, findings, and recommendations. The AV-1 contains sufficient textual information to enable a reader to select one architecture from among many to read in more detail. AV-1 serves two additional purposes:

1. In the initial phases of architecture development, it serves as a planning guide.
2. Upon completion of an architecture, the AV-1 provides summary textual information concerning that architecture. The AV-1 comprises a textual executive summary of a given architecture and documents the following descriptions.

Architecture Project Identification - identifies the architecture project name, the architect, and the organization developing the architecture. It also includes assumptions and constraints, identifies the approving authority and the completion date, and records the level of effort and costs (projected and actual) required to develop the architecture.

Scope - identifies 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, and so forth. Scope also identifies the organizations and Communities Of Interest (COI) that fall within the scope of the architecture. The scope also includes the COI that are related to the architecture.

Purpose and Viewpoint - explains the need for the architecture, what it should demonstrate, the types of analyses (e.g., Activity-Based Costing) that will be applied to it, who is expected to perform the analyses, what decisions are expected to be made on the basis of an analysis, who is expected to make those decisions, and what actions are expected to result. The viewpoint from which the architecture is developed is identified (e.g., planner or decision maker).

Context - describes the setting in which the architecture exists. It includes such things as mission, doctrine, relevant goals and vision statements, concepts of operation, scenarios, information assurance context (e.g., types of system data to be protected, such as classified or sensitive but unclassified, and expected information threat environment), other threats and environmental conditions, and geographical areas addressed, where applicable. Context also identifies authoritative sources for the rules, criteria, and conventions that were followed. The tasking for the architecture project and known or anticipated linkages to other architectures are identified.

Tools and File Formats Used - identifies the tool suite used to develop the architecture and file names and formats for the architecture and each product.

Findings - states the findings and recommendations that have been developed based on the architecture effort. Examples of findings include identification of shortfalls, recommended system implementations, and opportunities for technology insertion.

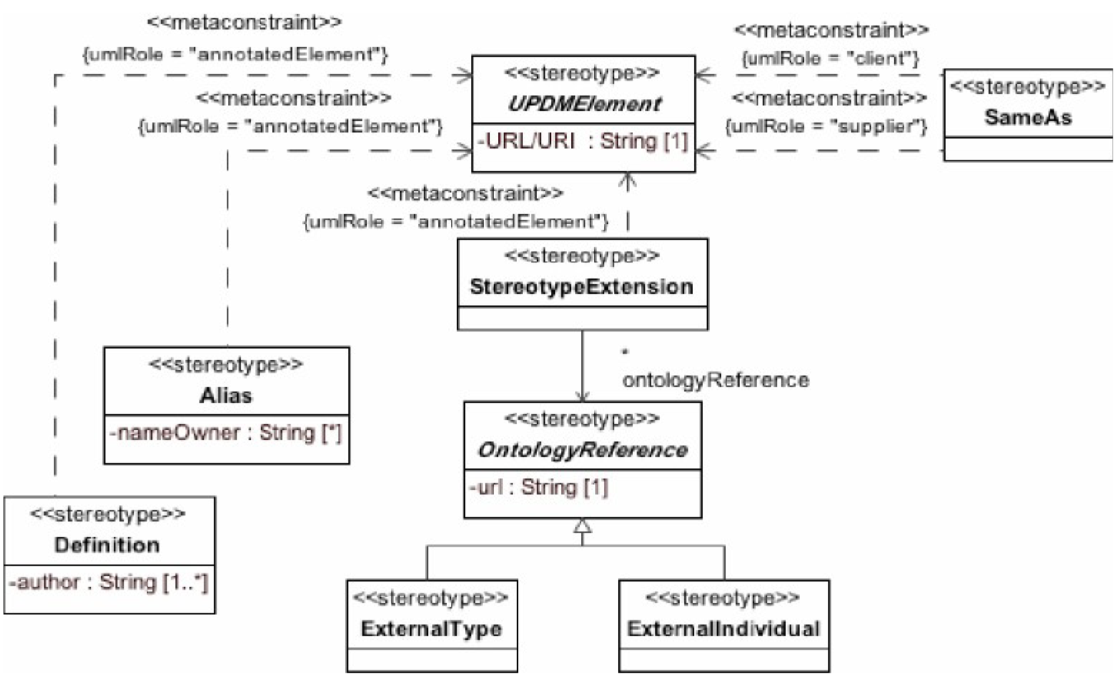


Figure 8.225 - Elements related to the AV-2 stereotype

The AV-2 contains definitions of terms used in the given architecture. It consists of textual definitions in the form of a glossary, a repository of architecture data, their taxonomies, and their metadata (i.e., data about architecture data), including metadata for tailored products, associated with the architecture products developed. Metadata are the architecture data types, possibly expressed in the form of a physical schema. In this document, architecture data types are referred to as architecture data elements.

An architecture via the AV-2 can also reference elements in an external ontology using the StereotypeExtension or import the element (as a form of OntologyReference) and map it to the appropriate element in the architecture using the generalization or SameAs relationship.

AV-2 provides a central repository for a given architecture's data and metadata. AV-2 enables the set of architecture products to stand alone, allowing them to be read and understood with minimal reference to outside resources. AV-2 is an accompanying reference to other products, and its value lies in unambiguous definitions. The key to long-term interoperability can reside in the accuracy and clarity of these definitions.

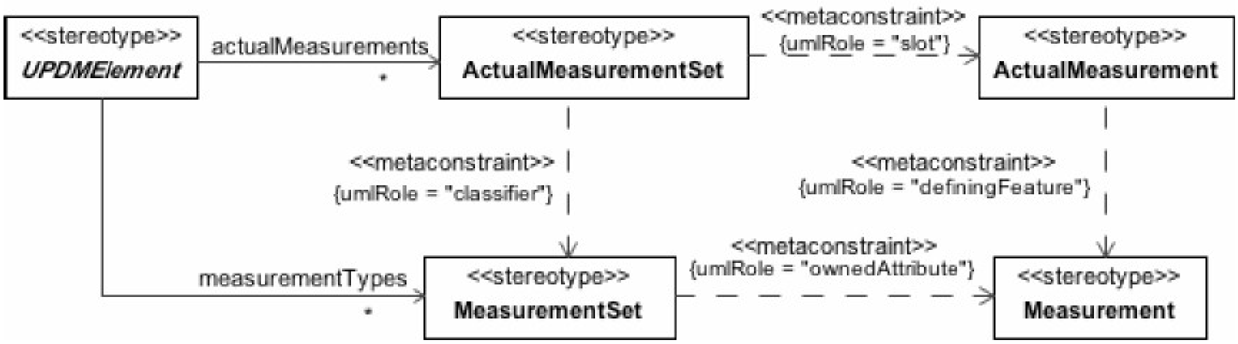


Figure 8.226 - Elements related to the AV-3 stereotype

The AV-3 defines the measureable properties of something in the physical world, expressed in amounts of a unit of measure that can be associated with a UPDMElement.

8.1.1.3.2.1 UPDM L1 ::UPDM L0::MODAF::AllElements::Ontology

Ontology elements from All Elements.

8.1.1.3.2.1.1 Alias

A UPDM Artifact used to define an alternative name for an element as used by DoDAF or MODAF.

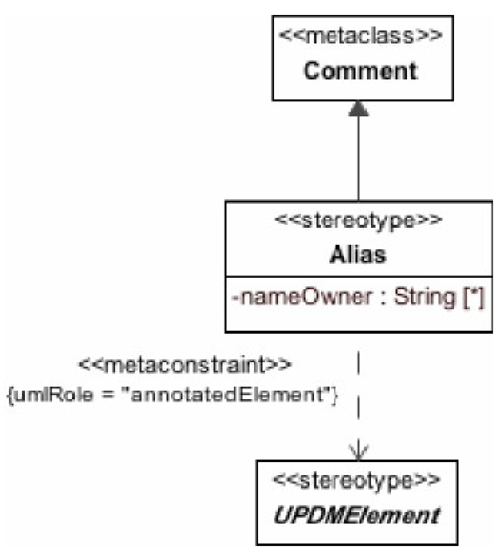


Figure 8.227 - Elements related to the Alias stereotype

Constraints

The following are constraints for Alias:

- Alias.annotatedElement - Value for the annotatedElement property must be stereotyped «UPDMElement» or its specializations.

Attribute

The following are attributes for Alias:

- nameOwner : String[*] - The person or organization that uses this alternative name.

Extensions

The following are extensions for Alias:

- Comment

Generalizations

The following are generalization relationships for Alias:

- UPDMElement

8.1.1.3.2.1.2 Definition

UPDM: A definition of an element in the architecture.

Note – Every element added by an architect must have a definition.

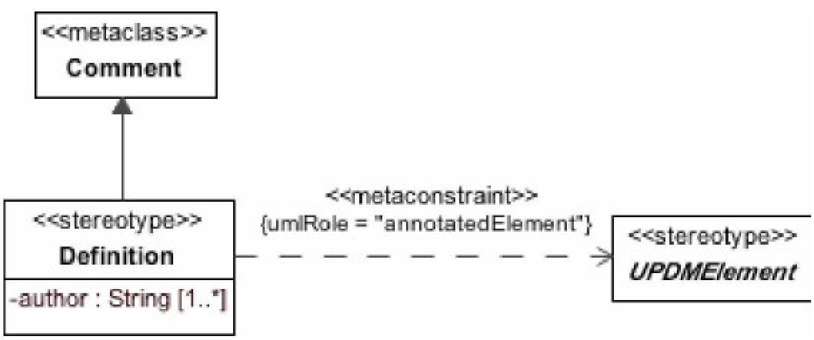


Figure 8.228 - Elements related to the Definition stereotype

Constraints

The following are constraints for Definition:

- Definition.annotatedElement - Value for the annotatedElement property must be stereotyped «UPDMElement» or its specializations.

Attribute

The following are attributes for Definition:

- author : String[1..*] - The original or current person (architect) responsible for the element.

Extensions

The following are extensions for Definition:

- Comment

Generalizations

The following are generalization relationships for Definition:

- UPDMElement

8.1.1.3.2.1.3 ExternalIndividual

MODAF: An individual (i.e., something which has spatial and temporal extent) defined by an external ontology.

DoDAF: NA

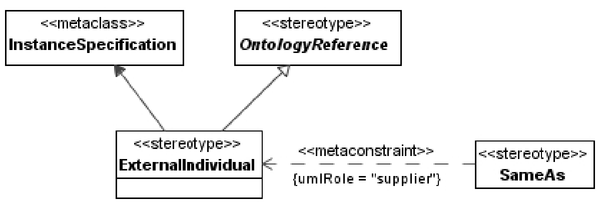


Figure 8.229 - Elements related to the ExternalIndividual stereotype

Extensions

The following are extensions for ExternalIndividual:

- InstanceSpecification

Generalizations

The following are generalization relationships for ExternalIndividual:

- OntologyReference

8.1.1.3.2.1.4 ExternalType

MODAF: A type defined by an external ontology.

DoDAF: NA

Note – This may be higher-order (i.e., a type of a type).

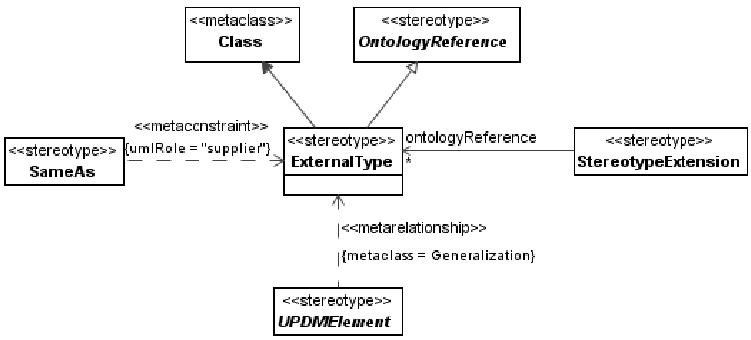


Figure 8.230 - Elements related to the ExternalType stereotype

Extensions

The following are extensions for ExternalType:

- Class

Generalizations

The following are generalization relationships for ExternalType:

- OntologyReference

8.1.1.3.2.1.5 OntologyReference

MODAF: A reference to an element in a recognized external ontology or taxonomy.

Note – OntologyReference is abstract.

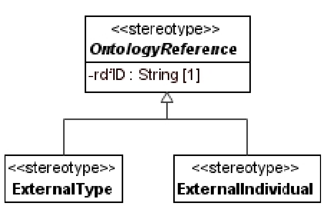


Figure 8.231 - Elements related to the abstract OntologyReference stereotype

Attribute

The following are attributes for OntologyReference:

- rfID : String[1] - Unique identifier for the element.

Generalizations

The following are generalization relationships for OntologyReference:

- UPDMElement

8.1.1.3.2.1.6 SameAs

MODAF: Asserts that two elements refer to the same real-world thing.

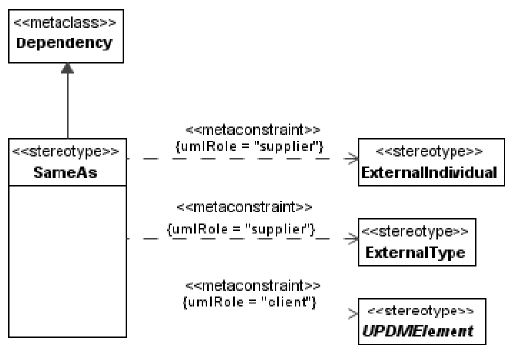


Figure 8.232 - Elements related to the SameAs stereotype

Constraints

The following are constraints for SameAs:

- SameAs.client - Values for the client property must be stereotyped «UPDMElement» or its specializations.
- SameAs.supplier - Values for the supplier property must be stereotyped «ExternalType» or its specializations.

Extensions

The following are extensions for SameAs:

- Dependency

Generalizations

The following are generalization relationships for SameAs:

- UPDMElement

8.1.1.3.2.1.7 StereotypeExtension

MODAF: Defines an additional stereotype used in the architecture which is not defined in this meta-model. The body attribute contains the name of the new stereotype. The extendedStereotype tagged value shall contain the name of the meta-model stereotype which is extended. The ontologyReference tagged value shall be populated with a reference to the external ontology element represented by the new stereotype.

DoDAF: NA

Note 1: This is effectively a short-hand method for representing ontology items in the architecture. New stereotype names can be created at will by the architect, provided that they reference an element in a recognized external ontology.

Note 2: Any stereotypes added by the architect which do not have a corresponding «StereotypeExtension» will be deemed non-compliant and ignored by tools importing data compliant to this meta-model.

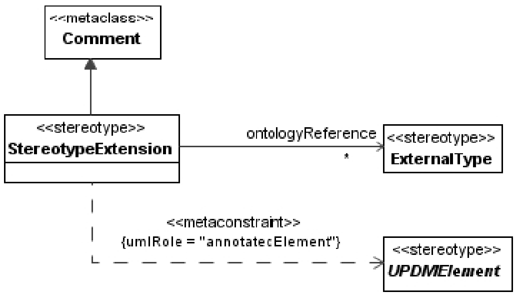


Figure 8.233 - Elements related to the StereotypeExtension stereotype

Constraints

The following are constraints for StereotypeExtension:

- StereotypeExtension.annotatedElement - Values for the annotatedElement property must be stereotyped «UPDMElement» or its specializations.

Attribute

The following are attributes for StereotypeExtension:

- ontologyReference : ExternalTypes [*] - Reference to the source ontology.

Extensions

The following are extensions for StereotypeExtension:

- Comment

Generalizations

The following are generalization relationships for StereotypeExtension:

- UPDMElement

8.1.1.3.3 UPDM L1 ::UPDM L0::MODAF::OperationalElements

The Operational View elements for MoDAF specific models.

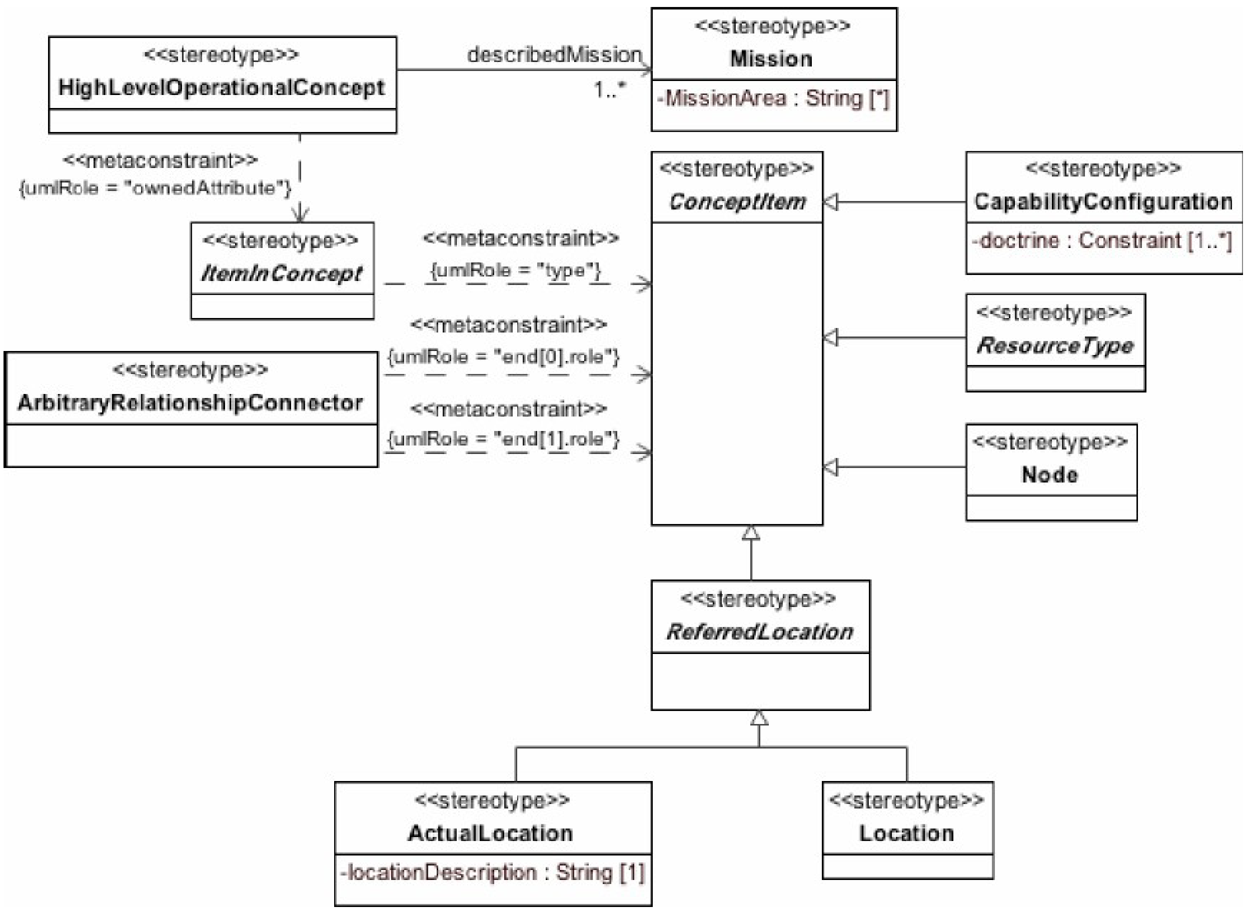


Figure 8.234 - Elements related to the OV-1 stereotype

The purpose of OV-1 is to provide a quick, high- level description of what the architecture is supposed to do, and how it is supposed to do it.

The purpose of High-level Operational Concept Graphic is to provide a high-level graphical and textual description of operational concept (high level organizations, missions, geographic configuration, connectivity) of what the architecture is supposed to do, and how it is supposed to do it. The OV-1, along with the corresponding AV-1 product is intended to serve as an executive summary of the architecture.

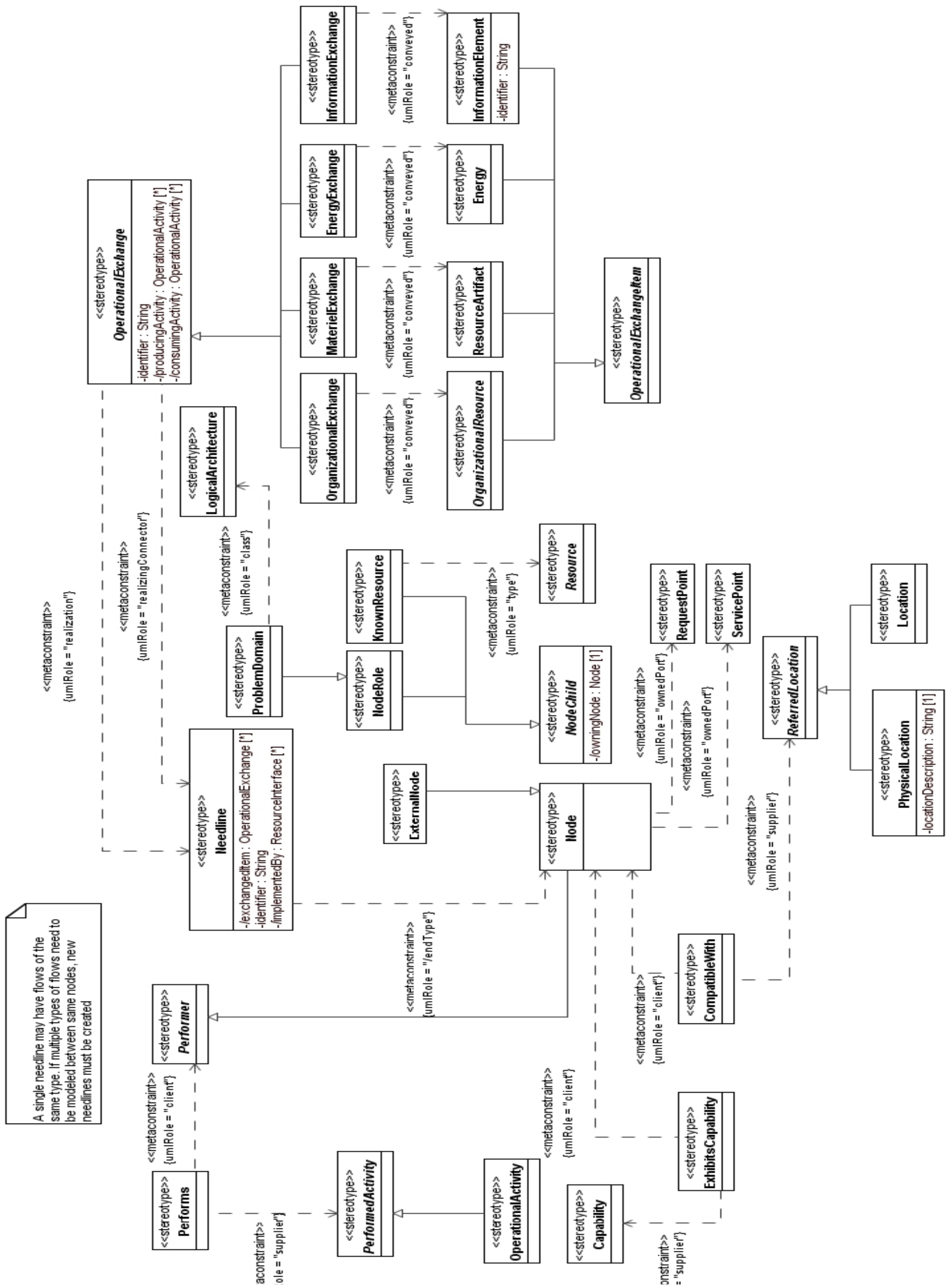


Figure 8.235 - Elements related to the OV-2 stereotype

The Operational Node Connectivity Description is intended to track the need to exchange information from specific operational nodes (that play a key role in the architecture) to others. OV-2 does not depict the connectivity between the nodes. MoDAF modifies the OV-2 is in two ways. First of all, it recommends that an OV-2 diagram (now OV-2a) shows the platforms or geographic locations at which operational nodes are deployed. Secondly it provides additional information (OV-2b) about each needline in the form of a requirements specification. There are now four types of needlines identified as follow:

1. InformationExchange
2. EnergyFlow
3. MaterielFlow
4. MovementOfPeople

In addition, MoDAF permits service-oriented architectures. Instead of needlines between nodes, it is possible simply to show which services the nodes provide and consume. Finally, MoDAF again permits known resources to be shown in an OV-2. However, this must be clearly shown as a KnownResource in an OV-2 model. The concept of a LogicalArchitecture is introduced - this is the container class for all the nodes and KnownResources.

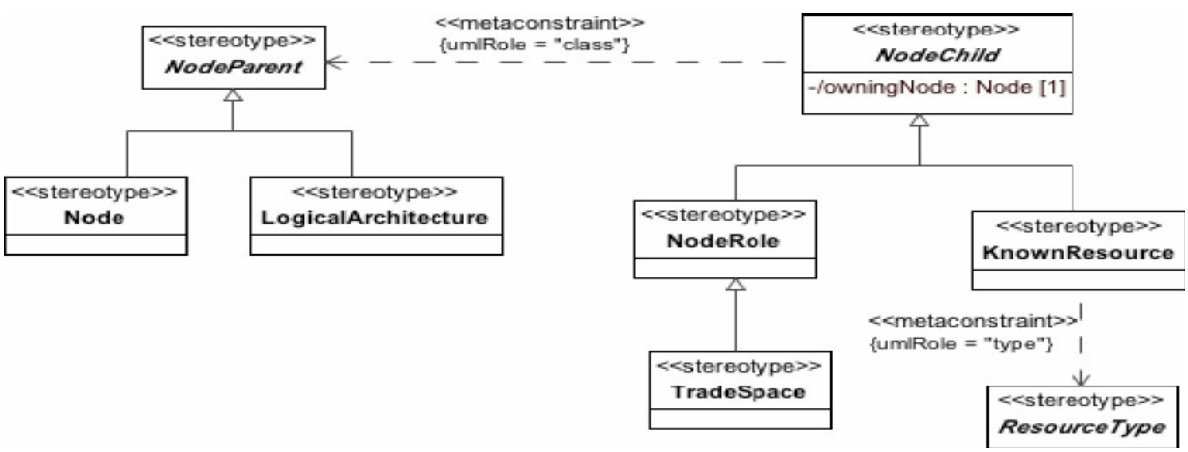


Figure 8.236 - Elements related to the OV-2 Nodes stereotype

Nodes related to OV-2 View.

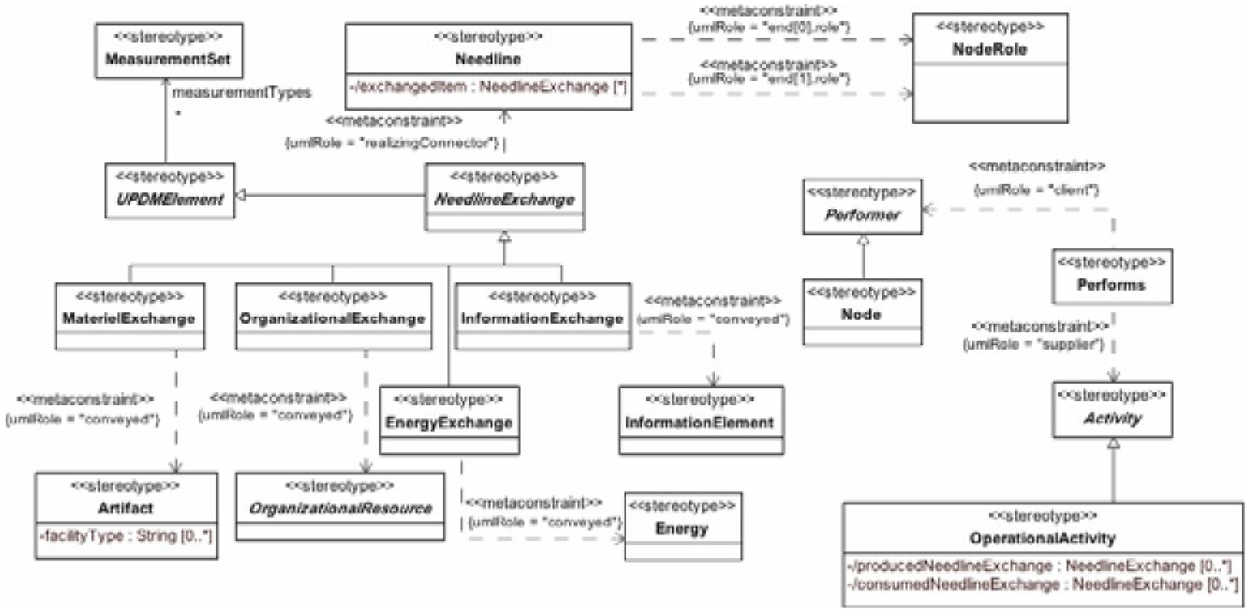


Figure 8.237 - Elements related to the OV-3 stereotype

Information exchanges express the relationship across the three basic architecture data elements of an OV (operational activities, operational nodes, and information flow) with a focus on the specific aspects of the information flow and the information content. The Information Exchanges of the OV-3 should remain at a high level of aggregation to represent actual information workflow products that are used at the operational nodes shown in OV-2 (and not their subordinate operational nodes).

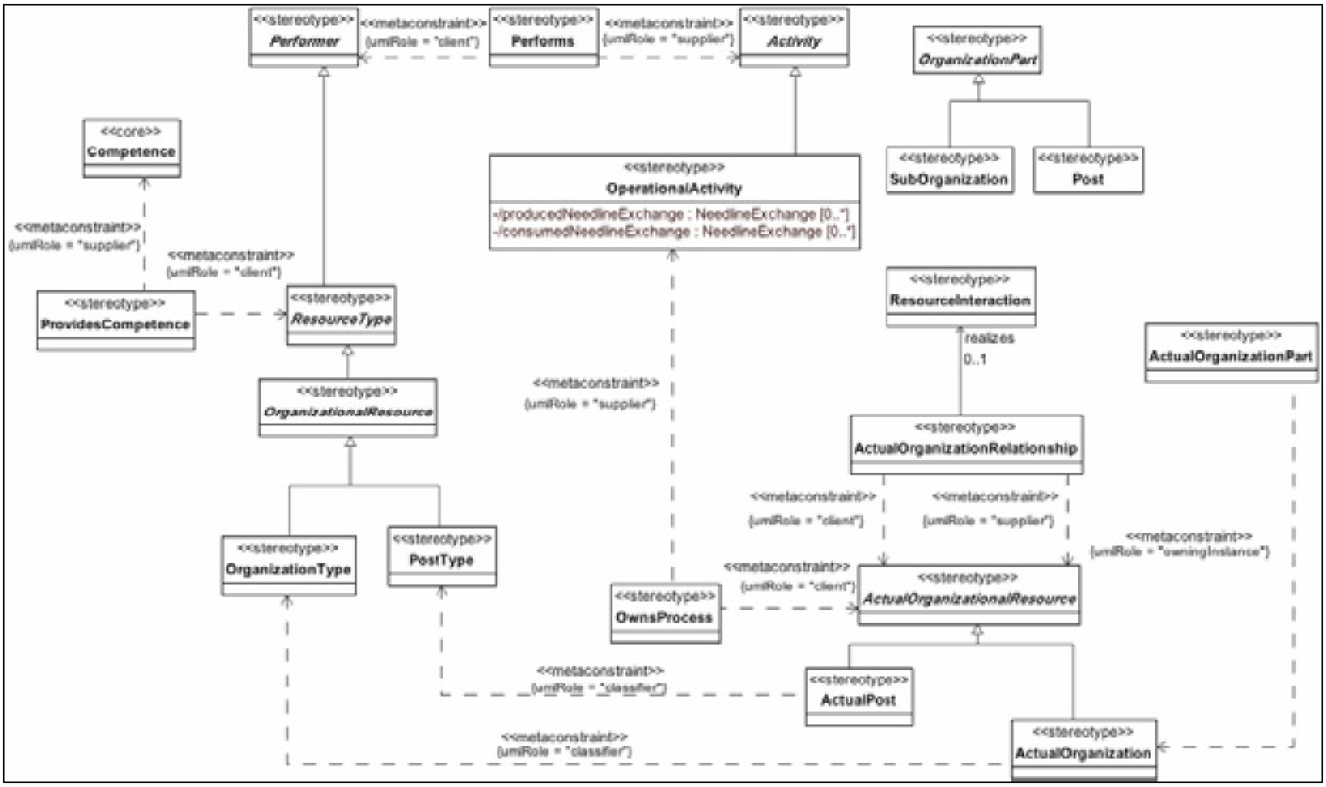


Figure 8.238 - Elements related to the OV-4 Actual stereotype

This is the OV-4 Actual View. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. MoDAF divides The OV-4 two views, an OV-4 Typical and an OV-4 Actual. The former is exactly as the DoDAF OV-4, while the latter is a special form of the SV-1; where the resources are restricted to being organizational.

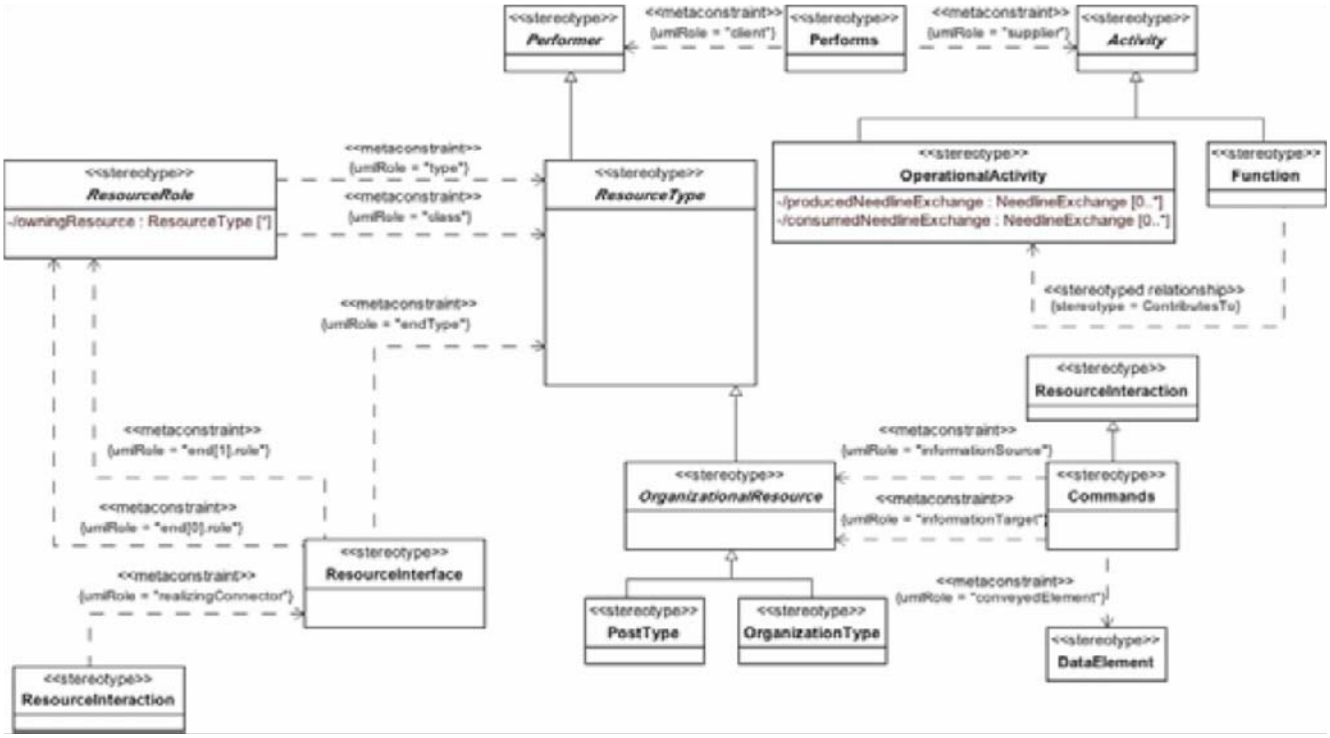


Figure 8.239 - Elements related to the OV-4 Typical stereotype

This is the OV-4 Typical View. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. MoDAF divides The OV-4 two views, an OV-4 Typical and an OV-4 Actual. The former is exactly as the DoDAF OV-4, while the latter is a special form of the SV- 1; where the resources are restricted to being organizational.

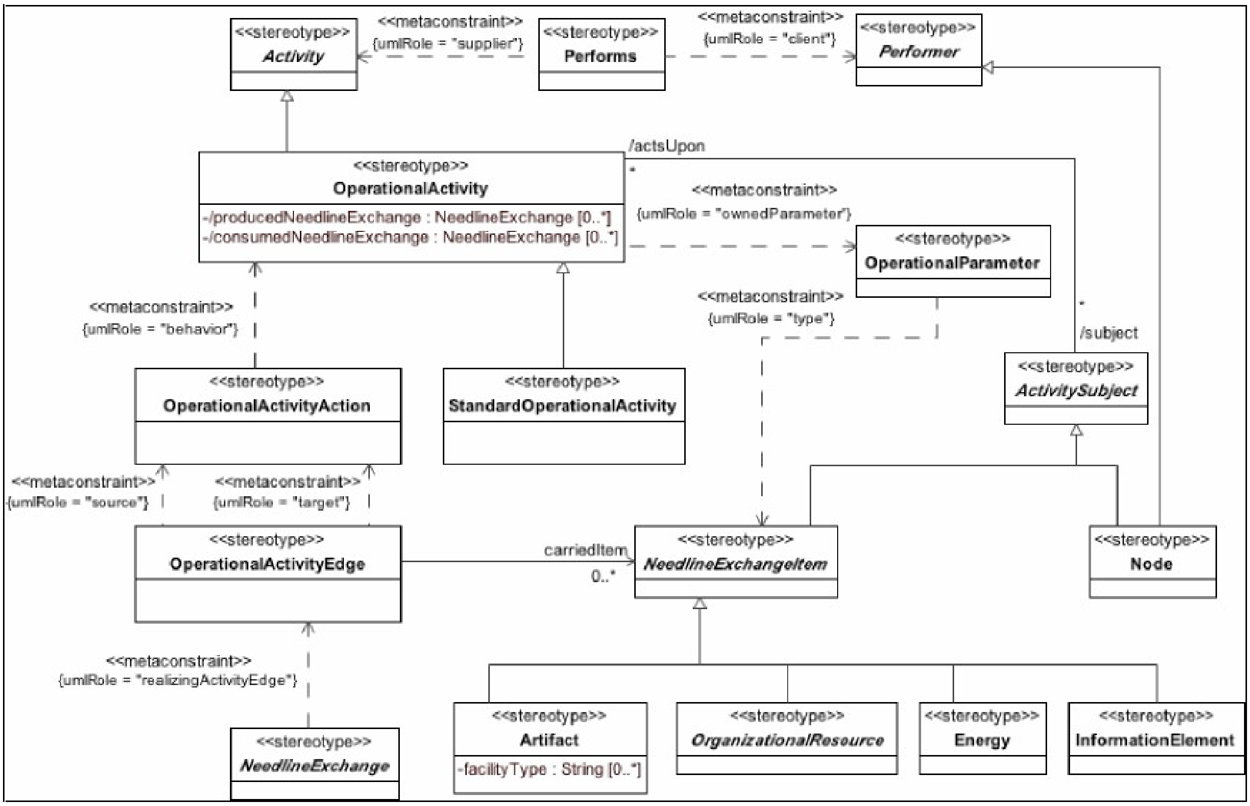


Figure 8.240 - Elements related to the OV-5 stereotype

The Operational Activity Model describes the operations that are normally conducted in the course of achieving a mission or a business goal, from a net-centric perspective. It describes capabilities, operational activities (or tasks), input and output (I/O) flows between activities, and I/O flows to/from activities that are outside the scope of the architecture. It is imperative that the levels-of-detail between the OV-2, OV-3, and OV-5 remain cohesive. For example, if one diagram of OV-2 operational nodes is developed that shows aggregated organizations only, then it is imperative that the corresponding OV-5 product be developed to show only those operational activities that are meaningful with respect to these operational nodes. Similarly, the information exchanges of OV-3 should remain at a high level of aggregation to represent actual information workflow products that are used at the operational nodes depicted in OV-2 (and not their subordinate operational nodes). The net-centric OV-5 may be used in the following ways:

- Delineate lines of dependency on external activities when coupled with OV-2.
- Highlight information flows to depict the status of the information's refinement (raw, pre-processed, fused).
- Provide the critical foundation for depicting Task, Post, Process, and Use (TPPU) activity sequencing and timing in the OV-6a, OV-6b, and OV-6c.
- Identify critical mission threads and operational information exchanges by annotating which activities are critical, i.e., identify the activities in the model that are critical.

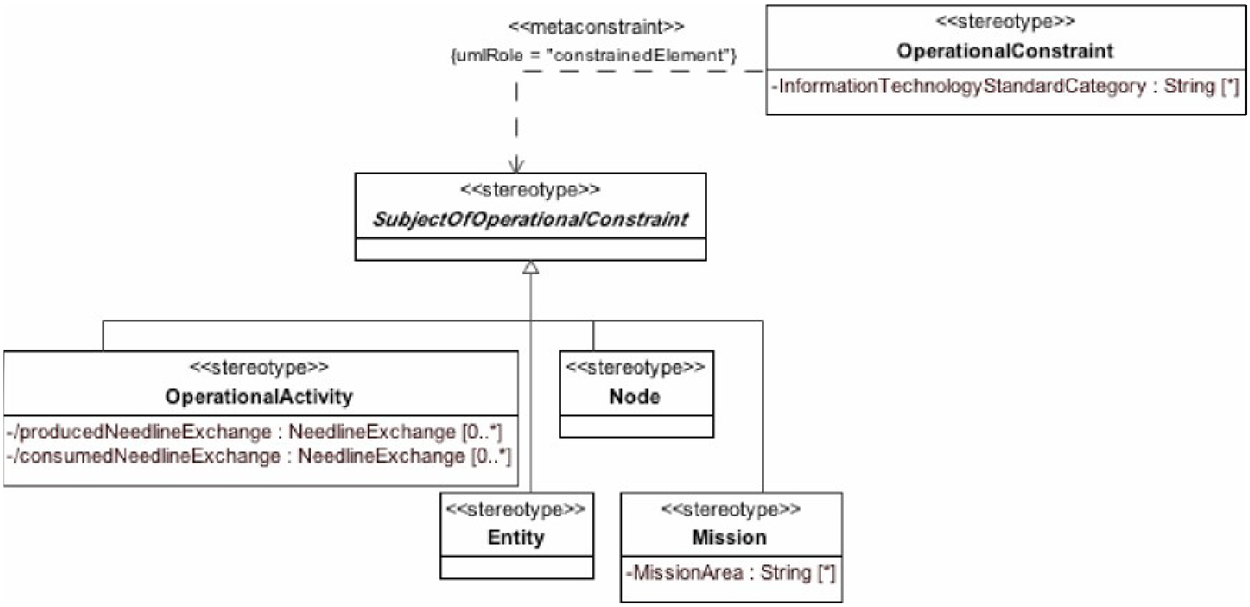


Figure 8.241 - Elements related to the OV-6a stereotype

The Operational Rules Model specifies operational or business rules that are constraints on an enterprise, a mission, operation, business, or an architecture. While other OV products (e.g., OV- 1, OV-2, and OV-5) describe the structure of a business-what the business can do-for the most part, they do not describe what the business must do, or what it cannot do.

At the mission level, OV-6a may consist of doctrine, guidance, rules of engagement, and so forth. At the operation level, rules may include such things as a military Operational Plan (OPLAN). 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).” At a top level, rules should at least embody the concepts of operations defined in OV- 1, and should provide guidelines for the development and definition of more detailed rules and behavioral definitions that will occur later in the architecture definition process.

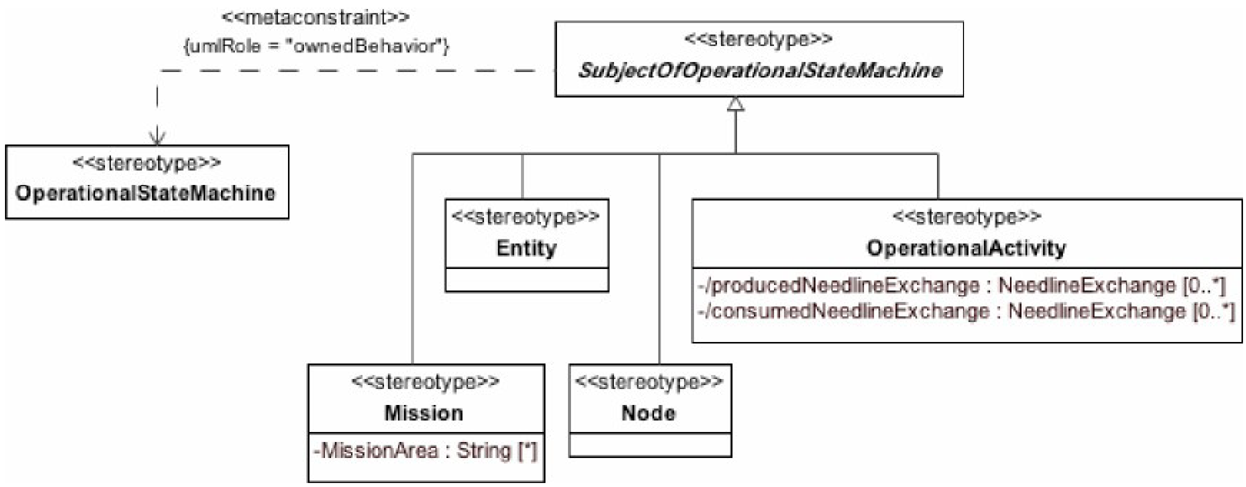


Figure 8.242 - Elements related to the OV-6b stereotype

The Operational State Transition Description is a graphical method of describing how an operational node or activity responds to various events by changing its state. The diagram represents the sets of events to which the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action. The explicit sequencing of activities in response to external and internal events is not fully expressed in OV-5. An OV-6b can be used to describe the explicit sequencing of the operational activities.

Alternatively, OV-6b can be used to reflect the explicit sequencing of actions internal to a single operational activity or the sequencing of operational activities with respect to a specific operational node. In a net-centric architecture, the OV6b is used to describe the set of state transitions for providers and consumers in the Net-Centric Environment (NCE) in response to the posting of information to the NCE or retrieving of information from the NCE.

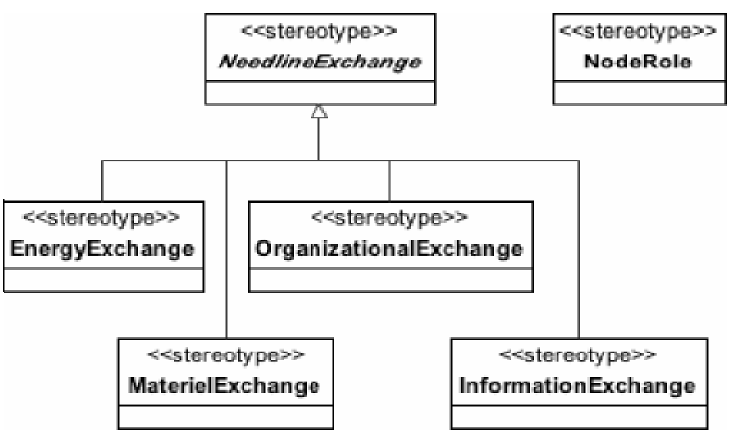


Figure 8.243 - Elements related to the OV-6c stereotype

The Operational Event-Trace Description provides a time-ordered examination of the information exchanges between participating operational nodes as a result of a particular operational thread or scenario. Each eventtrace diagram should have an accompanying description that defines the particular scenario or situation and represent a specific capability. The OV-6c is also used in conjunction with an OV-5 to depict process flow (such as an IDEF3 model). A process flow model captures precedence and causality relations between situations and events by providing a structured method for expressing knowledge about how a process or organization works. A process flow model should be annotated with the names of the operational nodes responsible for conducting those activities.

The net-centric OV-6c describes the business and mission processes that need to be executed to achieve Net-Centric Operations (NCO). The ability to discover, access, and understand information and capabilities from the NCE, where and when they are needed, is supported by the OV-6c and can be decomposed to the level of specificity required for the subject architecture. In the NCE, the OV-6c may depict the following:

- Exchanges between the Service Functionality Providers and Service Consumers, the Service Consumers and external Service Functionality Providers, and between the Service Functionality Providers and Unanticipated Users.
- Sequences that describe the timeline for the availability of information for any of its refinement states (raw, preprocessed, fused).
- Handling, methodologies, and the Enterprise Information Environment (EIE) infrastructure components that support the operational concepts of post before processing.
- Illustration of one-to-many, many-to-one, and many-to-many exchanges between Service Functionality Providers and Service Consumers found in the net-centric OV-3.

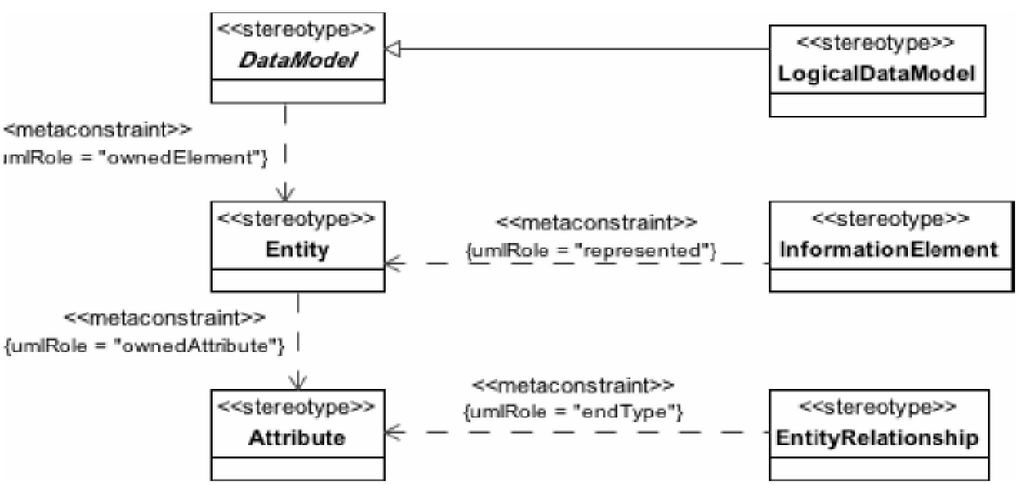


Figure 8.244 - Elements related to the OV-7 stereotype

The Logical Data Model describes the structure of an architecture domain's system data types and the structural business process rules (defined in the architecture's Operational View) that govern the system data. It provides a definition of architecture domain data types, their attributes or characteristics, and their interrelationships. OV-7, including the domain's system data types or entity definitions, is a key element in supporting interoperability between architectures, since these definitions may be used by other organizations to determine system data compatibility. Often, different organizations may use the same entity name to mean very different kinds of system data with different internal structure. This situation will pose significant interoperability risks, as the system data models may appear to be compatible, each having a Target Track data entity but having different and incompatible interpretations of what Target Track means.

In the NCE, the OV-7 describes the structure of data types (information elements) for information being made available or being consumed by the OV-5 activities and provides the organization and composition of metadata that can be used to characterize the information exchanged in the NCE.

8.1.1.3.3.1 UPDM L1 ::UPDM L0::MODAF::OperationalElements::Behavior

Behavior for Operational View elements for MoDAF specific models.

8.1.1.3.3.1.1 ActivitySubject

MODAF: Anything that is acted upon by an OperationalActivity

DoDAF: NA

Note – ActivitySubject is abstract.

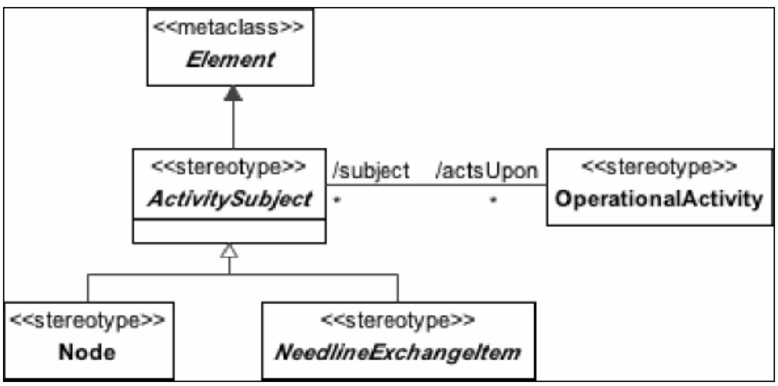


Figure 8.245 - Elements related to the abstract ActivitySubject stereotype

Attribute

The following are attributes for ActivitySubject:

- / actsUpon : OperationalActivity[*] - A list of all the OperationalActivities that act upon the ActivitySubject. This is the inverse of the OperationalActivity “subject” role.

Generalizations

The following are generalization relationships for ActivitySubject:

- UPDMElement

8.1.1.3.3.1.2 StandardOperationalActivity

MODAF: An OperationalActivity that is a standard procedure that is doctrinal.

Note – This is equivalent to what some defence organizations call JETLs.

DoDAF: Work, not specific to a single organization, weapon system, or individual that transforms inputs into outputs or changes their state (DoDAF:: Activity).

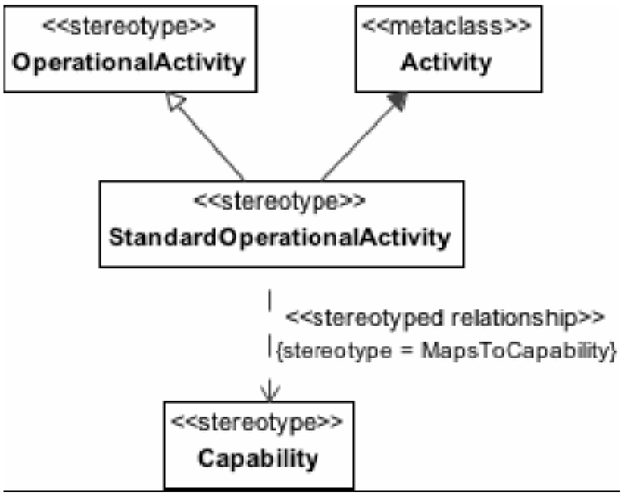


Figure 8.246 - Elements related to the StandardOperationalActivity stereotype

Extensions

The following are extensions for StandardOperationalActivity:

- Activity

Generalizations

The following are generalization relationships for StandardOperationalActivity:

- OperationalActivity

8.1.1.3.3.2 UPDM L1 ::UPDM L0::MODAF::OperationalElements::Data

Data for Operational View elements for MoDAF specific models.

8.1.1.3.3.2.1 LogicalDataModel

MODAF: A LogicalDataModel is a specification of business information requirements as a formal data structure, where relationships and classes (entities) are used to specify the logic that underpins the information.

DoDAF: A Logical Data Model allows analysis of an architecture's data definition aspect, without consideration of implementation specific or product specific issues.

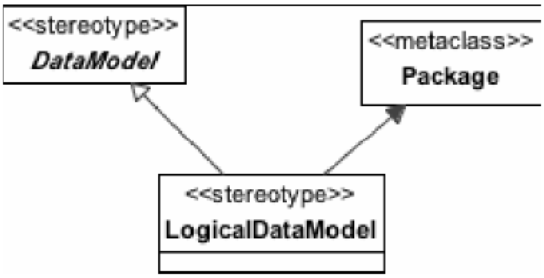


Figure 8.247 - Elements related to the LogicalDataModel stereotype

Extensions

The following are extensions for LogicalDataModel:

- Package

Generalizations

The following are generalization relationships for LogicalDataModel:

- DataModel

8.1.1.3.3.3 UPDM L1 ::UPDM L0::MODAF::OperationalElements::Flows

Flows for Operational View elements for MoDAF specific models.

8.1.1.3.3.3.1 Controls

MODAF: A type of ResourceInteraction where one Resource (source) controls another (target).

DoDAF: NA

Examples - the driver of a tank, one organization having operational control of another, a fire control system controlling a weapons system.

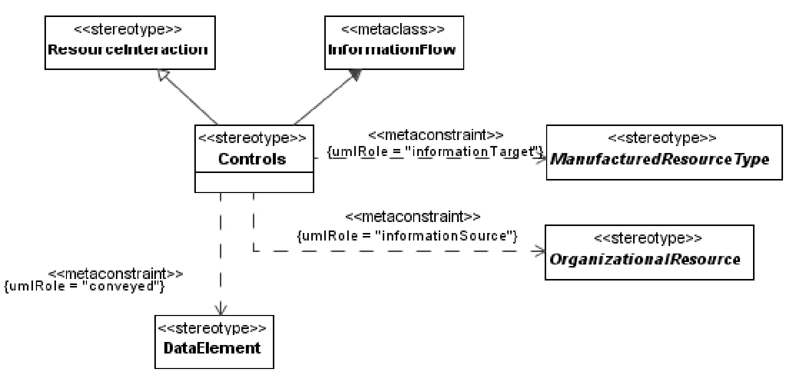


Figure 8.248 - Elements related to the Controls stereotype

Constraints

The following are constraints for Controls:

- Controls.informationSource - Value for the informationSource property must be stereotyped «OrganizationalResource» or its specializations.
- Controls.conveyed - Value for the conveyed property must be stereotyped «ManufacturedResourceType» or its specializations «OrganizationalResource» or its specializations.
- Controls.informationTarget - Value for the informationTarget property must be stereotyped «ManufacturedResourceType» or its specializations.

Extensions

The following are extensions for Controls:

- InformationFlow

Generalizations

The following are generalization relationships for Controls:

- ResourceInteraction

8.1.1.3.3.3.2 EnergyExchange

MODAF: (MODAF::EnergyFlow): A relationship specifying the need to exchange energy between nodes.

DoDAF: NA

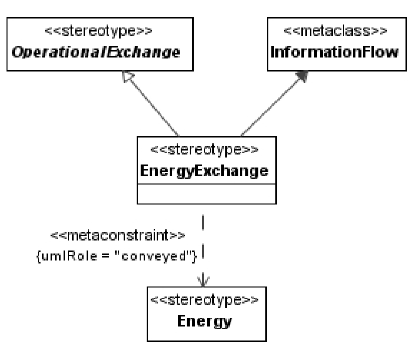


Figure 8.249 - Elements related to the EnergyExchange stereotype

Constraints

The following are constraints for EnergyExchange:

- EnergyExchange.conveyed - Value for the conveyed property must be stereotyped «Energy».

Extensions

The following are extensions for EnergyExchange:

- InformationFlow

Generalizations

The following are generalization relationships for EnergyExchange:

- OperationalExchange

8.1.1.3.3.3.3 MaterielExchange

UPDM: Material that is exchanged between Nodes.

MODAF: A MaterialExchange (MODAF::MaterielFlow) a relationship specifying the need to exchange materiel between nodes.

DoDAF: NA – this is a specialization of OperationalExchange (DoDAF::Interface).

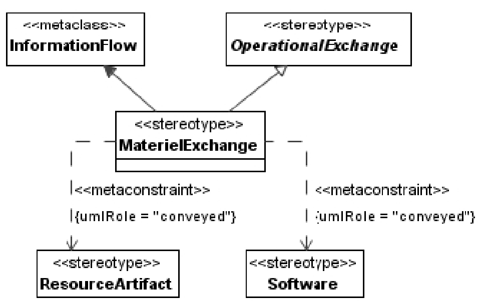


Figure 8.250 - Elements related to the MaterielExchange stereotype

Constraints

The following are constraints for MaterielExchange:

- MaterielExchange.conveyed - Value for the conveyed property must be stereotyped «ResourceArtifact».

Extensions

The following are extensions for MaterielExchange:

- InformationFlow

Generalizations

The following are generalization relationships for MaterielExchange:

- OperationalExchange

8.1.1.3.3.3.4 OrganizationalExchange

UPDM: A relationship specifying flow of people across organizations.

MODAF: An OrganizationExchange (MODAF::MovementOfPeople) relationship specifies the need to move people between nodes.

DoDAF: Interface: An overlap between Performers for the purpose of producing a Resource that is consumed by the other.

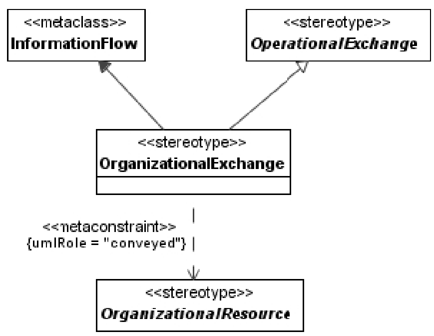


Figure 8.251 - Elements related to the OrganizationalExchange stereotype

Constraints

The following are constraints for OrganizationalExchange:

- OrganizationalExchange.conveyed - Value for the conveyed property must be stereotyped «OrganizationalResource».

Extensions

The following are extensions for OrganizationalExchange:

- InformationFlow

Generalizations

The following are generalization relationships for OrganizationalExchange:

- OperationalExchange

8.1.1.3.3.3.5 ConfigurationExchange

CapabilityConfigurations that are exchanged between Nodes.

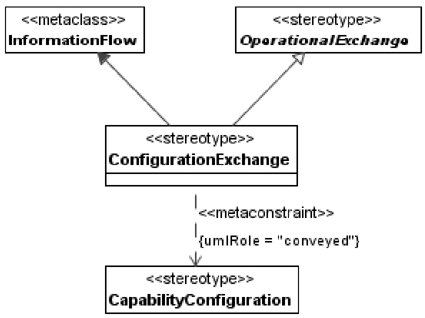


Figure 8.252 - Elements related to the ConfigurationExchange stereotype

Constraints

The following are constraints for ConfigurationExchange:

- ConfigurationExchange.conveyed - Value for the conveyed property must be stereotyped «CapabilityConfiguration».

Extensions

The following are extensions for ConfigurationExchange:

- InformationFlow

Generalizations

The following are generalization relationships for ConfigurationExchange:

- OperationalExchange

8.1.1.3.3.4 UPDM L1: :UPDM L0::MODAF: :OperationalElements: :Structure

Structure for Operational View elements for MoDAF specific models.

8.1.1.3.3.4.1 Energy

UPDM: Energy to be exchanged between Nodes.

MODAF: NA

DoDAF: NA

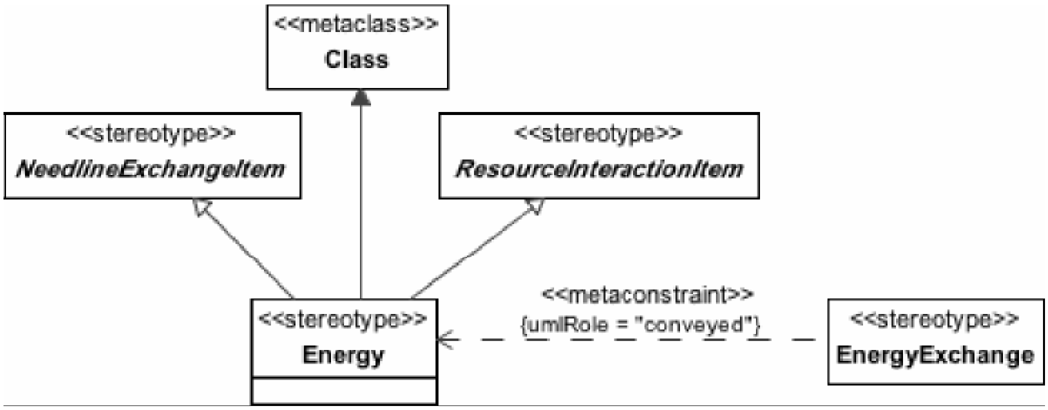


Figure 8.253 - Elements related to the Energy stereotype

Extensions

The following are extensions for Energy:

- Class

Generalizations

The following are generalization relationships for Energy:

- ResourceInteractionItem
- OperationalExchangeItem

8.1.1.3.3.4.2 ProblemDomain

MODAF: The boundary containing those Nodes, which may be realized by functional resources specified in SV-1. There may be more than one alternative solution for a given ProblemDomain specified as a set of SV suites. There may be only one ProblemDomain in a LogicalArchitecture.

DoDAF: NA – covered by the more general temporalWholePart element.

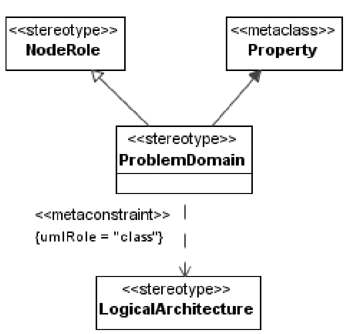


Figure 8.254 - Elements related to the ProblemDomain stereotype

Constraints

The following are constraints for ProblemDomain:

- ProblemDomain.class - Value for the class property must be stereotyped «LogicalArchitecture» or its specializations.

Extensions

The following are extensions for ProblemDomain:

- Property

Generalizations

The following are generalization relationships for ProblemDomain:

- NodeRole

8.1.1.3.3.5 UPDM L1: :UPDM L0::MODAF: :OperationalElements: :Alias

Structure for Operational View elements for MoDAF specific models.

8.1.1.3.3.5.1 MovementOfPeople

UPDM: MODAF alias for OrganizationalExchange.

MODAF: A relationship specifying the need to move people between nodes.

DoDAF: NA – a MODAF alias for a specialization of OperationalExchange (DoDAF::Interface).

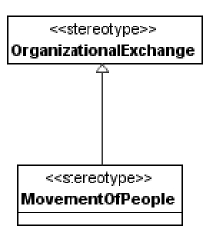


Figure 8.255 - Elements related to the MovementOfPeople stereotype

Generalizations

The following are generalization relationships for MovementOfPeople:

- OrganizationalExchange

8.1.1.3.4 UPDM L1 ::UPDM L0::MODAF::StrategicElements

The Strategic View elements for MoDAF specific models.

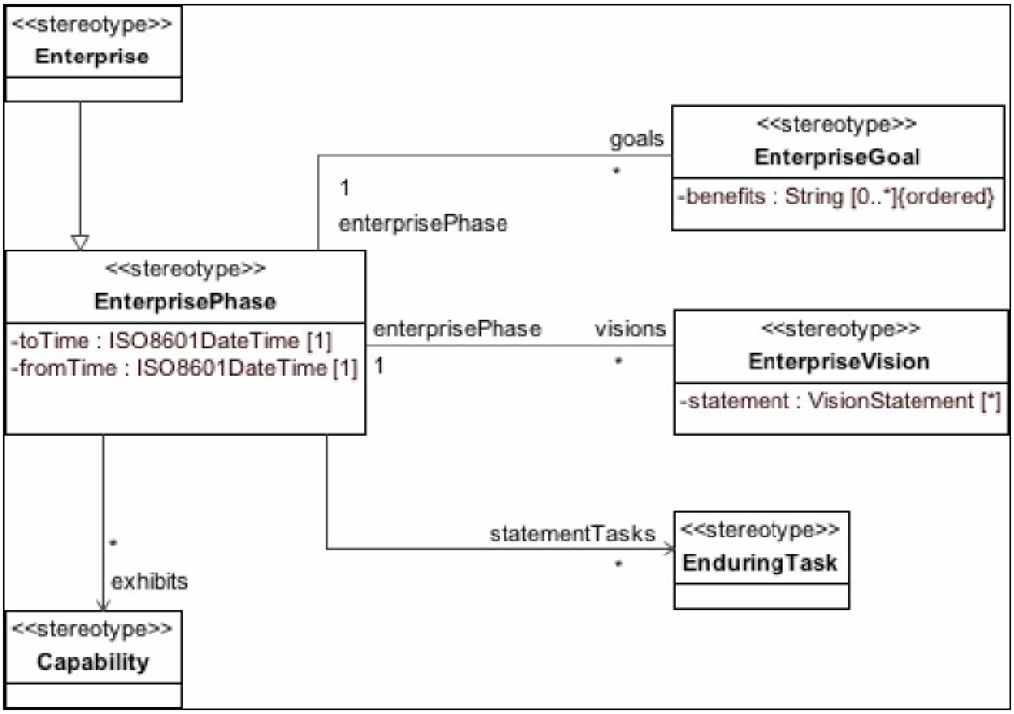


Figure 8.256 - Elements related to the StV-1 stereotype

MODAF: StV-1 addresses the enterprise concerns associated with the overall vision for transformational endeavors and thus defines the strategic context for a group of Enterprise capabilities.

DoDAF: CV-1: Vision: addresses the enterprise concerns associated with the overall vision for transformational endeavors and thus defines the strategic context for a group of capabilities.

StV-1 addresses the enterprise concerns associated with the overall vision for transformational endeavors and thus defines the strategic context for a group of Enterprise capabilities. 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. The Views are high-level and describe capabilities using terminology which is easily understood by non-technical readers (though they may make extensive use of military terminology and acronyms that are clearly defined in the AV-2 View).

Figure 8.257 - Elements related to the StV-2 stereotype

MODAF: The StV-2 Product models capability taxonomies.

DoDAF: The CV-2 DoDAF-described View models capability taxonomies.

The view presents a hierarchy of capabilities. These capabilities may be presented in context of an Enterprise Phase (i.e., it can show the required capabilities for current and future enterprises). StV-2 specifies all the capabilities that are reference throughout one or more architectures. In addition it can be used as a source document for the development of high-level use cases and Key User Requirements (KUR). The StV-2 also provides metrics against each capability that may be used to measure successfully fielded capability.

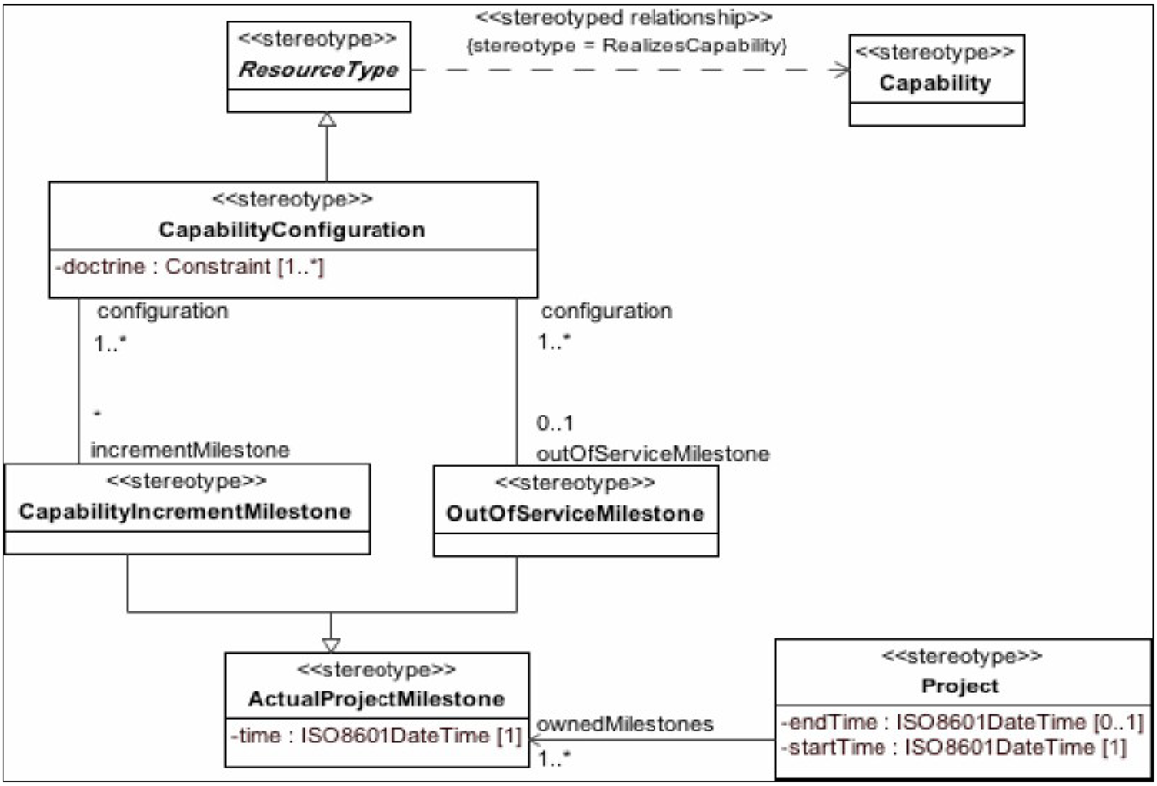


Figure 8.258 - Elements related to the StV-3 stereotype

MODAF: StV-3 addresses the planned achievement of capability at different points in time or during specific periods of time, i.e., capability phasing.

DoDAF: CV-3: Capability Phasing The CV-3 addresses the planned achievement of capability at different points in time or during specific periods of time, i.e., capability phasing.

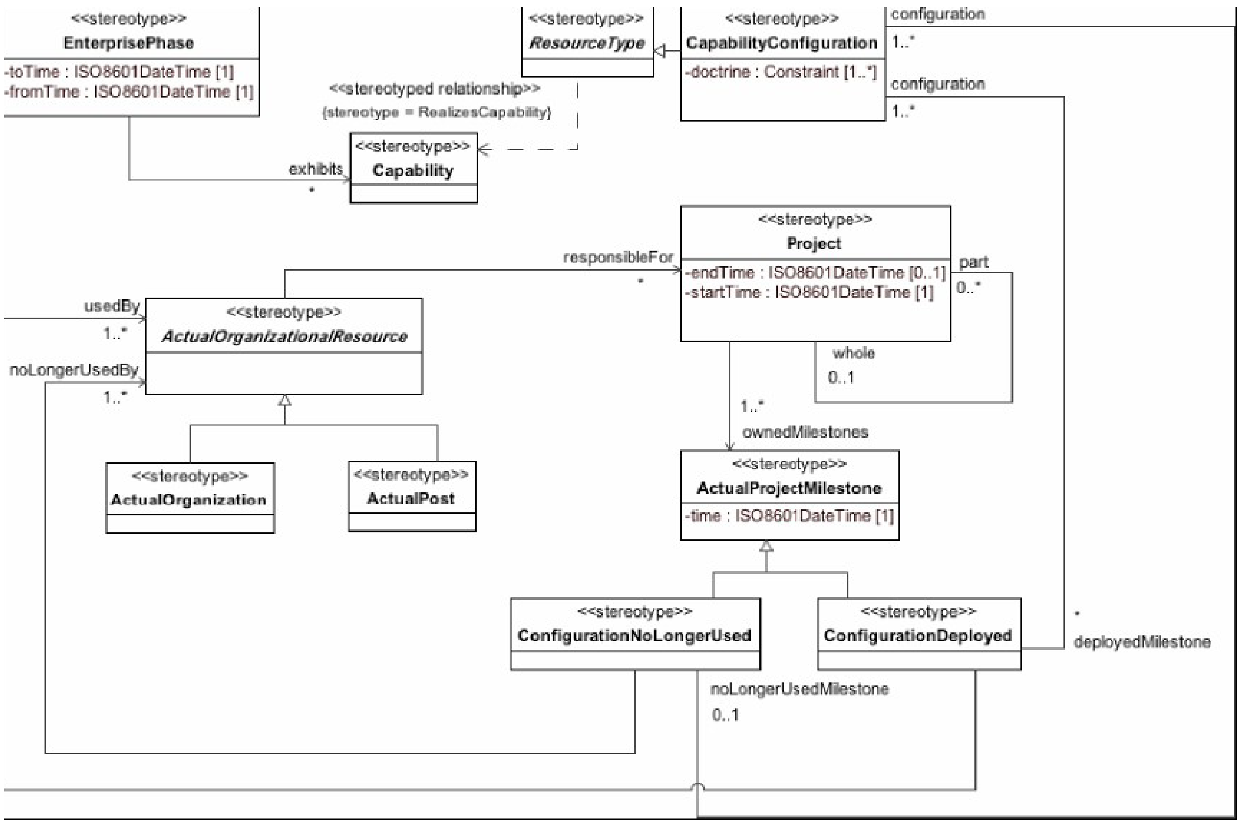


Figure 8.259 - Elements related to the StV-5 stereotype

MODAF: StV-5 addresses the fulfilment of capability requirements, in particular by network enabled capabilities.

DoDAF: CV-5: Capability to Organizational Development Mapping: The CV-5 addresses the fulfillment of capability requirements.

The StV-5 defines Capability to Organization Deployment Mapping. It addresses the fulfillment of capability requirements, in particular by network enabled capabilities. This view shows the planned capability deployment and interconnection for a particular EnterprisePhase. This view will provide a more detailed dependency analysis than is possible using StV-3. The StV-5 View is used to support the capability management process and, in particular, assist the planning of fielding.

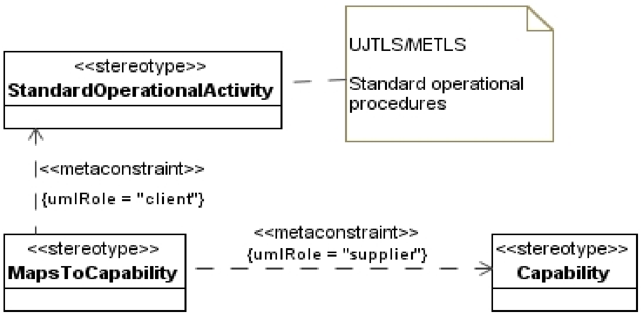


Figure 8.260 - Elements related to the StV-6 stereotype

MODAF: The StV-6 Product describes the mapping between the capabilities required by an Enterprise and the operational activities that those capabilities support.

DoDAF: CV-6: Capability to Operational Activities Mapping: The CV-6 DoDAF-described View describes the mapping between the capabilities required and the operational activities that those capabilities support.

8.1.1.3.4.1 UPDM L1 ::UPDM L0::MODAF::StrategicElements::Milestones

Milestone elements for Strategic View elements for MoDAF specific models.

8.1.1.3.4.1.1 DeployedMilestone

MODAF: Asserts that an ActualOrganizationResource started to use, or is slated to start using a CapabilityConfiguration from a specific point in time. This is used to describe capabilities going into service with specific organizations or posts.

DoDAF: NA

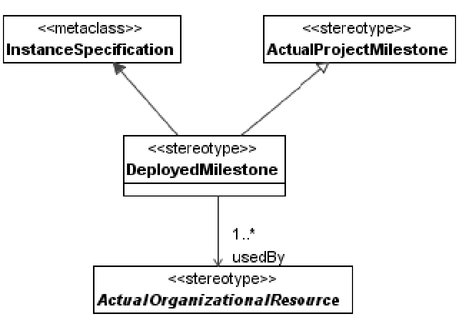


Figure 8.261 - Elements related to the ConfigurationDeployed stereotype

Attribute

The following are attributes for DeployedMilestone:

- usedBy : ActualOrganizationalResource[1 ..*] - ActualOrganizationalResources using CapabilityConfiguration deployed at this Milestone.

Extensions

The following are extensions for DeployedMilestone:

- InstanceSpecification

Generalizations

The following are generalization relationships for DeployedMilestone:

- ActualProjectMilestone

8.1.1.3.4.1.2 NoLongerUsedMilestone

MODAF:Asserts that an «ActualOrganisationResource» ceased to use or is slated to cease using a «CapabilityConfiguration» from a specific point in time.

This is used to describe capabilities going out of service with specific organizations or posts.

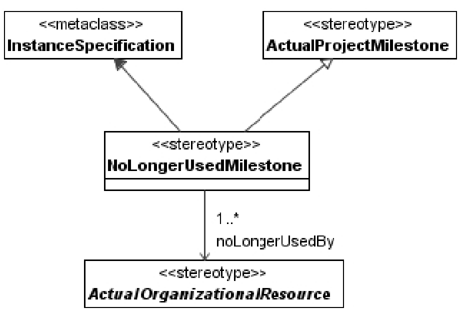


Figure 8.262 - Elements related to the NoLongerUsedMilestone stereotype

Attributes

The following are attributes for NoLongerUsedMilestone:

- noLongerUsedBy : ActualOrganizationalResource[1 ..*] - ActualOrganizationalResources that are no longer using.

Extensions

The following are extensions for NoLongerUsedMilestone:

- InstanceSpecification

Generalizations

The following are generalization relationships for NoLongerUsedMilestone:

- ActualProjectMilestone

8.1.1.3.4.2 UPDM L1: :UPDM L0::MODAF: :StrategicElements: :Structure

Structure elements for Strategic View elements for MoDAF specific models.

8.1.1.3.4.2.1 EnduringTask

MODAF: A type of behavior recognized by an enterprise as being essential to achieving its goals (i.e., a strategic specification of what the enterprise does).

DoDAF: NA

Note – This is equivalent to a task in an essential task list (JETL).

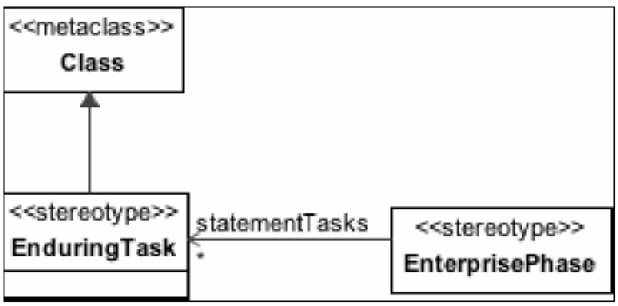


Figure 8.263 - Elements related to the EnduringTask stereotype

Extensions

The following are extensions for EnduringTask:

- Class

Generalizations

The following are generalization relationships for EnduringTask:

- UPDMElement

8.1.1.3.5 UPDM L1 ::UPDM L0::MODAF::SystemElements

The System View elements for MoDAF specific models.

8.1.1.3.5.1 UPDM L1 ::UPDM L0::MODAF::SystemElements::Structure

Structure for Systems View elements for MoDAF specific models.

Part IV - Annexes

This part contains the following annexes:

A - Domain Metamodel

B - UPDM Elements Traceability to DoDAF/MODAF Elements

C - Sample Problem

D - Bibliography

Annex A - Domain Metamodel (DMM)

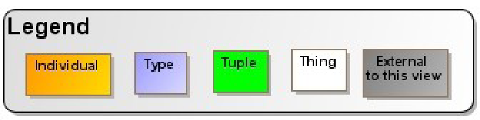
(non-normative)

This Annex comprises various diagrams which document the Domain Metamodel (DMM) that document the MoDAF 1.5 and MoDAF 1.2 integrated model. This model was used as a basis for creating the UPDM profile.

Note that the diagrams rely on color to aid the reader in understanding the model. Please refer to the legend in the various diagrams to understand the specific definitions.

Legend

In DMM the following color codes are used:



A.1 AcV

The AcquisitionElements describe project details, including dependencies between projects and capability integration. These Views guide the acquisition and fielding processes.

A.1.1 AcV-1

MODAF: AcV-1 view products represent an organizational perspective on projects.

DoDAF: AcV-1 view [DoDAF::Project Portfolio Relationships (PV-1) DoDAF-described View] represents an organizational perspective on programs, projects, or a portfolio of projects.

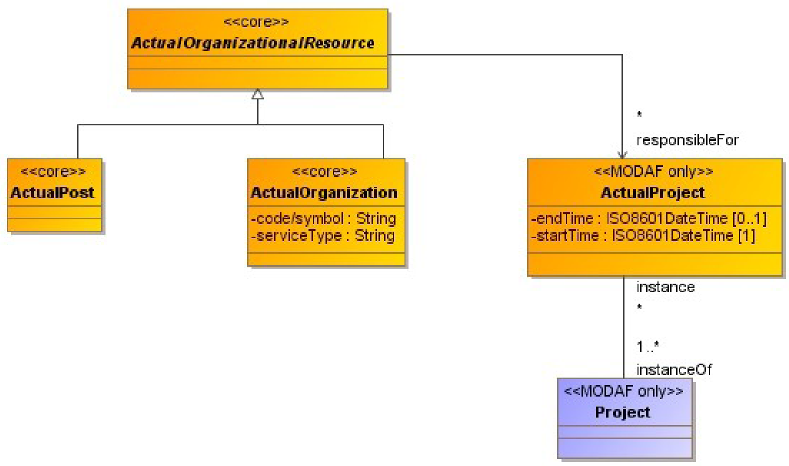


Figure A.1 - AcV-1

A.1.2 AcV-2

MODAF: AcV-2 view products provide a timeline perspective on projects.

DoDAF: AcV-2 (DoDAF::PV-2: Project Timelines DoDAF-described View) provides a timeline perspective on programs or projects.

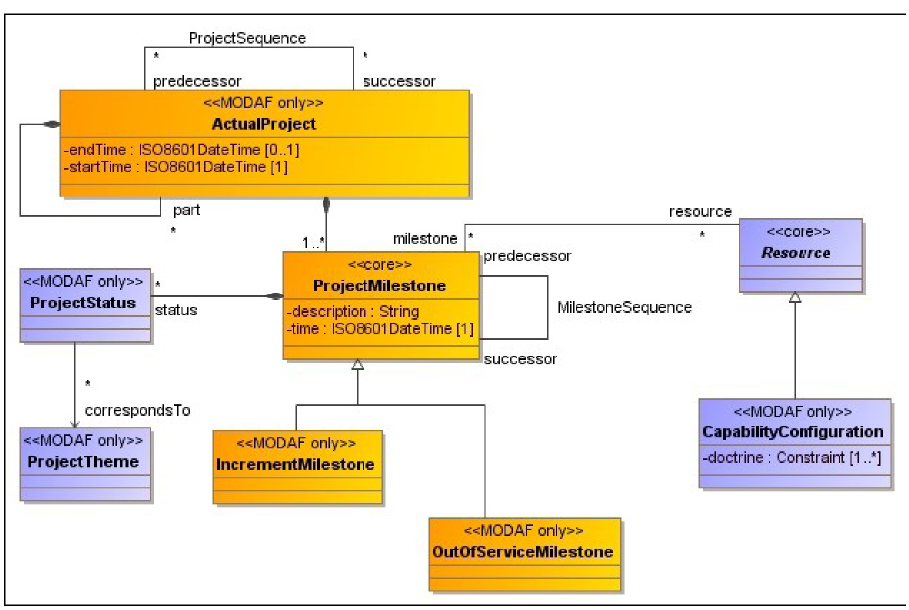


Figure A.2 - AcV-2

A.2 AV

Elements that are part of the All View. The All-Views (AVs) provide 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. They also provide a place to record any findings arising from the architecting 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. Since the AVs provide critical information for the future access and exploitation of an architectural model their population is essential whenever an architecture is created or modified. The AVs provide a critical input into the processes that provide architectural governance.

A.2.1 AV-1

MODAF: The overview and summary information contained within the AV-1 product provides executive-level summary information in a consistent form that allows quick reference and comparison between architectural descriptions. AV-1 includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.

DoDAF: The overview and summary information contained within the AV-1 DoDAF-described View provides executive level summary information in a consistent form that allows quick reference and comparison between architectural descriptions.

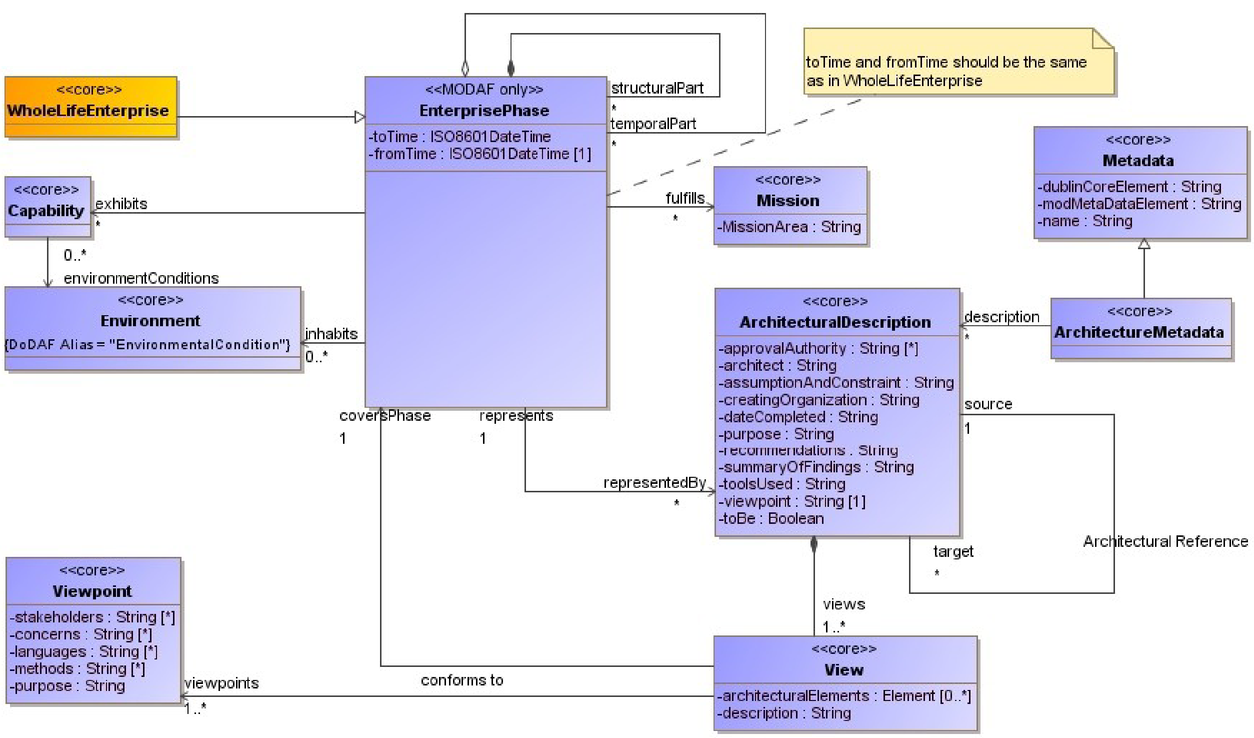


Figure A.3 - AV-1

A.2.2 AV-2

MODAF: AV-2 presents all the Elements used in an architecture as a stand alone structure. An AV-2 presents all the Elements as a specialization hierarchy, provides a text definition for each one, and references the source of the element (e.g., MODAF Ontology, IDEAS Model, local). 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.

DoDAF: The AV-2 presents all the metadata used in an architecture as a stand alone structure. An AV-2 presents all the metadata as a specialization hierarchy, provides a text definition for each one, and references the source of the element (e.g., DoDAF Meta-model, IDEAS, a published document or policy). An AV-2 shows elements from the DoDAF Meta-model that have been used in the architecture and new elements (i.e., not in the DoDAF Meta-model) that have been introduced by the architecture.

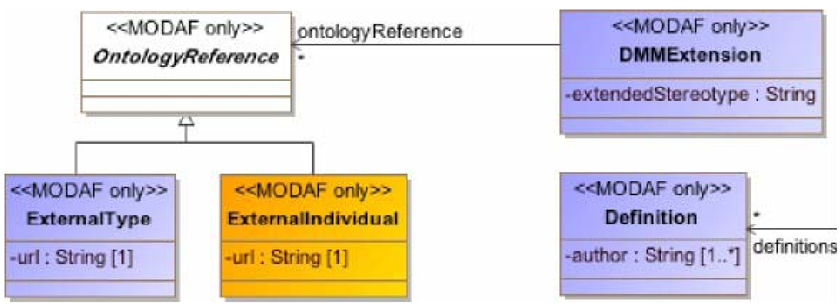


Figure A.4 - AV-2

A.2.3 AV-3Measurements

Shows the measureable properties of something in the physical world, expressed in amounts of a unit of measure that can be associated with a UPDMElement.

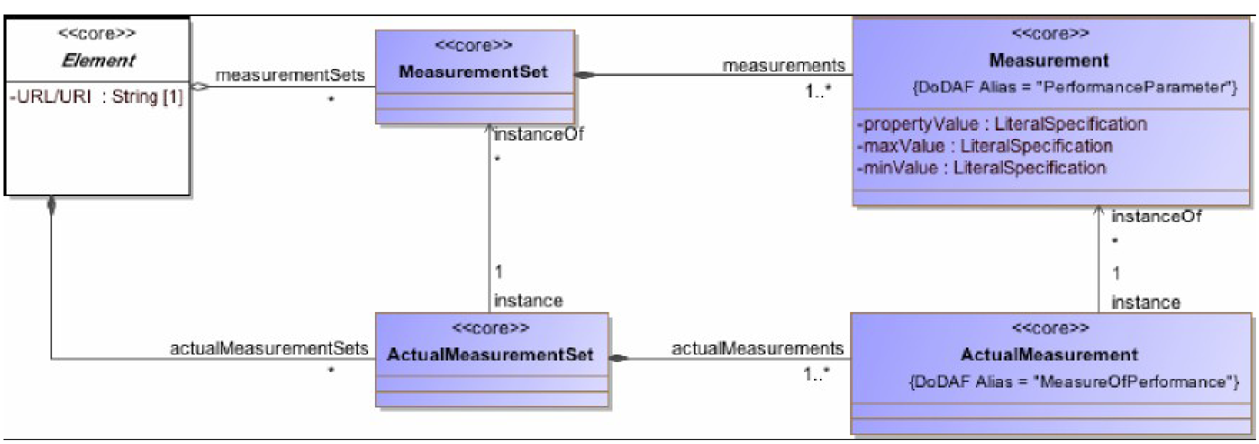


Figure A.5 - AV-3 Measurements

A.3 Layers

The layers part of the DMM show a view of the model based on each of the views. For instance, the OV layer documents the interrelated data from OV-1 through OV-7 views.

A.3.1 AcV

The Acquisition View (AcV) shows the elements that are part of AcV-1 and AcV-2.

The AcquisitionElements describe project details, including dependencies between projects and capability integration. These Views guide the acquisition and fielding processes.

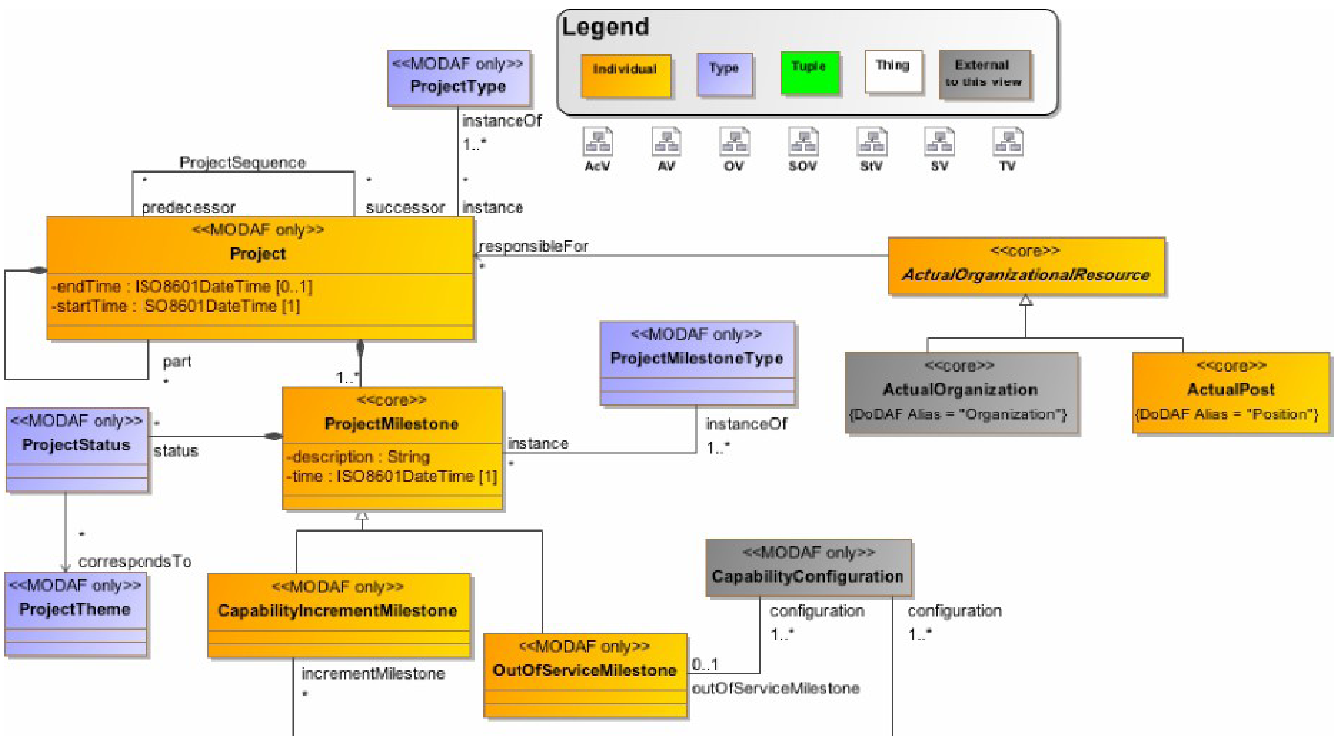


Figure A.6 - AcV

A.3.2 AV

The All View (AV) shows the elements that are part of AV- 1 through AV-2, and Measurements.

Elements that are part of the All View. The All-Views (AVs) provide 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. They also provide a place to record any findings arising from the architecting 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. Since the AVs provide critical information for the future access and exploitation of an architectural model their population is essential whenever an architecture is created or modified. The AVs provide a critical input into the processes that provide architectural governance.

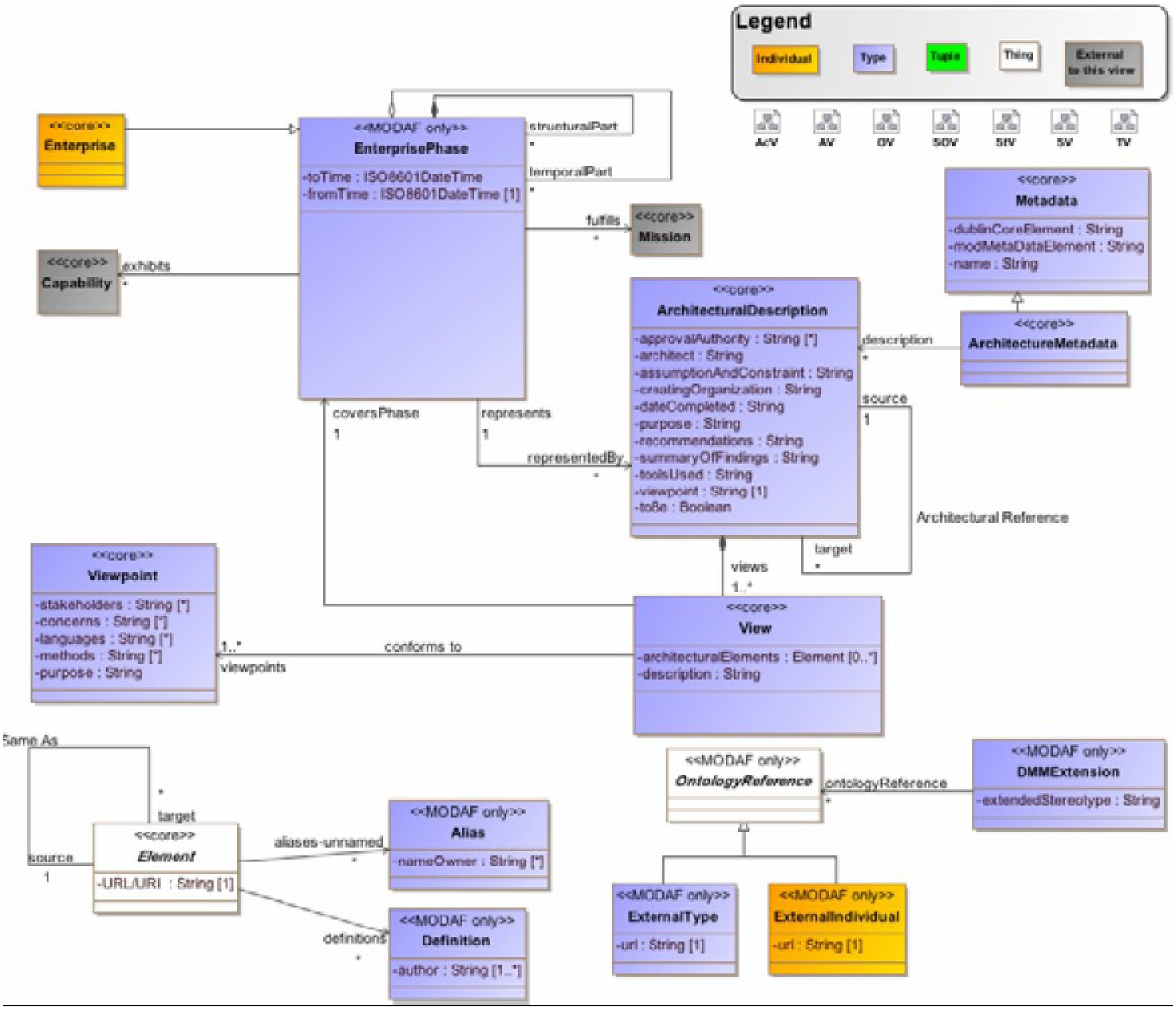


Figure A.7 - AV

A.3.3 OV

The Operational View (OV) shows the elements that are part of OV-1 through OV-7.

The Operational View is about real-world activities, the people and machinery that perform them, and the means by which they are performed. The Operational View is divided into nine products intended to answer the “who,” “what,” “when,” “where,” “why,” and “how” of a mission. They are summarized below.

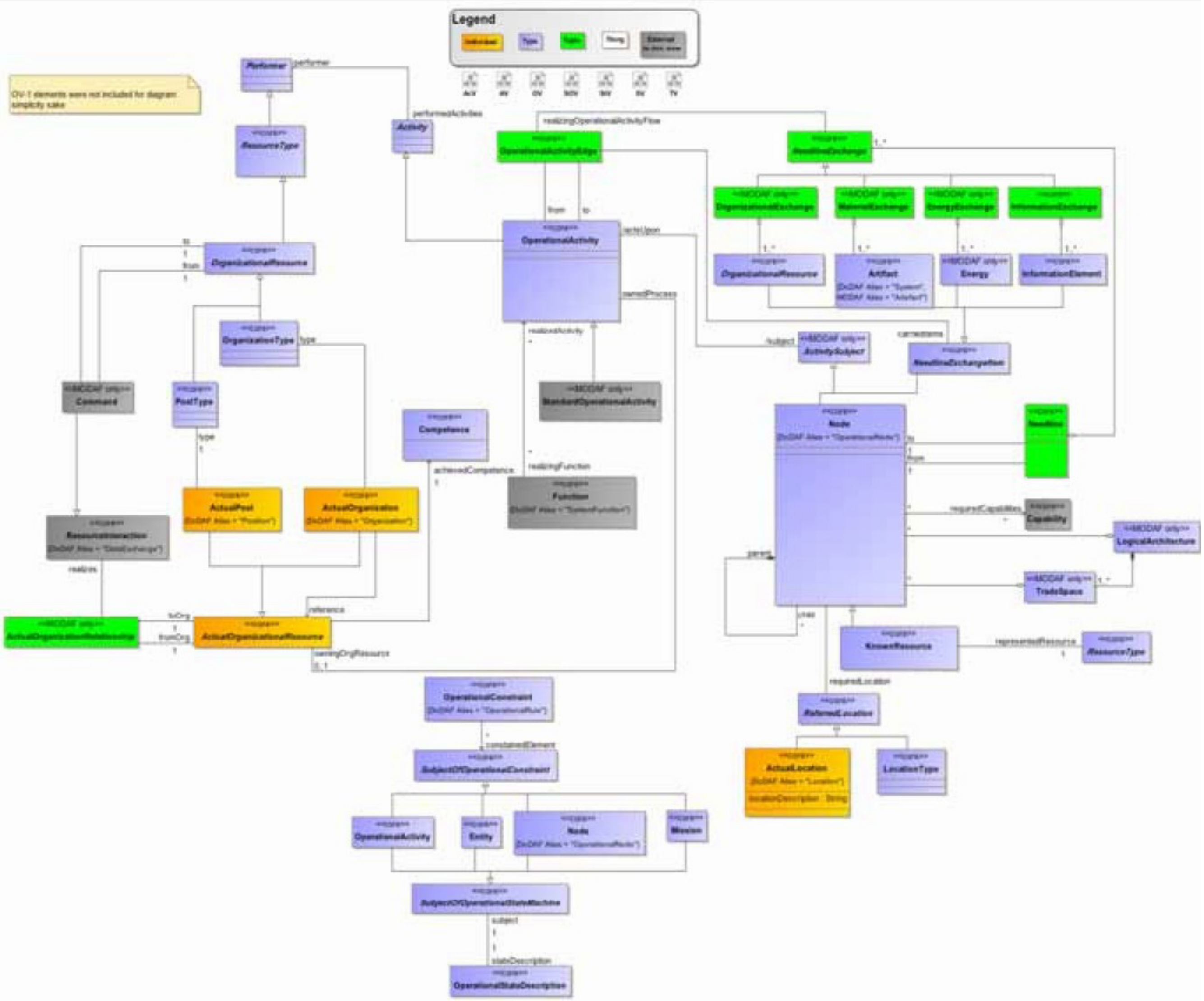


Figure A.8 - OV

A.3.4 SOV

The Services View (SOV) shows the elements that are part of SOV-1 through SOV-5.

The Service-Orientated View is a description of services needed to directly support the operational domain as described in the Operational View. A service is described as a unit of work through which a particular Resource provides a useful result to a consuming Resource.

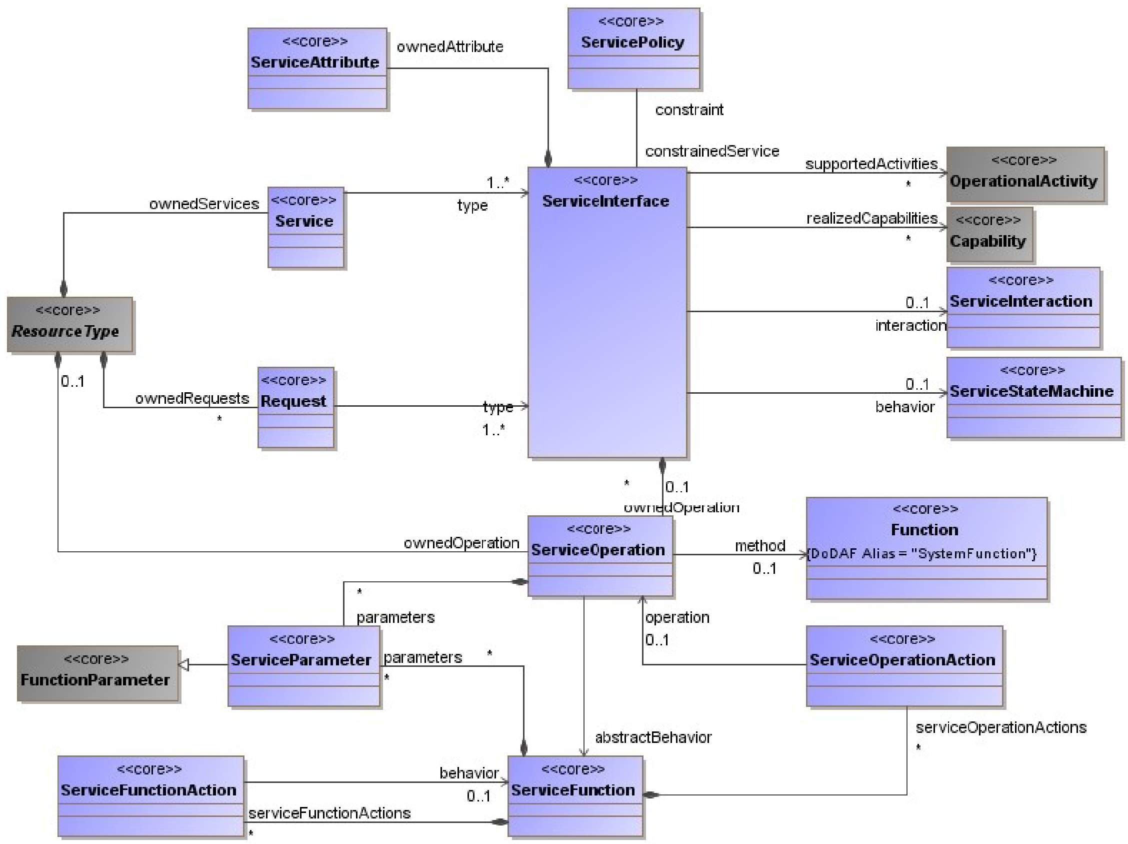


Figure A.9 - SOV

A.3.5 StV

The Strategic View (StV) shows the elements that are part of StV-1 through StV-6.

The Strategic Elements are used in the Strategic View that provides an overall Enterprise Architecture assessment of the Capabilities and their relationships facilitating Capability Management (e.g., capability introduction, integration, realignment, and removal). While an Enterprise will have a number of UPDM Architecture Descriptions that have the Operational, System, Technical Standards, and All Views, only one Strategic View will exist across a number of Architecture Descriptions.

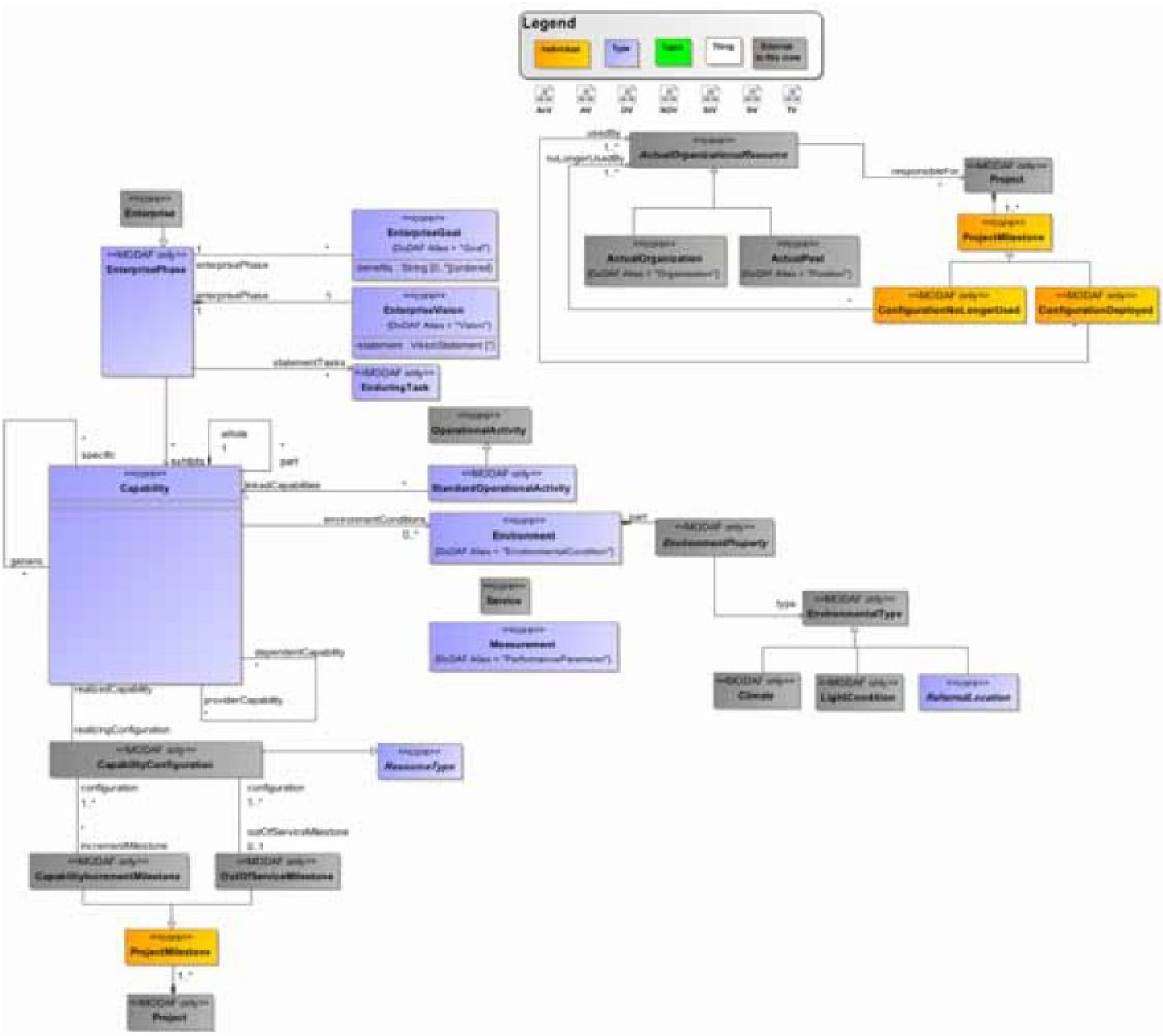


Figure A.10 - StV

A.3.6 SV

The System View (SV) shows the elements that are part of SV-1 through SV-12.

Models in the System Viewpoint represent alternate realizations in terms of equipment capability of the operational capabilities expressed through models in the Operational Viewpoint and in the User Requirements. The System Viewpoint primarily addresses the specification of the system capability needed (rather than implementation details). Significant changes originally made in MODAF improved the ability for modelers to represent configuration of capability that include people as well as systems and platforms.

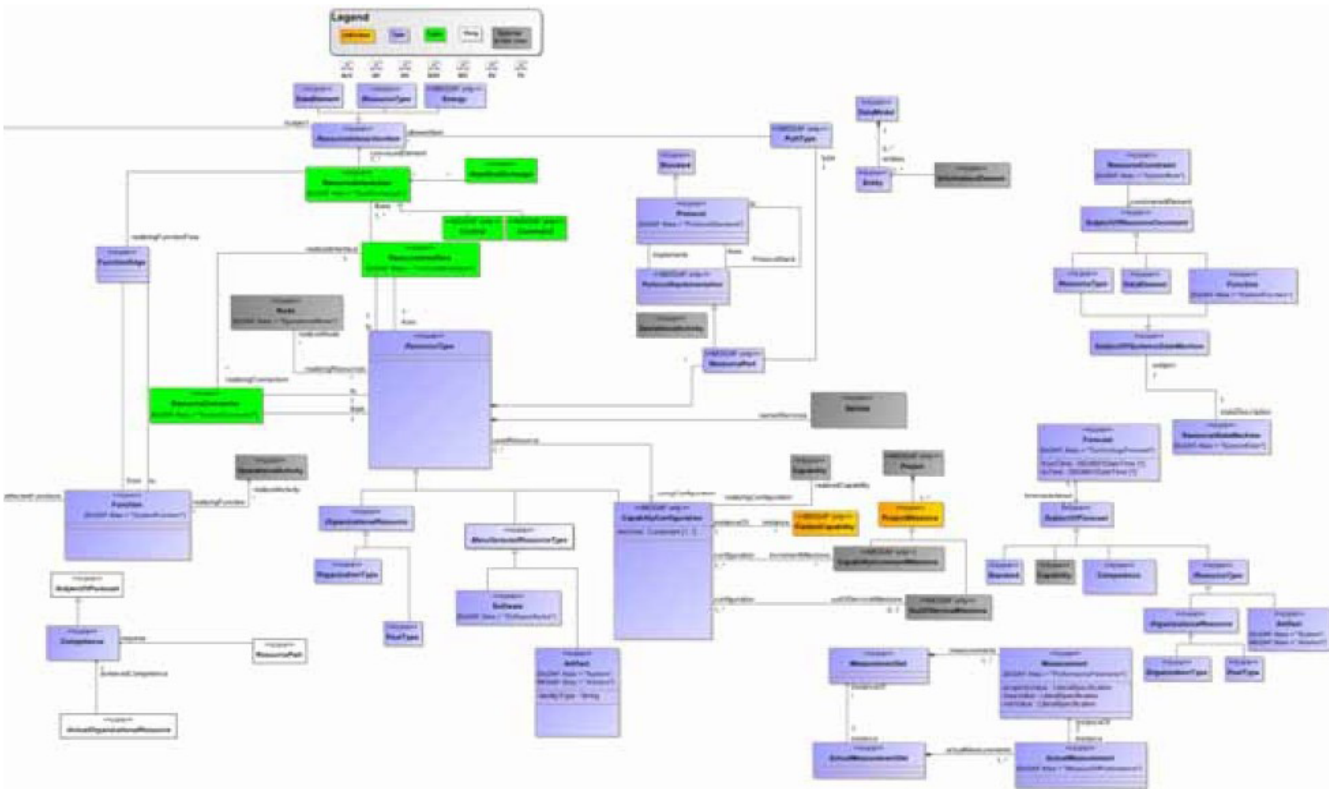


Figure A.11 - SV

A.3.7 TV

The Technical View (TV) shows the elements that are part of TV-1 through TV-3.

The Technical View is a set of products delineating standards, rules, notations, and conventions that apply to the implementation of the system architecture. When the standards profile is tied to the system elements to which they apply, TV-1 serves as the bridge between the SV and TV. SV-9 forecasts relate to the TV-1 in that a timed technology forecast may contribute to the decision to retire or phase out the use of a certain standard in connection with a system element. Similarly, SV-9 forecasts relate to TV-2 standards forecasts in that a certain standard may be adopted depending on a certain technology becoming available (e.g., the availability of Java Script may influence the decision to adopt a new HTML standard).

MODAF extends the core DoDAF Technical Standards Views to include non-technical standards and policies applicable to the architecture such as operational doctrine, industry process standards, etc. Additionally, the TV-1 may also document policies and standards applicable to the operational or business context. MoDAF also distinguishes between ‘applicability’ and ‘conformance’ with regard to architectural elements. If a standard is applicable to a given architecture, that architecture need not be fully conformant with the standard. The degree of conformance to a given standard may be judged on a risk basis at an approval point. An association between a Standard and an architectural element is not to be interpreted as stating the level of compliance of the element is fully compliant with that Standard. Additional evidences would need to be given (outside MODAF) to confirm the level of compliance. Finally, MoDAF adds the explicit requirement that any Standards cited in TV-1 View must, where appropriate, be in accordance with the trend towards open architectures (i.e., standards that encourage stove-piped systems are expressly prohibited).

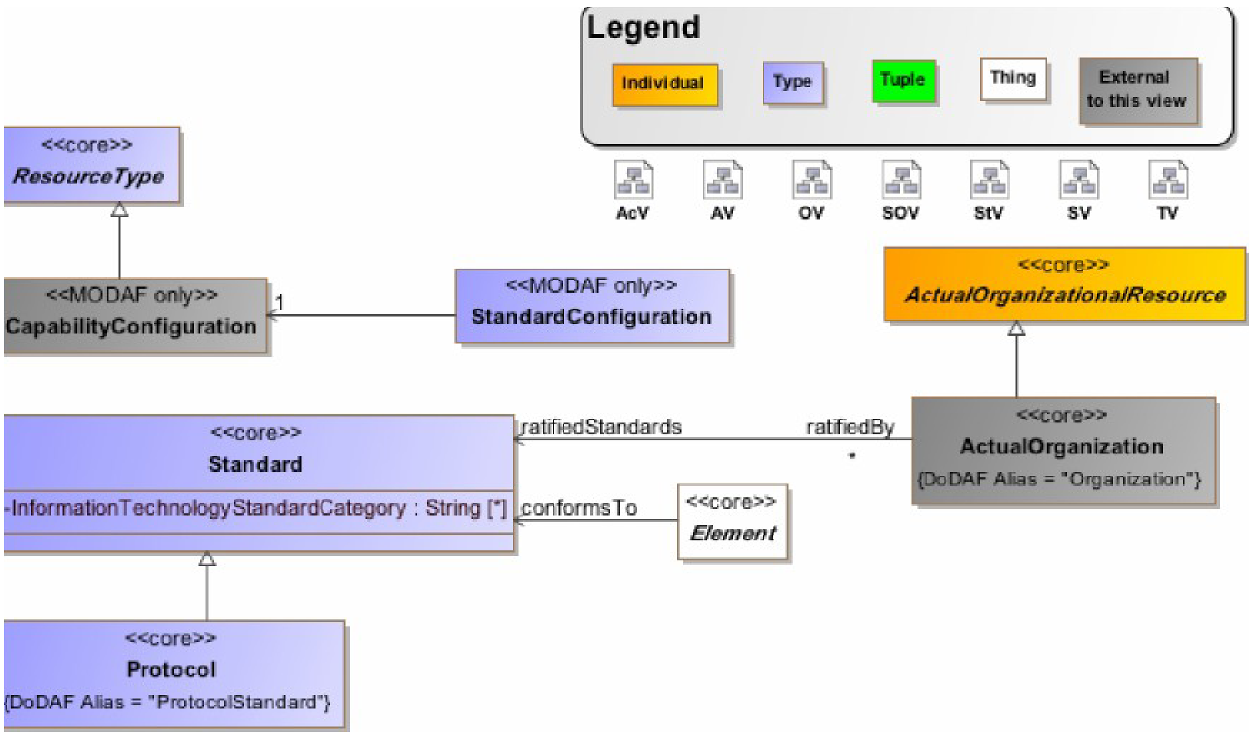


Figure A.12 - TV

A.4 OV

The Operational View is about real-world activities, the people and machinery that perform them, and the means by which they are performed. The Operational View is divided into nine products intended to answer the “who,” “what,” “when,” “where,” “why,” and “how” of a mission. They are summarized below.

A.4.1 OV-1

MODAF: OV-1 addresses the high level operational concepts related to one or more missions. An OV-1 describes a mission, class of mission, or scenario; and highlights the main operational elements and interesting or unique aspects of operations.

The OV-1 has two purposes. First, it provides a means of organizing the operational architecture models into distinct groups based on scenario context. Second, it communicates the essence of the scenario context in an essentially graphical form.

DoDAF: The OV-1 DoDAF-described View describes a mission, class of mission, or scenario. It shows the main operational concepts and interesting or unique aspects of operations. It describes the interactions between the subject architecture and its environment, and between the architecture and external systems. A textual description accompanying the graphic is crucial. Graphics alone are not sufficient for capturing the necessary architecture data.

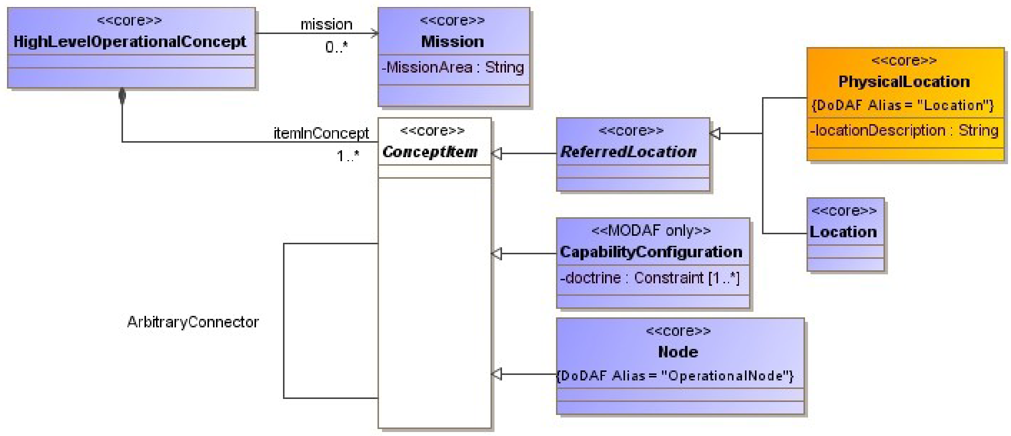


Figure A.13 - OV-1

A.4.2 OV-2

MODAF: The Operational Node Relationships Description (OV-2) addresses localization of operational capability.

DoDAF: The Operational Resource Description (OV-2) DoDAF-described View applies the context of the operational capability to a community of anticipated users.

The Operational Node Connectivity Description is intended to track the need to exchange information from specific operational nodes (that play a key role in the architecture) to others. OV-2 does not depict the connectivity between the nodes. MoDAF modifies the OV-2 is in two ways. First of all, it recommends that an OV-2 diagram (now OV-2a) shows the platforms or geographic locations at which operational nodes are deployed. Secondly it provides additional information (OV-2b) about each needline in the form of a requirements specification. There are now four types of needlines identified as follows:

1. InformationExchange
2. EnergyFlow
3. MaterielFlow
4. MovementOfPeople

In addition, MoDAF permits service-oriented architectures. Instead of needlines between nodes, it is possible simply to show which services the nodes provide and consume. Finally, MoDAF again permits known resources to be shown in an OV-2. However, this must be clearly shown as a KnownResource in an OV-2 model. The concept of a LogicalArchitecture is introduced - this is the container class for all the nodes and KnownResources.

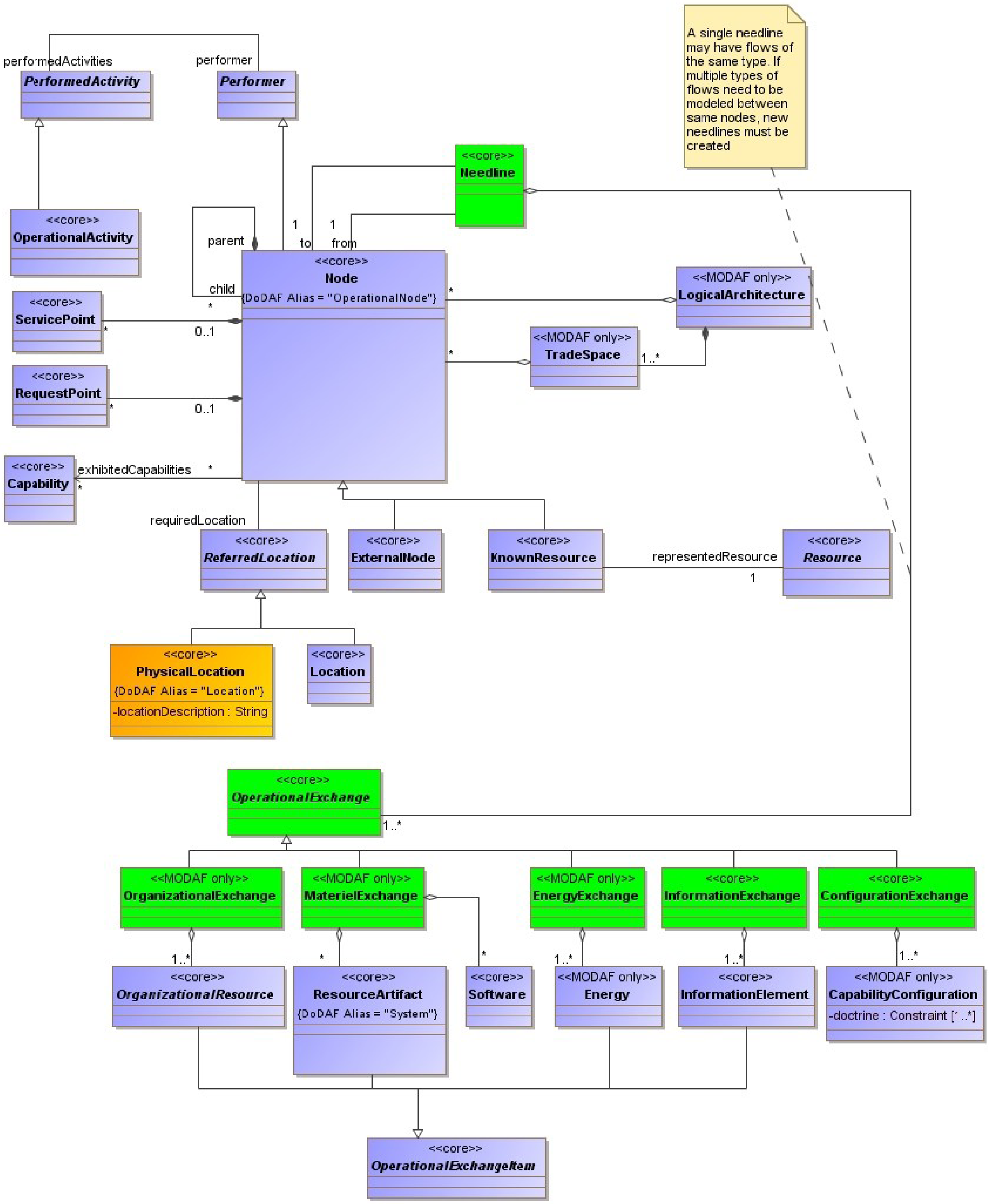


Figure A.14 - OV-2

A.4.3 OV-3

Information exchanges express the relationship across the three basic architecture data elements of an OV (operational activities, operational nodes, and information flow) with a focus on the specific aspects of the information flow and the information content.

The Information Exchanges of the OV-3 should remain at a high level of aggregation to represent actual information workflow products that are used at the operational nodes shown in OV-2 (and not their subordinate operational nodes).

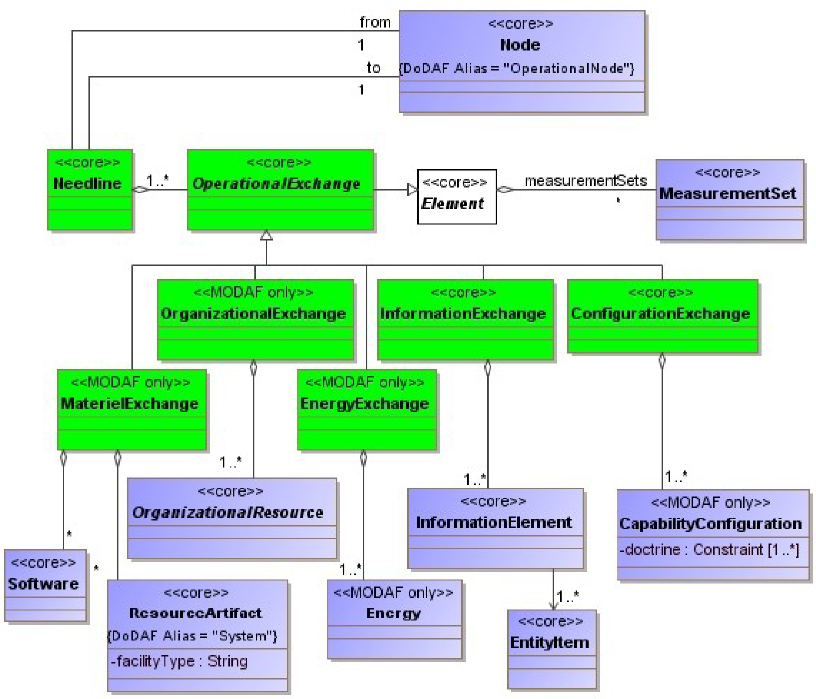


Figure A.15 - OV-3

A.4.4 OV-4 Actual

This is the OV-4 Actual View. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. MoDAF divides the OV-4 two views; an OV-4 Typical and an OV-4 Actual. The former is exactly as the DoDAF OV-4, while the latter is a special form of the SV-1; where the resources are restricted to being organizational.

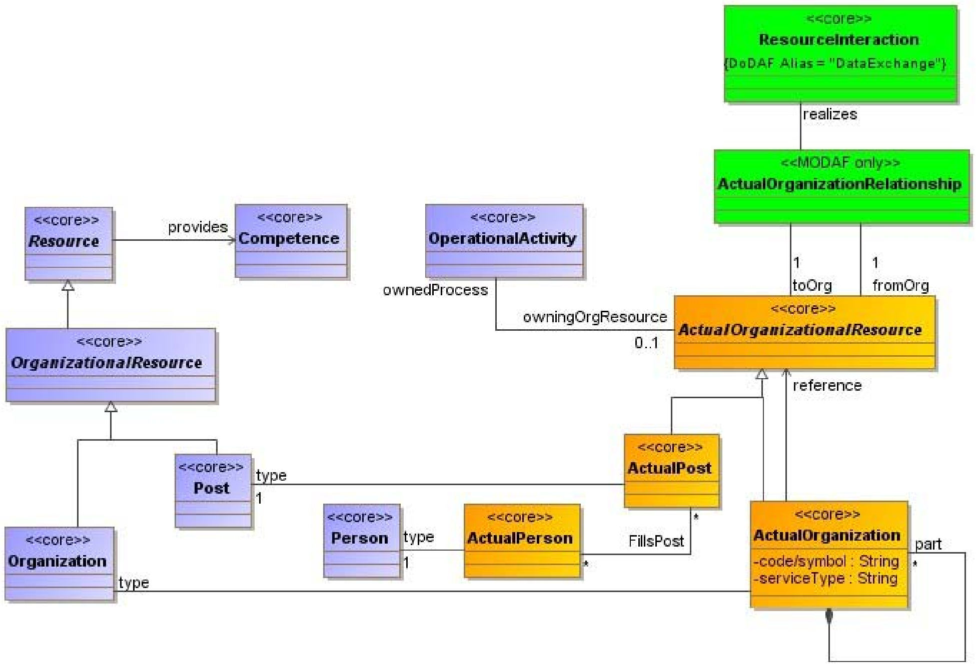


Figure A.16 - OV-4 Actual

A.4.5 OV-4 Typical

MODAF: The OV-4 shows organizational structures and interactions. The organizations shown may be civil or military. A typical OV-4 shows the possible relationships between organizational resources (organizations and posts).

DoDAF: DoDAF: The OV-4 DoDAF-described View shows organizational structures and interactions. The organizations shown may be civil or military. A typical OV-4 shows the possible relationships between organizational resources.

This is the OV-4 Typical View. The Organizational Relationships Chart illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in architecture. MoDAF divides the OV-4 two views; an OV-4 Typical and an OV-4 Actual. The former is exactly as the DoDAF OV-4, while the latter is a special form of the SV-1; where the resources are restricted to being organizational.

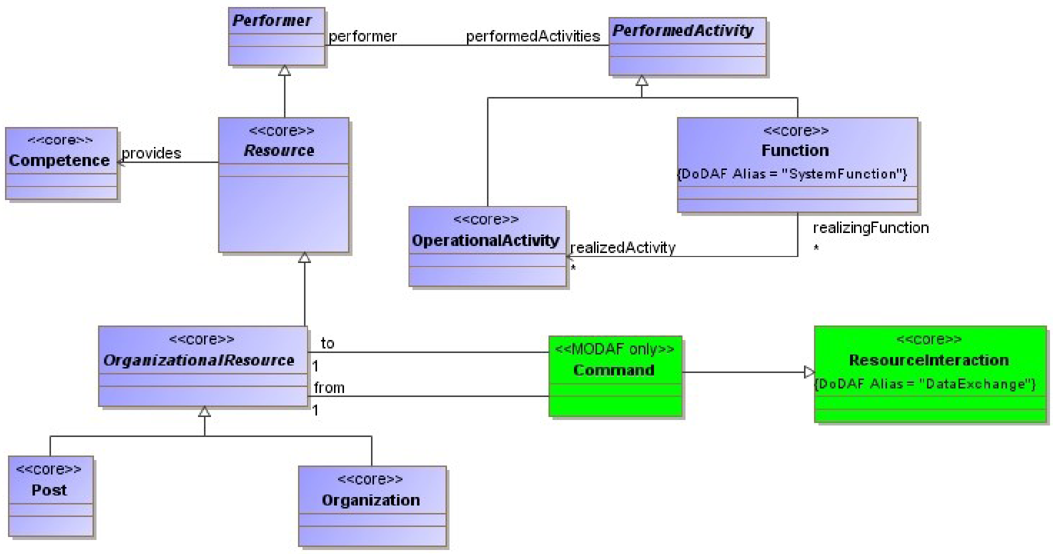


Figure A.17 - OV-4 Typical

A.4.6 OV-5

MODAF: The Operational Activity Model (OV-5) describes the operations that are normally conducted in the course of achieving a mission or a business goal. It describes operational activities (or tasks), Input/Output flows between activities, and to/from activities that are outside the scope of the Architecture.

DoDAF: The Operational Activity Model DoDAF-described View describes the operations that are normally conducted in the course of achieving a mission or a business goal. It describes operational activities (or tasks); Input/Output flows between activities, and to/from activities that are outside the scope of the Architecture.

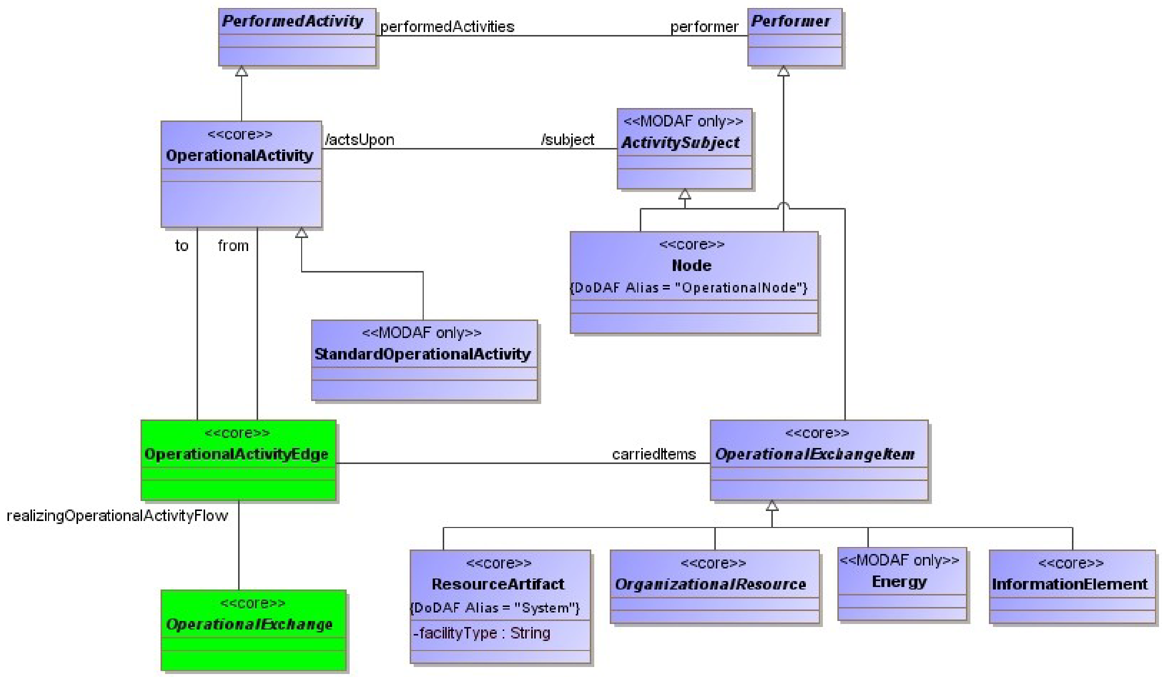


Figure A.18 - OV-5

A.4.7 OV-6a

MODAF: An Operational Rules Model (OV-6a) specifies operational or business rules that are constraints on the way that business is done in the enterprise.

DoDAF: An Operational Rules Model (OV-6a) DoDAF-described View specifies operational or business rules that are constraints on the way that business is done in the enterprise.

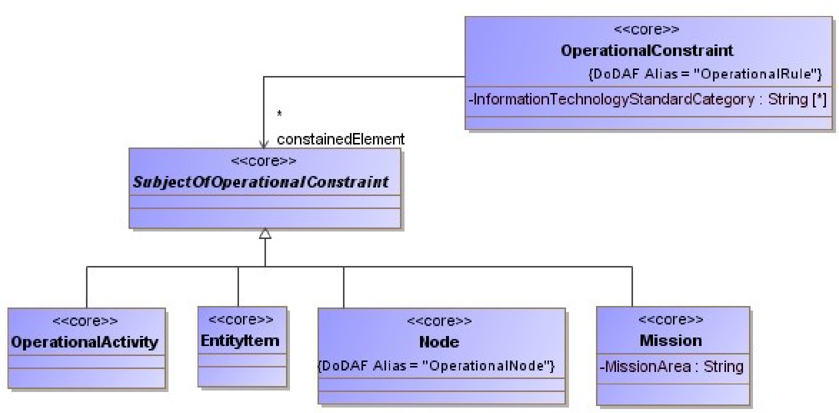


Figure A.19 - OV-6a

A.4.8 OV-6b

MODAF: OV-6b: The Operational State Transition Description is a graphical method of describing how an Operational Node or activity responds to various events by changing its state. The diagram represents the sets of events to which the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

DoDAF: The Operational State Transition Description (OV-6b) DoDAF-described View is a graphical method of describing how an Operational Activity responds to various events by changing its state. The diagram represents the sets of events to which the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

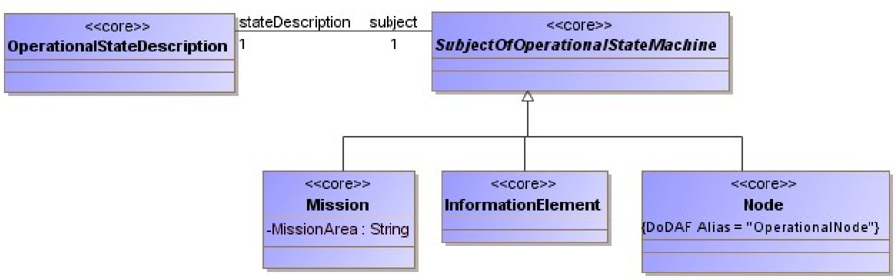


Figure A.20 - OV-6b

A.4.9 OV-6c

MODAF: OV-6c: The Operational Event-Trace Description provides a time-ordered examination of the information exchanges between participating Operational Nodes as a result of a particular scenario. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

DoDAF: The Operational Event-Trace Description (OV-6c) DoDAF-described View provides a time ordered examination of the resource flows as a result of a particular scenario. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

The net-centric OV-6c describes the business and mission processes that need to be executed to achieve Net-Centric Operations (NCO). The ability to discover, access, and understand information and capabilities from the NCE, where and when they are needed, is supported by the OV-6c and can be decomposed to the level of specificity required for the subject architecture. In the NCE, the OV-6c may depict the following:

- Exchanges between the Service Functionality Providers and Service Consumers, the Service Consumers and external Service Functionality Providers, and between the Service Functionality Providers and Unanticipated Users.
- Sequences that describe the timeline for the availability of information for any of its refinement states (raw, preprocessed, fused).
- Handling, methodologies, and the Enterprise Information Environment (EIE) infrastructure components that support the operational concepts of post before processing.
- Illustration of one-to-many, many-to-one, and many-to-many exchanges between Service Functionality Providers and Service Consumers found in the net-centric OV-3.

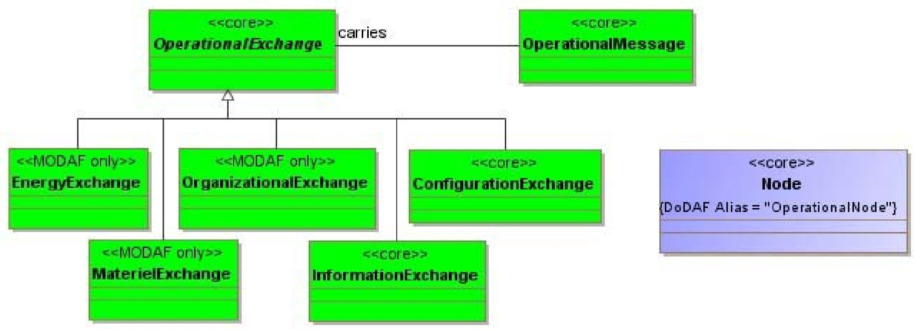


Figure A.21 - OV-6c

A.4.10 OV-7

The Logical Data Model describes the structure of an architecture domain's system data types and the structural business process rules (defined in the architecture's Operational View) that govern the system data. It provides a definition of architecture domain data types, their attributes or characteristics, and their interrelationships.

OV-7, including the domain's system data types or entity definitions, is a key element in supporting interoperability between architectures, since these definitions may be used by other organizations to determine system data compatibility. Often, different organizations may use the same entity name to mean very different kinds of system data with different internal structure. This situation will pose significant interoperability risks, as the system data models may appear to be compatible, each having a Target Track data entity but having different and incompatible interpretations of what Target Track means. In the NCE, the OV-7 describes the structure of data types (information elements) for information being made available or being consumed by the OV-5 activities and provides the organization and composition of metadata that can be used to characterize the information exchanged in the NCE.

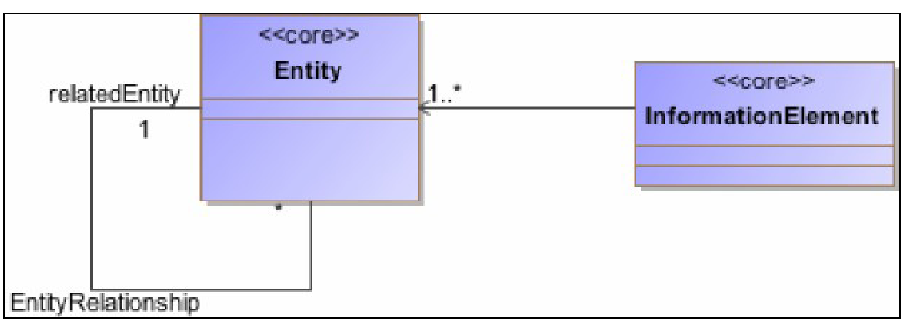


Figure A.22 - OV-7

A.5 Packages

This section shows the packaging of the DMM.

A.5.1 Package Structure

Package Structure of the DMM.

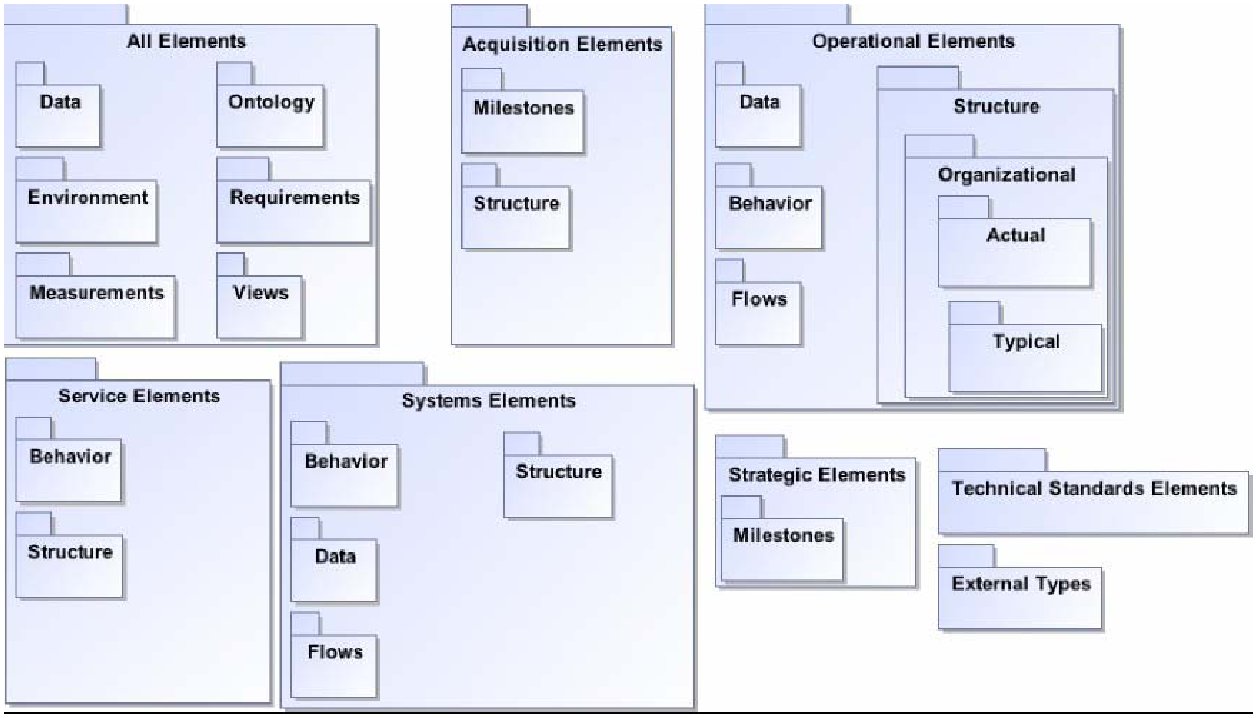


Figure A.23 - Package Structure

A.6 Products

This section documents each of the products of the DMM.

A.7 SOV

MODAF: The Service-Orientated View (SOV) is a description of services needed to directly support the operational domain as described in the Operational View. A service within MODAF is understood in its broadest sense, as a unit of work through which a provider provides a useful result to a consumer.

DoDAF: The Service Views within the Services Viewpoint describe services and interconnections providing for, or supporting, DoD functions. DoD functions include both warfighting and business functions. The Service Views associates service resources to the operational and capability requirements. These resources support the operational activities and facilitate the exchange of information.

A.7.1 SOV-1

The Service Taxonomy View (SOV-1) specifies a hierarchy of services. The elements in the hierarchy are service specifications (i.e., service interfaces), and the relationships between the elements are specializations (i.e., one Service is a special type of another). Along with SOV-2, it specifies a standard library of Service specifications for an enterprise, which Service implementers are expected to conform to.

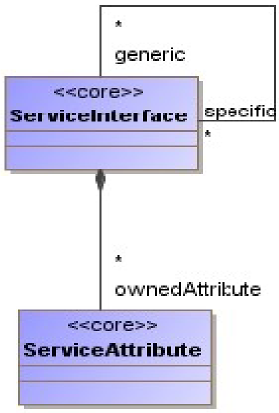


Figure A.24 - SOV-1

A.7.2 SOV-2

MODAF: The Service Taxonomy View (SOV-1) specifies a hierarchy of services. The elements in the hierarchy are service specifications (rather than service implementations), and the relationships between the elements are specializations (i.e., one Service is a special type of another).

DoDAF: NA

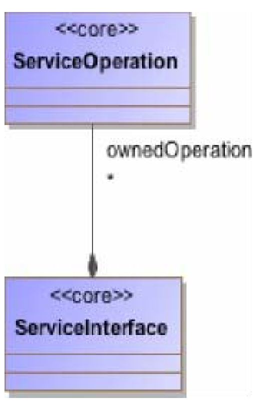


Figure A.25 - SOV-2

A.7.3 SOV-3

MODAF: The Capability to Service Mapping View (SOV-3) depicts which services contribute to the achievement of a capability.

DoDAF: The Operational Activity to Services Function Traceability Matrix (SvcV-5) DoDAF-described View addresses the linkage between service functions described in SvcV-4 and Operational Activities specified in OV-5.

Figure A.26 - SOV-3

A.7.4 SOV-4a

MODAF: The purpose of the Service Constraints View (SOV-4a) is to specify constraints that apply to implementations of services.

DoDAF: The SvcV-10a DoDAF-described View describes constraints on the resources, functions, data, and ports that make up the Service View physical architecture. The constraints are specified in text and may be functional or structural (i.e., nonfunctional).

Figure A.27 - SOV-4a

A.7.5 SOV-4b

MODAF: The purpose of the Service State Model View (SOV-4b) is to specify the possible states a service may have, and the possible transitions between those states.

DoDAF: The Services State Transition Description DoDAF-described View is a graphical method of describing a resource (or function) response to various events by changing its state. The diagram basically represents the sets of events to which the resources in the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

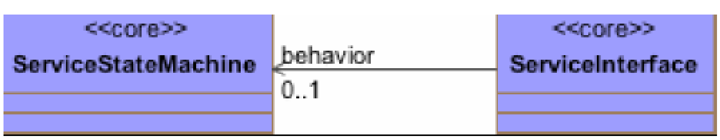


Figure A.28 - SOV-4b

A.7.6 SOV-4c

MODAF: The purpose of the Service Interaction Specification View (SOV-4c) is to specify how a service interacts with external agents, and the sequence and dependencies of those interactions.

DoDAF: The Services Event-Trace Description DoDAF-described View provides a time-ordered examination of the interactions between services functional resources. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

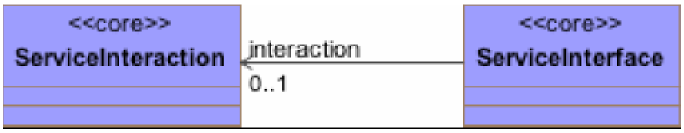


Figure A.29 - SOV-4c

A.7.7 SOV-5

MODAF: The Service Functionality View (SOV-5) defines the behavior of a service in terms of the functions it is expected to perform.

DoDAF: The Services Functionality Description provides detailed information regarding the Allocation of service functions to resources, and Flow of resources between service functions.

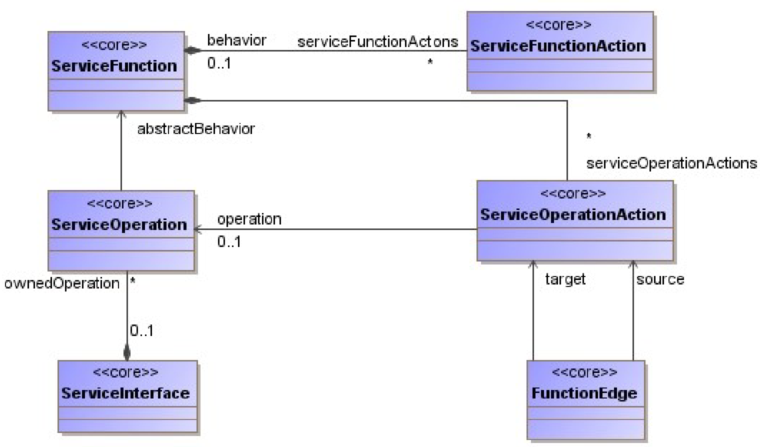


Figure A.30 - SOV-5

A.8 StV

The Strategic Elements are used in the Strategic View that provides an overall Enterprise Architecture assessment of the Capabilities and their relationships facilitating Capability Management (e.g., capability introduction, integration, realignment, and removal). While an Enterprise will have a number of UPDM Architecture Descriptions that have the Operational, System, Technical Standards, and All Views; only one Strategic View will exist across a number of Architecture Descriptions.

A.8.1 StV-1

MODAF: StV-1 addresses the enterprise concerns associated with the overall vision for transformational endeavors and thus defines the strategic context for a group of Enterprise capabilities.

DoDAF: CV-1: Vision: addresses the enterprise concerns associated with the overall vision for transformational endeavors and thus defines the strategic context for a group of capabilities.

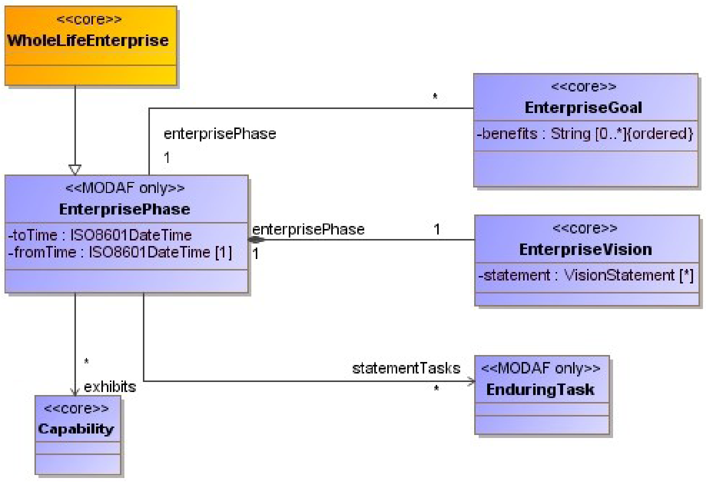


Figure A.31 - StV-1

A.8.2 StV-2

MODAF: The StV-2 Product models capability taxonomies.

DoDAF: The CV-2 DoDAF-described View models capability taxonomies.

The view presents a hierarchy of capabilities. These capabilities may be presented in context of an Enterprise Phase (i.e., it can show the required capabilities for current and future enterprises). StV-2 specifies all the capabilities that are referenced throughout one or more architectures. In addition it can be used as a source document for the development of high-level use cases and Key User Requirements (KUR). The StV-2 also provides metrics against each capability that may be used to measure successfully fielded capability.

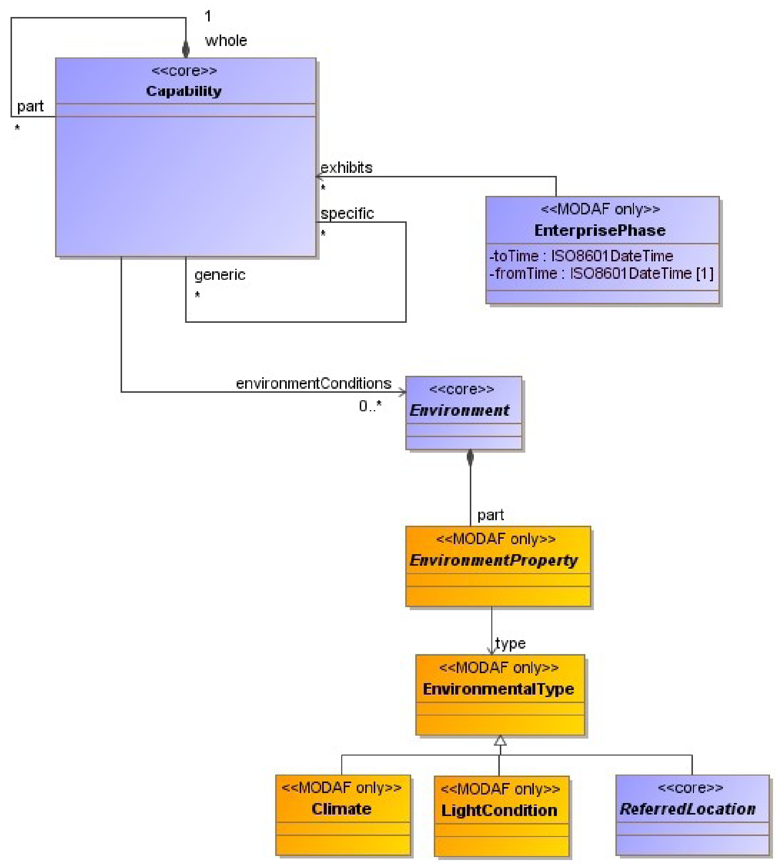


Figure A.32 - StV-2

A.8.3 StV-3

MODAF: StV-3 addresses the planned achievement of capability at different points in time or during specific periods of time (i.e., capability phasing).

DoDAF: CV-3: Capability Phasing The CV-3 addresses the planned achievement of capability at different points in time or during specific periods of time (i.e., capability phasing).

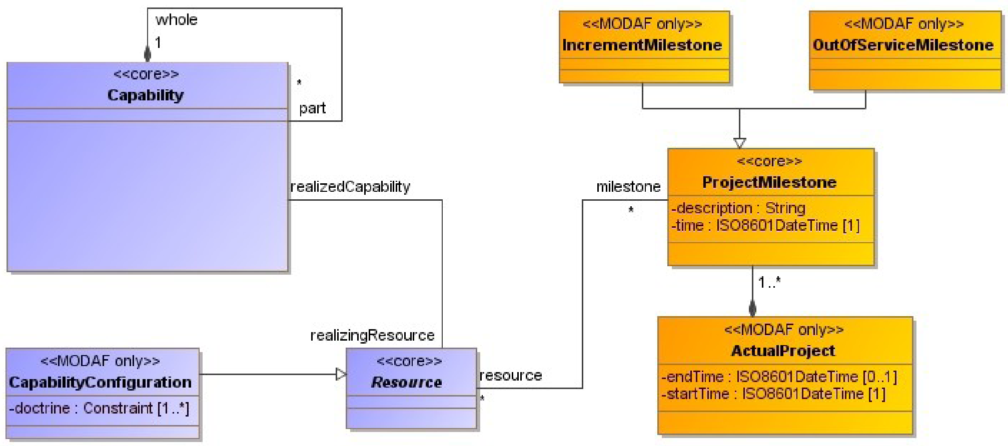


Figure A.33 - StV-3

A.8.4 StV-4

MODAF: The StV-4 Product describes the dependencies between planned capabilities. It also defines logical groupings of capabilities (capability clusters).

DoDAF: CV-4: Capability Dependencies: The CV-4 DoDAF-described View describes the dependencies between planned capabilities. It also defines logical groupings of capabilities.

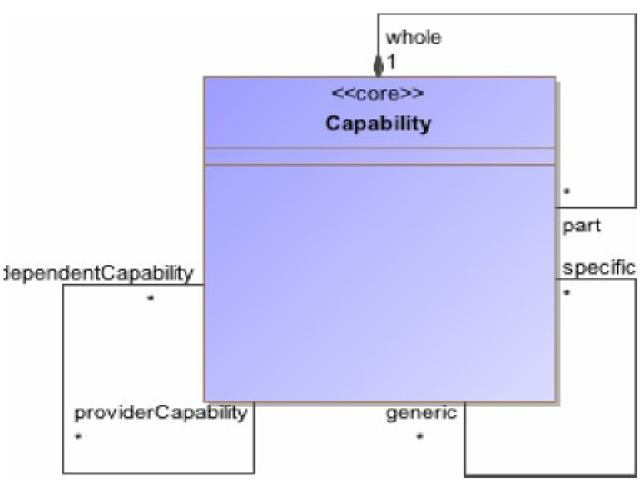


Figure A.34 - StV-4

A.8.5 StV-5

MODAF: StV-5 addresses the fulfillment of capability requirements, in particular by network enabled capabilities.

DoDAF: CV-5: Capability to Organizational Development Mapping: The CV-5 addresses the fulfillment of capability requirements.

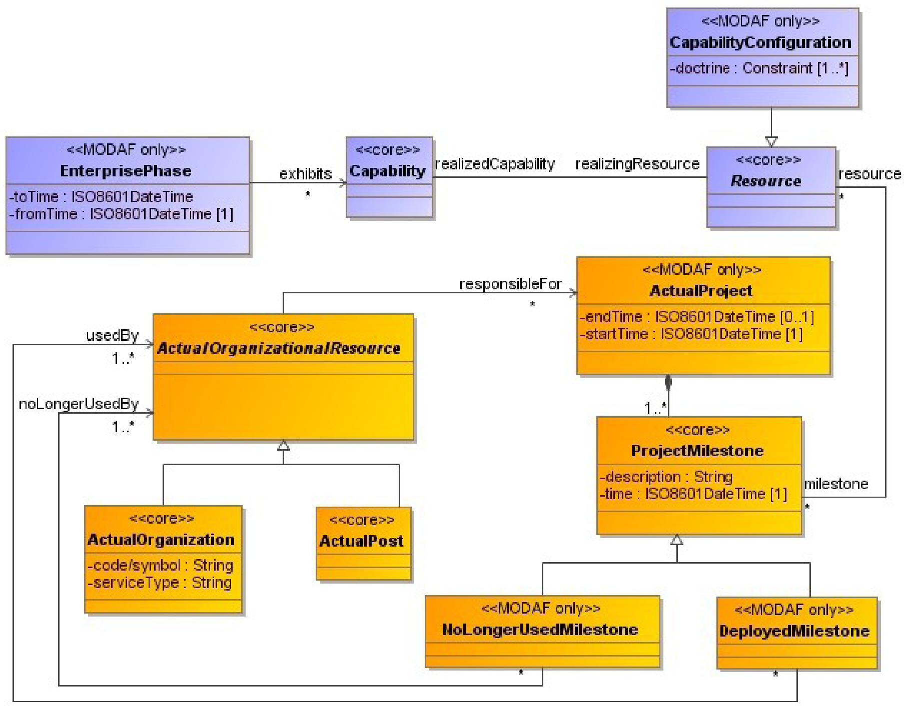


Figure A.35 - StV-5

A.8.6 StV-6

The StV-6 describes the mapping between the capabilities required by an Enterprise and the operational activities that those capabilities support.

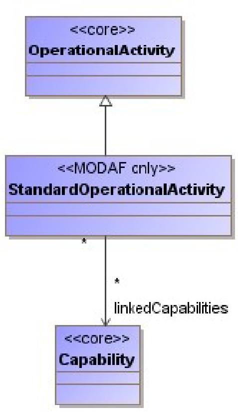


Figure A.36 - StV-6

A.9 SV

Models in the System Viewpoint represent alternate realizations in terms of equipment capability of the operational capabilities expressed through models in the Operational Viewpoint and in the User Requirements. The System Viewpoint primarily addresses the specification of the system capability needed (rather than implementation details). Significant changes originally made in MODAF improved the ability for modelers to represent configuration of capability that include people as well as systems and platforms.

A.9.1 SV-1

MODAF: Resource Interaction Specification (SV-1) addresses the composition and interaction of resources. From MODAF v1.1, SV-1 incorporates the human elements – Posts, Organizations, and Roles.

DoDAF: The Systems Interface Description (SV-1) DoDAF-described View addresses the composition and interaction of Systems. For DoDAF v2.0, the SV-1 incorporates the human elements as types of Performers - Organizations and Personnel Types.

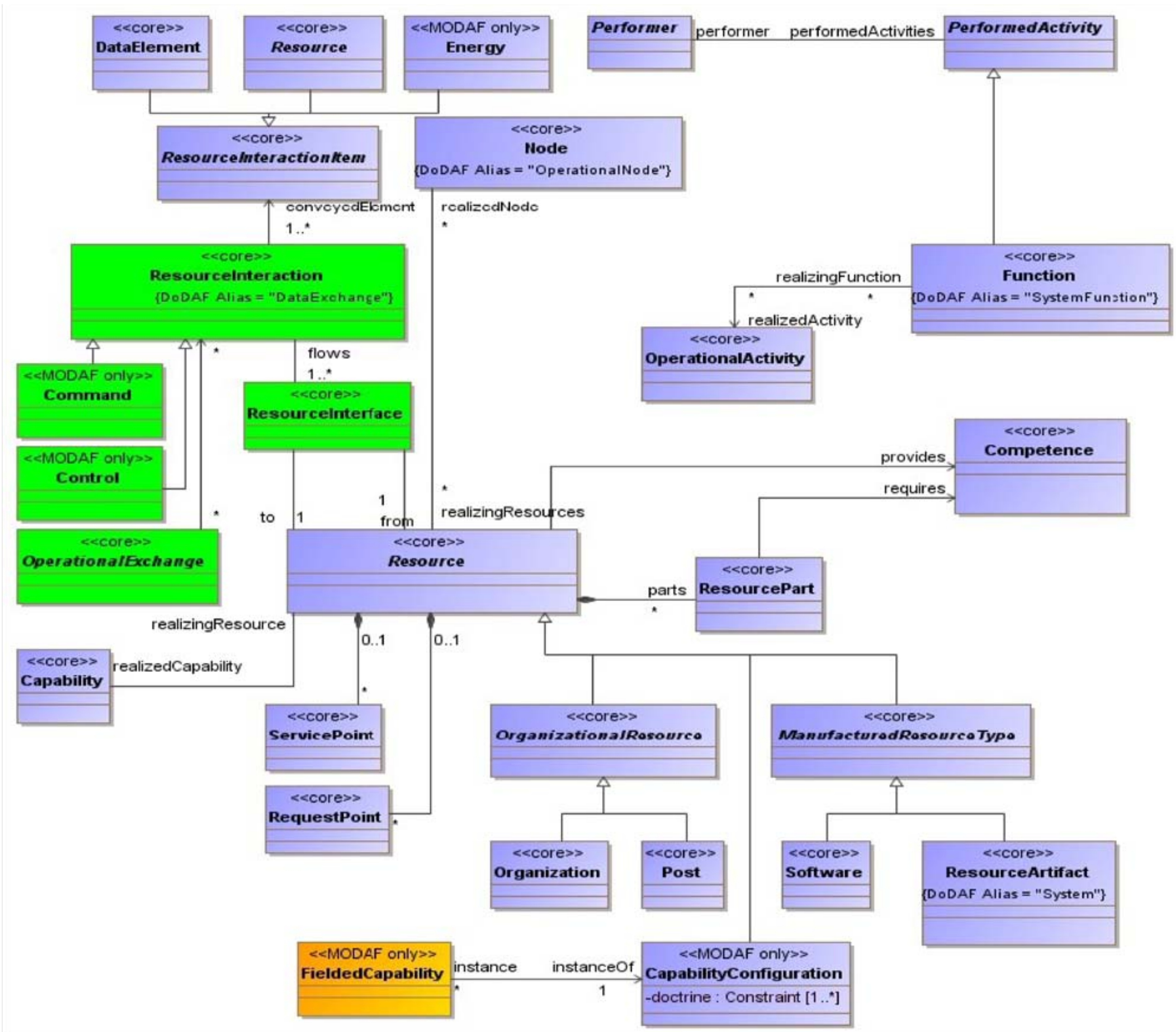


Figure A.37 - SV-1

A.9.2 SV-1 Resources

SV-1 Resource usages in detail.

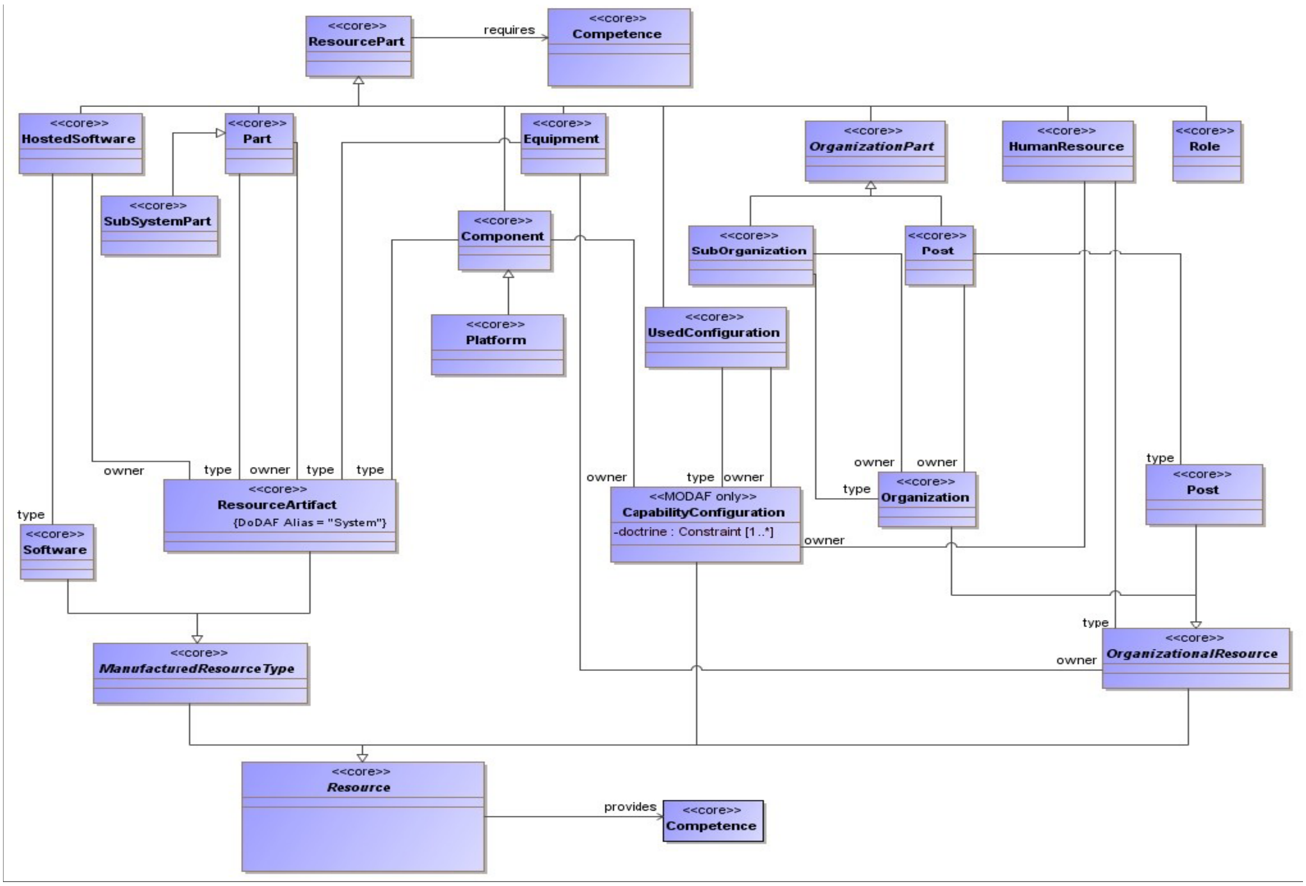


Figure A.38 - SV-1 Resources

A.9.3 SV-10a

MODAF: The purpose of this Product is to specify functional and non-functional constraints on the implementation aspects of the architecture (i.e., the structural and behavioral elements of the SV viewpoint).

DoDAF: The SV-10a Systems Rules Model DoDAF-described View 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).

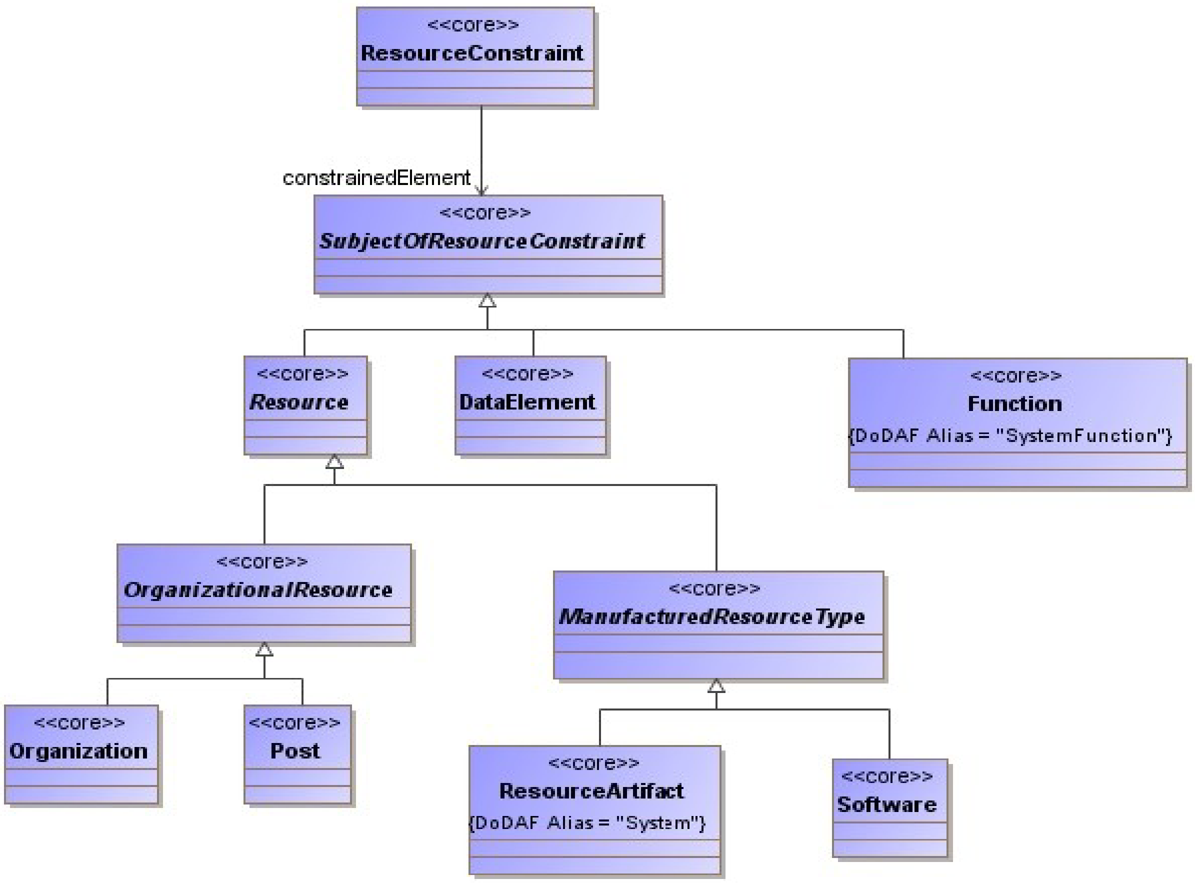


Figure A.39 - SV-10a

A.9.4 SV-10b

MODAF: The Resource State Transition Description is a graphical method of describing a resource (or function) response to various events by changing its state. The diagram basically represents the sets of events to which the Resources in the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

DoDAF: The Systems State Transition Description DoDAF-described View is a graphical method of describing a resource (or system function) response to various events by changing its state. The diagram basically represents the sets of events to which the resources in the Architecture will respond (by taking an action to move to a new state) as a function of its current state.

Each transition specifies an event and an action.

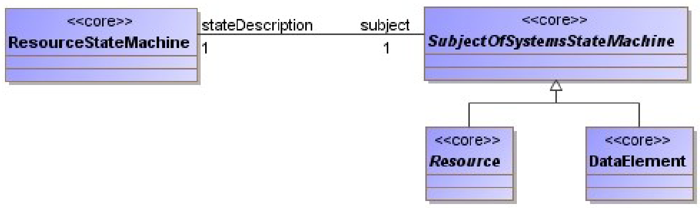


Figure A.40 - SV-10b

A.9.5 SV-10c

MODAF: The Resource Event-Trace Description 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.

DoDAF: The Systems Event-Trace Description provides a time-ordered examination of the interactions between functional resources. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

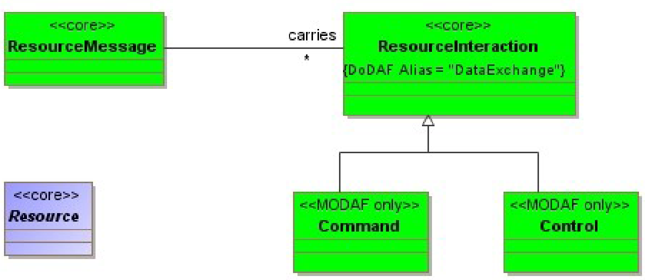


Figure A.41 - SV-10c

A.9.6 SV-11

MODAF: The SV-11 View defines the structure of the various kinds of system data that are utilized by the systems in the Architecture.

DoDAF: The DIV-3 Physical Data Model DoDAF-described view defines the structure of the various kinds of system or service data that are utilized by the systems or services in the Architecture.

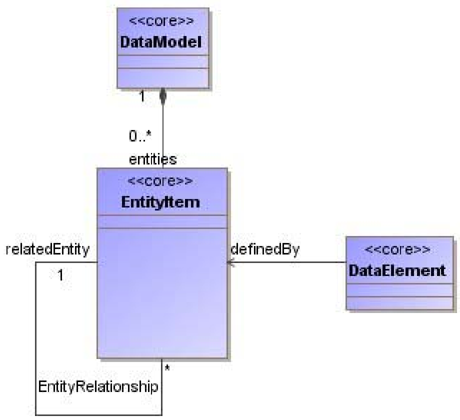


Figure A.42 - SV-11

A.9.7 SV-12

MODAF: The Service Provision View (SV-12) specifies configurations of resources that can deliver a service, and the levels of service those resources can deliver in different environments.

DoDAF: NA

Figure A.43 - SV-12

A.9.8 SV-2

MODAF: The Systems Communications Description (SV-2a/2b/2c) series of views is intended for the representation of communication networks and pathways that link communication systems, and provides details regarding their configuration.

DoDAF: A Systems Resource Flow Description (SV-2) DoDAF-described View specifies the resource flows between Systems and may also list the protocol stacks used in connections.

<<core>>

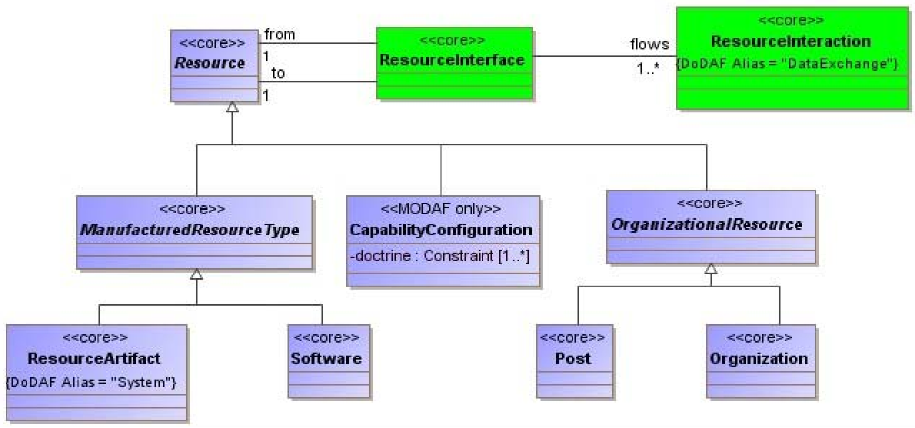


Figure A.45 - SV-3

A.9.10 SV-4

MODAF: Functionality Descriptions (SV-4) address human and system functionality.

DoDAF: The Systems Functionality Description (SV-4) DoDAF-described View addresses human and system functionality.

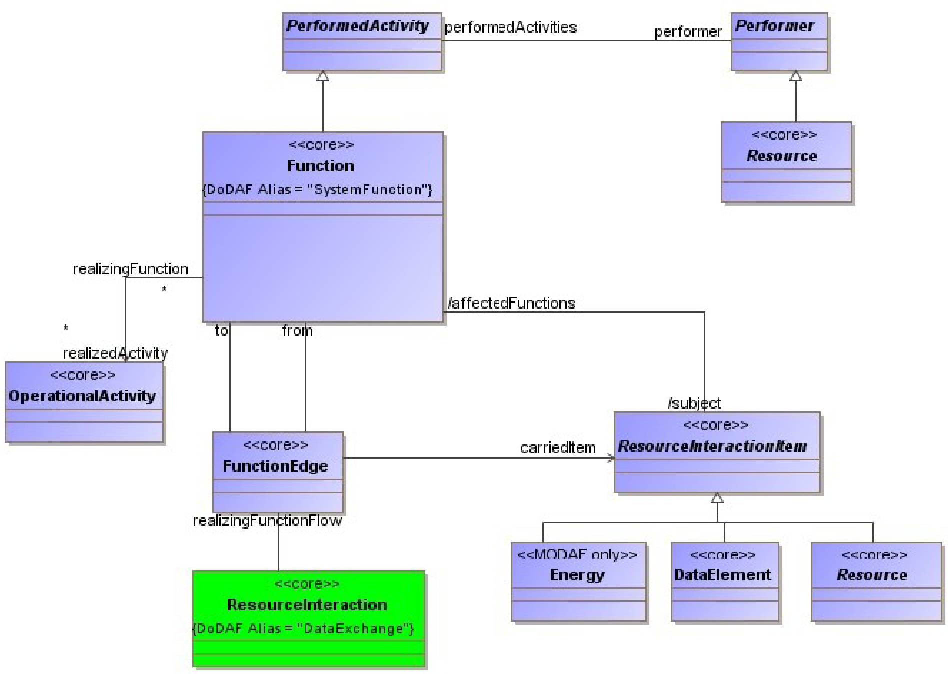


Figure A.46 - SV-4

A.9.11 SV-5

MODAF: This view has been expanded for the Service Orientated community by allowing for Service Functions as well as Operational Activities.

DoDAF: The Operational Activity to Systems Function Traceability Matrix (SV-5a) DoDAF-described View depicts the mapping of system functions (and, optionally, the capabilities and performers that provide them) to operational activities and thus identifies the transformation of an operational need into a purposeful action performed by a system or solution.

The Operational Activity to Systems Traceability Matrix (SV-5b) DoDAF-described View depicts the mapping of systems (and, optionally, the capabilities and performers that provide them) to operational activities and thus identifies the transformation of an operational need into a purposeful action performed by a system or solution.

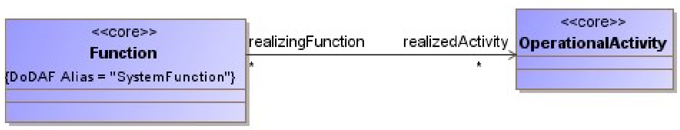


Figure A.47 - SV-5

A.9.12 SV-6

MODAF: The Systems Data Exchange Matrix specifies the characteristics of the system data exchanged between systems. The focus is on data crossing the system boundary.

DoDAF: The Systems Resource Flow Exchange Matrix DoDAF-described View specifies the characteristics of the system resource flows exchanged between systems. The focus is on resource crossing the system boundary.

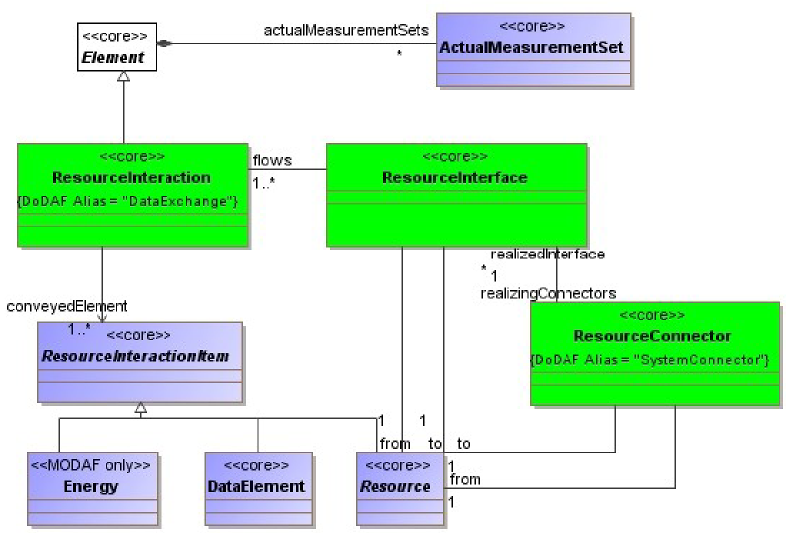


Figure A.48 - SV-6

A.9.13 SV-7

MODAF: The SV-7 is the Resource Performance Parameters Matrix and depicts the performance characteristics of a Resource (e.g., system, role, or capability configuration).

DoDAF: The SV-7 DoDAF-described View is the Systems Measures Matrix and depicts the measures (metrics) of resources.

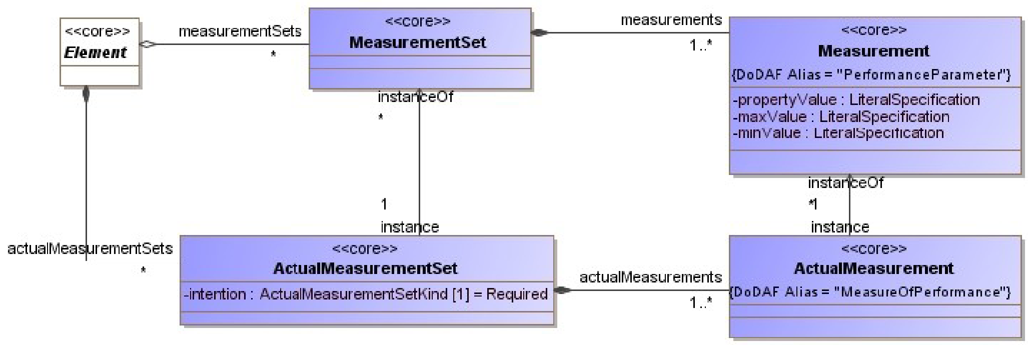


Figure A.49 - SV-7

A.9.14 SV-8

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

DoDAF: The Systems Evolution Description DoDAF-described View presents a whole lifecycle view of resources (systems), describing how it changes over time. It shows the structure of several resources mapped against a timeline.

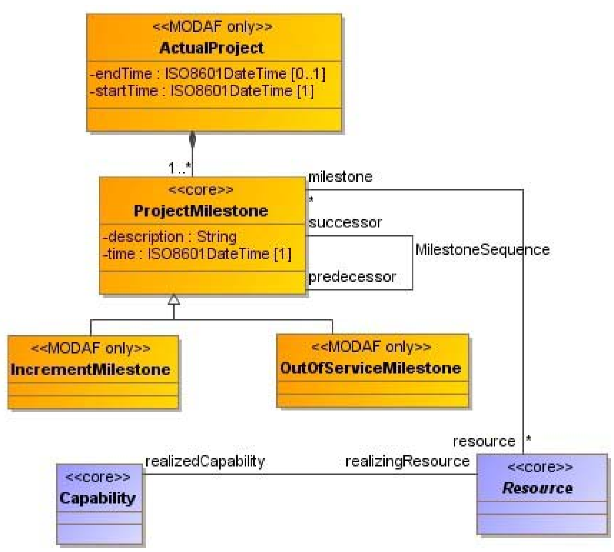


Figure A.50 - SV-8

A.9.15 SV-9

MODAF: The Technology & Skills Forecast defines the underlying current and expected supporting technologies and skills.

Expected supporting technologies and skills are those that can be reasonably forecast given the current state of technology and skills, and expected improvements / trends. New technologies and skills will be tied to specific time periods, which can correlate against the time periods used in SV-8 milestones and linked to Enterprise Phases.

DoDAF: The Technology & Skills Forecast defines the underlying current and expected supporting technologies and skills.

Expected supporting technologies and skills are those that can be reasonably forecast given the current state of technology and skills, and expected improvements / trends. New technologies and skills will be tied to specific time periods, which can correlate against the time periods used in SV-8 milestones and linked to Enterprise Phases.

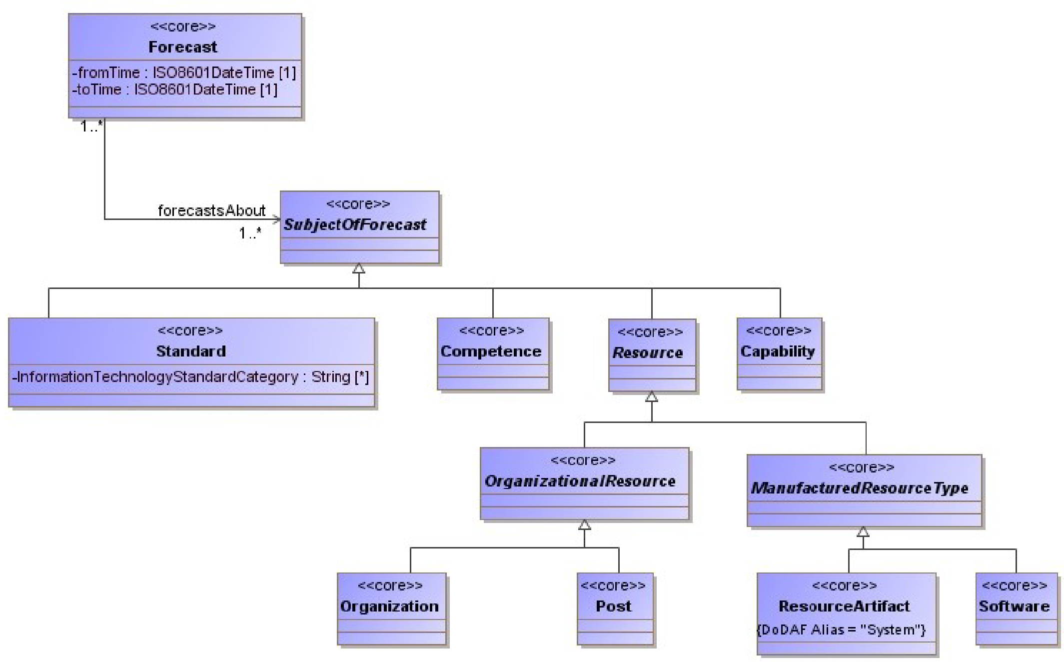


Figure A.51 - SV-9

A.10 TV

The Technical View is a set of products delineating standards, rules, notations, and conventions that apply to the implementation of the system architecture. When the standards profile is tied to the system elements to which they apply, TV-1 serves as the bridge between the SV and TV. SV-9 forecasts relate to the TV-1 in that a timed technology forecast may contribute to the decision to retire or phase out the use of a certain standard in connection with a system element. Similarly, SV-9 forecasts relate to TV-2 standards forecasts in that a certain standard may be adopted depending on a certain technology becoming available (e.g., the availability of Java Script may influence the decision to adopt a new HTML standard).

MODAF extends the core DoDAF Technical Standards Views to include non-technical standards and policies applicable to the architecture such as operational doctrine, industry process standards, etc. Additionally, the TV-1 may also document policies and standards applicable to the operational or business context. MoDAF also distinguishes between 'applicability' and 'conformance' with regard to architectural elements. If a standard is applicable to a given architecture, that architecture need not be fully conformant with the standard. The degree of conformance to a given standard may be judged on a risk basis at an approval point. An association between a Standard and an architectural element is not to be interpreted as stating the level of compliance of the element is fully compliant with that Standard. Additional evidences would need to be given (outside MODAF) to confirm the level of compliance. Finally, MoDAF adds the explicit requirement that any Standards cited in TV-1 View must, where appropriate, be in accordance with the trend towards open architectures (i.e., standards that encourage stove-piped systems are expressly prohibited).

A.10.1 TV-1&2&3

MODAF: Standards Profile (TV-1) defines the technical and non-technical standards, guidance and policy applicable to the architecture.

The Standards Forecast (TV-2) contains expected changes in technology-related standards and conventions, which are documented in the TV-1 Product.

DoDAF: The Standards Profile StdV-1 DoDAF-described View defines the technical, operational, and business standards, guidance and policy applicable to the architecture. The StdV-2 Standards Forecast DoDAF-described View contains expected changes in technology related standards, operational standards, or business standards and conventions, which are documented in the StdV-1 view.

Standards Profiles for a particular architecture must maintain full compatibility with the root standards they have been derived from. In addition, the TV-1 View may state a particular method of implementation for a standard, as compliance with a standard does not ensure interoperability. The standards cited are referenced as relationships to the systems, system functions, system data, hardware/software items, or communication protocols in SV-1, SV-2, SV-4, SV-6, OV-7, and SV-11 Products, where applicable. That is, each standard listed in the profile will be associated with the SV elements that implement or use the standard (e.g., SV-1, SV-2, SV-4, SV-6, OV-7 and SV-11 element standards, where applicable).

The Protocols referred to such as 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.

The Technical Standards Forecast (TV-2) provides a description of emerging standards and their potential impact on current Systems and Services View elements, within a set of time frames. The forecast for evolutionary changes in the standards should be correlated against the time periods as mentioned in the SV-8 and SV-9 products.

A TV-2 complements and expands on the Standards Profile (TV-1) product and should be used when more than one emerging standard time-period is applicable to the architecture.

The specific time periods selected (e.g., 6-month, 12-month, and 18-month intervals) and the standards being tracked will be coordinated with architecture transition plans (see SV-8). That is, insertion of new capabilities and upgrading of existing systems may depend on, or be driven by, the availability of new standards and products incorporating those standards. The forecast specifies potential standards and thus impacts current architectures and influences the development of transition and objective (i.e., target) architectures. The forecast will be tailored to focus on standards areas that are related to the purpose for which a given architecture is being described and should identify potential standards affecting that architecture. If interface standards are an integral part of the technologies important to the evolution of a given architecture, then it may be convenient to combine TV-2 with SV-9. For other projects, it may be convenient to combine all the standards information into one document, combining TV-2 with TV-1.

TV-2 delineates the standards that will potentially impact the relevant system elements (from SV-1, SV-2, SV-4, SV-6, and OV-7) and relates them to the time periods that are listed in the SV-8 and SV-9. A system's evolution, specified in SV-8, may be tied to a future standard listed in TV-2. A timed technology forecast from SV-9 is related to a TV-2 standards forecast in the following manner: a certain technology may be dependent on a TV-2 standard (i.e., a standard listed in TV-2 may not be adopted until a certain technology becomes available). This is how a prediction on the adoption of a future standard, as applicable to systems elements from SV-1, SV-2, SV-4, SV-6, and OV-7, may be related to standards listed in TV-1 through the SV-9.

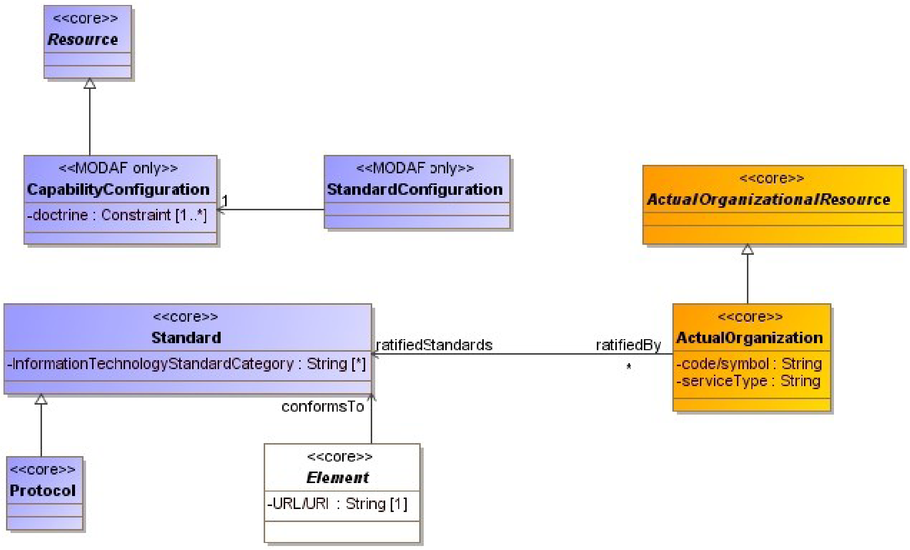


Figure A.52 - TV-1 &2&3

Annex B - UPDM Elements Traceability to DoDAF/MODAF Elements

(non-normative)

This Annex shows the traceability among UPDM stereotypes and DODAF/MODAF elements. Please note that not all DoDAF/MODAF elements have corresponding UPDM stereotype. Those DoDAF/MODAF elements are modeled by UML artifacts directly, which shows in the Metaclass column.

Viewpoint	MODAF 1.2 Model Elements	DoDAF 1.5 Model Elements	UPDM Stereotype	Metaclass
Acquisition	CapabilityIncrement	N/A	IncrementMilestone	InstanceSpecification
Acquisition	MilestoneInProject	N/A	ActualProjectMilestone	InstanceSpecification
Acquisition	MilestoneRelationship	N/A	MilestoneSequence	Dependency
Acquisition	OrganisationProjectRelationship	N/A		Dependency
Acquisition	OutOfService	N/A	OutOfServiceMilestone	InstanceSpecification
Acquisition	Project	N/A	Project/ProjectMilestone	InstanceSpecification
Acquisition	ProjectMilestone	N/A	ProjectMilestoneType	Class
Acquisition	ProjectOwnership	N/A		Dependency
Acquisition	ProjectSequence	N/A	ProjectSequence	InformationFlow
Acquisition	ProjectTheme	N/A	ProjectTheme	Property
Acquisition	ProjectType	N/A	ProjectType	Class
Acquisition	ProjectWholePart	N/A		Slot
Acquisition	RelatedProjectReference	N/A		InstanceValue
Acquisition	Status	N/A	ProjectStatus	Slot
Acquisition	StatusAtMilestone	N/A		Slot
Acquisition	StatusIndicators	N/A		Enumeration
Acquisition	StatusLiteral	N/A		LiteralString
All Views	Alias	Alias	Alias	Comment

All Views	ArchitecturalDescription	Architectural Description	Architectural Description	Package
All Views	ArchitecturalFramework	Architectural Framework		Package
All Views	ArchitecturalProduct	Architectural Product		Package
All Views	ArchitecturalReference	Architectural Reference	ArchitecturalReference	Dependency
All Views	Architecture	Architecture	DefinesArchitecture	Abstraction
All Views	ArchitectureMetaData	N/A	ArchitectureMetadata	Comment
All Views	AssignedProperty	N/A		Property
All Views	Climate	N/A	Climate	Class
All Views	Concern	Concern		UseCase
All Views	ConformsTo	ConformsTo	Conform	Dependency
All Views	Definition	Definition	Definition	Comment
All Views	EnterprisePhase	N/A	EnterprisePhase	Class
All Views	EnterpriseStructure	N/A		Property
All Views	EnterpriseTemporalPart	N/A		Property
All Views	Environment	Environment	Environment	Class
All Views	EnvironmentalProperty	Environmental Property	EnvironmentProperty	Property
All Views	ExternalIndividual	ExternalIndividuall	ExternalIndividual	InstanceSpecification
All Views	ExternalType	ExternalType	ExternalType	Class
All Views	FrequencyRange	N/A		Slot
All Views	InformationModel	InformationModel		Package
All Views	ISO8601DateTime	N/A	ISO8601DateTime	LiteralString
All Views	LightCondition	N/A	LightCondition	Class
All Views	Matrix	Matrix		Package
All Views	MeasurableProperty	Measurable Property	MeasureOfPerformance	Slot
All Views	MetaData	MetaData	Metadata	Comment
All Views	Ontology	Ontology		Package
All Views	OntologyReference	OntologyReference	OntologyReference	Extension

All Views	ProductOfView	ProductOfView		Instantiate
All Views	QualitativeProperty	Qualitative Property	QualitativeProperty	Property
All Views	SameAs	N/A	SameAs	Dependency
All Views	StakeholderHasConcern	StakeholderHas Concern		Dependency
All Views	Standard	Standard	Standard	Class
All Views	StereotypeExtension	Stereotype Extension	StereotypeExtension	Comment
All Views	TextProduct	TextProduct		Package
All Views	View	Product	View	Package
All Views	WholeLifeEnterprise	Enterprise	Enterprise	Class
Operational	ActivityComposition	Activity Composition	Activity	Activity
Operational	ActivitySubject	ActivitySubject	ActivitySubject	Class
Operational	ActsUpon	ActsUpon		Association
Operational	ActualCompetence	N/A		InstanceSpecification
Operational	PhysicalLocation	N/A	PhysicalLocation	DataType
Operational	ActualOrganisation	Organization	ActualOrganization	InstanceSpecification
Operational	ActualOrganisationalResource	N/A	ActualOrganizational Resource	InstanceSpecification
Operational	ActualOrganisationRelationship	N/A	ActualOrganization Relationship	InformationFlow
Operational	ActualOrganizationComposition	N/A	ActualOrganizationRole	Slot
Operational	ActualPost	Billet	ActualPost	InstanceSpecification
Operational	ArbitraryConnection	N/A	ArbitraryRelationship Connector	Relationship
Operational	CapabilityForNode	N/A		Property
Operational	Competence	N/A	Competence	Class
Operational	CompetenceForRole	N/A	ProvidesCompetence/ RequiresCompetence	Dependency
Operational	ConceptDescription	ConceptDescription		Comment
Operational	ConceptItem	ConceptItem	ConceptItem	Element
Operational	Consumes	N/A		Dependency

Operational	EnergyFlow	N/A	EnergyExchange	InformationFlow
Operational	HighLevelOperationalConcept	Operational Concept	HighLevelOperational Concept	Class
Operational	InformationElement	Information Element	InformationElement	InformationItem
Operational	InformationExchange	Information Exchange	InformationExchange	InformationFlow
Operational	InformationExchangeMessage	Information ExchangeMessage	OperationalExchangeItem	Class
Operational	InstanceInConcept	InstanceInConcept		Property
Operational	ItemInConcept	ItemInConcept	ItemInConcept	Property
Operational	KnownResource	KnownResource	KnownResource	Property
Operational	Location	Location	Location	Class
Operational	LogicalArchitecture	N/A	LogicalArchitecture	Class
Operational	LogicalDataModel	LogicalData Model	LogicalDataModel	Package
Operational	LogicalFlow	LogicalFlow		InformationFlow
Operational	MaterielFlow	N/A	MaterielExchange	InformationFlow
Operational	Mission	Mission	Mission	UseCase
Operational	MovementOfPeople	N/A		InformationFlow
Operational	Needline	Needline	Needline	Connector Association
Operational	Node	OperationalNode	Node/OperationalNode	Class
Operational	NodeEnvironment	N/A		Class
Operational	NodeHasBehaviour	N/A		InstanceSpecification
Operational	NodeParent	N/A	NodeParent	Element
Operational	NodeType	OperationalNode	Node	Class
Operational	OpActivityInputPin	N/A		InputPin
Operational	OpActivityOutputPin	N/A		OutputPin
Operational	OperationalActivity	Operational Activity	OperationalActivity	Activity
Operational	OperationalActivityAction	N/A	OperationalActivity Action	CallBehaviorAction
Operational	OperationalActivityFlow	N/A	OperationalActivityEdge	ActivityEdge

Operational	OperationalConstraint	Operational Constraint	OperationalConstraint/ OperationalRule	Constraint
Operational	OperationalInteractionSpecificatio	N/A		Interaction
Operational	OperationalNodeLifeline	OperationalNode Lifeline		Lifeline
Operational	OperationalStateDescription	OperationalState Description	OperationalStateMachine	StateMachine
Operational	OperationalSwimlane	N/A		ActivityPartition
Operational	OrgResourceReference	N/A		InstanceValue
Operational	ProblemDomain	ProblemDomain		Class
Operational	ProcessOwner	N/A	OwnsProcess	Dependency
Operational	Provides	N/A		Dependency
Operational	ReferredLocation	Location	ReferredLocation	Class
Operational	RequiredNodeLocation	RequiredNode Location		Dependency
Operational	SubjectOfOperationalConstraint	SubjectOf Operational Constraint	SubjectOfOperational Constraint	Classifier
Services	AsynchronousMessage	N/A		DataType
Services	MessageHandler	N/A		Class
Services	Service	Service	Service	UseCase
Services	ServiceAimsToAchieve	ServiceAimsTo Achieve		Realization
Services	ServiceAttribute	ServiceAttribute	ServiceAttribute	Property
Services	ServiceComposition	Service Composition	Service	UseCase
Services	ServiceConnectorEnd	ServiceConnector End		Role
Services	ServiceConsumer	ServiceConsumer	ServiceConsumer	Actor
Services	ServiceFunction	N/A	ServiceFunction	Activity
Services	ServiceGeneralisation	Service Generalization		Generalisation
Services	ServiceInteractionSpecification	ServiceInteraction Specification	ServiceInteraction Specification	Interaction
Services	ServiceInterface	ServiceInterface	ServiceInterface	Port

Services	ServiceInterfaceDefinition	ServiceInterface Definition	ServiceInterface Definition	Interface
Services	ServiceInterfaceOperation	ServiceInterface Operation	ServiceOperation	Operation
Services	ServiceInterfaceParameter	ServiceInterface Parameter	ServiceParameter	Parameter
Services	ServiceInterfaceSchema	ServiceInterface Schema		Class
Services	ServiceLevel	ServiceLevel	ServiceLevel	InstanceSpecification
Services	ServiceLifeLine	N/A		Lifeline
Services	ServiceNeedline	N/A	ServiceNeedline	Connector
Services	ServiceParameterType	ServiceParameter Type	ServiceParameterType	DataType
Services	ServicePolicy	ServicePolicy	ServicePolicy	Constraint
Services	ServiceProvision	N/A		Package
Services	ServiceStateMachine	N/A	ServiceStateMachine	StateMachine
Services	ServiceSupportsActivity	ServiceSupports Activity	SupportsOperational Activity	Usage
Services	UsesService	N/A		Dependency
Strategic	ActivityMapsToCapability	N/A	MapsToCapability	Denpendency
Strategic	Capability	Capability	Capability	Class
Strategic	CapabilityComposition	N/A	Capability	Class
Strategic	CapabilityDependency	N/A	RequiresCapability	Dependency
Strategic	CapabilitySpecialisation	N/A		
Strategic	ConfigurationDeployed	N/A	DeployedMilestone	InstanceSpecification
Strategic	ConfigurationNoLongerUsed	N/A	NoLongerUsedMilestone	InstanceSpecification
Strategic	EnduringTask	N/A	EnduringTask	UseCase
Strategic	EnterpriseGoal	N/A	EnterpriseGoal	Class
Strategic	EnterpriseVision	N/A	EnterpriseVision	Class
Strategic	EnvironmentalConditions	N/A	Environment	Class
Strategic	StandardOperationalActivity	N/A	StandardOperational Activity	Activity
Strategic	VisionStatement	VisionStatement	VisionStatement	Comment

Systems	ActivityToFunctionMapping	ActivityToFunctio n Mapping		Denpendency
Systems	Artefact	System	Artefact	Class
Systems	CapabilityConfiguration	SystemNode	CapabilityConfiguration	Class
Systems	CapabilityRealisation	N/A		Realization
Systems	Commands	Commands	Commands	InformationFlow
Systems	Controls	Controls	Controls	InformationFlow
Systems	DataElement	DataElement	DataElement	Class
Systems	FieldedCapability	N/A	FieldedCapability	InstanceSpecification
Systems	Forecast	Forecast	Forecast	Comment
Systems	Function	SystemFunction	Function	Activity
Systems	FunctionFlow	N/A	FunctionEdge	ActivityEdge
Systems	FunctionProvision	N/A	FunctionParameter	Parameter
Systems	FunctionsUpon	N/A	FunctionAction	CallBehaviorAction
Systems	HostedSoftware	N/A	HostedSoftware	Property
Systems	HumanResource	HumanResource	HumanResource	Property
Systems	ImplementsDataModel	ImplementsData Model	DataModel	Package
Systems	NodeRealisation	N/A		InstanceSpecification
Systems	OrganisationalResource	N/A	OrganizationalResource	Class
Systems	OrganisationType	Organization	Organization	Class
Systems	Part	N/A	OrganizationPart/Part	Property
Systems	PhysicalArchitecture	N/A		Class
Systems	PhysicalAsset	Platform		Class
Systems	PhysicalDataModel	PhysicalData Model	PhysicalDataModel	Package
Systems	Platform	Platform	Platform	Property
Systems	Post	Billet	Post	Property
Systems	PostType	N/A	PostType	Class
Systems	RadioFrequencyPortConnector	N/A		Denpendency
Systems	ResourceConfiguration	N/A		Class
Systems	ResourceConstraint	N/A	ResourceConstraint	Constraint

Systems	ResourceInteraction	N/A	ResourceInteraction	InformationFlow
Systems	ResourceInteractionSpecification	N/A		Comments
Systems	ResourceLifeLine	N/A		Lifeline
Systems	ResourceLifelineItem	N/A		ConnectableElement
Systems	ResourcePartition	N/A		SwimLane
Systems	ResourceStateMachine	N/A	ResourceStateMachine	StateMachine
Systems	ResourceStateMachineOwner	N/A		Actor
Systems	ResourceType	N/A	ResourceType	Class
Systems	ResourceUsage	N/A		Usage
Systems	Role	Role	Role	Property
Systems	RoleType	N/A		Class
Systems	ServiceFunctionToFunction Mappin g	N/A		Denpendency
Systems	Software	Software	Software	Class
Systems	SoftwareComponent	Software Component		Class
Systems	SubjectOfForecast	SubjectOfForecast	SubjectOfForecast	Element
Systems	SubjectOfResourceConstraint	N/A	SubjectOfResource Constraint	Element
Systems	SubOrganisation	N/A	SubOrganization	Property
Systems	System	System	System	Class
Systems	SystemPort	SystemPort		Port
Systems	SystemPortConnector	SystemPort Connector	SystemConnector	Connector
Systems	SystemPortConnectorEnd	SystemPort ConnectorEnd		Comments
Systems	SystemStructureModel	SystemStructure Model		Package
Systems	UsedConfiguration	N/A		Class
Systems	VersionOfConfiguration	N/A		Class
Systems	WholeLifeConfiguration	N/A		Class
Technical	Attribute	Attribute	Attribute	Property
Technical	DataModel	DataModel	DataModel	Package
Technical	Entity	Entity	Entity	DataType

Technical	EntityRelationship	EntityRelationship	EntityRelationship	Association
Technical	ImplementsProtocol	Implements Protocol		Dependency
Technical	Protocol	Protocol	Protocol	Class
Technical	ProtocolImplementation	Protocol Implementation	ProtocolImplementation	Connector
Technical	ProtocolStack	ProtocolStack		Property
Technical	RatificationBody	N/A		Dependency
Technical	SpectrumAllocation	N/A		Dependency
Technical	StandardConfiguration	Standard Configuration	StandardsConfiguration	Comment
Technical	SubtypeRelationship	Subtype Relationship		Dependency

Annex C - Sample Problem

(non-normative)

C.1 Purpose

The purpose of this annex is to illustrate how UPDM can support DODAF and MoDAF requirements for organizations developing Network Enabled Capability (NEC) systems using some of the basic features of the specification. This example provides a model that illustrates a sample of DoDAF and MoDAF views addressing the problem space described below.

C.2 Scope

The scope of this example is to provide a diagram for the views that are most used and most requested by the defense community. The intent is to select portions of the sample problem to illustrate how the diagrams can be applied, and demonstrate some of the possible interrelationships among the model elements in the different diagrams. The sample problem does not highlight all of the features of the specification.

C.3 Problem Scenario

C.3.1 Problem Domain Suitability

The problem domain is civilian maritime search and rescue (SAR). Civilian SAR was selected for several reasons:

- UK MODAF 1.1 has previously used this domain to illustrate its framework[1] .
- The scenario and modeling was easily updated to include UPDM concepts including US DoDAF 1.5.
- SAR is internationally recognized problem domain with easy-to-recognize typical scenarios.
- SAR is based on publicly available International Agreements[2] and implementing or conforming National Plans including the US[3] and the UK[4].

1. See Acknowledgements
2. See for example, International Aeronautical and Maritime Search and Rescue (IAMSAR) Manua l, 2007 ed., 6th ed. London: IMO; Montreal: ICAO, 2007. IAMSAR Manual is by jointly published by the International Maritime Organization (IMO) and the International Civil Aviation Organization (ICAO). It consists of a three volume set: Volume I is Organization and Management; Volume II is Mission Co-ordination; & Volume III is Mobile Facilities.
3. See for example, U.S. National Search and Rescue Supplement (NSS) to the International Aeronautical and Maritime Search and Rescue Manual. National Search and Rescue Plan of the United States (US National SAR Plan). http://www.uscg.mil/hq/cg5/cg534/manuals/Natl_SAR_Plan(2007).pdf
4. See for example, Search and Rescue Framework for the United Kingdom of Great Britain and Northern Ireland, Queen's Printer and Controller, June 2002. (Published by MCGA - Maritime & Coastguard Agency, Spring Place, 105 Commercial Road, Southampton. SO15 1EG.) "The organization for Search and Rescue (SAR) in the UK is an amalgam of separate Governments Departments, the emergency services, and other organizations. A number of charities and voluntary organizations dedicated to SAR also play a significant role. The purpose of this document is to provide a management framework for SAR in the UK. (back cover)". http://www.mcga.gov.uk/c4mca/mcga-uk_sar_framework_document.pdf

- The documentation is generally unclassified as opposed to many equivalent defense or military plans.
- Subject matter experts and periodicals are readily available.[1]
- The domain is sufficiently large and complex involving mixed human, software, and hardware solutions. As such, it will support the current specification that includes parametric modeling from systems engineering (SysML)[2] as well as future evolutions of UPDM that may include more national and multinational architecture frameworks. Several of the countries share usage of the same automated information systems and sensors.

C.3.2 Acknowledgements

The scenario is derived from the UK Search and Rescue framework, which is publicly available on the internet[3]. The sample problem is based on a concept derived by VEGA under contract for the UK MOD[4] . The UPDM Group acknowledges its debt owed to the authors of the original problem:

- Ian Bailey of Model Futures,
- Peter Martin of Logica CMG, and
- Paul King of Vega

C.3.3 Summary

C.3.4 The “Yacht in Distress” Scenario

The Sample Problem applies UPDM to a common scenario in civilian maritime Search and Rescue (SAR) operations -- a yacht in distress. A monitoring unit picks up the distress signal from the yacht and passes it on to the Command and Control (C2) Center. The C2 Center coordinates the search and rescue operation among helicopters, a naval ship, and a Royal National Lifeboat Institution (RNLI) Lifeboat. This section is structured to show each diagram in the context of how it might be used in such an example problem.

C.4 Diagrams

C.4.1 Package Overview (Structure of the Sample Model)

Acronyms

The table below provides definitions for acronyms used in this sample problem.

1. See for example, ON SCENE - The Journal of U. S. Coast Guard Search and Rescue . Summer 2008, “Exceptional SAR Stories”, pp. 29 – 40 for more detailed scenarios similar to the Problem Scenario and Fall 2003, “SPECIAL SECTION - SAR Case Studies: A Review”, pp. 18 -28 regarding performance standards.
2. See U SCG, “SAR System Performance Benchmark ” – “ Percent of lives saved from imminent danger in the maritime environment” and subbenchmarks. http://uscg.mil/hq/cg5/cg534/SAR_Program_Info.asp (Current as of 29 April 2009).
3. See “MODAF: Examples: Search and Rescue Example” and the corresponding files are at http://www.modaf.org.uk/file_download/33/SAR.zip (as of 29 April 2009)
4. http://www.modaf.org.uk/vExamples/163/search-and-rescue-example

Table C-1: Acronyms

DoT	Department of Transport
NIMROD	Aircraft name
MRA	Maritime Role Aircraft
ESM	Electronic Signal Monitoring
RN ASR Helo	Royal Navy Approach Surveillance Radar Helicopter
RNLI	Royal National Lifeboat Institution
HMG	Her Majesty’s Government
TDM	Time Division Multiplex
MRT	Maritime Rescue Team
SAR	Search and Rescue
C2	Command and Control

Flow of SAR Example Models

Figure C.1 shows the flow of the SAR example models through the different viewpoints. Beginning with the All Viewpoint, the natural progression is through the key Strategic Views, the key Operational Views, the key Service Oriented Views, the key Systems Views, and finally to the Acquisition Views.

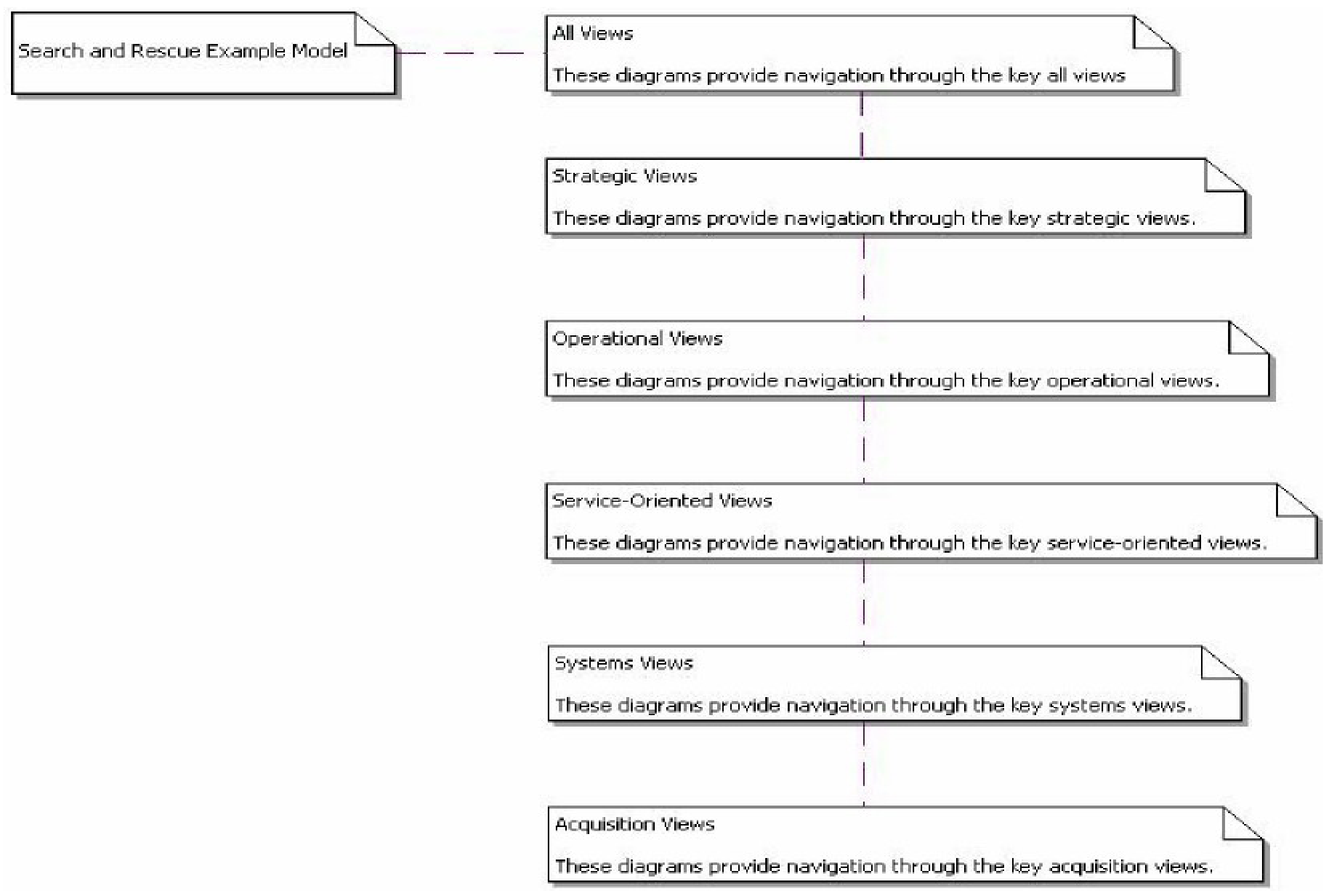


Figure C.1- Diagram Flow

C.5 All Views

The All Views provide overview and summary information as well as an integrated dictionary. This information is provided in a consistent form that allows quick reference and comparison among architectures.

C.5.1 AV-1 Enterprise Definition

Figure C.2 shows the top level context item, the Search and Rescue Enterprise, broken down into Temporal Phases with start and end dates.

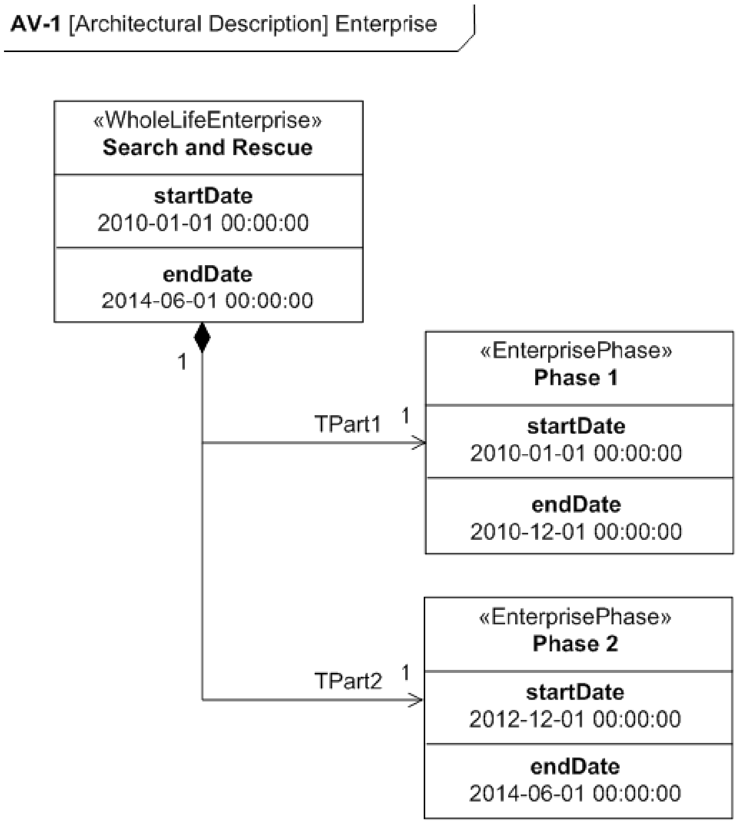


Figure C.2 - AV-1

Table C-1 is an alternate way of showing the AV-1. It provides executive-level summary information in a consistent form that allows quick reference and comparison between architectural descriptions. It includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.

Table C-2: AV-1 Alternate

AV-1 Overview and Summary Information

- Architecture Project Identification
  - Name: SAR Satellite Aid Tracking System
  - Architect: Coastguard Agency Architecture 4
  - Organization Developing the Architecture: Maritime & Coastguard Agency
  - Assumptions and Constraints: None
  - Approval Authority: Howard Overtree, Project Manager
  - Date Completed
- Scope: Architecture Views and Products Identification

- Views and Products Developed: AV1, StV1, StV2, StV4, StV5, StV6, OV1, OV2, OV3, OV4, OV5, OV6b, OV7, StV1, StV2, StV4, StV6, AcV1, AcV3, SOV1, SOV2, SOV3, SOV4a, SOV5, SV1, SV2, SV4, SV7
    - Time Frames Addressed: Present
    - Organizations Involved: Dept. of Transport, Maritime & Coastguard Agency
- Purpose and Viewpoint
    - Purpose of the Architecture: To detect and locate mariners, aviators and recreational enthusiasts in distress
    - From Whose Viewpoint Architecture is Developed: Users of System
- Context
    - Mission: Manage, coordinate and implement SAR activities
    - Tasking for Architecture Project: Tasking by UK SAR Steering Committee
    - Tools and File Formats Used: Together, Word, Excel

C.5.2 AV Measurements Definition

Figure C.3 shows the class diagram version of the measurements diagram. This provides a means of defining types of measurements that are important to the system. These consist of measureable quantitative measurements. It defines the measurements that are important to the capabilities in the strategic view such as find time and persistence, shown later.

These concepts are defined in All Views, as they can pertain to all elements in all views of the model. Metrics specific to System elements are addressed in the SV-7. As there is no diagram MODAF or DoDAF in All Views for expressing this information, we have created a new diagram. This could be called AV-n, Measurements Definition or other suitable name. This is an example of the extensibility features provided by UML and SysML.

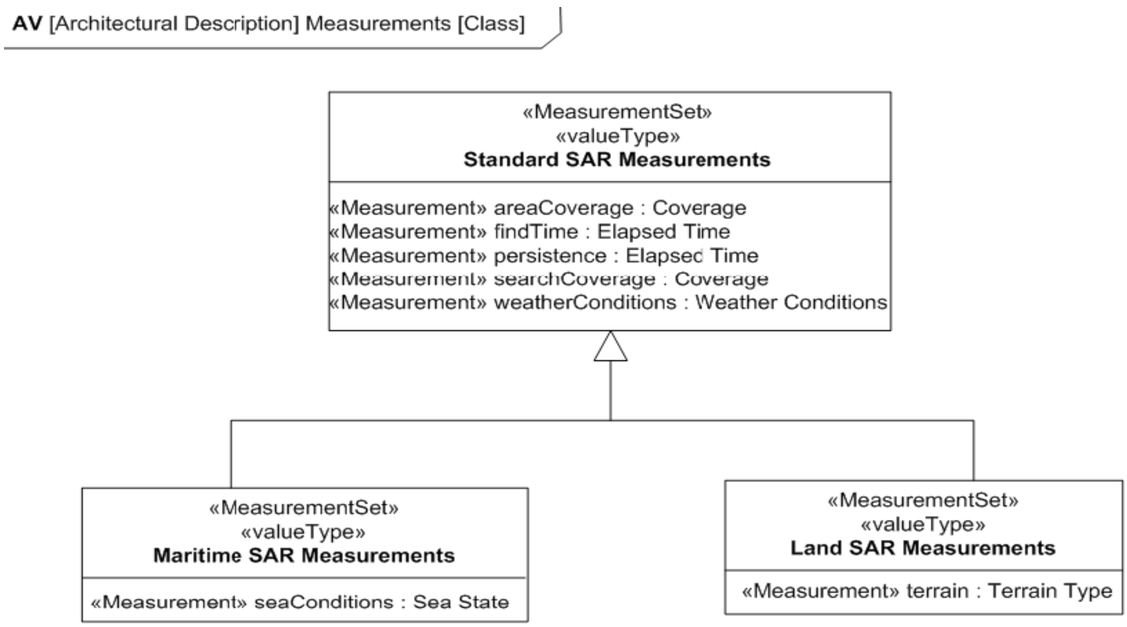


Figure C.3 - AV Measurements Class Diagram

C.5.3 AV Measurements Instances

Figure C.4 shows the instance diagram version of the measurements diagram. Instances of the measurements can be created and associated with architecture elements. In this case, they define the initial, required, and final values for SAR capabilities.

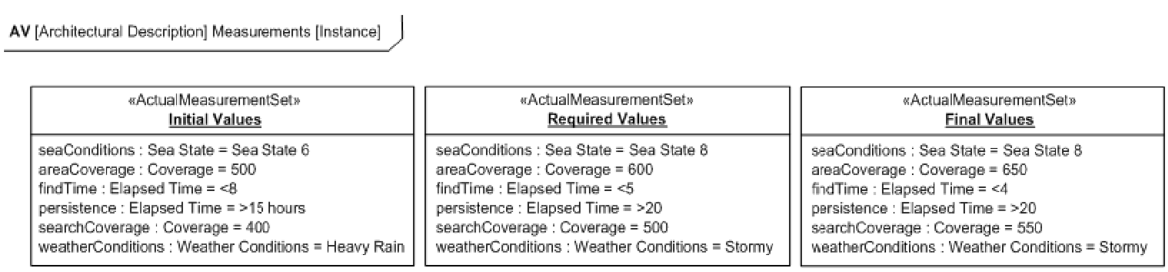


Figure C.4 AV - Measurements Instance Diagram

C.6 Strategic Views

The diagrams in the Strategic View provide a capability view of the SAR operation. These views will show the relationships between capabilities and the resources required to realize them.

C.6.1 StV-1 Capability Vision

Figure C.5 describes the strategic context for Search and Rescue Capabilities. It outlines the vision for a capability area over a specified period of time. It describes how high level goals and strategy are to be delivered in terms of capability.

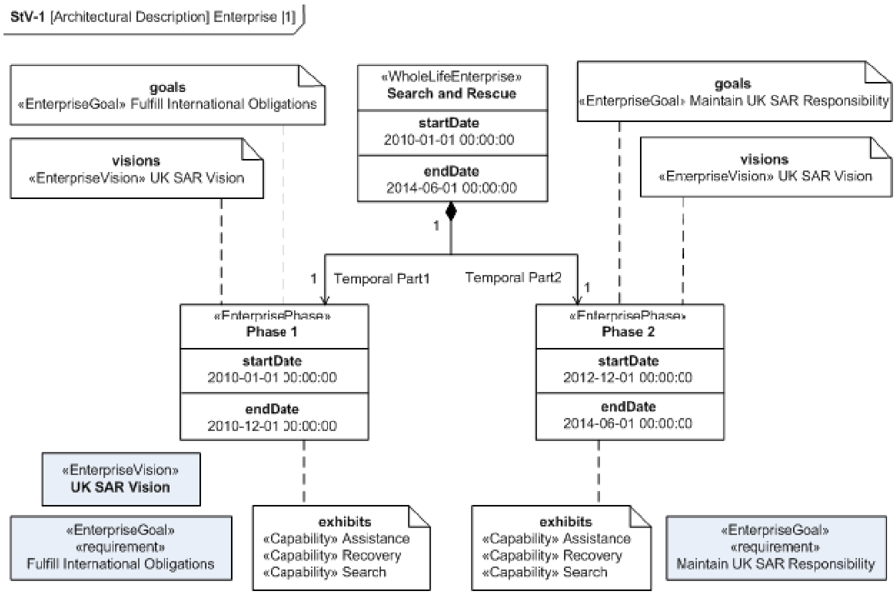


Figure C.5 - StV-1

C.6.2 StV-2 Capability Taxonomy

Capabilities need to be characterized in terms of the properties they need to exhibit, which enables the enterprise to use them to achieve the enterprise goals, as well as their relationships in an inheritance hierarchy. In Figure C.6 we have characterized Maritime SAR in terms of required values. These are defined in Figure C.4 and include the length of a Maritime SAR operation, the sea conditions in which Maritime SAR must be deliverable, the search area covered by an operation, and the time to find a victim.

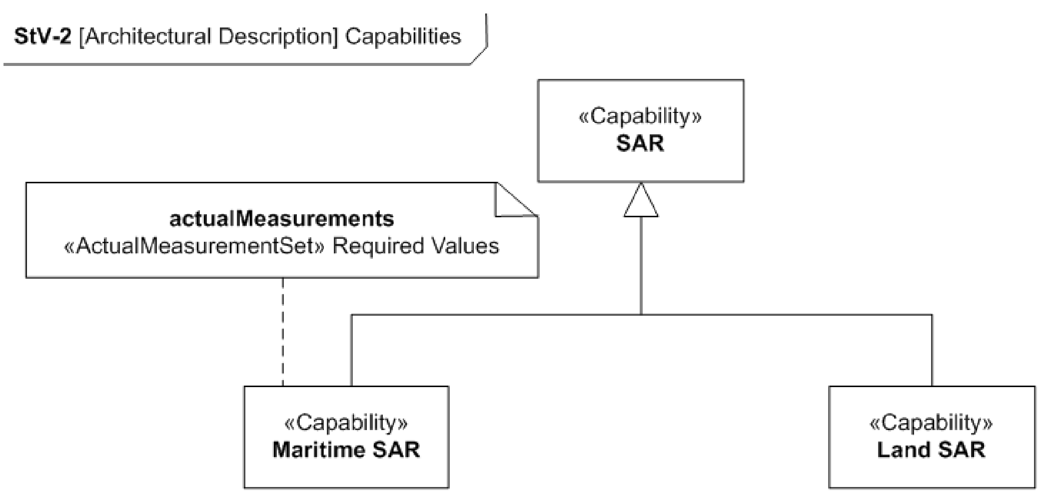


Figure C.6 - StV-2 Capability Taxonomy

C.6.3 StV-4 Capability Clusters

This view, Figure C.7, addresses the logical grouping of capabilities and the dependencies between them. In many circumstances, SAR Command and Control depends on the Military C2 Capability. Similarly, the Assistance, Search, and Recovery Capabilities are dependent upon the SAR C2 Capability, which in turn is dependent upon the Distress Signal Monitoring Capability.

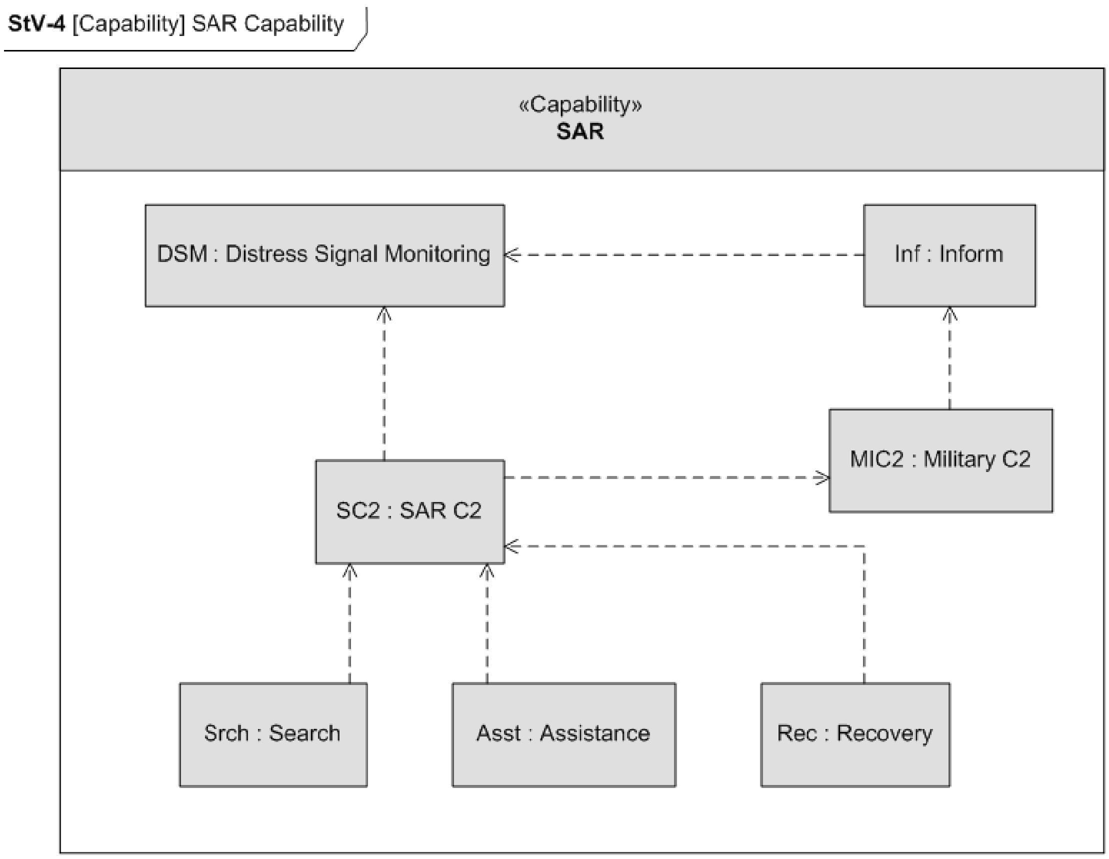


Figure C.7 - StV-4

C.6.4 StV- 4 Capability Clusters Class Diagram

Figure C.8 shows the class diagram version of the capability clusters.

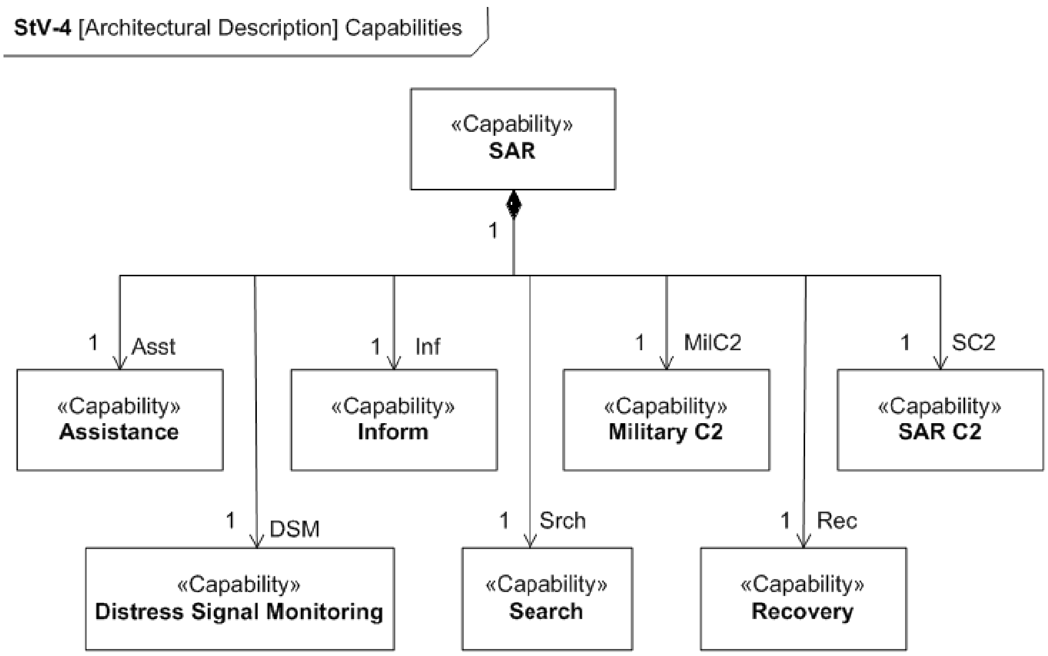


Figure C.8 – StV-4 Alternative View

C.6.5 StV- 5 Capability to Organization Deployment

Figure C.9 shows the generated StV-5 table. The StV-5 defines Capability to Organization Deployment Mapping. It shows the planned capability deployment for a resource and the responsible organization. The StV-5 View is used to support the capability management process and, in particular, assist the planning of fielding. For example, the Assistance Capability is supported by the Maritime Rescue Unit. The RNLI and Maritime and Coastguard Agency are responsible for them.

[Architectural Description] Capabilities [StV-5]

Capabilities

Organizational Resources	«Capability» Assistance	«Capability» Inform	«Capability» Recovery	«Capability» Search
«ActualOrganization» RNLI	Maritime Rescue Unit Maritime Rescue Unit V2	Maritime Rescue Unit Maritime Rescue Unit V2	Maritime Rescue Unit Maritime Rescue Unit V2	Maritime Rescue Unit Maritime Rescue Unit V2
«ActualOrganization» Maritime & Coastguard Agency	Maritime Rescue Unit Maritime Rescue Unit V2	Maritime Rescue Unit Maritime Rescue Unit V2	Maritime Rescue Unit Maritime Rescue Unit V2	Maritime Rescue Unit Maritime Rescue Unit V2

Figure C.9 - StV-5

C.6.6 StV-6 Operational Activity to Capability Mapping

This view, Figure C.10, identifies how operational activities can be performed using various available capability elements. Figure C.10 shows that in order to achieve Search and Assistance Capabilities, certain Standard Operational Activities must be performed, including Monitor Health and Provide Medical Assistance.

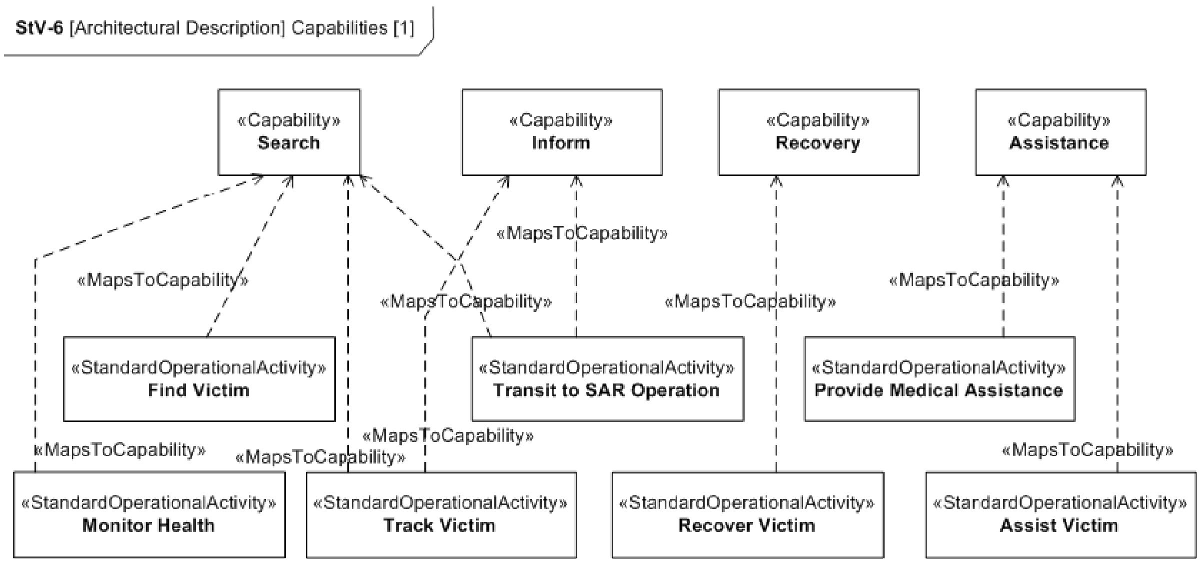


Figure C.10 - StV-6

C.7 Operational Views

The Operational Views identify what needs to be accomplished in the SAR operation and who needs to accomplish it. These views describe the tasks and activities, operational elements, and exchanges of information that are required to conduct the operation.

C.7.1 OV-1 Operational Context Graphic

This diagram, Figure C.11, of the Maritime rescue sets the context by illustrating the search and rescue operation at sea involving a yacht in distress. The diagram shows that the monitoring unit picks up the distress calls of the yacht and sends them to a Command and Control (C2) center, which coordinates the operation among helicopters, a naval ship, and an RNLI lifeboat.

In the OV-1, each model element depicted may include a graphical depiction to help convey its intended meaning. The spatial relationships of the elements on the diagram sometimes convey their relative position, although this is not specifically captured in the semantics. A brief description of the interactions between the elements is provided. It may represent abstract conceptual relationships and will be refined in subsequent diagrams.

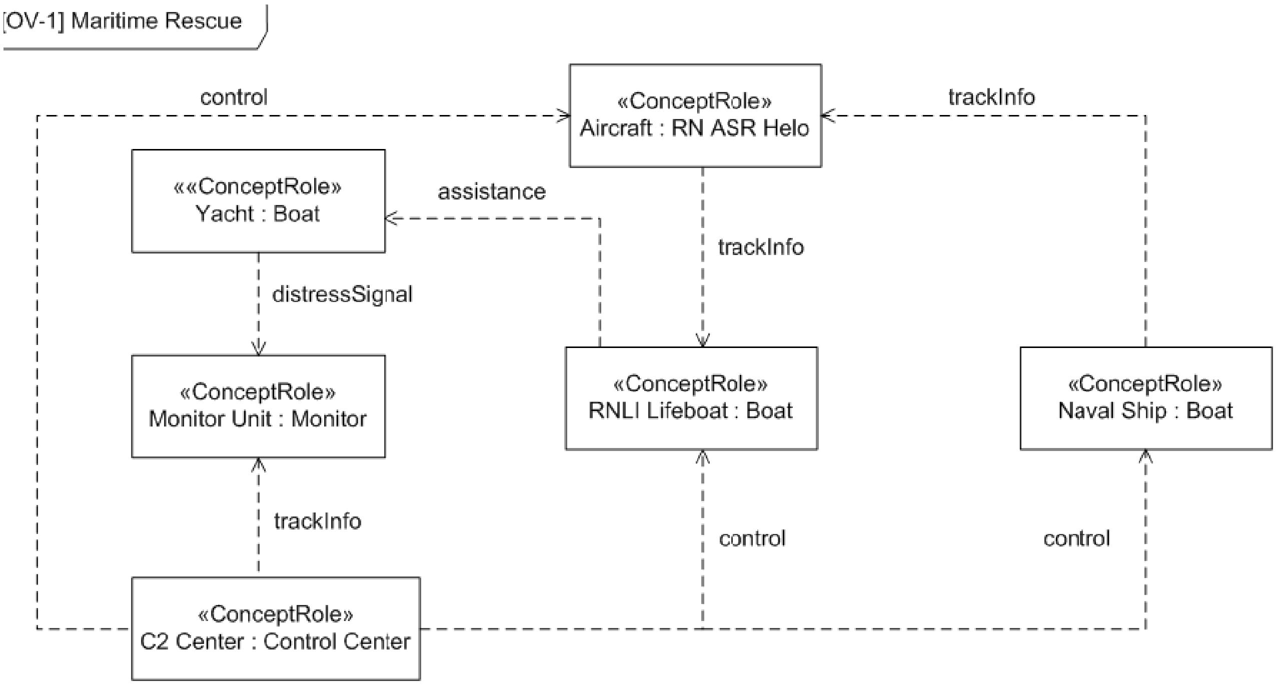


Figure C.11 - OV-1

As shown below, a pictorial background can be included to provide additional context.

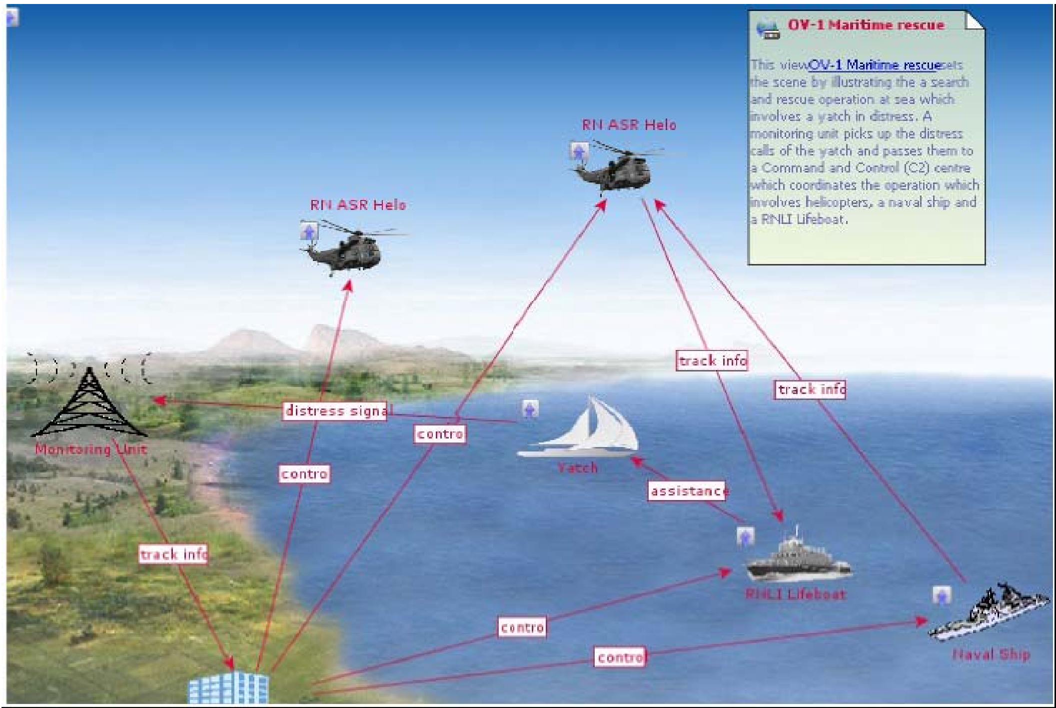


Figure C.12 - Alternate OV-1

C.7.2 OV-2 Operational Node Connectivity Description

Figure C.13 depicts the key players in the SAR operation and the interactions for information exchange. It identifies the different types of nodes in the SAR operation: Person in Distress, Monitoring Node, Tactical C2 Node, SAR Asset Controller, Search Node, Rescue Node, and Place of Safety. This diagram indicates the need to exchange information between the operational nodes and also shows the interactions between these nodes. The OV-5 view shows the operational activities undertaken by a few select nodes. Figure C.13 is the class diagram version of the OV-2.

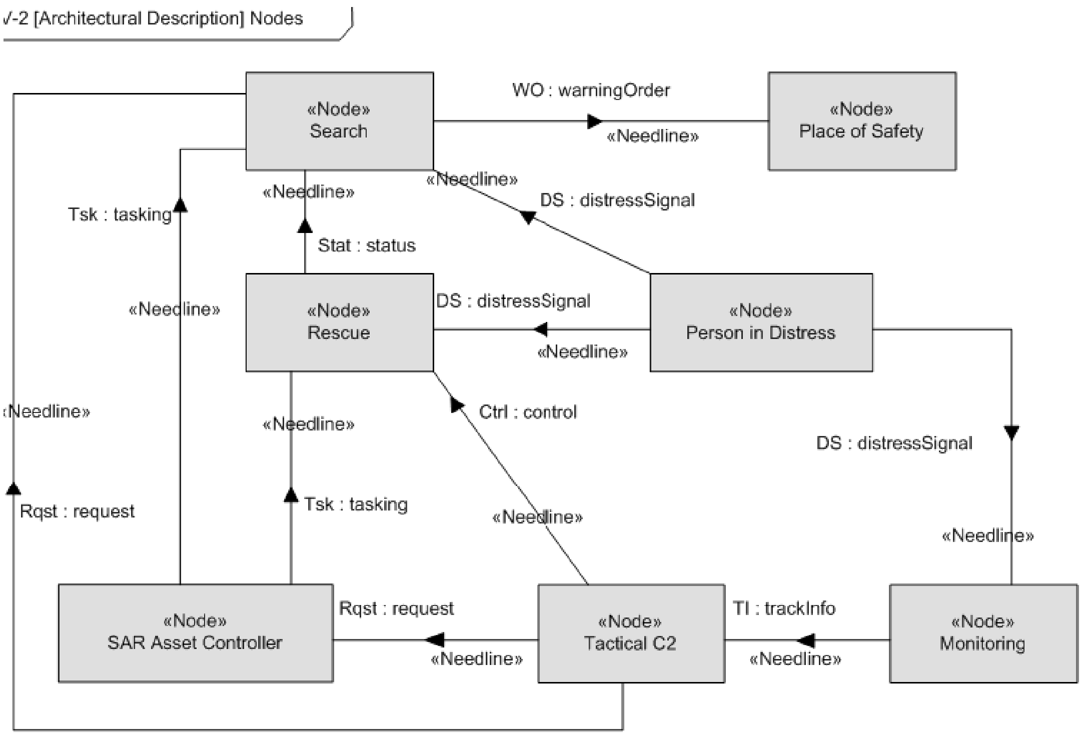


Figure C.13 - OV-2

Figure C.14 shows an alternate way to display the OV-2. It can be illustrated as above with IO Flows directed relationships or as below using connectors and SysML Item Flows with or without flow ports as in Figure C.14 or with flow ports as in Figure C.15.

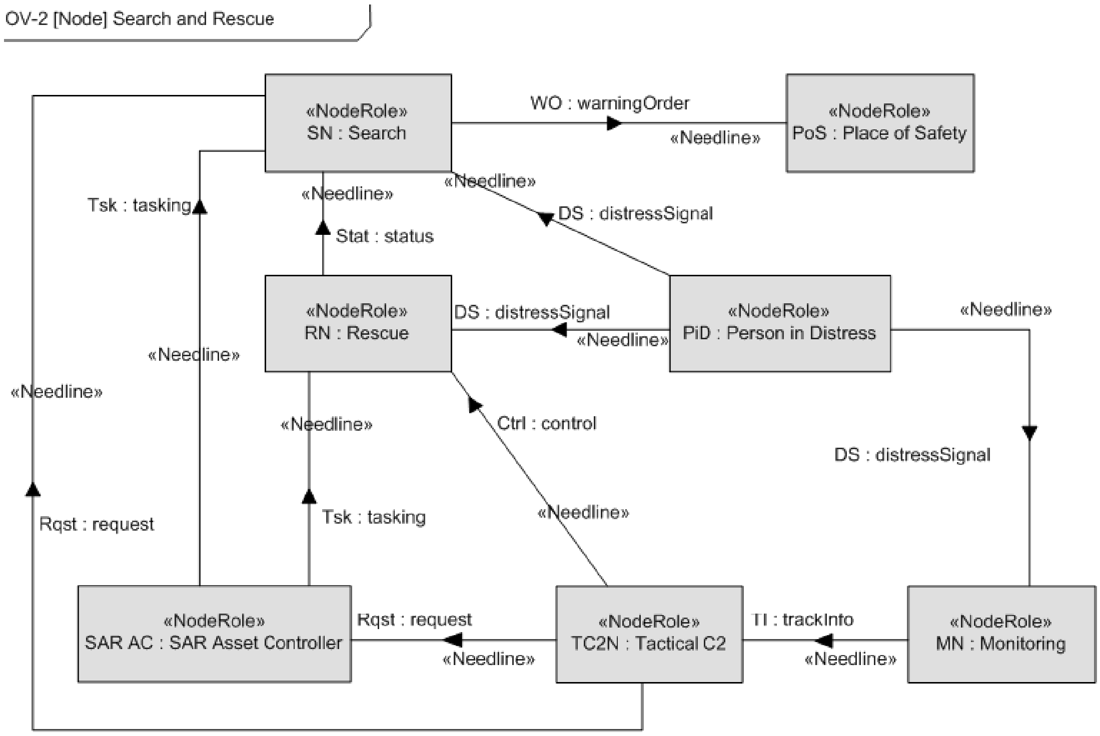


Figure C.14 - Alternate OV-2 SysML Version Without Flow Ports

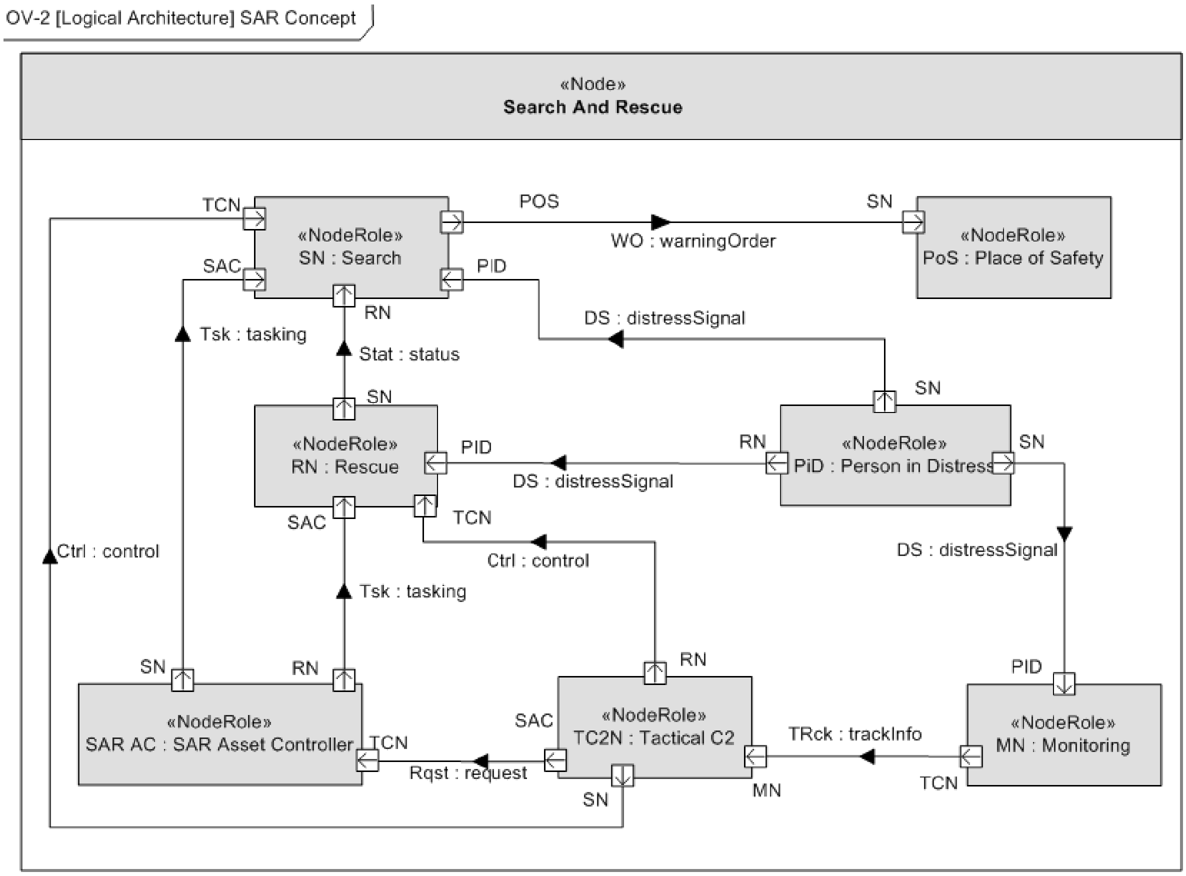


Figure C.15 - Alternate OV-2 with SysML Flow Ports

Figure C.15 shows the SysML version with Flow Ports and Item Flows. The typed ports mean that the user can constrain the elements that can flow in and out of the port. This means that consistency checks can be performed on the ports to ensure that the flows correspond to the allowed elements. The stereotypes have also been removed to aid readability.

C.7.3 OV-3 Operational Information Exchange

Table C.3 shows the operational information exchanges between nodes. The OV-3 can include InformationExchanges associated with a Needline as well as Information Elements carried by one or more InformationExchange.

Table C.3 - OV-3

[Architectural Description] Structure [OV-3]

Information Exchange		Producer		Needline	Consumer	
Name	Conveyed	Node	Operational Activity	Name	Node	Operational Activity
Ctrl	«Information Element» control	«Node» Tactical C2		TC2N - RN	«Node» Rescue	
Ctrl	«Information Element» control	«Node» Tactical C2		TC2N - SN	«Node» Search	
DS	«Information Element» distressSignal	«Node» Person in Distress	Send Distress Signal	PiD – MN	«Node» Monitoring	Receive Distress Signal
DS	«Information Element» distressSignal	«Node» Person in Distress	Send Distress Signal	PiD – SN	«Node» Search	Receive Distress Signal
DS	«Information Element» distressSignal	«Node» Person in Distress	Send Distress Signal	PiD - RN	«Node» Rescue	Receive Distress Signal
Rqst	«Information Element» request	«Node» Tactical C2		TC2N - SAR AC	«Node» SAR Asset Controller	
Stat	«Information Element» status	«Node» Search	Monitor Health	SN - RN	«Node» Rescue	Provide Medical Assistance
TI	«Information Element» trackInfo	«Node» Monitoring		MN - TC2N	«Node» Tactical C2	
Tsk	«Information Element» tasking	«Node» SAR Asset Controller		SAR AC - RN	«Node» Rescue	
Tsk	«Information Element» tasking	«Node» SAR Asset Controller		SAR AC - SN	«Node» Search	
WO	«Information Element» warningOrder	«Node» Search	Send Warning Order	SN - PoS	«Node» Place of Safety	Process Warning Order

C.7.4 OV-4 Organizational Relationships Chart

Figure C.16 illustrates the command structure or relationships (as opposed to relationships with respect to a business process flow) among human roles, organizations, or organization types that are the key players in the SAR operation.

The OV-4 exists in two forms - typical (typical command structure) and actual (organization chart for a department or agency). Figure C.16, the typical OV-4, shows the possible relationships between organizations and posts. It is also possible to define types of people who are capable of filling these posts. For example, a Qualified Lifeguard could become an MRT Swimmer.

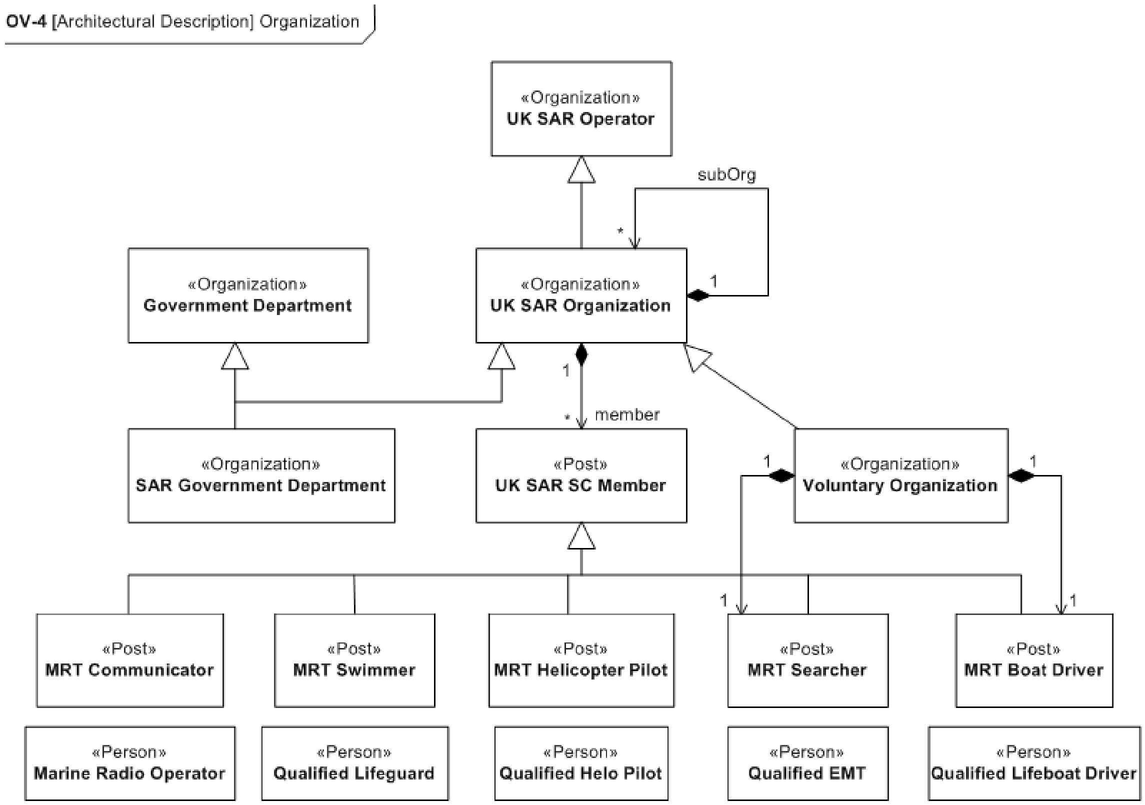


Figure C.16 - OV-4 Typical

The actual OV-4, shown in Figure C.17, depicts the structure of the organization, the actual posts, and the actual persons who fill those posts. The diagram can also be annotated with the start and end dates for this.

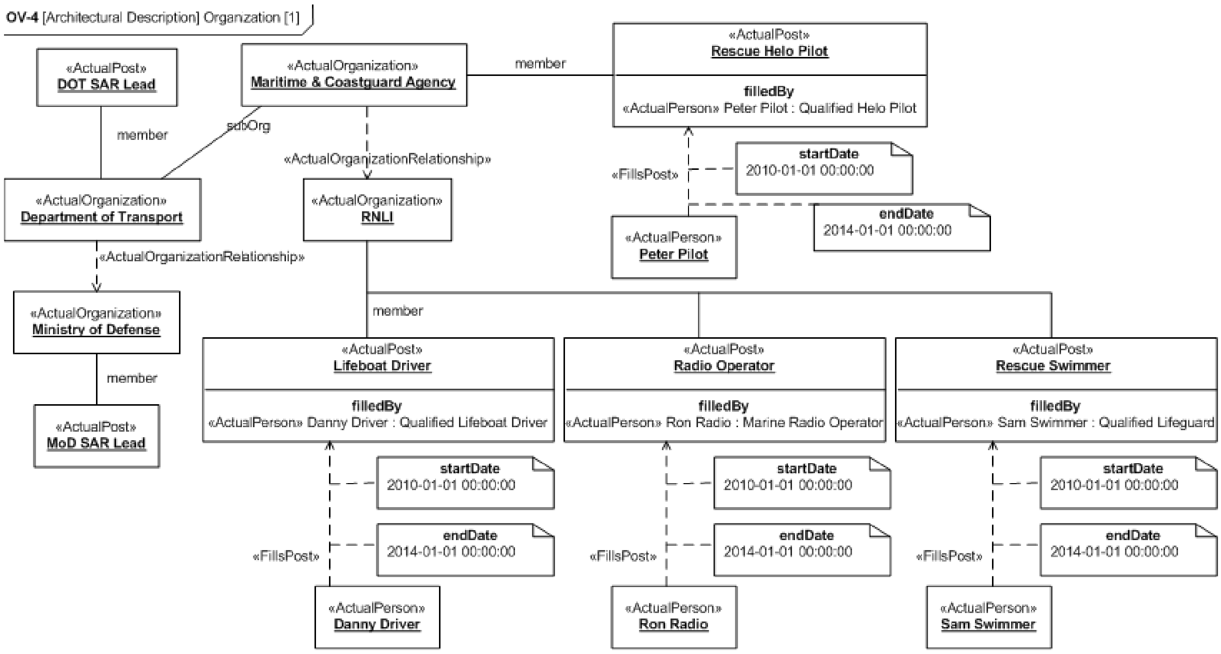


Figure C.17 - OV-4 Actual

C.7.5 OV-5 Operational Activity Model

Figure C.18 describes the operations that are normally conducted in the different nodes of a Search and Rescue operation. This view shows the operational activities which are performed in the Search Node and Rescue Node. This view shows which nodes undertake the operations.

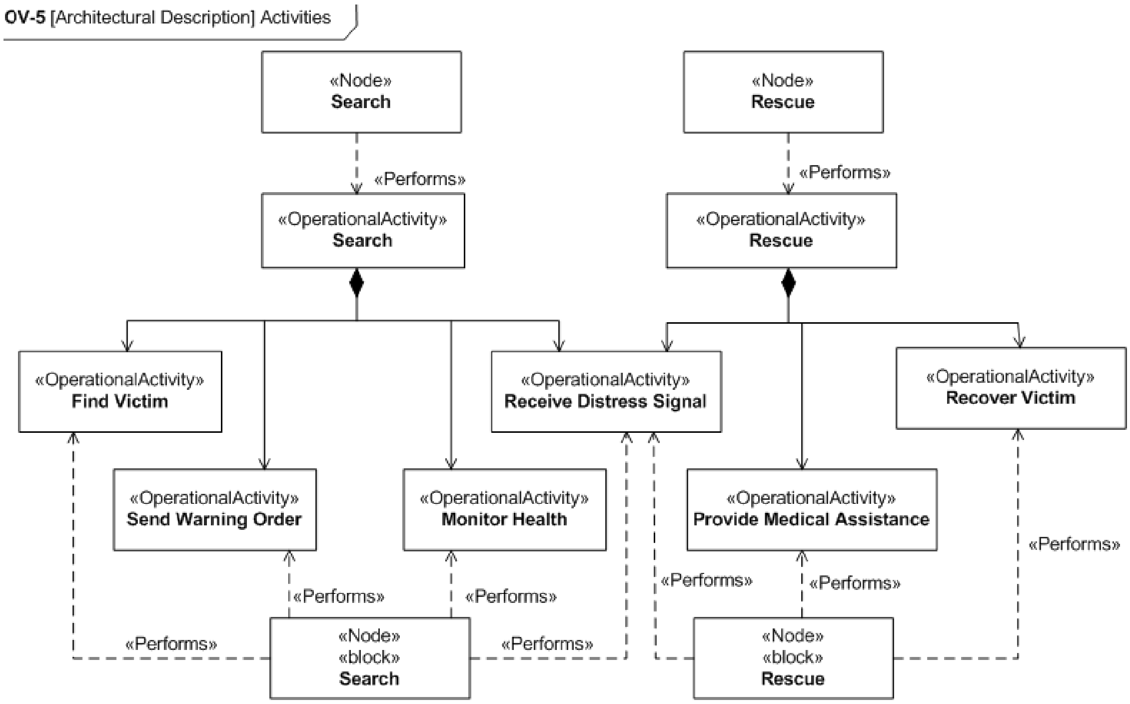


Figure C.18 - OV-5

Figure C.19 shows the OV-5 as an Activity Diagram. It describes Operational Activity Actions, Input/Output flows between activities, and to/from activities that are outside the scope of the Architecture.

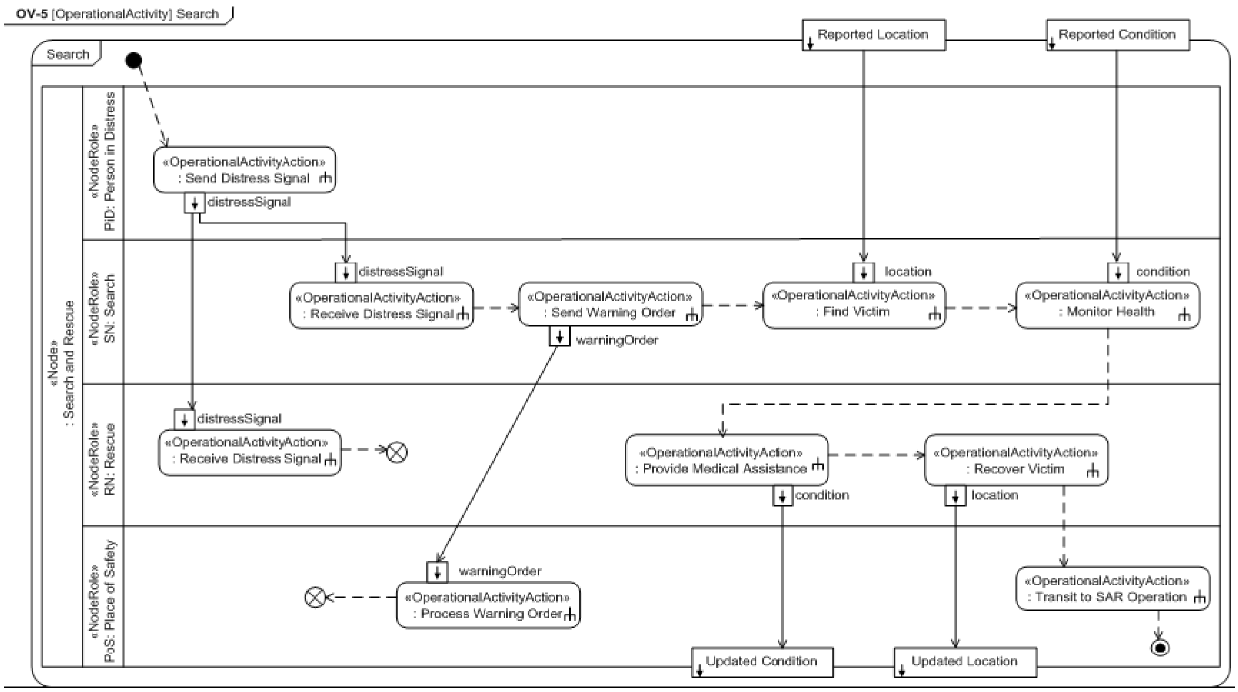


Figure C.19 - Alternate OV-5

C.7.6 OV- 6 Operational States

Figure C.20 describes the operational states of the Search Node, the behaviors that take place within those states, the transitions between the states and the events and guards that cause those transitions to take place. For example, the search node is waiting for a distress signal. When one is received, the warning order is sent out and the search node transitions to searching for victim.

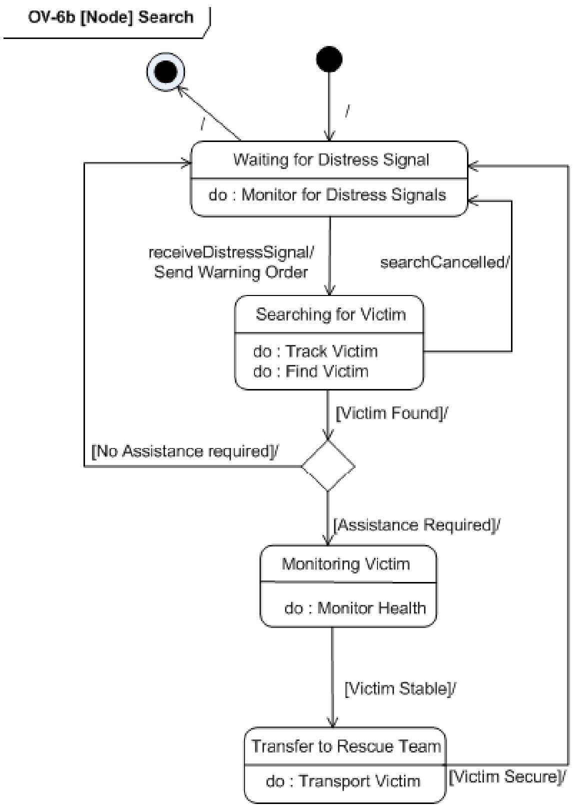


Figure C.20 - OV-6b

C.7.7 OV-7 Logical Data Model

This view describes the information that is associated with the Warning Order information element. The boxes show the information items and the lines represent their inter-relationships. Attributes can be used to show the characteristics of the information items. The call-out notes show the information elements that represent the entity items. These are used on the OV-2 and other diagrams.

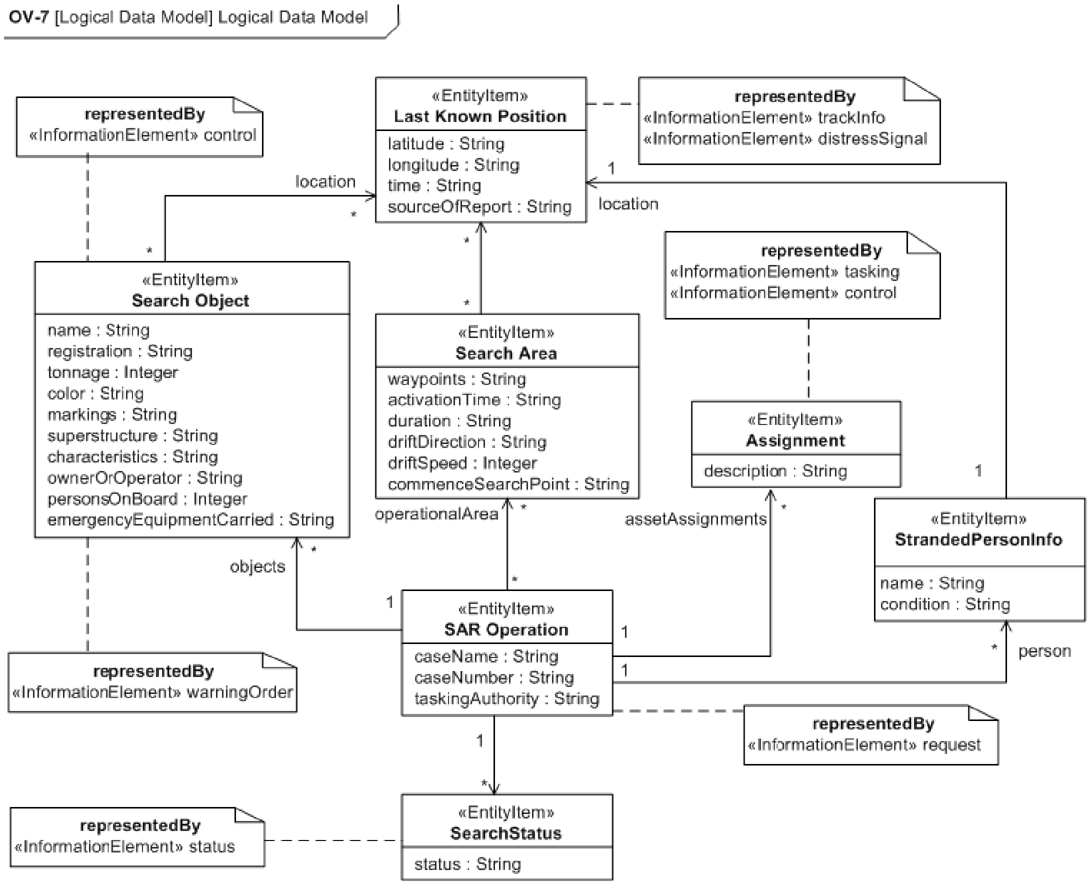


Figure C.21 - OV-7

C.8 Service Oriented Views

The Service Oriented views describe the services needed to directly support the Search and Rescue operations described in the Operational View. The role of the services is to support the Operational Activities.

C.8.1 SOV-1 Service Taxonomy

This view specifies the hierarchy of services as well as the relationships between them. Figure C.22 shows the hierarchy of services within the Search and Rescue Service with Land and Maritime Search and Rescue Services as specializations of the SAR Service.

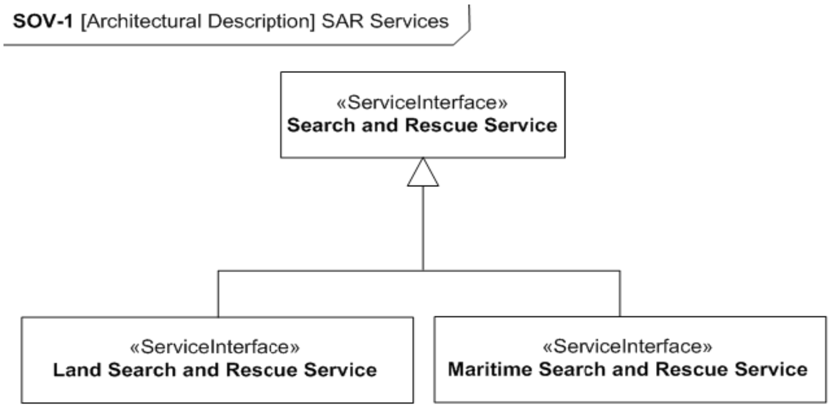


Figure C.22 - SOV-1

C.8.2 SOV-2 Service Interface Specification

Figure C.23 defines the interfaces that will provide access to the services.

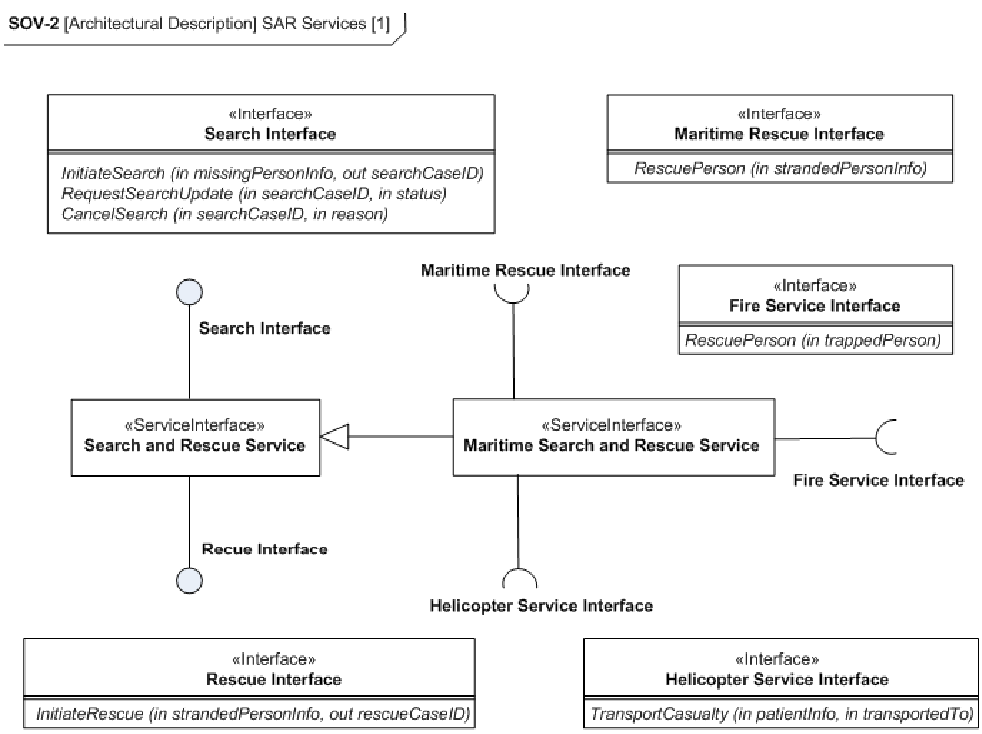


Figure C.23 - SOV-2

C.8.3 SOV-3 Capability to Service Mapping

Figure C.24 shows which services contribute to the achievement of a capability. In this example, the Land Search and Rescue Service aims to achieve the Land SAR Capability. Likewise, the Maritime Search and Rescue Service aims to achieve the Maritime SAR.

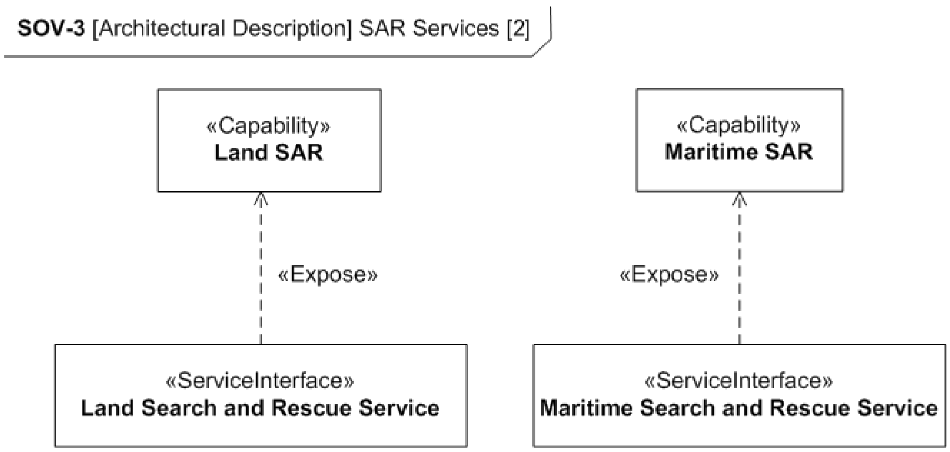


Figure C.24 - SOV-3

As shown below, the SOV-3 can also be presented as a matrix with capabilities on one axis and services on the other.

Table C.4 - Alternate SOV-3

	«Capability» **Land SAR**	«Capability» **Maritime SAR**
«ServiceInterface» **Land Search and Rescue Service**	**X**	
«ServiceInterface» **MaritimeSearchand RescueService**		**X**

C.8.4 SOV-4 Service Behaviors and Constraints

Figure C.25 defines constraints that must be adhered to by Consumers and Providers of the Services via Service Policies.

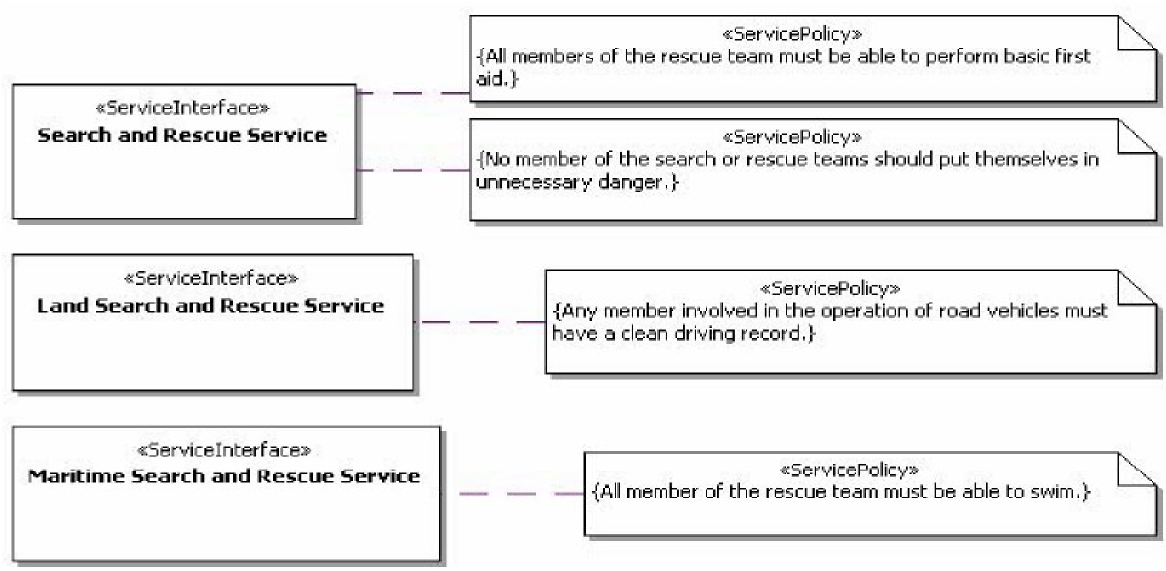


Figure C.25 - SOV-4

Table C.5 - Alternate SOV-4a

[Architectural Description] SAR Services [SOV-4a]

Service Interface	**Service Policy**	
Name	**Name**	**Text**
Land Search and Rescue Service	Driving Record	Any member involved in the operation of road vehicles must have a clean driving record.
Maritime Search and Rescue Service	Swim	All members of the rescue team must be able to swim.
Search and Rescue Service	First Aid	All members of the rescue team must be able to perform basic first aid.
	Danger	No member of the search and rescue team should put themselves in unnecessary danger.

Table C.5 is an alternate representation.

C.8.5 SOV-5 Service Functionality

Figure C.26 defines the Service Functions to describe the abstract behavior of each Service Operation. It specifies the set of functions that the service implementation is expected to perform.

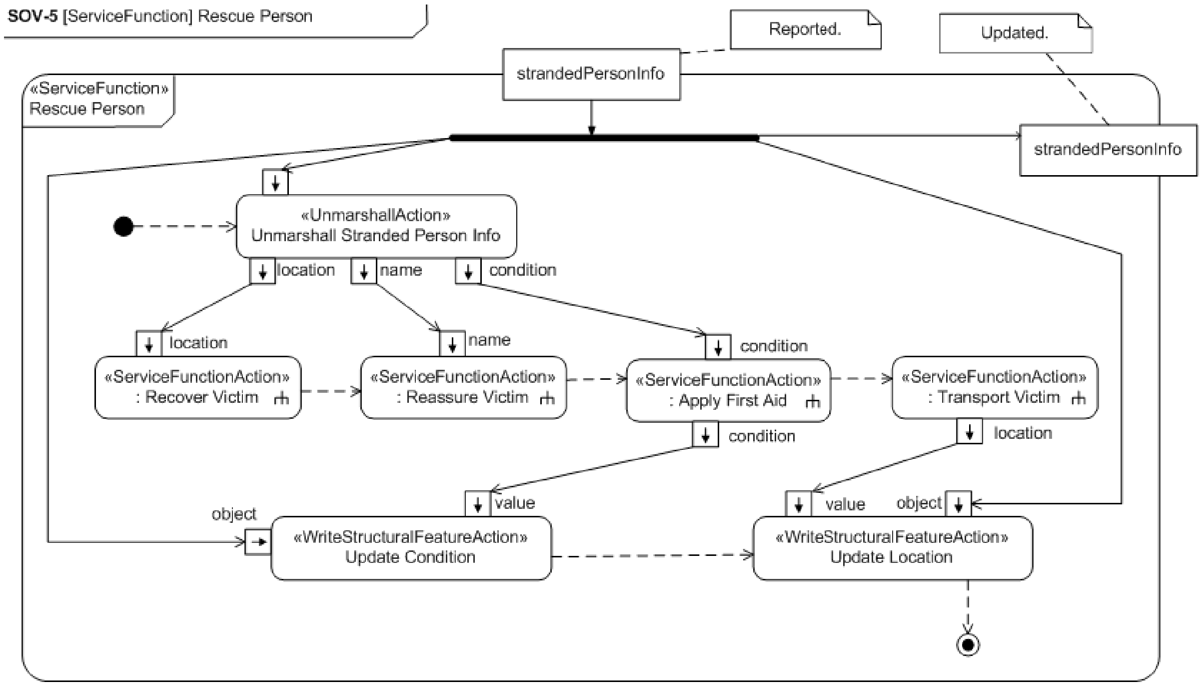


Figure C.26 - SOV-5

C.9 Systems Views

These views describe the resources that realize the SAR capabilities. They describe resource functions, interactions between resources, and can provide detailed system interface models.

C.9.1 SV-1 Resource Interaction Specification

This view defines the structure and internal flows of the Capability Configuration. Figure C.27 shows the Capability Configuration of a Maritime Rescue Unit. The Maritime Rescue Unit is comprised of the Maritime Rescue Team (MRT), and the roles that make up the MRT, as well as the components that enable them to fulfill their role. This example shows that the Role of Driver is filled by an MRT Member who must interact with an MR Boat.

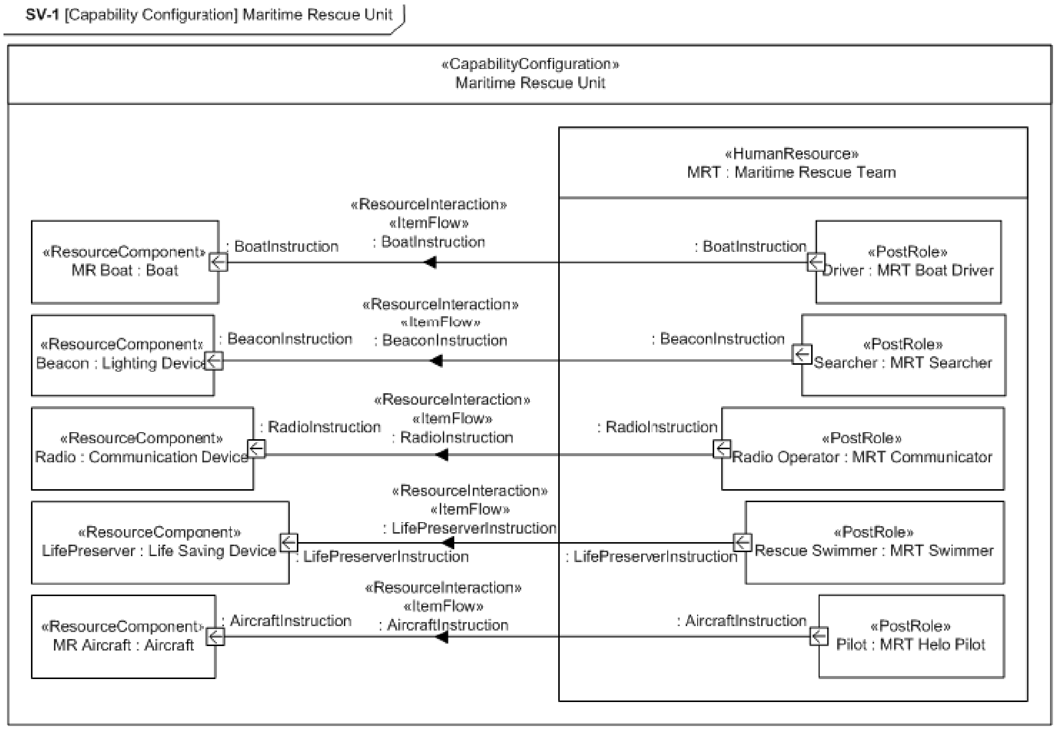


Figure C.27 - SV-1

C.9.2 SV-2 Systems Communications Description

Figure C.28 shows systems interconnections for a number of entities in a maritime search and rescue scenario.

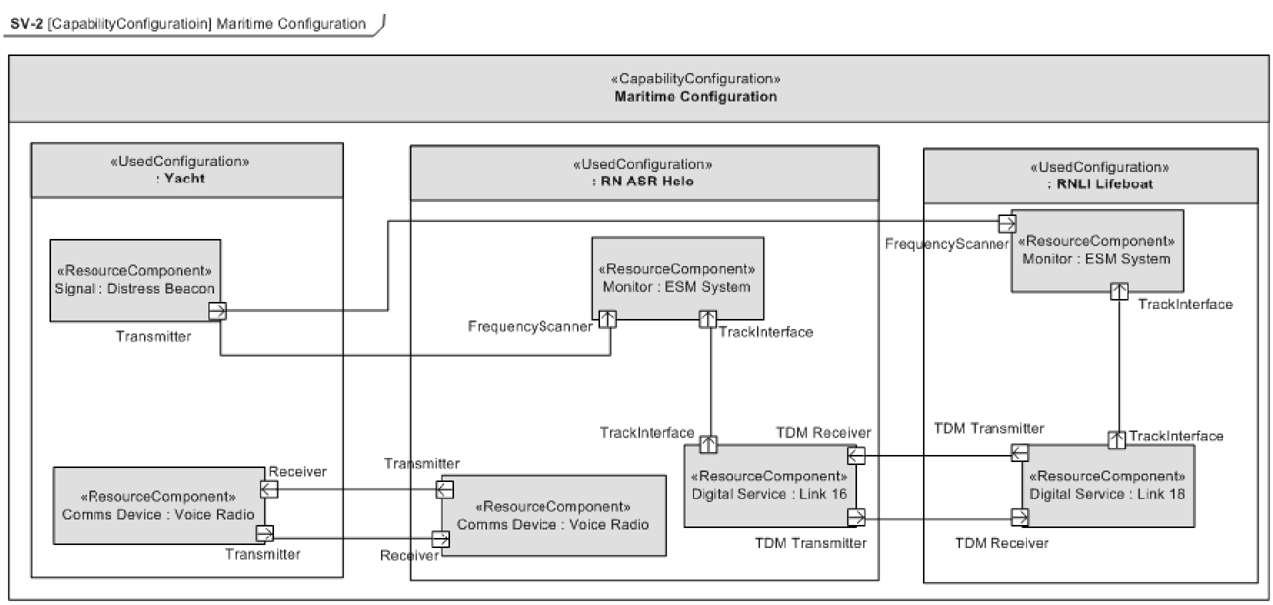


Figure C.28 - SV-2

C.9.3 SV-4 Functionality Description

This view uses Functions to describe Resources. Figure C.29 is a mapping of resource usage to function. It shows that the Maritime Rescue Team, which is part of the Maritime Rescue Unit, consists of two Roles - Radio Operator and Rescue Swimmer, both of which are filled by MRT Members. The Rescue Swimmer performs the Function of Recover Victim. The Radio Operator performs the functions of Send Message, Receive Message, and Broadcast Message.

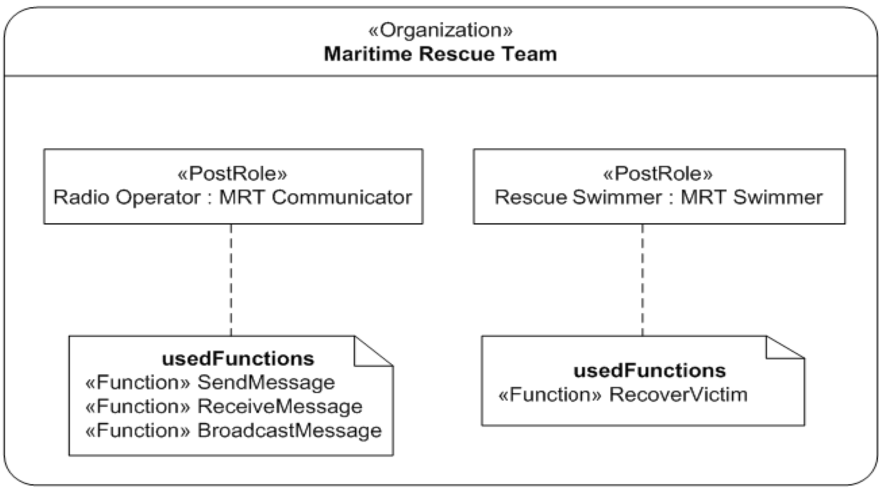


Figure C.29 - SV-4

The below diagram describes the Resources using Functions. It shows the operational step-by-step workflows and the overall flow of control.

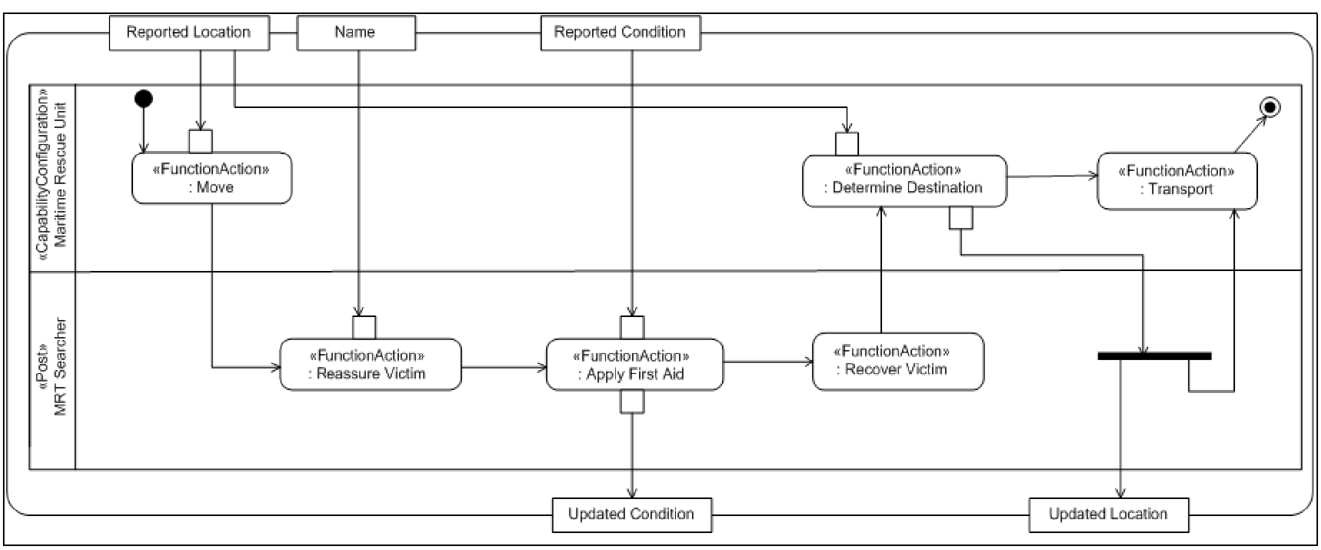


Figure C.30 - SV-4 Activity Diagram

C.9.4 SV-7 Resource Performance Parameters

This view defines the types of measurements that are important to the Capabilities. It consists of measurable, qualitative properties. Figure C.31 shows the Capability Configurations that are linked to the various measurements.

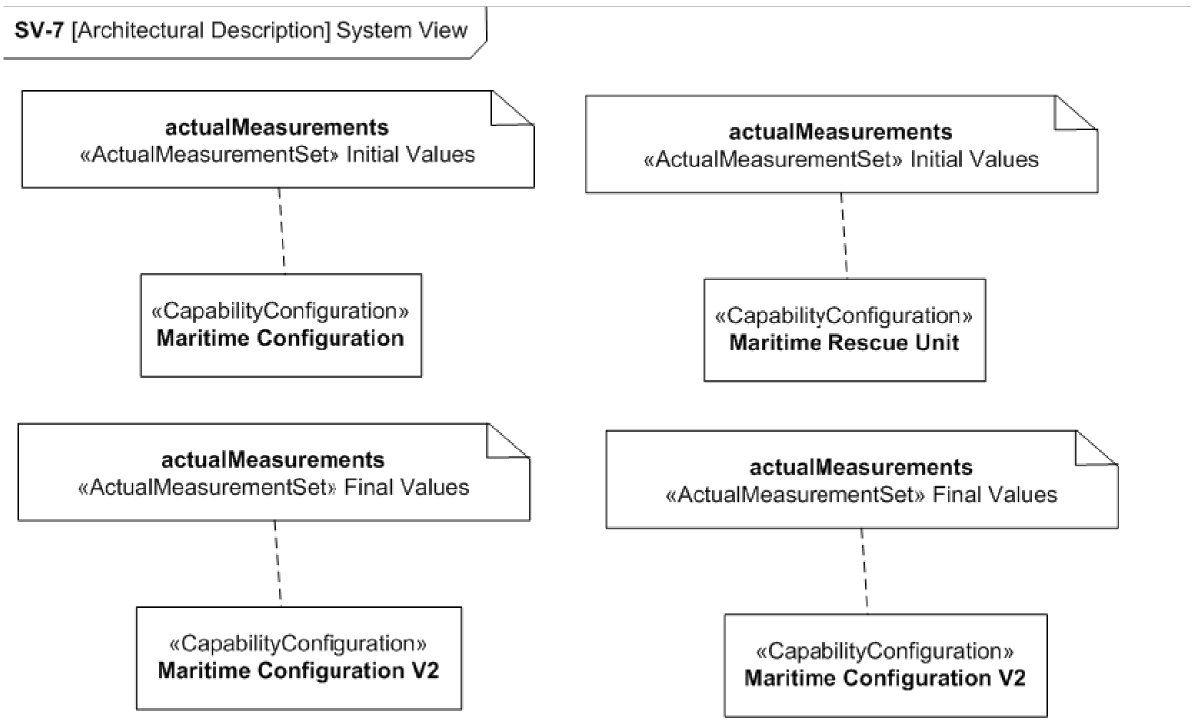


Figure C.31 - SV-7

As shown below, the SV-7 can also be in tabular format, specifying qualitative and quantitative characteristics of resources.

Table C.6 - Alternate SV-7

[Architectural Description] System View [SV-7]

Resource		Actual Measurement Set				
Type	**Name**	**Name**	**Measurement**	**Actual Value**	**Unit**	**Dimension**
«CapabilityConfiguration»	Maritime Configuration	Initial Values	seaConditions	Sea State 6	Meter	Wave Height
			areaCoverage	500	SquareKilometers	Area
			findTime	<8	Hours	Time
			persistence	>15	Hours	Time
			searchCoverage	400	SquareKilometers	Area
			weatherConditions	Heavy Rain	Weather Severity Index	
«CapabilityConfiguration»	Maritime Configuration V2	Final Values	seaConditions	sea state 8	Meter	Wave Height
			areaCoverage	650	SquareKilometers	Area
			findTime	<4	Hours	Time
			persistence	>20	Hours	Time
			searchCoverage	550	SquareKilometers	Area
			weatherConditions	Stormy	Weather Severity Index	
«CapabilityConfiguration»	Maritime Rescue Unit	Initial Values	seaConditions	Sea State 6	Meter	Wave Height
			areaCoverage	500	SquareKilometers	Area
			findTime	<8	Hours	Time
			persistence	>15	Hours	Time
			searchCoverage	400	SquareKilometers	Area
			weatherConditions	Heavy Rain	Weather Severity Index	
«CapabilityConfiguration»	Maritime Rescue Unit V2	Final Values	seaConditions	sea state 8	Meter	Wave Height
			areaCoverage	650	SquareKilometers	Area
			findTime	<4	Hours	Time
			persistence	>20	Hours	Time
			searchCoverage	550	SquareKilometers	Area
			weatherConditions	Stormy	Weather Severity Index	

C.10 Acquisition Views

The Acquisition views identify top-level tasks in the acquisition process. They help you understand how resources, assets and capabilities are acquired during the life of the project. It gives you the ability to perform analysis to determine if the resources can be obtained, if they are available in the time they are needed, and the overall effect on the schedule.

C.10.1 AcV-1 System of Systems Acquisition Clusters

The AcV- 1 represents an organizational perspective of the program. It allows the user to model the organizational structures needed to manage a portfolio of projects. The below diagram shows who is responsible for the SAR Project, as well as the project type.

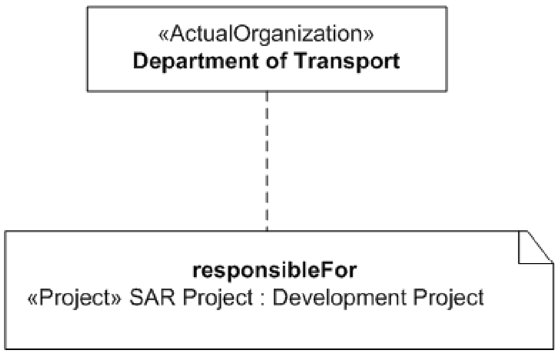


Figure C.32 - AcV-1

C.10.2 AcV-1 Project Definitions

The AcV-1 class diagram provides a means of defining projects and project types. In this case, the development project can contain other Development projects contain milestones containing project themes corresponding to DOTMLPF themes.

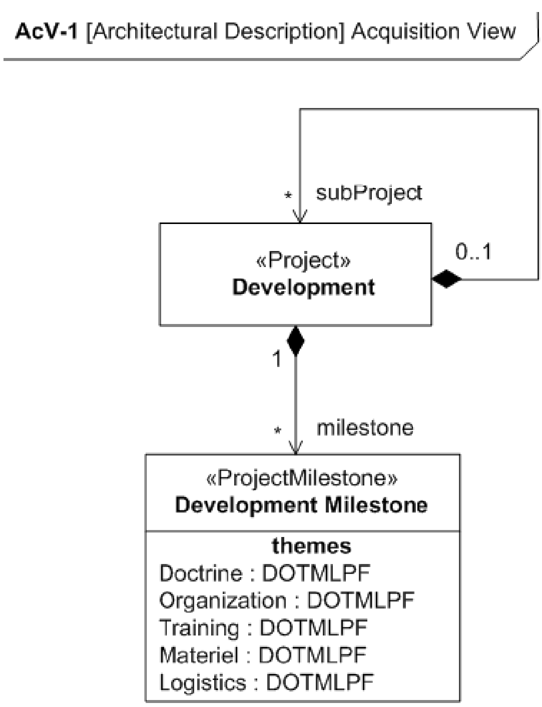


Figure C.33 - AcV-1 Class Diagram

C.10.3 AcV-1 Project Instance

The AcV-1 provides a means of defining actual projects and actual project milestones. The SAR project and Flood Response projects are both part of an Emergency Response Enhancement project. The SAR project milestones are also shown.

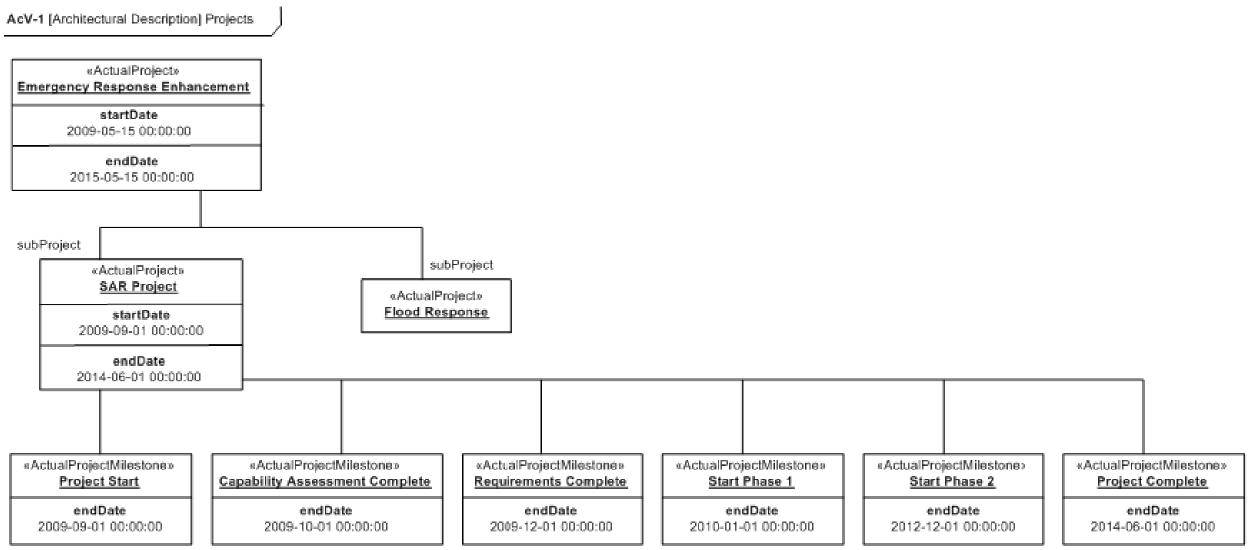


Figure C.34 - AcV-1 Typical

The project also contains increment and deployment milestones that provide a means of showing when resources are deployed and rendered out of service as well as capability increments, as shown in Figure C.35.

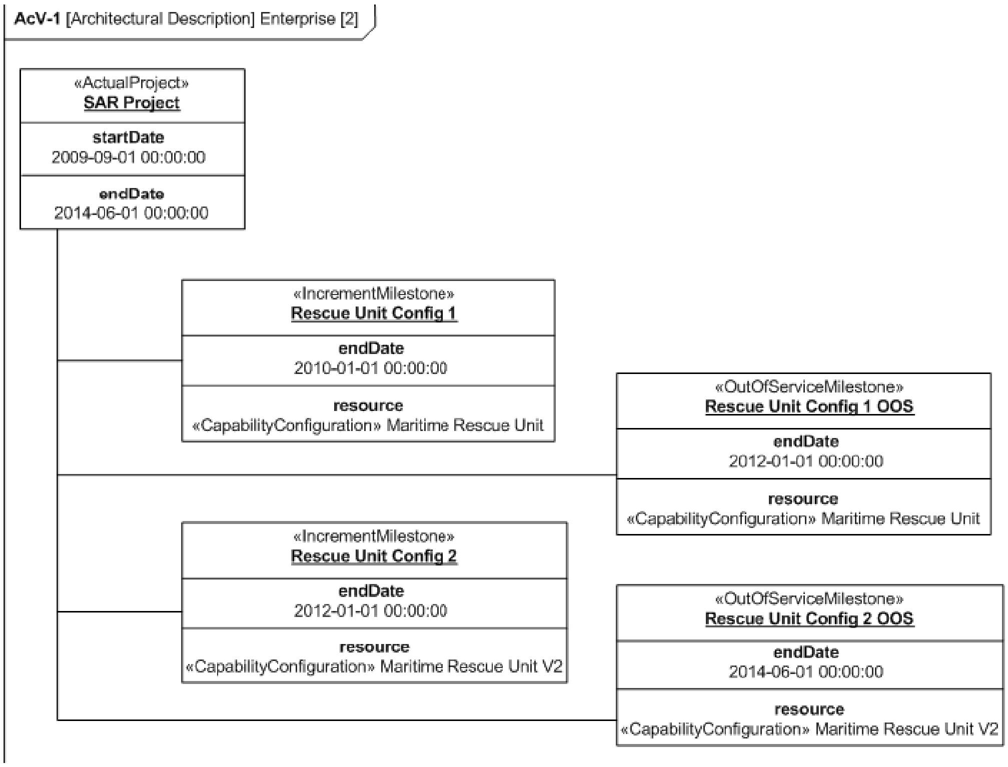


Figure C.35 – AcV-1 Additional Milestone Types

C.11 A Simple Example of SysML Parametrics

C.11.1 Definition of SysML Typedefs

The SysML Block Definition Diagram (BDD) provides a means of defining value types, units and dimensions, as shown below. The value types are the types for the measurements in Figure C.4. The units and dimensions are displayed in Table 6. This provides a clearer definition of different values being measured as opposed to real or float definitions.

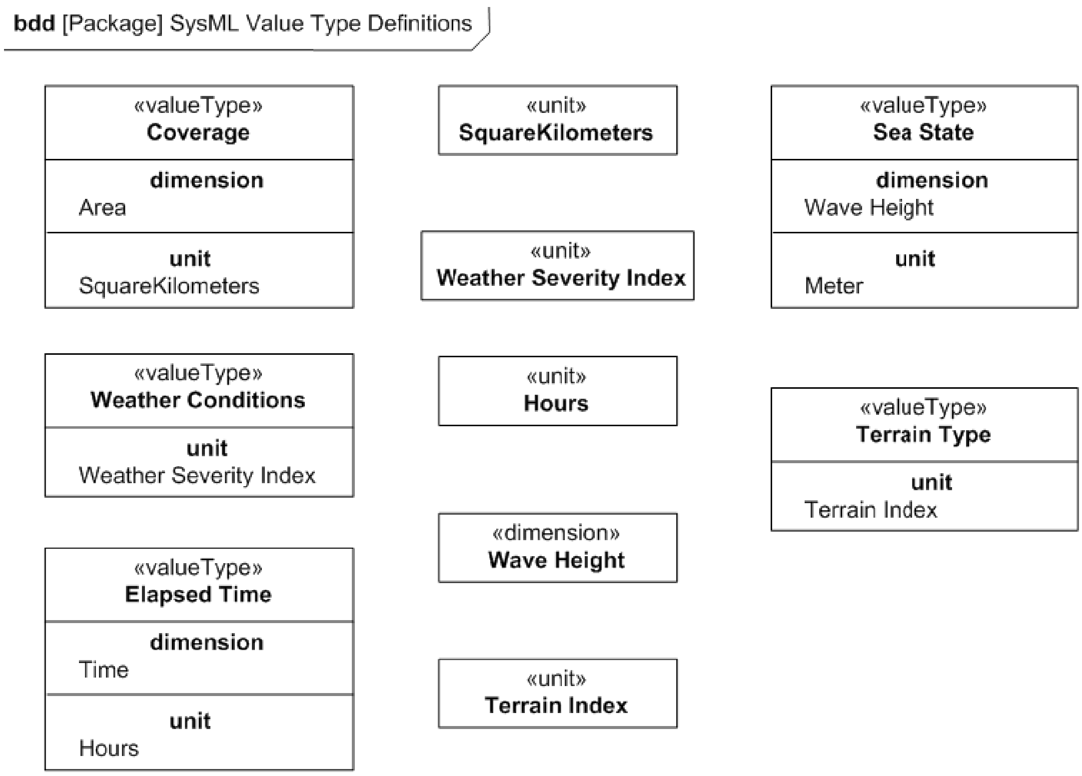


Figure C.36 – Block Definition Diagram

C.11.2 SysML Parametrics

The parametric diagram represents constraints on system parameter values such as performance, reliability and mass properties to support engineering analysis. The Parametric Diagram is a specialized variant of an internal block diagram that restricts diagram elements to represent constraint blocks, their parameters and the block properties that they bind to. An example of the parametric diagram is shown in Figure C.37.

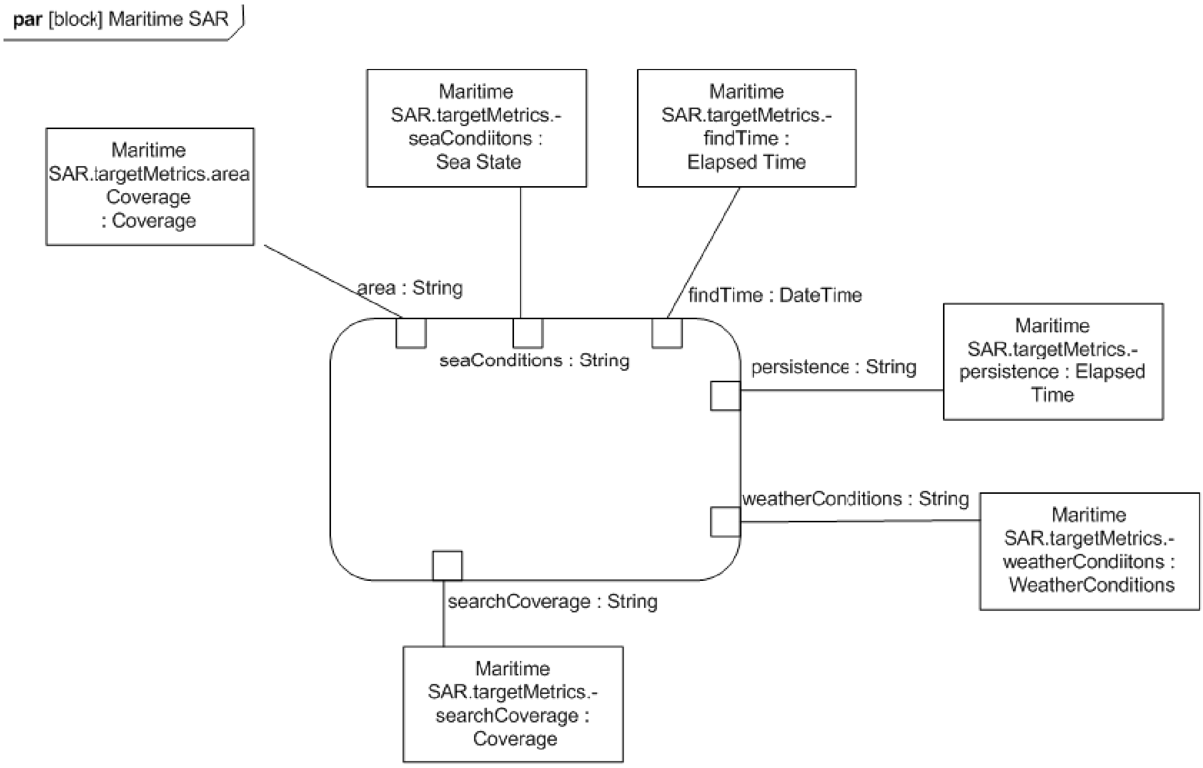


Figure C.37 - SysML Parametric Diagram

Annex D - Bibliography

MOD Architectural Framework, Version 1.2, 23 June 2008, Office of Public Sector Information, http:// www.modaf.org.uk/

DoD Architecture Framework Version 1.5, Volume I: Definitions and Guidelines, 23 April 2007, DoD Architecture Framework Working Group

DoD Architecture Framework Version 1.5, Volume II: Product Descriptions, 23 April 2007, DoD Architecture Framework Working Group

DoD Architecture Framework Version 1.5, Volume III: Product Descriptions, 23 April 2007, DoD Architecture Framework Working Group

DoD Dictionary of Military Terms, http://www.dtic.mil/doctrine/jel/doddict/index.html

Joint Doctrine Joint Force Employment Briefing, http://www.dtic.mil/doctrine/jrm/plans.pdf

Joint Forces Staff College, Joint Planning and Operations Course, http://www.jfsc.ndu.edu/schools_programs/jpoc course_materials/default.asp

Doctrine for Joint Operations, http://www.dtic.mil/doctrine/jel/new_pubs/jp3_0.pdf

Joint Doctrine for countering air and missile threats, http://www.dtic.mil/doctrine/jel/new_pubs/jp3_01 .pdf

Joint Communications System, http://www.dtic.mil/doctrine/j el/new_pubs/jp6_0.pdf

Federal Information Processing Standards Publication (FIPS PUB) 183 -- INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0), issued by the National Institute of Standards and Technology after approval by the Secretary of Commerce pursuant to Section 111(d) of the Federal Property and Administrative Services Act of 1949 as amended by the Computer Security Act of 1987, Public Law 100-235.

INDEX