Unified Architecture Framework (UAF) Domain Metamodel

Version 1.1

OMG Document Number: formal/19-11-05
Release Date: April 2020
Standard document URL: https://www.omg.org/spec/UAF/1.1
Normative Machine Consumable File(s):
https://www.omg.org/spec/UAF/20190615/UAF.xmi
https://www.omg.org/spec/UAF/20190615/Measurements_Library.xmi
USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any
DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS


For a complete list of trademarks, see: https://www.omg.org/legal/tm_list.htm. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.
**Table of Contents**

1. **SCOPE** .......................................................................................................................................................... 3
   1.1 INTRODUCTION ........................................................................................................................................... 3
   1.2 UAF BACKGROUND ..................................................................................................................................... 4
   1.3 INTENDED USAGE ....................................................................................................................................... 4
   1.4 RELATED DOCUMENTS .............................................................................................................................. 4

2. **CONFORMANCE** ......................................................................................................................................... 5

3. **REFERENCES** ................................................................................................................................................ 6
   3.1 NORMATIVE REFERENCES .......................................................................................................................... 6
   3.2 OMG DOCUMENTS (NORMATIVE REFERENCES) .......................................................................................... 6
   3.3 OTHER NORMATIVE REFERENCES ............................................................................................................. 6
   3.4 INFORMATIVE REFERENCES ...................................................................................................................... 7

4. **TERMS AND DEFINITIONS** .......................................................................................................................... 8

5. **SYMBOLS** ................................................................................................................................................... 9

6. **ADDITIONAL INFORMATION** ....................................................................................................................... 11
   6.1 CHANGES TO ADOPTED OMG SPECIFICATIONS ......................................................................................... 11
   6.2 LANGUAGE ARCHITECTURE ........................................................................................................................ 11
   6.3 PHILOSOPHY .............................................................................................................................................. 11
   6.4 CORE PRINCIPLES ....................................................................................................................................... 11

7. **UAF GRID** ................................................................................................................................................... 13
   7.1 DESCRIPTIONS OF DOMAINS AND MODEL KINDS ..................................................................................... 15
   7.2 DOMAIN INTERRELATIONSHIPS ................................................................................................................ 17

8. **DOMAIN METAMODEL DIAGRAM LEGEND** .............................................................................................. 17

9. **DOMAIN METAMODEL DIAGRAMS** ........................................................................................................... 19
   9.1 VIEW SPECIFICATIONS ............................................................................................................................... 19
      9.1.1 View Specifications::Metadata .................................................................................................................. 19
      9.1.2 View Specifications::Strategic .................................................................................................................... 20
      9.1.3 View Specifications::Operational ............................................................................................................... 28
      9.1.4 View Specifications::Services ..................................................................................................................... 38
      9.1.5 View Specifications::Personnel ................................................................................................................... 47
      9.1.6 View Specifications::Resources .................................................................................................................. 62
      9.1.7 View Specifications::Security .................................................................................................................... 74
      9.1.8 View Specifications::Projects .................................................................................................................... 82
      9.1.9 View Specifications::Standards ................................................................................................................ 88
      9.1.10 View Specifications::Actual Resources .................................................................................................. 91
      9.1.11 View Specifications::Dictionary ............................................................................................................... 94
      9.1.12 View Specifications::Summary & Overview .......................................................................................... 95
      9.1.13 View Specifications::Requirements ....................................................................................................... 96
      9.1.14 View Specifications::Information ........................................................................................................... 97
      9.1.15 View Specifications::Parameters .......................................................................................................... 98
      9.1.16 View Specifications::Other ................................................................................................................... 101
10. DOMAIN METAMODEL (DMM) ELEMENTS ................................................................................................ 105

10.1 DOMAIN METAMODEL........................................................................................................................................ 105
10.1.1 Domain MetaModel::Metadata ........................................................................................................... 105
10.1.2 Domain MetaModel::Strategic ............................................................................................................ 119
10.1.3 Domain MetaModel::Operational ........................................................................................................ 130
10.1.4 Domain MetaModel::Services .............................................................................................................. 144
10.1.5 Domain MetaModel::Personnel ........................................................................................................... 152
10.1.6 Domain MetaModel::Resources ........................................................................................................... 159
10.1.7 Domain MetaModel::Security .............................................................................................................. 177
10.1.8 Domain MetaModel::Projects .............................................................................................................. 192
10.1.9 Domain MetaModel::Standards ........................................................................................................... 199
10.1.10 Domain MetaModel::Actual Resources ........................................................................................ 201
10.1.11 Domain MetaModel::Dictionary ........................................................................................................ 207
10.1.12 Domain MetaModel::Summary & Overview .................................................................................... 209
10.1.13 Domain MetaModel::Information ..................................................................................................... 213
10.1.14 Domain MetaModel::Parameters ....................................................................................................... 214
TABLE OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>UAF Grid</td>
<td>13</td>
</tr>
<tr>
<td>7.2</td>
<td>Domain Interrelationships</td>
<td>17</td>
</tr>
<tr>
<td>7.3</td>
<td>Legend of color codes for element types defined in UAF</td>
<td>18</td>
</tr>
<tr>
<td>8.2</td>
<td>Strategic Structure</td>
<td>22</td>
</tr>
<tr>
<td>8.3</td>
<td>Strategic Connectivity</td>
<td>23</td>
</tr>
<tr>
<td>8.4</td>
<td>Strategic States</td>
<td>24</td>
</tr>
<tr>
<td>8.5</td>
<td>Strategic Constraints</td>
<td>25</td>
</tr>
<tr>
<td>8.6</td>
<td>Strategic Roadmap: Deployment</td>
<td>26</td>
</tr>
<tr>
<td>8.7</td>
<td>Strategic Roadmap: Phasing</td>
<td>27</td>
</tr>
<tr>
<td>8.8</td>
<td>Strategic Traceability</td>
<td>28</td>
</tr>
<tr>
<td>8.9</td>
<td>Operational Taxonomy</td>
<td>29</td>
</tr>
<tr>
<td>8.10</td>
<td>Operational Structure</td>
<td>30</td>
</tr>
<tr>
<td>8.11</td>
<td>Operational Connectivity</td>
<td>31</td>
</tr>
<tr>
<td>8.12</td>
<td>Operational Processes</td>
<td>32</td>
</tr>
<tr>
<td>8.13</td>
<td>Operational Processes BPMN Semantics</td>
<td>34</td>
</tr>
<tr>
<td>8.14</td>
<td>Operational States</td>
<td>35</td>
</tr>
<tr>
<td>8.15</td>
<td>Operational Interaction Scenarios</td>
<td>36</td>
</tr>
<tr>
<td>8.16</td>
<td>Operational Constraints</td>
<td>37</td>
</tr>
<tr>
<td>8.17</td>
<td>Operational Traceability</td>
<td>38</td>
</tr>
<tr>
<td>8.18</td>
<td>Services Taxonomy</td>
<td>39</td>
</tr>
<tr>
<td>8.19</td>
<td>Services Structure</td>
<td>40</td>
</tr>
<tr>
<td>8.20</td>
<td>Services Connectivity</td>
<td>41</td>
</tr>
<tr>
<td>8.21</td>
<td>Services Processes</td>
<td>42</td>
</tr>
<tr>
<td>8.22</td>
<td>Services Processes BPMN Semantics</td>
<td>43</td>
</tr>
<tr>
<td>8.23</td>
<td>Services States</td>
<td>44</td>
</tr>
<tr>
<td>8.24</td>
<td>Services Interaction Scenarios</td>
<td>45</td>
</tr>
<tr>
<td>8.25</td>
<td>Services Constraints</td>
<td>46</td>
</tr>
<tr>
<td>8.26</td>
<td>Services Roadmap</td>
<td>46</td>
</tr>
<tr>
<td>8.27</td>
<td>Services Traceability</td>
<td>47</td>
</tr>
<tr>
<td>8.28</td>
<td>Personnel Taxonomy</td>
<td>48</td>
</tr>
<tr>
<td>8.29</td>
<td>Personnel Structure</td>
<td>49</td>
</tr>
<tr>
<td>8.30</td>
<td>Personnel Connectivity</td>
<td>50</td>
</tr>
<tr>
<td>8.31</td>
<td>Personnel Processes</td>
<td>51</td>
</tr>
<tr>
<td>8.32</td>
<td>Personnel States</td>
<td>52</td>
</tr>
<tr>
<td>8.33</td>
<td>Personnel Interaction Scenarios</td>
<td>53</td>
</tr>
<tr>
<td>8.34</td>
<td>Personnel Constraints: Competence</td>
<td>55</td>
</tr>
<tr>
<td>8.35</td>
<td>Personnel Constraints: Drivers</td>
<td>56</td>
</tr>
<tr>
<td>8.36</td>
<td>Personnel Constraints: Performance</td>
<td>57</td>
</tr>
<tr>
<td>8.37</td>
<td>Personnel Roadmap: Availability</td>
<td>59</td>
</tr>
<tr>
<td>8.38</td>
<td>Personnel Roadmap: Evolution</td>
<td>60</td>
</tr>
<tr>
<td>8.39</td>
<td>Personnel Roadmap: Forecast</td>
<td>61</td>
</tr>
<tr>
<td>8.40</td>
<td>Personnel Traceability</td>
<td>62</td>
</tr>
<tr>
<td>8.41</td>
<td>Resources Taxonomy</td>
<td>63</td>
</tr>
<tr>
<td>8.42</td>
<td>Resources Structure</td>
<td>64</td>
</tr>
<tr>
<td>8.43</td>
<td>Resources Connectivity</td>
<td>65</td>
</tr>
<tr>
<td>8.44</td>
<td>Resources Processes</td>
<td>67</td>
</tr>
<tr>
<td>8.45</td>
<td>Resources Processes BPMN Semantics</td>
<td>68</td>
</tr>
<tr>
<td>8.46</td>
<td>Resources States</td>
<td>69</td>
</tr>
<tr>
<td>8.47</td>
<td>Resources Interaction Scenarios</td>
<td>70</td>
</tr>
<tr>
<td>8.48</td>
<td>Resources Constraints</td>
<td>71</td>
</tr>
<tr>
<td>8.49</td>
<td>Resources Roadmap: Evolution</td>
<td>72</td>
</tr>
</tbody>
</table>
TABLE OF TABLES

Table 1:1 - Table of Related Documents................................................................................................................ 4
Table 5:1 - Description of acronyms used in this specification.................................................................................. 9
Table 7:1 - Definitions for the Domains.................................................................................................................... 15
Table 7:2 - Definitions of the Model Kinds................................................................................................................... 16
Preface

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 https://www.omg.org/.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from this URL: https://www.omg.org/spec

Specifications are organized by the following categories:

Business Modeling Specifications

Middleware Specifications

- CORBA/IIOP
- Data Distribution Services
- Specialized CORBA IDL/Language Mapping Specifications

Modeling and Metadata Specifications

- UML, MOF, CWM, XMI
- UML Profile Specifications

Platform Independent Model (PIM) - Platform Specific Model (PSM) - Interface Specifications

- CORBAServices
- CORBAFacilities
- OMG Domain Specifications
- CORBA Embedded Intelligence Specifications
- CORBA 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 109 Highland Avenue, 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

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

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

**Helvetic/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
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 https://www.omg.org, under Documents, Report a Bug/Issue (https://issues.omg.org/issues/create-new-issue).
1 Scope

1.1 Introduction

There are four parts to this specification, two are normative and two informative. The normative parts are:

1. The UAF Domain Metamodel (DMM) (this document) that provides the definition of concepts, relationships and viewpoints for the framework. The UAF DMM is the basis for any implementation of UAF including non-UML/SysML implementations.

2. The UAF Profile (UAFP) (see document dtc/19-06-15) is a UML/SysML implementation of the UAF DMM.

The informative parts are:

3. The UAF Traceability, Annex A (see document dtc/19-06-17), which details the mappings between the UAF and the various frameworks and languages that contribute to the UAF.

4. The UAF Example Model, Annex B (see document dtc/19-06-18), which illustrates a practical usage of UAF.

1.2 UAF Background

UAF evolved from the Unified Profile for DoDAF and MODAF (UPDM), version 2.1. UAF extends the scope of UPDM and generalizes it to make it applicable to commercial as well as military architectures. The intent of UAF is to provide a standard representation for describing enterprise architectures using a Model Based Systems Engineering (MBSE) approach.

The core concepts in the UAF are based upon the DoDAF 2.0.2 Domain Metamodel (DM2) and the MODAF ontological data exchange mechanism (MODEM), Security Views from Canada's Department of National Defense Architecture Framework (DNDAF) and the North Atlantic Treaty Organization (NATO) Architecture Framework (NAF) v 4.

UAF models describe a system1 from a set of stakeholders' concerns such as security or information through a set of predefined viewpoints. Developed models can also reflect custom viewpoints or users can develop more formal extensions for new viewpoints.

The UAFP can be used to develop architectures compliant with:

- Department of Defense Architecture Framework (DoDAF) version 2.02
- Ministry of Defence Architecture Framework (MODAF) version 1.3
- North Atlantic Treaty Organization (NATO) Architecture Framework (NAF) version 3.1
- North Atlantic Treaty Organization (NATO) Architecture Framework (NAF) version 4

UAF v 1.1 supports 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 (SoS) down to lower levels of design and implementation,
- support the analysis, specification, design, and verification of complex systems; and
- improve the ability to exchange architecture information among related tools that are SysML based.

---

1 The term system is used from: “Systems and software engineering -- Architecture description,” http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508
### 1.3 Intended Usage

The UAF enables the modeling of strategic capabilities, operational scenarios, services, resources, personnel, security, projects, standards, measures and requirements; which supports best practices through, separation of concerns and abstractions. In addition, the UAF enables the modeling of related architecture concepts such as:

- System of Systems (SoS),
- information exchanges consistent with the National Information Exchange Model (NIEM),
- DoD's doctrine, organization, training material, leadership & education, personnel, and facilities (DOTMLPF)
- UK Ministry of Defence Lines of Development (DLOD) elements,
- Human Computer Interfaces (HCI).

Further, The UAF conforms to terms defined in the ISO/IEC/IEEE 42010 standard for architecture description, where the terms: architecture, architecture description (AD), architecture framework, architecture view, architecture viewpoint, concern, environment, model kind, stakeholder [ISO/IEC/IEEE 42010:2011] form correspondence rules specified as constraints on UAF.

### 1.4 Related Documents

The specification includes a metamodel and description as separate documents. Other appendices are also provided as separate documents. The table below provides a listing of these documents:

| dtc/19-06-16 | The UAF Domain MetaModel (DMM) |
| dtc/19-06-15 | The UAF Profile (UAFP) |
| dtc/19-06-17 | Appendix A that contains a separate traceability subsection from UAFP to each of the frameworks listed in Section 1.2 of this specification |
| dtc/19-06-18 | Appendix B: An example of how the language can be used to represent a UAFP architecture |
| dtc/19-06-19 | UAF XMI file |
| dtc/19-06-20 | UAF XMI Measurements library |
| dtc/19-05-14 | Attachments |
2 Conformance

UAF specifies four types of conformance.

*Type 1 Conformance: - UAF View specification conformance.* A tool demonstrating view specification conformance shall implement a version of all the view specifications defined in the UAF Grid, with the exception of the view specifications in the Metadata Domain. Optionally the tool vendor can implement other donor framework viewpoints, for instance DoDAF, MODAF or NAF based upon the mapping between them and UAF provided in Appendix A (dtc/19-06-17)

*Type 2 Conformance: - UAF Conceptual Syntax Conformance.* A tool demonstrating conceptual syntax conformance is consistent with the concepts, relationships and constraints defined in the UAF DMM (this document). UAF Conceptual Syntax Conformance implies Type 1 Conformance.

*Type 3 Conformance: - UAF Formal Syntax Conformance.* A tool demonstrating formal syntax conformance:
- enables instances of concrete UAFP stereotypes defined in the UAFP (dtc/19-06-15)
- complies with the constraints defined in the UAFP (dtc/19-06-15)
- complies with the SysML version 1.5 Concrete Syntax Conformance (formal/17-05-01)

UAF Formal Syntax Conformance implies Type 2 Conformance.

*Type 4 Conformance: - UAF Model interchange conformance.* A tool demonstrating model interchange conformance can import and export conformant XMI for all valid UAFP models. Model interchange conformance implies Type 3 Conformance.
3 References

3.1 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.

3.2 OMG Documents (Normative References)

- Unified Profile for DoDAF and MODAF (UPDM), 2.1, August 2013, https://www.omg.org/spec/UPDM
- Ontology Definition Metamodel (ODM), 1.1, September 2014, https://www.omg.org/spec/ODM

3.3 Other Normative References

- DM2 - DoDAF Meta-Model,
- IDEAS Foundation v1.0 as XMI File (zipped), http://www.ideasgroup.org/7Documents/
- MODAF Ontological Data Exchange Mechanism (MODEM)
- NATO Architecture Framework (NAF),
3.4 Informative References

- Object Management Group (OMG), Metamodel Extension Facility, Initial submission, ad/12-02-01, https://www.omg.org/cgi-bin/doc?ad/12-02-01 (Requires OMG Member Access)
4 Terms and Definitions

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

For the purposes of this specification, the following List of symbols/abbreviations apply.

Table 5:1 - Description of acronyms used in this specification

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcV-*2</td>
<td>Acquisition View</td>
</tr>
<tr>
<td>AD</td>
<td>Architecture Description</td>
</tr>
<tr>
<td>AV-*</td>
<td>All View</td>
</tr>
<tr>
<td>BMM</td>
<td>Business Motivation Model</td>
</tr>
<tr>
<td>BPMN</td>
<td>Business Process Modeling Notation</td>
</tr>
<tr>
<td>C4ISR</td>
<td>Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance</td>
</tr>
<tr>
<td>CaT</td>
<td>Capability Team</td>
</tr>
<tr>
<td>COI</td>
<td>Communities of Interest</td>
</tr>
<tr>
<td>CV-*</td>
<td>Capability View</td>
</tr>
<tr>
<td>DIV-*</td>
<td>Data and Information Views</td>
</tr>
<tr>
<td>DLOD</td>
<td>Defence Lines of Development</td>
</tr>
<tr>
<td>DM2</td>
<td>DoDAF Meta Model</td>
</tr>
<tr>
<td>DMM</td>
<td>Domain Meta Model</td>
</tr>
<tr>
<td>DNDAF</td>
<td>Department National Defence and Canadian Forces (DND/ CF) Architecture Framework</td>
</tr>
<tr>
<td>DoD</td>
<td>United States Department of Defense</td>
</tr>
<tr>
<td>DoDAF</td>
<td>Department of Defense Architecture Framework</td>
</tr>
<tr>
<td>DOTMLP</td>
<td>Doctrine, Organization, Training, Material, Leadership, Personnel, Facilities</td>
</tr>
<tr>
<td>EIE</td>
<td>Enterprise Information Environment</td>
</tr>
<tr>
<td>IDEAS</td>
<td>International Defense Enterprise Architecture Specification for Exchange</td>
</tr>
<tr>
<td>IDEF</td>
<td>Integrated DEFinition Methods</td>
</tr>
<tr>
<td>INCOSE</td>
<td>International Council Of Systems Engineering</td>
</tr>
<tr>
<td>JCIDS</td>
<td>Joint Capabilities Integration and Development System</td>
</tr>
<tr>
<td>MISIG</td>
<td>Model Interchange Special Interest Group</td>
</tr>
<tr>
<td>MOD</td>
<td>United Kingdom Ministry of Defence</td>
</tr>
<tr>
<td>MODAF</td>
<td>Ministry of Defence Architecture Framework</td>
</tr>
<tr>
<td>MODEM</td>
<td>MODAF Ontological Data Exchange Mechanism</td>
</tr>
<tr>
<td>NAF</td>
<td>NATO Architecture Framework</td>
</tr>
<tr>
<td>OASIS</td>
<td>Organization for the Advancement of Structured Information Standards</td>
</tr>
<tr>
<td>OSLC</td>
<td>Open Services for Lifecycle Collaboration</td>
</tr>
<tr>
<td>OV-*</td>
<td>Operational View</td>
</tr>
<tr>
<td>PES</td>
<td>DoDAF Physical Exchange Specification</td>
</tr>
<tr>
<td>POC</td>
<td>Proof of Concept</td>
</tr>
<tr>
<td>PV-*</td>
<td>Project View</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>SoaML</td>
<td>Service orientated architecture Modeling Language</td>
</tr>
<tr>
<td>SoS</td>
<td>System of Systems</td>
</tr>
</tbody>
</table>

2 * denotes a wildcard
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOV-*</td>
<td>Service Oriented View</td>
</tr>
<tr>
<td>StdV-*</td>
<td>Standards View in DoDAF 2.02 compare TV-* in UAF</td>
</tr>
<tr>
<td>STV-*</td>
<td>Strategic View</td>
</tr>
<tr>
<td>SV-*</td>
<td>System View</td>
</tr>
<tr>
<td>SvcV-*</td>
<td>Service View</td>
</tr>
<tr>
<td>TEPID</td>
<td>Training, Equipment, Personnel, Information, Concepts and Doctrine, Organisation, Infrastructure, Logistics</td>
</tr>
<tr>
<td>OIL</td>
<td></td>
</tr>
<tr>
<td>TOGAF</td>
<td>The Open Group Architectural Framework©</td>
</tr>
<tr>
<td>TPPU</td>
<td>Task, Post, Process, and Use</td>
</tr>
<tr>
<td>TV-*</td>
<td>Technical View</td>
</tr>
<tr>
<td>UAF</td>
<td>Unified Architecture Framework</td>
</tr>
<tr>
<td>UAFP</td>
<td>Unified Architecture Framework Profile</td>
</tr>
<tr>
<td>UPDM</td>
<td>Unified Profile for DoDAF/MODAF</td>
</tr>
</tbody>
</table>
6 Additional Information

6.1 Changes to Adopted OMG Specifications
This specification completely replaces Unified Architecture Framework (UAF), version 1.0
https://www.omg.org/spec/UAF/About-UAF/

6.2 Language Architecture
The UAF specification reuses a subset of UML 2.5.1 and SysML 1.5 and provides additional extensions needed to address requirements in the UPDM 3.0 RFP Mandatory Requirements. Those requirements form the basis for this specification. This specification documents the language architecture in terms of UML 2.5.1 and SysML 1.5 and specifies how to implement UAF. This clause explains design principles and how they are applied to define the UAF language architecture.

6.3 Philosophy
The UAF development uses a model-driven approach. A simple description of the work process is:

- A Domain Metamodel (DMM) uses UML Class models to represent individuals, types and tuples that aggregate the concepts defined in DoDAF, MODEM, NAF, DNDAF and other frameworks.
- The aligned and renamed viewpoints from the various frameworks provide a common generic name for each viewpoint. It should be noted that the term viewpoint is in the context of ISO 42010 where a viewpoint is the specification of a view. The UAF viewpoints are mapped to the corresponding viewpoint in the relevant contributing framework. It is the viewpoints described in the DMM that provides the basis for the Unified Architecture Framework (UAF).
- The UAF provides an abstraction layer that separates the underlying UAF metamodel from the presentation layer. The results of this mapping are given in Appendix A (see document dtc/19-06-17 and an overview of the viewpoints in a grid format are given in this document.
- The intent of the UAF is to provide a Domain MetaModel usable by non-UML/SysML tool vendors who may wish to implement the UAF within their own tool and metalanguage.
- The Unified Architecture Framework Profile (UAFP) is the standard implementation of the UAF DMM. It was created by mapping the UAF concepts and relationships to corresponding stereotypes in the UAFP.
- The UAFP analysis and refactoring reflects language architecture, tool implementation, and reuse considerations.
- The specification is generated from the UML model used to describe the UAF DMM and UAFP. This approach allows the team to concentrate on architecture issues rather than documentation production. The UML tool automatically maintains consistency. The UML tool improves maintenance and enables traceability between the UAF and the UAFP where every stereotype is linkable to the UAF element using UML Abstraction relationship.

6.4 Core Principles
The fundamental design principles for UAF DMM are:

- **Requirements-driven**: UAF is intended to satisfy the requirements of the UPDM 3.0 RFP Mandatory Requirements.
- **Influence from donor Frameworks**: The DMM was based upon an aggregation of concepts and relationships from the donor frameworks.
- **IDEAS Ontology driven**: The DMM was based upon a simplified version of the IDEAS ontology, see chapter 8.
• **DMM Notation**: The DMM was expressed using UML class diagram notation.

• **Reusability of UML Metamodel concepts**: The UAF DMM reuses a number of concepts from the UML Metamodel, such as Statemachines, Activities and Interactions. The explicit relationship to these concepts enables the UAF DMM to reuse UML semantics instead of reinventing its own semantics.

• **Reusability of BPMN concepts**: The UAF DMM reuses a number of concepts from BPMN, such as processes. The explicit relationship to these concepts enables the UAF DMM to reuse BPMN semantics instead of reinventing its own semantics.
7 UAF Grid

Due to the complexity of managing the multiple viewpoints with overlapping concerns and metamodels, the standard viewpoints are refactored as described in the donor frameworks into a more manageable format. This decision led to the development of the UAF grid which is described below.

The grid is a way of showing how the various viewpoints (known as view specifications in the rest of the document) correspond to domains (horizontal rows) and the model kinds (the columns) that describe the view specification. The intent of the grid is not to be complete, but to capture the information that is present in the frameworks that contributes to the UAF, consequently, some gaps are evident.

<table>
<thead>
<tr>
<th>Metadata Md</th>
<th>Tauxonomy Tx</th>
<th>Structure St</th>
<th>Connectivity Cn</th>
<th>Processes Pr</th>
<th>States St</th>
<th>Interaction Scenarios Ss</th>
<th>Information Cst</th>
<th>Parameter</th>
<th>Constraints Cst</th>
<th>Roadmap Rm</th>
<th>Traceability Tr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>Metadata</td>
<td>Metadata</td>
<td>Metadata</td>
<td>Metadata</td>
<td>Metadata</td>
<td>-</td>
<td>Metadata</td>
<td>-</td>
<td>Metadata</td>
<td>-</td>
<td>Metadata</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Processes</td>
<td>States</td>
<td>Interaction</td>
<td>Constraints</td>
<td>Parameter</td>
<td>Roadmap</td>
<td>Traceability</td>
<td>Tauxonomy</td>
</tr>
<tr>
<td></td>
<td>Md-Md</td>
<td>Md-Md</td>
<td>Md-Md</td>
<td>Pr</td>
<td>St</td>
<td>Ss</td>
<td>Cst</td>
<td></td>
<td>Rm</td>
<td>Tr</td>
<td>Md-Md</td>
</tr>
<tr>
<td>Strategic</td>
<td>Strategic</td>
<td>Strategic</td>
<td>Strategic</td>
<td>-</td>
<td>Strategic</td>
<td>-</td>
<td>Strategic</td>
<td></td>
<td>Strategic</td>
<td>-</td>
<td>Strategic</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>States</td>
<td>States</td>
<td>Interaction</td>
<td>Tauxonomy</td>
<td></td>
<td>Tauxonomy</td>
<td>-</td>
<td>Tauxonomy</td>
</tr>
<tr>
<td></td>
<td>St-St</td>
<td>St-St</td>
<td>St-St</td>
<td>St-St</td>
<td>St-St</td>
<td>Ss</td>
<td>St</td>
<td></td>
<td>St-St</td>
<td>St-St</td>
<td>St-St</td>
</tr>
<tr>
<td>Operational</td>
<td>Operational</td>
<td>Operational</td>
<td>Operational</td>
<td>Operational</td>
<td>Operational</td>
<td>Operational</td>
<td>Operational</td>
<td></td>
<td>Operational</td>
<td>-</td>
<td>Operational</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Processes</td>
<td>States</td>
<td>Services</td>
<td>Cst</td>
<td></td>
<td>Cst</td>
<td>-</td>
<td>Cst</td>
</tr>
<tr>
<td>Services</td>
<td>Service</td>
<td>Service</td>
<td>Service</td>
<td>Service</td>
<td>Service</td>
<td>Service</td>
<td>Service</td>
<td></td>
<td>Service</td>
<td>Service</td>
<td>Service</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Processes</td>
<td>States</td>
<td>Services</td>
<td>Information</td>
<td></td>
<td>Measurements</td>
<td>Measurements</td>
<td>Information</td>
</tr>
<tr>
<td></td>
<td>Sv-Sv</td>
<td>Sv-Sv</td>
<td>Sv-Sv</td>
<td>Pr</td>
<td>St</td>
<td>Sv-Sv</td>
<td>Sv</td>
<td></td>
<td>Pr-We</td>
<td>Pr-We</td>
<td>Sv-Pr</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Processes</td>
<td>Status</td>
<td>Processes</td>
<td>Information</td>
<td></td>
<td>Constraints</td>
<td>Evolution</td>
<td>Trcability</td>
</tr>
<tr>
<td></td>
<td>Pr-Pr</td>
<td>Pr-Pr</td>
<td>Pr-Pr</td>
<td>Pr</td>
<td>Pr</td>
<td>Pr</td>
<td>Pr</td>
<td></td>
<td>Rm</td>
<td>Rm</td>
<td>Tr</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Processes</td>
<td>States</td>
<td>Processes</td>
<td>Information</td>
<td></td>
<td>Constraints</td>
<td>Evolution</td>
<td>Trcability</td>
</tr>
<tr>
<td></td>
<td>Rs-Rs</td>
<td>Rs-Rs</td>
<td>Rs-Rs</td>
<td>Rs</td>
<td>Rs</td>
<td>Rs</td>
<td>Rs</td>
<td></td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Processes</td>
<td>States</td>
<td>Processes</td>
<td>Processes</td>
<td></td>
<td>Processes</td>
<td>Processes</td>
<td>Processes</td>
</tr>
<tr>
<td></td>
<td>Sc-Sc</td>
<td>Sc-Sc</td>
<td>Sc-Sc</td>
<td>Sc-Pj</td>
<td>Sc-Pj</td>
<td>Sc-Pj</td>
<td>Sc-Pj</td>
<td></td>
<td>Sc-Pj</td>
<td>Sc-Pj</td>
<td>Sc-Pj</td>
</tr>
<tr>
<td>Projects</td>
<td>Project</td>
<td>Project</td>
<td>Project</td>
<td>Project</td>
<td>Project</td>
<td>Project</td>
<td>-</td>
<td></td>
<td>Project</td>
<td>Project</td>
<td>Project</td>
</tr>
<tr>
<td></td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Tauxonomy</td>
<td>Processes</td>
<td>Processes</td>
<td>Processes</td>
<td>-</td>
<td></td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
</tr>
<tr>
<td>Standards</td>
<td>Standard</td>
<td>Standard</td>
<td>Standard</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>Standards</td>
<td>Standards</td>
<td>Standards</td>
</tr>
<tr>
<td>Actual</td>
<td>Actual</td>
<td>Actual</td>
<td>Actual</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Resources</td>
<td>Transactions</td>
<td>Resources</td>
<td>Connectivity</td>
<td>-</td>
<td>-</td>
<td>Simulation</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Actual</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>Ar</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
d. The parameters column captures the measures and environments across the architecture in all the different domains.

e. The expectation is that the physical schema model would not be defined in the UAF. Any tool implementing the framework provides a means to import or link-to representations of the physical model.

f. The Metadata Taxonomy view specification provides a means to extend the framework to other domains.

The detailed mapping between the view specifications of the UAF shown in the grid and the viewpoints from the donor frameworks is described in dte'2019-06-17. A definition for each view specification in the grid is described in the following chapters.
7.1 Descriptions of Domains and Model Kinds

Table 7:1 - Definitions for the Domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>Md</td>
<td>Identifies the metadata required to develop a suitable architecture that is fit for its purpose.</td>
</tr>
<tr>
<td>Strategic</td>
<td>St</td>
<td>Capability management process. Describes the capability taxonomy, composition, dependencies and evolution.</td>
</tr>
<tr>
<td>Operational</td>
<td>Op</td>
<td>Illustrates the Logical Architecture of the enterprise. Describes the requirements, operational behavior, structure, and exchanges required to support (exhibit) capabilities. Defines all operational elements in an implementation/solution independent manner.</td>
</tr>
<tr>
<td>Services</td>
<td>Sv</td>
<td>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. DoD: The Service Views within the Services Viewpoint describe the design for service-based solutions to support operational development processes (JCIDS) and Defense Acquisition System or capability development within the Joint Capability Areas.</td>
</tr>
<tr>
<td>Personnel</td>
<td>Pr</td>
<td>Defines and explores organizational resource types. Shows the taxonomy of types of organizational resources as well as connections, interaction and growth over time.</td>
</tr>
<tr>
<td>Resources</td>
<td>Rs</td>
<td>Captures a solution architecture consisting of resources, e.g., organizational, software, artifacts, capability configurations, and natural resources that implement the operational requirements. Further design of a resource is typically detailed in SysML or UML.</td>
</tr>
<tr>
<td>Security</td>
<td>Sc</td>
<td>Security assets and security enclaves. Defines the hierarchy of security assets and asset owners, security constraints (policy, laws, and guidance) and details where they are located (security enclaves).</td>
</tr>
<tr>
<td>Projects</td>
<td>Pj</td>
<td>Describes projects and project milestones, how those projects deliver capabilities, the organizations contributing to the projects and dependencies between projects.</td>
</tr>
<tr>
<td>Standards</td>
<td>Sd</td>
<td>MODAF: Technical Standards Views are extended from the core DoD: views to include non-technical standards such as operational doctrine, industry process standards, etc. DoD: The Standards Views within the Standards Viewpoint are the set of rules governing the arrangement, interaction, and interdependence of solution parts or elements.</td>
</tr>
<tr>
<td>Actual</td>
<td>Ar</td>
<td>The analysis, e.g., evaluation of different alternatives, what-if, trade-offs, V&amp;V on the actual resource configurations. Illustrates the expected or achieved actual resource configurations.</td>
</tr>
<tr>
<td>Resources</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Kind</td>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Taxonomy</td>
<td>Tx</td>
<td>Presents all the elements as a standalone structure. Presents all the elements as a specialization hierarchy, provides a text definition for each one and references the source of the element.</td>
</tr>
<tr>
<td>Structure</td>
<td>Sr</td>
<td>Describes the definitions of the dependencies, connections, and relationships between the different elements.</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Cn</td>
<td>Describes the connections, relationships, and interactions between the different elements.</td>
</tr>
<tr>
<td>Processes</td>
<td>Pr</td>
<td>Captures activity based behavior and flows. It describes activities, their Inputs/Outputs, activity actions and flows between them.</td>
</tr>
<tr>
<td>States</td>
<td>St</td>
<td>Captures state-based behavior of an element. It is a graphical representation of states of a structural element and how it responds to various events and actions.</td>
</tr>
<tr>
<td>Interaction</td>
<td>Is</td>
<td>Expresses a time ordered examination of the exchanges as a result of a particular scenario. Provides a time-ordered examination of the exchanges between participating elements as a result of a particular scenario.</td>
</tr>
<tr>
<td>Scenarios</td>
<td>If</td>
<td>Address the information perspective on operational, service, and resource architectures. Allows analysis of an architecture’s information and data definition aspect, without consideration of implementation specific issues.</td>
</tr>
<tr>
<td>Information</td>
<td>Ct</td>
<td>Details the measurements that set performance requirements constraining capabilities. Also defines the rules governing behavior and structure.</td>
</tr>
<tr>
<td>Constraints</td>
<td>Rm</td>
<td>Addresses how elements in the architecture change over time. Also, how at different points in time or different periods of time.</td>
</tr>
<tr>
<td>Traceability</td>
<td>Tr</td>
<td>Describes the mapping between elements in the architecture. This can be between different viewpoints within domains as well as between domains. It can also be between structure and behaviors.</td>
</tr>
</tbody>
</table>
7.2 Domain Interrelationships

Although the grid is the primary means of expressing the relationship between the Domains, Model Kinds and View Specifications, because of its two-dimensional nature it is not adequate to explain the abstract interrelationships that exist between the domains. The following diagram is an indication of how the domains are interrelated.

Figure 7:2 - Domain Interrelationships

Where a Domain is shown vertically the intent is to show that the Domain is a cross cutting concern that goes across the levels of abstraction in the architecture.

Where a Domain is shown horizontally the intent is to show that the Domain exists in a layer of abstraction between the Domains above and below it and there is an interrelationship with the Domains either side of it.

7.3 Domain Metamodel Diagram Legend

This Annex comprises 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.

Note that the diagrams rely on color to aid the reader in understanding the model. Please refer to the legend below to understand the diagrams.

The following is the legend of element colors used in the DMM and what they denote.
The meaning of the element types in the UAF are based upon concepts put forth in the International Defence Enterprise Architecture Specification (IDEAS).

- An Individual denotes a single instance of an element.
- A Type denotes a set of Individuals.
- A Tuple denotes a relationship that exists between elements.
- An Abstract denotes that the element has no direct use but is a means of construction.
- An Enumeration is a complete, ordered listing of all the items in a collection.
- An External Type is an element that exists outside of the core DMM but is referenceable by elements in the DMM.
8  Domain Metamodel Diagrams

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.

8.1  View Specifications

This section documents each of the view specifications of UAF.

8.1.1  View Specifications::Metadata

Stakeholders: Enterprise Architects, Technical Managers.
Concerns: architecture development process, architecture traceability, metamodel and its extensions, architecture versioning.
Definition: Identifies the metadata required to develop a suitable architecture that is fit for its purpose.

View Specifications::Metadata::Taxonomy
Stakeholders: Enterprise Architects, Technical Managers.
Concerns: metamodel and its extensions.
Definition: captures user defined metamodel extensions
Recommended Implementation: UML Profile Diagram, SysML Block Definition Diagram

View Specifications::Metadata::Structure
Stakeholders: Enterprise Architects, Technical Managers.
Concerns: domains, model kinds, and view specifications that are used to describe the architecture.
Definition: (i) lists predefined and custom domains, model kinds, and view specifications (ii) and identify the key stakeholders and their concerns.
Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram.

View Specifications::Metadata::Connectivity
Stakeholders: Enterprise Architects, people who want to understand relationships to related architectural descriptions, Technical Managers.
Concerns: high-level dependencies between architectural descriptions.
Definition: depicts and analyzes all relevant dependencies between architectural descriptions, e.g., reference architectures, as-is to to-be architectures.
Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram, matrix format.

View Specifications::Metadata::Processes
Stakeholders: Enterprise Architects, people who want to understand the architecture development process, Technical Managers.
Concerns: methodology used.
Definition: methodology used in developing the architecture.
Recommended Implementation: SysML Activity Diagram, text.

View Specifications::Metadata::States
Stakeholders: Enterprise Architects, people who want to understand the architecture governance, Technical Managers.
Concerns: architecture status.
Definition: captures version number and approval workflow of the architecture.
Recommended Implementation: SysML State Machine Diagram, state table, text.

**View Specifications::Metadata::Constraints**

Stakeholders: Enterprise Architects, people who want to understand constraints for the architecture, Technical Managers.
Concerns: architectural constraints.
Definition: captures assumptions and constraints on the architecture.
Recommended Implementation: tabular format, text.

**View Specifications::Metadata::Roadmap**

Stakeholders: Enterprise Architects, people who want to understand the architecture development plan, Technical Managers.
Concerns: architecture release schedule.
Definition: captures project timeline for the architecture.
Recommended Implementation: timeline, text.

**View Specifications::Metadata::Traceability**

Stakeholders: Enterprise Architects, people who want to understand impact of change across the architecture supporting assets, Technical Managers.
Concerns: reuse of architectures.
Definition: shows references to asset libraries, legacy architectures, and external sources, e.g., documents.
Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram, tabular format.

**8.1.2 View Specifications::Strategic**

Stakeholders: Capability Portfolio Managers.
Concerns: capability management process.
Definition: describe capability taxonomy, composition, dependencies and evolution.

**View Specifications::Strategic::Taxonomy**

Contains the diagrams that document the Strategic Taxonomy Viewpoint.

**View Specifications::Strategic::Taxonomy::Strategic Taxonomy**

Stakeholders: PMs, Enterprise Architects, Executives.
Concerns: capability needs.
Definition: shows the taxonomy of capabilities.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8.1 - Strategic Taxonomy

Elements
- Capability
- CapabilityGeneralization
- CapabilityRole

View Specifications::Strategic::Structure

Contains the diagrams that document the Strategic Structure Viewpoint.

View Specifications::Strategic::Structure::Strategic Structure

Stakeholders: PMs, Enterprise Architects, Executives.
Concerns: capability needs.
Definition: shows the relationship between EnterprisePhases and the Capabilities that are intended to be developed during the enterprise phases, and the organizations involved in the enterprise.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8.2 - Strategic Structure

Elements

- **ActualEnduringTask**
- **ActualEnterprisePhase**
- **ActualOrganization**
- **ActualResponsibleResource**
- **Capability**
- **CapableElement**
- **EnduringTask**
- **EnterpriseGoal**
- **EnterprisePhase**
- **EnterpriseVision**
- **Exhibits**
- **OperationalArchitecture**
- **OrganizationInEnterprise**
- **ResourceArchitecture**
- **StructuralPart**
- **TemporalPart**
- **WholeLifeEnterprise**
**View Specifications::Strategic::Connectivity**
Contains the diagrams that document the Strategic Connectivity Viewpoint.

**View Specifications::Strategic::Connectivity::Strategic Connectivity**
Stakeholders: PMs, Executives, Enterprise Architects.
Concerns: capability dependencies.
Definition: describes the dependencies between planned capabilities.
Recommended Implementation: SysML Block Definition Diagram. SysML Internal Block Diagram.

**Figure 8:3 - Strategic Connectivity**
Elements
- Capability
- CapabilityDependency
- CapabilityRole
- CapabilityRoleDependency

**View Specifications::Strategic::States**
Contains the diagrams that document the Strategic States Viewpoint.

**View Specifications::Strategic::States::Strategic States**
Stakeholders: PMs, Enterprise Architects.
Concerns: effects that the implementation(s) of capabilities are expected to deliver.
Definition: captures the relationships between capability(ies) and desired effect(s) that implementation(s) of capability(ies) should achieve.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:4 - Strategic States

Elements

- AchievedEffect
- Achiever
- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResource
- ActualResponsibleResource
- ActualState
- Capability
- CapabilityConfiguration
- DesiredEffect
- Desirer
- FieldedCapability
- NaturalResource
- PhysicalResource
- Post
- ResourceArchitecture

Unified Architecture Framework (UAF) Domain Metamodel Version 1.1
View Specifications::Strategic::Constraints
Contains the diagrams that document the Strategic Constraints Viewpoint.

View Specifications::Strategic::Constraints::Strategic Constraints
Stakeholders: PMs, Enterprise Architects.
Concerns: capability constraints.
Definition: details the measurements that set performance requirements constraining capabilities.
Recommended Implementation: tabular format, SysML Block Definition Diagram.

Figure 8:5 - Strategic Constraints
Elements
- Capability
- Measurement
- PropertySet

View Specifications::Strategic::Roadmap
Contains the diagrams that document the Strategic Roadmap Viewpoint.

View Specifications::Strategic::Roadmap::Deployment

View Specifications::Strategic::Roadmap::Deployment::Strategic Roadmap: Deployment
Stakeholders: PMs, Executives, Enterprise Architects.
Concerns: capability deployment to organizations over time.
Definition: addresses the deployment of capability(ies) to actual organizations over time.
Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.
Figure 8:6 - Strategic Roadmap: Deployment

Elements

- ActualEnterprisePhase
- ActualOrganization
- ActualPerson
- ActualPost
- ActualProject
- ActualProjectMilestone
- ActualResource
- ActualResponsibleResource
- Capability
- CapabilityConfiguration
- CapableElement
- EnterprisePhase
- Exhibits
- ResourceArchitecture
- ResourcePerformer
- ResponsibleFor
- VersionedElement

View Specifications::Strategic::Roadmap::Phasing

View Specifications::Strategic::Roadmap::Phasing::Strategic Roadmap: Phasing

Stakeholders: PMs, Executives, Enterprise Architects.
Concerns: capability(ies) achievement over time.
Definition: the planned achievement of capability(ies) at different points in time or during specific periods of time.
Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.
Figure 8:7 - Strategic Roadmap: Phasing

Elements

- ActualProject
- ActualProjectMilestone
- ActualResource
- Capability
- CapabilityConfiguration
- CapableElement
- Exhibits
- FieldedCapability
- Project
- ResourceArchitecture
- ResourcePerformer
- VersionedElement

View Specifications::Strategic::Traceability

Contains the diagrams that document the Strategic Traceability Viewpoint.

View Specifications::Strategic::Traceability::Strategic Traceability

Stakeholders: PMs, Enterprise Architects, Business Architects.
Concerns: traceability between capabilities and operational activities.
Definition: describes the mapping between the capabilities required by an Enterprise and the supporting operational activities.
Recommended Implementation: matrix format, SysML Block Definition Diagram.
Figure 8:8 - Strategic Traceability

Elements
- ActualEnduringTask
- Capability
- CapabilityForTask
- Function
- Implements
- MapsToCapability
- OperationalActivity
- Process
- StandardOperationalActivity

8.1.3 View Specifications::Operational

Stakeholders: Business Architects, Executives.
Concerns: illustrate the Logical Architecture of the enterprise.
Definition: describe the requirements, operational behavior, structure, and exchanges required to support (exhibit) capabilities. Defines all operational elements in an implementation/solution independent manner.

View Specifications::Operational::Taxonomy

Contains the diagrams that document the Operational Taxonomy Viewpoint.

View Specifications::Operational::Taxonomy::Operational Taxonomy

Stakeholders: Business Architects, Systems Engineers, Enterprise Architects, Owners responsible for Operational Agents.
Concerns: OperationalAgent types.
Definition: shows the taxonomy of types of OperationalAgents.
Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.
Figure 8:9 - Operational Taxonomy

Elements

- ArbitraryConnector
- Asset
- CapabilityConfiguration
- ConceptItem
- HighLevelOperationalConcept
- Location
- NaturalResource
- OperationalAgent
- OperationalAsset
- OperationalPerformer
- Organization
- OrganizationalResource
- PhysicalResource
- Post
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourcePerformer
- Software

View Specifications::Operational::Structure

Contains the diagrams that document the Operational Structure Viewpoint.

View Specifications::Operational::Structure::Operational Structure

Stakeholders: Business Architects, Systems Engineers, Enterprise Architects, Owners responsible for Operational Agents.
Concerns: identifies the operational exchange requirements between nodes.
Definition: defines operational architecture and exchange requirements necessary to support a specific set of Capability(ies). Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.

**Figure 8:10 - Operational Structure**

**Elements**
- ActualEnvironment
- ActualLocation
- Asset
- Capability
- CapableElement
- Exhibits
- IsCapableToPerform
- KnownResource
- LocationHolder
- OperationalActivity
- OperationalAgent
- OperationalAsset
- OperationalPerformer
- OperationalRole
- ProblemDomain
- OperationalArchitecture

**View Specifications::Operational::Connectivity**
Contains the diagrams that document the Operational Connectivity Viewpoint.
View Specifications::Operational::Connectivity::Operational Connectivity

Stakeholders: Systems Engineers, Architects, Solution Providers.
Concerns: capture the interfaces between Operational Performers.
Definition: summarizes logical exchanges between Operational Performers of information, systems, personnel, energy etc. and the logical activities that produce and consume them. Measurements can optionally be included.
Recommended Implementation: SysML Internal Block Diagram, tabular format.

Figure 8:11 - Operational Connectivity

Elements

- CapabilityConfiguration
- Exchange
- GeoPoliticalExtentType
- InformationElement
- IsCapableToPerform
- MeasurableElement
- MeasurementSet
- NaturalResource
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalActivity
- OperationalAgent
- OperationalConnector
- OperationalExchange
- OperationalExchangeItem
- OperationalInterface
- OperationalPerformer
- OperationalPort
View Specifications::Operational::Processes

Contains the diagrams that document the Operational Processes Viewpoint.

View Specifications::Operational::Processes::Operational Processes

Stakeholders: Business Architect, Systems Engineers, Enterprise Architects
Concerns: captures activity based behavior and flows.
Definition: describes the activities that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them.
Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.

Figure 8:12 - Operational Processes
Elements

- ActivityPerformableUnderCondition
- ActualCondition
- ActualMeasurementSet
- ActualService
- IsCapableToPerform
- MeasurableElement
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod
- OperationalParameter
- OperationalPerformer
- OperationalRole
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- RequiredServiceLevel
- ServiceSpecification
- StandardOperationalActivity
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

View Specifications::Operational::Processes::Operational Processes BPMN Semantics

Stakeholders: Business Architect, Enterprise Architects.
Concerns: captures activity based behavior and flows using BPMN notation.
Definition: describes the BPMN processes that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them using BPMN notation.
Recommended Implementation: BPMN Process Diagram.
Figure 8.13 - Operational Processes BPMN Semantics

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- ActualEnduringTask
- ActualMeasurementSet
- ActualService
- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- EnduringTask
- Exchange
- ExchangeItem
- Implements
- IsCapableToPerform
- MeasurableElement
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod
- OperationalParameter
- OperationalPerformer
- OperationalRole
- PerformsInContext
- Process
- ProcessEdge
- ProcessUsage
- RequiredServiceLevel
- ServiceSpecification
- StandardOperationalActivity

**View Specifications::Operational::States**

Contains the diagrams that document the Operational States Viewpoint.

**View Specifications::Operational::Interaction Scenarios**

Contains the diagrams that document the Operational Interaction Scenarios Viewpoint.

**Figure 8:14 - Operational States**

Elements

- OperationalAgent
- OperationalStateDescription
- StateDescription
- UML2.5Metamodel::StateMachine

**Definition:**
It is a graphical representation of states of an operational OperationalPerformer and how that operational OperationalPerformer responds to various events and actions.

**Recommended Implementation:**
SysML State Machine Diagram.
**View Specifications::Operational::Interaction Scenarios::Operational Interaction Scenarios**

Stakeholders: Systems Engineers, Business Architects.
Concerns: express a time ordered examination of the operational exchanges as a result of a particular operational scenario.
Definition: provides a time-ordered examination of the operational exchanges between participating nodes (OperationalPerformer roles) as a result of a particular operational scenario.
Recommended Implementation: SysML Sequence Diagram, BPMN Collaboration Diagram.

**Figure 8:15 - Operational Interaction Scenarios**

Elements
- InteractionMessage
- InteractionRole
- InteractionScenario
- OperationalActivity
- OperationalAgent
- OperationalExchange
- OperationalInteractionScenario
- OperationalMessage
- OperationalMethod
- OperationalPerformer
- OperationalRole
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message
**View Specifications::Operational::Constraints**

Contains the diagrams that document the Operational Constraints Viewpoint.

**View Specifications::Operational::Constraints::Operational Constraints**

Stakeholders: Systems Engineers, Architects, Program Sponsors
Concerns: define operational limitations, constraints and performance parameters for the enterprise.
Definition: specifies traditional textual operational or business rules that are constraints on the way that business is done in the enterprise. The addition of SysML parametrics provides a computational means of defining operational constraints across the enterprise or within a specific operational context.
Recommended Implementation: tabular format, SysML Block Definition Diagram, SysML Parametric Diagram.

![Operational Constraints Diagram](image)

**Figure 8:16 - Operational Constraints**

Elements

- DataModel
- InformationElement
- OperationalActivity
- OperationalAgent
- OperationalConstraint
- OperationalExchange
- OperationalPerformer
- Rule
- SubjectOfOperationalConstraint

**View Specifications::Operational::Traceability**

Contains the diagrams that document the Operational Traceability Viewpoint.

**View Specifications::Operational::Traceability::Operational Traceability**

Stakeholders: PMs, Enterprise Architects, Business Architects.
Concerns: traceability between capabilities and operational activities and capabilities and operational agents.
Definition: describes the mapping between the capabilities required by an Enterprise and the supporting operational...
activities and operational agents.
Recommended Implementation: matrix format, SysML Block Definition Diagram.

Figure 8:17 - Operational Traceability

Elements

- **Capability**
- **CapabilityElement**
- **Exhibits**
- **MapsToCapability**
- **OperationalActivity**
- **OperationalAgent**
- **OperationalArchitecture**
- **OperationalPerformer**
- **Process**

### 8.1.4 View Specifications::Services

Concerns: specifications of services required to exhibit a Capability.
Definition: shows Service Specifications and required and provided service levels of these specifications required to exhibit a Capability or to support an Operational Activity.

**View Specifications::Services::Taxonomy**

Contains the diagrams that document the Services Taxonomy Viewpoint.

**View Specifications::Services::Taxonomy::Services Taxonomy**

Concerns: service specification types and required and provided service levels of these types.
Definition: shows the taxonomy of types of services and the level of service that they are expected to provide or are required to meet through the display of ActualMeasurements associated with the Provided and Required Service Level.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:18 - Services Taxonomy

Elements

- **ActualMeasurement**
- **ActualMeasurementSet**
- **ActualService**
- **Measurement**
- **PropertySet**
- **ProvidedServiceLevel**
- **RequiredServiceLevel**
- **ServicePolicy**
- **ServiceSpecification**
- **ServiceSpecificationGeneralization**

**View Specifications::Services::Structure**

Contains the diagrams that document the Services Structure Viewpoint.

**View Specifications::Services::Structure::Services Structure**

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: combination of services required to exhibit a capability.
Definition: shows the composition of services and how services are combined into a higher level service required to exhibit a capability or support an operational activity.
Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.
Figure 8:19 - Services Structure

Elements

- InformationElement
- Measurement
- OperationalExchangeItem
- PropertySet
- ServiceConnector
- ServiceInterface
- ServiceMethod
- ServiceParameter
- ServicePort
- ServiceSpecification
- ServiceSpecificationRole

View Specifications::Services::Connectivity

Contains the diagrams that document the Services Connectivity Viewpoint.

View Specifications::Services::Connectivity::Services Connectivity

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: interoperability among services
Definition: specifies service interfaces, e.g., provided and required service operations, to ensure compatibility and reusability of services.
Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram, tabular format.
Figure 8:20 - Services Connectivity

Elements

- **ServiceConnector**
- **ServiceInterface**
- **ServiceMethod**
- **ServiceParameter**
- **ServicePort**
- **ServiceSpecification**
- **ServiceSpecificationRole**

**View Specifications::Services::Processes**

Contains the diagrams that document the Services Processes Viewpoint.

**View Specifications::Services::Processes::Services Processes**

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: the behavior of a service in terms of the operational activities it is expected to support.
Definition: provides detailed information regarding the allocation of service functions to service specifications, and data flows between service functions.
Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.
Figure 8:21 - Services Processes

Elements

- IsCapableToPerform
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- ServiceFunction
- ServiceFunctionAction
- ServiceFunctionEdge
- ServiceMethod
- ServiceParameter
- ServiceSpecification
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Process
- UML2.5Metamodel::ProcessEdge
- UML2.5Metamodel::ProcessOperation
- UML2.5Metamodel::ProcessParameter
- UML2.5Metamodel::ProcessUsage
- UML2.5Metamodel::ServiceMethod
- UML2.5Metamodel::ServiceParameter
- UML2.5Metamodel::ServiceSpecification
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamet
Figure 8:22 - Services Processes BPMN Semantics

Elements

- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- InteractionRole
- IsCapableToPerform
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessUsage
- ServiceFunction
- ServiceFunctionAction
- ServiceFunctionEdge
- ServiceMethod
- ServiceParameter
- ServiceSpecification
- ServiceSpecificationRole

View Specifications::Services::States

Contains the diagrams that document the Services States Viewpoint.

View Specifications::Services::States::Services States

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: the behavior of a service specification in terms of states and events causing transitions between states.
Definition: specifies the possible states a service specification may have, and the possible transitions between those states.
Recommended Implementation: SysML State Machine Diagram.
Figure 8:23 - Services States

Elements

- ServiceSpecification
- ServiceStateDescription
- StateDescription
- UML2.5Metamodel::StateMachine

View Specifications::Services::Interaction Scenarios

Contains the diagrams that document the Services Interaction Scenarios Viewpoint.

View Specifications::Services::Interaction Scenarios::Services Interaction Scenarios

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: the behavior of a service specification in terms of expected time-ordered examination of the interactions between service roles.
Definition: specifies how a service roles interact with each other, service providers and consumers, and the sequence and dependencies of those interactions.
Recommended Implementation: SysML Sequence Diagram.
Figure 8.24 - Services Interaction Scenarios

Elements

- InteractionMessage
- InteractionRole
- InteractionScenario
- ServiceFunction
- ServiceInteractionScenario
- ServiceMessage
- ServiceMethod
- ServiceSpecification
- ServiceSpecificationRole
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

View Specifications::Services::Constraints

Contains the diagrams that document the Services Constraints Viewpoint.

View Specifications::Services::Constraints::Services Constraints

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: service policies that apply to implementations of service specifications.
Definition: specifies traditional textual service policies that are constraints on the way that service specifications are implemented within resources. The addition of SysML parametrics provide a computational means of defining service policies across the enterprise or within a specific service configuration.
Recommended Implementation: tabular format, SysML Parametric Diagram.
Figure 8:25 - Services Constraints

Elements

- Rule
- ServicePolicy
- ServiceSpecification

View Specifications::Services::Roadmap

Contains the diagrams that document the Services Roadmap Viewpoint.

View Specifications::Services::Roadmap::Services Roadmap

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: service specification changes over time.
Definition: provides an overview of how a service specification changes over time. It shows the combination of several service specifications mapped against a timeline.
Recommended Implementation: timeline, SysML Block Definition Diagram, SysML Internal Block Diagram.

Figure 8:26 - Services Roadmap

Elements

- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- ServiceSpecification
View Specifications::Services::Traceability

Contains the diagrams that document the Services Traceability Viewpoint.

View Specifications::Services::Traceability::Services Traceability

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: traceability between operational activities and service specifications that support them.
Definition: depicts the mapping of service specifications to operational activities and how service specifications contribute to the achievement of a capability.
Recommended Implementation: tabular or matrix format.

Figure 8:27 - Services Traceability

Elements

- ActualService
- Capability
- CapableElement
- Consumes
- Exhibits
- OperationalActivity
- ServiceSpecification

8.1.5 View Specifications::Personnel

Stakeholders: Human resources, Solution Providers, PMs.
Concerns: human factors.
Definition: aims to clarify the role of Human Factors (HF) when creating architectures in order to facilitate both Human Factors Integration (HFI) and systems engineering (SE).

View Specifications::Personnel::Taxonomy

Contains the diagrams that document the Personnel Taxonomy Viewpoint.
View Specifications::Personnel::Taxonomy::Personnel Taxonomy

Stakeholders: Human resources, Solution Providers, PMs.
Concerns: organizational resource types.
Definition: shows the taxonomy of types of organizational resources.
Recommended Implementation: SysML Block Definition Diagram.

![Personnel Taxonomy Diagram]

Elements
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- Responsibility

View Specifications::Personnel::Structure

Contains the diagrams that document the Personnel Structure Viewpoint.

View Specifications::Personnel::Structure::Personnel Structure

Stakeholders: Human resources, Solution Providers, PMs.
Concerns: typical organizational structure used to support a capability(ies).
Definition: shows organizational structures and possible interactions between organizational resources.
Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.
Figure 8.29 - Personnel Structure

Elements

- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PostRole
- ResourcePerformer
- ResourceRole
- Responsibility
- SubOrganization

View Specifications::Personnel::Connectivity

Contains the diagrams that document the Personnel Connectivity Viewpoint.

View Specifications::Personnel::Connectivity::Personnel Connectivity

Stakeholders: Solution providers.
Concerns: interaction of organizational resources.
Definition: captures the possible interactions between organizational resources, including command and control relationships. Interactions typically illustrate the fundamental roles and management responsibilities.
Recommended Implementation: tabular format.
Figure 8:30 - Personnel Connectivity

Elements

- Command
- Control
- DataElement
- Environment
- Exchange
- Function
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PropertySet
- Resource
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceInterface
- ResourcePerformer
- ResourcePort
- ResourceRole
- Responsibility
View Specifications::Personnel::Processes

Contains the diagrams that document the Personnel Processes Viewpoint.

View Specifications::Personnel::Processes::Personnel Processes

Stakeholders: Systems engineers, Solution providers.
Concerns: functions that have to be carried out by organizational resources.
Definition: specifies organizational resource functions in relation to resource definitions.
Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram, BPMN Process Diagram as described in the Resources Processes section.

Figure 8:31 - Personnel Processes

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- DataElement
- Function
- FunctionAction
- FunctionEdge
- IsCapableToPerform
- Organization
- OrganizationalResource
View Specifications::Personnel::States

Contains the diagrams that document the Personnel States Viewpoint.

View Specifications::Personnel::States::Personnel States

Stakeholders: Systems Engineers, Software Engineers.
Concerns: capture state-based behavior of an organizational resource.
Definition: it is a graphical representation of states of an organizational resource and how that organizational resource responds to various events and actions.
Recommended Implementation: SysML State Machine Diagram.
View Specifications::Personnel::Interaction Scenarios

Contains the diagrams that document the Personnel Interaction Scenarios Viewpoint.

View Specifications::Personnel::Interaction Scenarios::Personnel Interaction Scenarios

Stakeholders: Software Engineers, Systems Engineers.
Concerns: interactions between organizational resources (roles).
Definition: provides a time-ordered examination of the interactions between organizational resources.
Recommended Implementation: SysML Sequence Diagram, BPMN Collaboration Diagram.

Figure 8:33 - Personnel Interaction Scenarios
Elements

- DataElement
- Function
- InteractionMessage
- InteractionRole
- InteractionScenario
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- ResourceExchange
- ResourceExchangeItem
- ResourceInteractionScenario
- ResourceMessage
- ResourceMethod
- ResourcePerformer
- ResourceRole
- Responsibility
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

**View Specifications::Personnel::Constraints**

Contains the diagrams that document the Personnel Constraints Viewpoint.

**View Specifications::Personnel::Constraints::Personnel Constraints: Competence**

Stakeholders: Systems engineers, Solution providers.
Concerns: allocation of competencies to actual posts.
Definition: specifies requirements for actual organizational resources – by linking competencies and actual posts.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:34 - Personnel Constraints: Competence

Elements

- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResponsibility
- ActualResponsibleResource
- Competence
- CompetenceForRole
- CompetenceToConduct
- Function
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ProvidesCompetence
- RequiresCompetence
- ResourcePerformer
- ResourceRole
- Responsibility

**View Specifications::Personnel::Constraints::Personnel Constraints: Drivers**

Stakeholders: Systems engineers, Solution providers, Human resources.
Concerns: optimization of organizational resource behavior.
Definition: captures the factors that affect, constrain and characterize organizational resource behavior as the basis for
Performance predictions at the level of actual persons and actual organizations. It creates a bridge between static architectural definitions and behavior predictions through executable models. Recommended Implementation: tabular format, SysML Parametric Diagram, SysML Block Definition Diagram.

**Figure 8:35 - Personnel Constraints: Drivers**

Elements

- `ActualMeasurement`
- `ActualMeasurementSet`
- `Function`
- `IsCapableToPerform`
- `MeasurableElement`
- `Measurement`
- `MeasurementSet`
- `Organization`
- `OrganizationalResource`
- `Person`
- `PhysicalResource`
- `Post`
- `Process`
- `ResourceConstraint`
- `ResourcePerformer`
- `Responsibility`
- `Rule`
- `SubjectOfResourceConstraint`

**View Specifications::Personnel::Constraints::Personnel Constraints: Performance**

Stakeholders: Human resources, solution providers.

Concerns: how well an actual organizational resource matches the needs of the actual organization.
Definition: provides a repository for human-related measures (i.e., quality objectives and performance criteria (HFI values)), targets and competences.
Recommended Implementation: SysML Block Definition Diagram.

Figure 8:36 - Personnel Constraints: Performance

Elements
- ActivityPerformableUnderCondition
- ActualCondition
- ActualMeasurement
- ActualMeasurementSet
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualState
- DesiredEffect
- Desirer
- Function
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- **Organization**
- **OrganizationalResource**
- **Person**
- **PhysicalResource**
  - **Post**
  - **Process**
  - **ResourcePerformer**
  - **Responsibility**

**View Specifications::Personnel::Roadmap**

Contains the diagrams that document the Personnel Roadmap Viewpoint.

**View Specifications::Personnel::Roadmap::Personnel Roadmap: Availability**

Concerns: the staffing and training of resources.
Definition: defines the requirements and functions to ensure that actual persons with the right competencies, and in the right numbers, are available to fulfill actual posts.
Recommended Implementation: Timeline, SysML Block Definition Diagram.
Figure 8:37 - Personnel Roadmap: Availability

Elements

- **ActualMeasurement**
- **ActualOrganizationalResource**
- **ActualPerson**
- **ActualPost**
- **ActualProject**
- **ActualProjectMilestone**
- **ActualPropertySet**
- **ActualResource**
- **ActualResponsibleResource**
- **ActualState**
- **Asset**
- **FillsPost**
- **Measurement**
View Specifications::Personnel::Roadmap::Personnel Roadmap: Evolution

Stakeholders: Human resources, Solution Providers.
Concerns: organizational structure changes over time.
Definition: provides an overview of how an organizational structure changes over time. It shows the structure of several organizational structures mapped against a timeline.
Recommended Implementation: timeline, SysML Block Definition Diagram, SysML Internal Block Diagram.

Figure 8:38 - Personnel Roadmap: Evolution

Elements

- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- Responsibility
- VersionedElement
- VersionOfConfiguration

OrganizationalResource
Person
PhysicalResource
Post
Project
ProjectMilestone
PropertySet
ResourceAsset
ResourcePerformer
• VersionSuccession
• WholeLifeConfiguration

**View Specifications::Personnel::Roadmap::Personnel Roadmap: Forecast**

Stakeholders: Human resources, Logisticians, Solution Providers.
Concerns: competencies and skills forecast.
Definition: defines the underlying current and expected supporting competencies and skills of organizational resources.
Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

![Diagram](image)

**Figure 8:39 - Personnel Roadmap: Forecast**

Elements

- ActualEnterprisePhase
- ActualPropertySet
- ActualState
- Asset
- Competence
- Forecast
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- Responsibility
- SubjectOfForecast

**View Specifications::Personnel::Traceability**

Contains the diagrams that document the Personnel Traceability Viewpoint.
**View Specifications::Personnel::Traceability::Personnel Traceability**

Concerns: traceability between operational activities and functions that implements them.
Definition: depicts the mapping of functions (performed by organizational resources) to operational activities and thus identifies the transformation of an operational need into a purposeful function performed by an organizational resource or solution.
Recommended Implementation: Matrix format, SysML Block Definition Diagram.

![Diagram of Personnel Traceability](image)

**Figure 8:40 - Personnel Traceability**

Elements

- **Function**
- **Implements**
- **OperationalActivity**
- **ServiceFunction**

8.1.6 **View Specifications::Resources**

Concerns: definition of solution architectures to implement operational requirements.
Definition: captures a solution architecture consisting of resources, e.g., organizational, software, artifacts, capability configurations, natural resources that implement the operational requirements. Further design of a resource is typically detailed in SysML or UML.

**View Specifications::Resources::Taxonomy**

Contains the diagrams that document the Resources Taxonomy Viewpoint.

**View Specifications::Resources::Taxonomy::Resources Taxonomy**

Stakeholders: Solution Providers, Systems Engineers, IT Architects, Implementers.
Concerns: resource types.
Definition: shows the taxonomy of types of resources.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:41 - Resources Taxonomy

Elements

- Asset
- CapabilityConfiguration
- Implements
- Measurement
- NaturalResource
- OperationalAgent
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PropertySet
- PropertySetGeneralization
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourceExchange
- ResourceMitigation
- ResourcePerformer
- ResourceRole
- Responsibility
- Software
- System
**View Specifications::Resources::Structure**

Contains the diagrams that document the Resources Structure Viewpoint.

**View Specifications::Resources::Structure::Resources Structure**

Concerns: reference the resource structure, connectors and interfaces in a specific context.
Definition: defines the physical resources, e.g., capability configuration(s)/system(s) and interactions necessary to implement a specific set of OperationalPerformer(s). Can be used to represent communications networks and pathways that link communications resources and provides details regarding their configuration.
Recommended Implementation: SysML Internal Block Diagram, SysML Bock Definition Diagram.

**Figure 8:42 - Resources Structure**

Elements

- Asset
- CapabilityConfiguration
- Measurement
- NaturalResource
- PhysicalResource
- PropertySet
- Protocol
- ProtocolImplementation
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourceConnector
- ResourceExchange
- ResourceInterface
- ResourceMitigation
- ResourcePerformer
- ResourcePort

**Figure 8:42 - Resources Structure**

Elements
View Specifications::Resources::Connectivity

Contains the diagrams that document the Resources Connectivity Viewpoint.

View Specifications::Resources::Connectivity::Resources Connectivity

Stakeholders: Systems Engineers, IT Architects, Solution Providers, Implementers.
Concerns: capture the interactions between resources.
Definition: summarizes interactions between resources of information, systems, personnel, natural resources etc. and the functions that produce and consume them. Measurements can optionally be included.
Recommended Implementation: SysML Internal Block Diagram, tabular format.

Figure 8:43 - Resources Connectivity

Elements
- CapabilityConfiguration
- DataElement
- Exchange
- Function
- FunctionAction
- FunctionEdge
- GeoPoliticalExtentType
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- NaturalResource
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- PropertySet
- Resource
- ResourceArchitecture
- ResourceArtifact
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceInterface
- ResourceMitigation
- ResourcePerformer
- ResourcePort
- ResourceRole
- ResourceSignal
- Software
- Technology

View Specifications::Resources::Processes
Contains the diagrams that document the Resources Processes Viewpoint.

View Specifications::Resources::Processes::Resources Processes
Stakeholders: Solution Providers, Systems Engineers, IT Architects.
Concerns: captures activity based behavior and flows.
Definition: describes the functions that are normally conducted in the course of implementing operational activity(ies) in support of capability(ies). It describes the functions, their Inputs/Outputs, function actions and flows between them.
Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.
Figure 8.44 - Resources Processes

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- DataElement
- Function
- FunctionAction
- FunctionEdge
- Implements
- OperationalActivity
- PerformsInContext
- PhysicalResource
- Process
- ProcessEdge
- ProcessParameter
- ProcessUsage
- ResourceArchitecture
- ResourceExchange
- ResourceExchangeItem
- ResourceParameter
- ResourcePerformer
- ResourceRole
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
**View Specifications::Resources::Processes::Resources Processes BPMN Semantics**

Stakeholders: Solution Providers, IT Architects.
Concerns: captures activity based behavior and flows using BPMN.
Definition: describes the functions that are normally conducted in the course of implementing operational activity(ies) in support of capability(ies). It describes the functions, their Inputs/Outputs, function actions and flows between them using BPMN.
Recommended Implementation: BPMN Process Diagram.

**Figure 8:45 - Resources Processes BPMN Semantics**

Elements

- ActivityPerformableUnderCondition
- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- DataElement
View Specifications::Resources::States

Contains the diagrams that document the Resources States Viewpoint.

View Specifications::Resources::States::Resources States

Stakeholders: Systems Engineers, Software Engineers.
Concerns: capture state-based behavior of a resource.
Definition: it is a graphical representation of states of a resource and how that resource responds to various events and actions.
Recommended Implementation: SysML State Machine Diagram.

Figure 8:46 - Resources States

Elements

- ResourcePerformer
- ResourceStateDescription
- StateDescription
- UML2.5Metamodel::StateMachine
View Specifications::Resources::Interaction Scenarios

Contains the diagrams that document the Resources Interaction Scenarios Viewpoint.

View Specifications::Resources::Interaction Scenarios::Resources Interaction Scenarios

Stakeholders: Software Engineers, Systems Engineers.
Concerns: interactions between resources (roles).
Definition: provides a time-ordered examination of the interactions between resources.
Recommended Implementation: SysML Sequence Diagram.

Figure 8:47 - Resources Interaction Scenarios

Elements
- Function
- InteractionMessage
- InteractionRole
- InteractionScenario
- ResourceExchange
- ResourceInteractionScenario
- ResourceMessage
- ResourcePerformer
- ResourceRole
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

View Specifications::Resources::Constraints

Contains the diagrams that document the Resources Constraints Viewpoint.
View Specifications::Resources::Constraints::Resources Constraints

Stakeholders: Systems Engineers, IT Architects, Solution Providers, Implementers.
Concerns: define limitations, constraints and performance parameters for resources, their interactions, performed functions, and data.
Definition: specifies traditional textual rules/non-functional requirements that are constraints on resources, their interactions, performed functions, and data. The addition of SysML parametrics provide a computational means of defining resource constraints within a specific context.
Recommended Implementation: tabular format, SysML Block Definition Diagram, SysML Parametric Diagram, OCL.

Figure 8:48 - Resources Constraints
Elements
- ActualResource
- CapabilityConfiguration
- DataElement
- Function
- NaturalResource
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourceArchitecture
- ResourceArtifact
- ResourceConstraint
- ResourcePerformer
- ResourceRole

Unified Architecture Framework (UAF) Domain Metamodel Version 1.1
View Specifications::Resources::Roadmap

Contains the diagrams that document the Resources Roadmap Viewpoint.

View Specifications::Resources::Roadmap::Resources Roadmap: Evolution

Stakeholders: Systems Engineers, IT Architects, Solution Providers, Implements.
Concerns: resource structure changes over time.
Definition: provides an overview of how a resource structure changes over time. It shows the structure of several resources mapped against a timeline.
Recommended Implementation: timeline, SysML Block Definition Diagram, SysML Internal Block Diagram.

Figure 8:49 - Resources Roadmap: Evolution

Elements

- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- ResourcePerformer
- VersionedElement
- VersionOfConfiguration
- VersionSuccession
- WholeLifeConfiguration
- WholeLifeConfigurationKind

Unified Architecture Framework (UAF) Domain Metamodel Version 1.1
**View Specifications::Resources::Roadmap::Resources Roadmap: Forecast**

Stakeholders: Solution Providers, Systems Engineers, IT Architects.
Concerns: technology forecast.
Definition: defines the underlying current and expected supporting technologies. Expected supporting technologies are those that can be reasonably forecast given the current state of technology, and expected improvements / trends.
Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

**Figure 8:50 - Resources Roadmap: Forecast**

Elements

- `ActualEnterprisePhase`
- `ActualPropertySet`
- `ActualState`
- `CapabilityConfiguration`
- `Forecast`
- `NaturalResource`
- `PhysicalResource`
- `ResourceArtifact`
- `ResourceArchitecture`
- `ResourceMitigation`
- `Software`
- `Technology`
- `forecastPeriod`
- `forecastedUsage`
- `forecastSubject`
- `forecastPeriod`
- `forecastedUsage`
- `forecastSubject`

**View Specifications::Resources::Traceability**

Contains the diagrams that document the Resources Traceability Viewpoint.
View Specifications::Resources::Traceability::Resources Traceability

Concerns: traceability between operational activities and functions that implements them.
Definition: depicts the mapping of functions to operational activities and thus identifies the transformation of an operational need into a purposeful function performed by a resource or solution.
Recommended Implementation: Matrix format, SysML Block Definition Diagram.

Figure 8:51 - Resources Traceability

Elements
- Capability
- CapableElement
- Exhibits
- Function
- Implements
- IsCapableToPerform
- OperationalActivity
- OperationalAgent
- ResourcePerformer
- ServiceFunction

8.1.7 View Specifications::Security

Concerns: addresses the security constraints and information assurance attributes that exist on exchanges between resources and OperationalPerformers
Definition: illustrates the security assets, security constraints, security controls, families, and measures required to address specific security concerns.

View Specifications::Security::Taxonomy

Contains the diagrams that document the Security Taxonomy Viewpoint.
View Specifications::Security::Taxonomy::Security Taxonomy

Stakeholders: Security Architects, Security Engineers.
Concerns: Security assets and security enclaves.
Definition: Defines the hierarchy of security assets and asset owners that are available to implement security, security constraints (policy, guidance, laws and regulations) and details where they are located (security enclaves).
Recommended Implementation: tabular format, SysML Block Definition Diagram.

Figure 8:52 - Security Taxonomy

Elements

- ActualLocation
- Asset
- DataElement
- InformationElement
- LocationHolder
- Measurement
- MeasurementSet
- OperationalAgent
- OperationalArchitecture
- OperationalAsset
- OperationalMitigation
- OperationalPerformer
- PropertySet
- ResourceArchitecture
- ResourceAsset
- ResourceMitigation
- ResourcePerformer
View Specifications::Security::Structure
Contains the diagrams that document the Security Structure Viewpoint.

View Specifications::Security::Structure::Security Structure
Stakeholders: Security Architects, Security Engineers.
Concerns: The structure of security information and where it is used at the operational and resource level.
Definition: Captures the allocation of assets (operational and resource, information and data) across the security enclaves, shows applicable security controls necessary to protect organizations, systems and information during processing, while in storage (bdd), and during transmission (flows on an ibd). This view also captures Asset Aggregation and allocates the usage of the aggregated information at a location through the use of the SecurityProperty.
Recommended Implementation: SysML Internal Block Diagram, SysML Block Definition Diagram.

Figure 8:53 - Security Structure

Elements
- Asset
- DataElement
- DataRole
- InformationElement
- InformationRole
- OperationalAgent
- OperationalAsset
- OperationalConnector
- OperationalExchange
- **OperationalPerformer**
- **OperationalPort**
- **OperationalRole**
- **ResourceAsset**
- **ResourceConnector**
- **ResourceExchange**
- **ResourcePerformer**
- **ResourcePort**
- **ResourceRole**
- **SecurityCategory**

### View Specifications::Security::Connectivity

Contains the diagrams that document the Security Connectivity Viewpoint.

### View Specifications::Security::Connectivity::Security Connectivity

**Stakeholders:** Security Architects, Security Engineers.

**Concerns:** Addresses the security constraints and information assurance attributes that exist on exchanges across resources and across performers.

**Definition:** Lists security exchanges across security assets; the applicable security controls; and the security enclaves that house the producers and consumers of the exchanges. Measurements can optionally be included.

**Recommended Implementation:** SysML Internal Block Diagram, tabular format.

---

**Figure 8:54 - Security Connectivity**
Elements

- Caveat
- MeasurableElement
- MeasurementSet
- OperationalAgent
- OperationalConnector
- OperationalExchange
- OperationalExchangeItem
- OperationalInterface
- OperationalPerformer
- OperationalPort
- OperationalRole
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceInterface
- ResourcePerformer
- ResourcePort
- ResourceRole
- SecurityConstraint
- SubjectOfSecurityConstraint

**View Specifications::Security::Processes**

Contains the diagrams that document the Security Processes Viewpoint.

**View Specifications::Security::Processes::Security Processes**

Stakeholders: Security Architects, Security Engineers.
Concerns: The specification of the Security Control families, security controls, and measures required to address a specific security baseline.
Definition: Provides a set of Security Controls and any possible enhancements as applicable to assets. The activity diagram describes operational or resource level processes that apply (operational level) or implement (resource level) security controls/enhancements to assets located in enclaves and across enclaves. This Security Process view can be instantiated either as a variant of an activity/flow diagram or as a hierarchical work breakdown structure.
Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram, BPMN Process Diagram as described in Operational Processes and Resources Processes sections.
Figure 8:55 - Security Processes

Elements

- Function
- FunctionAction
- IsCapableToPerform
- MeasurableElement
- MeasurementSet
- OperationalActivity
- OperationalActivityAction
- OperationalAgent
- OperationalRole
- PerformsInContext
- ResourcePerformer
- ResourceRole
- SecurityProcess
- SecurityProcessAction

View Specifications::Security::Constraints

Contains the diagrams that document the Security Constraints Viewpoint.
**View Specifications::Security::Constraints::Security Constraints**

**Stakeholders:** Security Architects, Security Engineers, Risk Analysts.

**Concerns:** (i) Security-related policy, guidance, laws and regulations as applicable to assets, (ii) threats, vulnerabilities, and risk assessments as applicable to assets.

**Definition:** (i) Specifies textual rules/non-functional requirements that are security constraints on resources, information and data (e.g. security-related in the form of rules (e.g. access control policy)). A common way of representing access control policy is through the use of XACML (eXtensible Access Control Markup Language), it is expected that implementations of UAF allow users to link security constraints to external files represented in XACML. (ii) Identifies risks, specifies risk likelihood, impact, asset criticality, other measurements and enables risk assessment.

**Recommended Implementation:** tabular or Matrix format, SysML Block Definition Diagram, SysML Parametric Diagram, or OCL.

**Figure 8:56 - Security Constraints**

**Elements**
- **ActualMeasurement**
- **ActualPropertySet**
- **ActualResource**
• ActualResponsibleResource
• ActualRisk
• Affects
• AffectsInContext
• Asset
• AssetRole
• EnhancedSecurityControl
• Enhances
• Measurement
• Mitigates
• OperationalAgent
• OperationalRole
• OrganizationalResource
• OwnsRisk
• OwnsRiskInContext
• PropertySet
• Protects
• ProtectsInContext
• ResourcePerformer
• ResourceRole
• Risk
• Rule
• Satisfy
• SecurityConstraint
• SecurityControl
• SecurityControlFamily
• SecurityProcess
• SubjectOfSecurityConstraint

**View Specifications::Security::Traceability**

Contains the diagrams that document the Security Traceability Viewpoint.

**View Specifications::Security::Traceability::Security Traceability**

Concerns: traceability between risk and risk owner, risk mitigations, and affected asset roles.
Definition: depicts the mapping of a risk to each of the following: risk owner, risk mitigations, and affected asset roles.
Recommended Implementation: Matrix format, SysML Block Definition Diagram.
Figure 8:57 - Security Traceability

Elements

- Affects
- AffectsInContext
- Asset
- AssetRole
- DataRole
- InformationRole
- Mitigates
- OperationalRole
- OwnsRiskInContext
- Protects
- ProtectsInContext
- ResourceRole
- Risk
- Satisfy
- SecurityControl

8.1.8 View Specifications::Projects

Stakeholders: PMs, Project Portfolio Managers, Enterprise Architects.
Concerns: project portfolio, projects and project milestones.
Definition: describes projects and project milestones, how those projects deliver capabilities, the organizations contributing to the projects and dependencies between projects.

View Specifications::Projects::Taxonomy

Contains the diagrams that document the Project Taxonomy Viewpoint.
View Specifications::Projects::Taxonomy::Project Taxonomy

Stakeholders: PMs, Project Portfolio Managers, Enterprise Architects. Concerns: types of projects and project milestones. Definition: shows the taxonomy of types of projects and project milestones. Recommended Implementation: SysML Block Definition Diagram.

![Diagram of Project Taxonomy](image)

Figure 8:58 - Project Taxonomy

Elements
- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- Project
- ProjectMilestone
- ProjectMilestoneRole
- ProjectSequence

View Specifications::Projects::Structure

Contains the diagrams that document the Project Structure Viewpoint.

View Specifications::Projects::Structure::Project Structure

Stakeholders: PMs.
Concerns: relationships between types of projects and project milestones.
Definition: provides a template for an actual project(s) road map(s) to be implemented.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8: Project Structure

Elements

- ActualOrganization
- ActualPost
- ActualProject
- ActualPropertySet
- ActualResponsibleResource
- ActualState
- Project
- ProjectMilestone
- ProjectMilestoneRole
- ResponsibleFor
- StatusIndicators
- ResourceRole

View Specifications::Projects::Connectivity

Contains the diagrams that document the Project Connectivity Viewpoint.

View Specifications::Projects::Connectivity::Project Connectivity

Stakeholders: PMs.
Concerns: relationships between projects and project milestones.
Definition: shows how projects and project milestones are related in sequence.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:60 - Project Connectivity

Elements

- **Project**
- **ProjectMilestone**
- **ProjectMilestoneRole**
- **ResourcePerformer**

**View Specifications::Projects::Processes**

Contains the diagrams that document the Project Processes Viewpoint.

**View Specifications::Projects::Processes::Project Processes**

Stakeholders: PMs.
Concerns: captures project tasks (ProjectActivities) and flows between them.
Definition: describes the ProjectActivities that are normally conducted in the course of projects to support capability(ies) and implement resources. It describes the ProjectActivities, their Inputs/Outputs, ProjectActivityActions and flows between them.
Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram, BPMN Process Diagram as described in Resources Processes section.
Figure 8:61 - Project Processes

Elements

- ActualProject
- DataElement
- Function
- FunctionAction
- FunctionEdge
- GeoPoliticalExtentType
- IsCapableToPerform
- Organization
- OrganizationalResource
- PerformsInContext
- PhysicalResource
- Post
- Project
- ProjectActivity
- ProjectActivityAction
- ResourceExchange
- ResourceExchangeItem
- ResourcePerformer
- ResourceRole
- ResourceSignal

View Specifications::Projects::Roadmap

Contains the diagrams that document the Project Roadmap Viewpoint.
View Specifications::Projects::Roadmap::Project Roadmap

Stakeholders: PMs, Capability Owners, Solution Providers, Enterprise Architects.
Concerns: the product portfolio management; a planning of capability delivery.
Definition: provides a timeline perspective on programs or projects
Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

Figure 8:62 - Project Roadmap

Elements
- ActualProject
- ActualProjectMilestone
- ActualProjectMilestoneRole
- ActualPropertySet
- ActualResource
- ActualState
- CapabilityConfiguration
- FieldedCapability
- MilestoneDependency
- Project
- ProjectMilestone
- ProjectMilestoneRole
- ProjectSequence
- ProjectStatus
- ProjectTheme
- ResourceArchitecture
- ResourcePerformer
- StatusIndicators

View Specifications::Projects::Traceability

Contains the diagrams that document the Project Traceability Viewpoint.
**View Specifications::Projects::Traceability::Project Traceability**

Stakeholders: PMs, Project Portfolio Managers, Enterprise Architects.
Concerns: traceability between capabilities and projects that deliver them.
Definition: depicts the mapping of projects to capabilities and thus identifies the transformation of a capability(ies) into a purposeful implementation via projects.
Recommended Implementation: Matrix format, SysML Block Definition Diagram.

![Figure 8:63 - Project Traceability](image)

Elements
- ActualProject
- ActualProjectMilestone
- ActualResource
- Capability
- CapableElement
- Exhibits
- ResourcePerformer

**8.1.9 View Specifications::Standards**

Concerns: technical and non-technical Standards applicable to the architecture.
Definition: shows the technical, operational, and business Standards applicable to the architecture. Defines the underlying current and expected Standards.

**View Specifications::Standards::Taxonomy**

Contains the diagrams that document the Standards Taxonomy Viewpoint.

**View Specifications::Standards::Taxonomy::Standards Taxonomy**

Concerns: technical and non-technical standards, guidance and policy applicable to the architecture.
Definition: shows the taxonomy of types of technical, operational, and business standards, guidance and policy applicable to the architecture.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:64 - Standards Taxonomy

Elements
- ActualOrganization
- CapabilityConfiguration
- Protocol
- ProtocolStack
- ResourceArchitecture
- ResourcePerformer
- Standard
- StandardOperationalActivity
- UAFEElement

View Specifications::Standards::Structure
Contains the diagrams that document the Standards Structure Viewpoint.

View Specifications::Standards::Structure::Standards Structure
Concerns: the specification of the protocol stack used in the architecture.
Definition: shows the composition of standards required to achieve the architecture's objectives.
Recommended Implementation: SysML Internal Block Diagram.
Figure 8:65 - Standards Structure

Elements

- Protocol
- ProtocolLayer
- ProtocolStack
- Standard

View Specifications::Standards::Roadmap

Contains the diagrams that document the Standards Roadmap Viewpoint.

View Specifications::Standards::Roadmap::Standards Roadmap

Concerns: expected changes in technology-related standards and conventions, operational standards, or business standards and conventions.
Definition: defines the underlying current and expected standards. Expected standards are those that can be reasonably forecast given the current state of technology, and expected improvements / trends.
Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

Figure 8:66 - Standards Roadmap

Elements

- ActualEnterprisePhase
- Forecast
- Protocol
- Standard
- SubjectOfForecast
**View Specifications::Standards::Traceability**

Contains the diagrams that document the Standards Traceability Viewpoint.

**View Specifications::Standards::Traceability::Standards Traceability**

Concerns: standards that need to be taken in account to ensure the interoperability of the implementation of architectural elements.
Definition: shows the applicability of standards to specific elements in the architecture.
Recommended Implementation: tabular format, matrix format, SysML Block Definition Diagram.

---

**Figure 8:67 - Standards Traceability**

Elements
- **Protocol**
- **Standard**
- **UAFEElement**

---

**8.1.10 View Specifications::Actual Resources**

Stakeholders: Solution Providers, Systems Engineers, Business Architects, Human Resources.
Concerns: the analysis, e.g., evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations.
Definition: illustrates the expected or achieved actual resource configurations and actual relationships between them.

**View Specifications::Actual Resources::Structure**

Contains the diagrams that document the Actual Resources Structure Viewpoint.

**View Specifications::Actual Resources::Structure::Actual Resources Structure**

Stakeholders: Solution Providers, Systems Engineers, Business Architects.
Concerns: the analysis, e.g., evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations as it provides a means to capture different solution architectures. The detailed analysis (trade-off, what-if etc.) is carried out using the Resource Constraints view.
Definition: illustrates the expected or achieved actual resource configurations required to meet an operational need.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:68 - Actual Resources Structure

Elements
- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResource
- ActualResponsibility
- ActualResponsibleResource
- CapabilityConfiguration
- FieldedCapability
- Organization
- Person
- Post
- ResourcePerformer
- Responsibility

View Specifications::Actual Resources::Connectivity

Contains the diagrams that document the Actual Resources Connectivity Viewpoint.

View Specifications::Actual Resources::Connectivity::Actual Resources Connectivity

Stakeholders: Solution Providers, Systems Engineers, Business Architects.
Concerns: the communication of actual resource.
Definition: illustrates the actual resource configurations and actual relationships between them.
Recommended Implementation: tabular format, SysML Block Definition Diagram, SysML Internal Block Diagram, SysML Sequence Diagram.
Figure 8:69 - Actual Resources Connectivity

Elements

- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResource
- ActualResourceRelationship
- ActualResponsibility
- ActualResponsibleResource
- FieldedCapability

View Specifications::Actual Resources::Traceability

Contains the diagrams that document the Actual Resources Traceability Viewpoint.

View Specifications::Actual Resources::Traceability::Actual Resources Traceability

Concerns: traceability between operational activities and functions that implements them.
Definition: depicts the mapping of functions to operational activities and thus identifies the transformation of an operational need into a purposeful function performed by a resource or solution.
Recommended Implementation: Matrix format, SysML Block Definition Diagram.
Elements

- ActualResource
- Capability
- CapableElement
- Exhibits

8.1.11 View Specifications::Dictionary

Stakeholders: Architects, users of the architecture, Capability Owners, Systems Engineers, Solution Providers.
Concerns: Definitions for all the elements in the architecture, libraries of environments and measurements.
Definition: Presents all the elements used in an architecture. Can be used specifically to capture:
   a. Elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.
   b. Measurable properties that can be used to support analysis such as KPIs, MoEs, TPIs etc.

View Specifications::Dictionary::Dictionary

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: provides a central reference for a given architecture’s data and metadata. It enables the set of architecture description to stand alone, with minimal reference to outside resources.
Definition: contains definitions of terms used in the given architecture. It consists of textual definitions in the form of a glossary, their taxonomies, and their metadata (i.e., data about architecture data), including metadata for any custom-tailored views. Architects should use standard terms where possible (i.e., terms from existing, approved dictionaries, glossaries, and lexicons).
Recommended Implementation: text, table format.
8.1.12 View Specifications::Summary & Overview

Stakeholders: Executives, PMs, Enterprise Architects.
Concerns: executive-level summary information in a consistent form.
Definition: provides executive-level summary information in a consistent form that allows quick reference and comparison between architectural descriptions. Includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.

View Specifications::Summary & Overview::Summary & Overview

Stakeholders: Decision makers, Solution Providers, Systems Engineers, Software Architects, Business Architects.
Concerns: quick overview of an architecture description and summary of analysis. In the initial phases of architecture development, it serves as a planning guide. Upon completion of an architecture, it provides a summary of findings, and any conducted analysis.
Definition: provides executive-level summary information in a consistent form that allows quick reference and comparison among architectures. The Summary and Overview includes assumptions, constraints, and limitations that may affect high-level decision processes involving the architecture.
Recommended Implementation: text, free form diagram, table format.
Figure 8.72 - Summary & Overview

Elements

- ActualEnterprisePhase
- ActualOrganizationalResource
- ArchitecturalDescription
- ArchitecturalReference
- Architecture
- ArchitectureMetadata
- Concern
- EnterprisePhase
- Metadata
- OperationalArchitecture
- OrganizationalResource
- ResourceArchitecture
- Stakeholder
- View
- Viewpoint
- WholeLifeEnterprise

8.1.13 View Specifications::Requirements


Concerns: requirements traceability.

Definition: used to represent requirements, their properties, and relationships (trace, verify, satisfy, refine) to UAF architectural elements.
View Specifications::Requirements::Requirements

Concerns: provides a central reference for a set of stakeholder needs expressed as requirements, their relationship (via traceability) to more detailed requirements and the solution described by the architecture that will meet those requirements.
Definition: used to represent requirements, their properties, and relationships (trace, verify, satisfy, refine) between each other and to UAF architectural elements.
Recommended Implementation: SysML Requirement Diagram, tabular format, matrix format.

Figure 8:73 - Requirements

Elements
- Refine
- Requirement
- Satisfy
- Trace
- UAFElement
- Verify

8.1.14 View Specifications::Information

Stakeholders: Data Modelers, Software Engineers, Systems Engineers
Concerns: address the information perspective on operational, service, and resource architectures.
Definition: allows analysis of an architecture’s information and data definition aspect, without consideration of implementation specific issues.
Recommended Implementation: SysML Block Definition Diagram.

View Specifications::Information::Information Model

Stakeholders: Data Modelers, Software Engineers, Systems Engineers
Concerns: address the information perspective on operational, service, and resource architectures.
Definition: allows analysis of an architecture’s information and data definition aspect, without consideration of implementation specific issues.
Recommended Implementation: SysML Block Definition Diagram.
8.1.15 View Specifications::Parameters

Stakeholders: Capability owners, Systems Engineers, Solution Providers.
Concerns: identifies measurable properties that can be used to support engineering analysis and environment for the Capabilities
Definition: Shows the measurable properties of something in the physical world and elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.

**View Specifications::Parameters::Parameters: Environment**

Stakeholders: Capability owners, Systems Engineers, Solution Providers.
Concerns: defines the environment for the capabilities.
Definition: shows the elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:75 - Parameters: Environment

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- ActualEnvironment
- ActualLocation
- ActualPropertySet
- ActualResource
- Asset
- Condition
- Environment
- EnvironmentProperty
- GeoPoliticalExtentType
- Location
- LocationHolder
- OperationalRole
- Process
- ResourcePerformer
- ResourceRole

View Specifications::Parameters::Parameters: Measurements

Stakeholders: Capability owners, Systems Engineers, Solution Providers.
Concerns: identifies measurable properties that can be used to support analysis such as KPIs, MoEs, TPIs etc.
Definition: Shows the measurable properties of something in the physical world, expressed in amounts of a unit of measure that can be associated with any element in the architecture.
Recommended Implementation: SysML Block Definition Diagram.
Figure 8:76 - Parameters: Measurements

Elements

- ActualMeasurement
- ActualMeasurementSet
- ActualPropertySet
- ActualService
- ActualState
- Capability
- Competence
- Condition
- EnterprisePhase
- MeasurableElement
- Measurement
- MeasurementSet
- PropertySet
- ProvidedServiceLevel
- RequiredServiceLevel
- Resource
- ServiceInterface
- ServiceSpecification
8.1.16 View Specifications::Other
Contains the diagrams that document the use of BPMN, NIEM, IEPPV in the context of UAF.

View Specifications::Other::BPMN
Stakeholders: Business Architects, Enterprise Architects
Concerns: captures activity based behavior and flows.
Definition: describes the activities that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them using BPMN.
Recommended Implementation: BPMN Process Diagram.

![BPMN Diagram]

Figure 8:77 - BPMN
Elements
- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- Exchange
• ExchangeItem
• InteractionMessage
• InteractionRole
• InteractionScenario
• Process
• ProcessEdge
• ProcessUsage

View Specifications::Other::IEPPV

Concerns: information exchanges, information interfaces, information interoperability, information sharing and safeguarding.
Definition: UAFP supports information modeling and traceability to IEPPV model elements using the IEPPV-defined elements: Message, SemanticElement, and FilteredSemanticElement, used to represent data, properties/attributes, structure, format, and relationships. The IEPPV profile enables the specification of the policies, rules and constraints governing the packaging (assembly, transformation, marking, redaction) of data elements conforming to information sharing and safeguarding requirements. The IEPPV profile also governs the processing (parsing, transformation, and marshalling) received information and data element.
Recommended Implementation: UML Class Diagram, SysML Block Diagram.

Figure 8:78 - IEPPV

Elements
• Abstraction
• FilteredSemanticElement
• InformationElement
• InformationSpecification
• Message
• OperationalExchange
• OperationalExchangeItem

Conveys
**View Specifications::Other::NIEM**


Concerns: information exchanges, information interoperability, data schema.

Definition: A specification representing the structure, semantics, and relationships of data objects that satisfy an information exchange requirement. Used for organizing and packaging Model Package Descriptions (MPDs) and Information Exchange Package Documentation (IEPD) as defined by the National Information Exchange Model (NIEM).

An IEPD is a type of MPD. The NIEM MPD defines an Enterprise Information Exchange Model (EIEM) as an MPD that contains NIEM-conforming schemas that define and declare data components to be consistently reused in the IEPDs of an enterprise. An EIEM is a collection of schemas organized into a collection of subset schemas and one or more extension schemas. An information sharing enterprise creates and maintains an EIEM.

Recommended Implementation: UML Class Diagram, SysML Block Diagram.

---

**Figure 8:79 - NIEM**

Elements

- Abstraction
- DataElement
- DataModel
- DataObject
- InformationElement
- InformationModel
9 Domain Metamodel (DMM) Elements

9.1 Domain MetaModel

This package contains the elements of the DMM.

9.1.1 Domain MetaModel::Metadata

Stakeholders: Enterprise Architects, people who want to discover the architecture, Technical Managers.

Concerns: Captures meta-data relevant to the entire architecture

Definition: Provide information pertinent to the entire architecture. Present supporting information rather than architectural models.

Domain MetaModel::Metadata::Taxonomy

ArchitectureMetadata

Package: Taxonomy

isAbstract: No

Generalization: Metadata

Description

Information associated with an ArchitecturalDescription, that supplements the standard set of tags used to summarize the Architecture. It states things like what methodology was used, notation, etc.

Figure 9:1 - ArchitectureMetadata

InteractionScenarioGeneralization

Package: Taxonomy

isAbstract: No

Generalization: UML2.5Metamodel::Generalization, MeasurableElement

Description

An InteractionScenarioGeneralization is a taxonomic relationship between a more general InteractionScenario and a more specific InteractionScenario.
**Metadata**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** *MeasurableElement*

**Description**

A comment that can be applied to any element in the architecture. The attributes associated with this element details the relationship between the element and its related dublinCoreElement, metaDataScheme, category and name. This allows the element to be referenced using the Semantic Web.

**Attributes**

- **category** : String [0..1]  
  Defines the category of a Metadata element example: http://purl.org/dc/terms/abstract.
- **dublinCoreTag** : String [0..1]  
  A metadata category that is a DublinCore tag.
- **metaDataScheme** : String [0..1]  
  A representation scheme that defines a set of Metadata.
- **name** : String [0..1]  
  The name of the Metadata.

**ProcessGeneralization**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** *UML2.5Metamodel::Generalization, MeasurableElement*
Description
A ProcessGeneralization is a taxonomic relationship between a more general Process and a more specific Process.

Figure 9:4 - ProcessGeneralization

PropertySetGeneralization
Package: Taxonomy
isAbstract: No
Generalization: UML2.5Metamodel::Generalization, MeasurableElement

Description
A PropertySetGeneralization is a taxonomic relationship between a more general PropertySet and a more specific PropertySet.

Figure 9:5 - PropertySetGeneralization

StateDescriptionGeneralization
Package: Taxonomy
isAbstract: No
Generalization: UML2.5Metamodel::Generalization, MeasurableElement
A StateDescriptionGeneralization is a taxonomic relationship between a more general StateDescription and a more specific StateDescription.

![Diagram of StateDescriptionGeneralization](image)

**Domain MetaModel::Metadata::Structure**

**EnvironmentProperty**
- **Package:** Structure
- **isAbstract:** No
- **Generalization:** MeasurableElement

**Description**
A property of an Environment that is typed by a Condition. The kinds of Condition that can be represented are Location, GeoPoliticalExtentType and Environment.

![Diagram of EnvironmentProperty](image)

**Domain MetaModel::Metadata::Connectivity**

**Exchange**
- **Package:** Connectivity
- **isAbstract:** Yes
- **Generalization:** MeasurableElement, BPMN2Metamodel::MessageFlow, SubjectOfSecurityConstraint

**Description**
Abstract tuple, grouping OperationalExchanges and ResourceExchanges that exchange Resources.
Figure 9:8 - Exchange

**ExchangeItem**

**Package**: Connectivity  
**isAbstract**: Yes  
**Generalization**: BPMN2Metamodel::BPMNMessage

**Description**
An abstract grouping for elements that defines the types of elements that can be exchanged between Assets and conveyed by an Exchange.

Figure 9:9 - ExchangeItem

**Resource**

**Package**: Connectivity  
**isAbstract**: Yes  
**Generalization**: PropertySet

**Description**
Abstract type grouping all elements that can be conveyed by an Exchange.
Domain MetaModel::Metadata::Processes

ActivityPerformableUnderCondition

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description
The ActualCondition under which an Activity is performed.

IsCapableToPerform

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description
A tuple defining the traceability between the structural elements to the Activities that they can perform.
**IsCapableToPerform**

**Package:** Processes

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple that relates an OperationalAction to an OperationalRole, or a FunctionAction to a ResourceRole. It indicates that the action can be carried out by the role when used in a specific context or configuration.

**PerformsInContext**

**Package:** Processes

**isAbstract:** Yes

**Generalization:** MeasurableElement, UML2.5Metamodel::Activity, BPMN2Metamodel::Process

**Description**

An abstract type that represents a behavior or process (i.e., a Function or OperationalActivity) that can be performed by a Performer.
**ProcessEdge**

**Package:** Processes

**isAbstract:** Yes

**Generalization:** MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::ActivityEdge, BPMN2Metamodel::Process

**Description**

An abstract type that represents a behavior or process (i.e., a Function or OperationalActivity) that can be performed by a Performer.
Figure 9:16 - ProcessEdge

**ProcessOperation**

**Package:** Processes  
**isAbstract:** Yes  
**Generalization:** MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::Operation

**Description**  
An abstract type that represents a behavior or process (i.e., a Function or OperationalActivity) that can be performed by a Performer.

Figure 9:17 - ProcessOperation

**ProcessParameter**

**Package:** Processes  
**isAbstract:** Yes  
**Generalization:** MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::CallBehaviorAction, UML2.5Metamodel::Parameter

**Description**  
An abstract type that represents a behavior or process (i.e., a Function or OperationalActivity) that can be performed by a Performer.
**ProcessParameter**

**Package:** Processes  
**isAbstract:** Yes  
**Generalization:** MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::CallBehaviorAction, BPMN2Metamodel::CallActivity  
**Description**
An abstract type that represents a behavior or process (i.e., a Function or OperationalActivity) that can be performed by a Performer.

**Domain MetaModel::Metadata::States**

**StateDescription**

**Package:** States  
**isAbstract:** Yes  
**Generalization:** UML2.5Metamodel::StateMachine  
**Description**
An abstract type that represents a state machine (i.e., an OperationalStateDescription or ResourceStateDescription), depicting how the Asset responds to various events and the actions.
Domain MetaModel::Metadata::Interaction Scenarios

InteractionMessage
Package: Interaction Scenarios
isAbstract: Yes
Generalization: MeasurableElement, UML2.5Metamodel::Activity, BPMN2Metamodel::Process, UML2.5Metamodel::Interaction, UML2.5Metamodel::Message

Description
An abstract type that groups several types of messages used in the InteractionScenario.

InteractionRole
Package: Interaction Scenarios
isAbstract: Yes
Generalization: BPMN2Metamodel::ResourceRole

Description
An abstract type that represents an individual participant in the InteractionScenario.
**InteractionScenario**

**Package:** Interaction Scenarios  
**isAbstract:** Yes  
**Generalization:** MeasurableElement, UML2.5Metamodel::Activity, BPMN2Metamodel::Process, UML2.5Metamodel::Interaction

**Description**

An abstract type that specifies interactions between Assets, like ResourcePerformers, and ServiceSpecifications.

---

**Domain MetaModel::Metadata::Information**

**Information**

**Package:** Information  
**isAbstract:** No  
**Generalization:** MeasurableElement

**Description**

A comment that describes the state of an item of interest in any medium or form -- and is communicated or received.
**Domain MetaModel::Metadata::Constraints**

**Rule**

**Package:** Constraints  
**isAbstract:** Yes  
**Generalization:** MeasurableElement

**Description**

An abstract type for all types of constraint (i.e., an OperationalConstraint could detail the rules of accountancy best practice).
Domain MetaModel::Metadata::Traceability

ArchitecturalReference

**Package**: Traceability

**isAbstract**: No

**Generalization**: MeasurableElement

**Description**

A tuple that specifies that one architectural description refers to another.

![ArchitecturalReference Diagram]

**Figure 9:25 - ArchitecturalReference**

**Implements**

**Package**: Traceability

**isAbstract**: No

**Generalization**: MeasurableElement

**Description**

A tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (for example tracing the Functions to the OperationalActivities) in the lower level of abstraction.
9.1.2 Domain MetaModel::Strategic

Domain MetaModel::Strategic::Taxonomy

**Capability**

- **Package:** Taxonomy
- **isAbstract:** No
- **Generalization:** PropertySet, Desirer

**Description**

A high-level specification of the enterprise's ability to execute a specified course of action.
Figure 9:27 - Capability

**CapabilityGeneralization**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** [PropertySetGeneralization](#)

**Description**  
A CapabilityGeneralization is a taxonomic relationship between a more general Capability and a more specific Capability.

---

Figure 9:28 - CapabilityGeneralization

**Domain MetaModel::Strategic::Structure**

**ActualEnduringTask**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** [CapabilityElement](#), [ActualPropertySet](#)

**Description**  
An actual undertaking recognized by an enterprise as being essential to achieving its goals - i.e., a strategic specification of what the enterprise does.
**ActualEnterprisePhase**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** CapableElement, ActualPropertySet, Achiever  
**Description**  
An individual that describes the phase of an actual enterprise endeavor.

**CapabilityRole**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** PropertySet, Desirer, MeasurableElement  
**Description**  
A high-level specification of the enterprise's ability to execute a specified course of action.
**EnduringTask**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** PropertySet

**Description**

A type of template behavior recognized by an enterprise as being essential to achieving its goals - i.e., a template for a strategic specification of what the enterprise does.

**EnterpriseGoal**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** PropertySet

**Description**

A statement about a state or condition of the enterprise to be brought about or sustained through appropriate Means. An EnterpriseGoal amplifies an EnterpriseVision that is, it indicates what must be satisfied on a continuing basis to effectively attain the EnterpriseVision. BMM: OMG dtc-13-08-24.
Attributes

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**

**Package**: Structure  
**isAbstract**: No  
**Generalization**: PropertySet  

**Description**

A type of a current or future state of the enterprise.

Figure 9:34 - EnterprisePhase

**EnterpriseVision**

**Package**: Structure  
**isAbstract**: No  
**Generalization**: PropertySet  

**Description**

A Vision describes the future state of the enterprise, without regard to how it is to be achieved. BMM: OMG dtc-13-08-24.

Figure 9:35 - EnterpriseVision

**StructuralPart**

**Package**: Structure  
**isAbstract**: No
Generalization: **MeasurableElement**

Description

A current or future state of the wholeLifeEnterprise or another EnterprisePhase.

![Diagram](image)

**Figure 9:36 - StructuralPart**

**TemporalPart**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** **MeasurableElement**

Description

A current or future state of the wholeLifeEnterprise or another EnterprisePhase.

![Diagram](image)

**Figure 9:37 - TemporalPart**

**VisionStatement**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** **MeasurableElement**

Description

A type of comment that describes the future state of the enterprise, without regard to how it is to be achieved. BMM: OMG dtc-13-08-24.
**WholeLifeEnterprise**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** EnterprisePhase

**Description**

A WholeLifeEnterprise is a purposeful endeavor of any size involving people, organizations and supporting systems. It is made up of TemporalParts and StructuralParts.

---

**Domain MetaModel::Strategic::Connectivity**

**CapabilityDependency**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** MeasurableElement

**Description**

A tuple that asserts that one CapabilityDependency is dependent from another.
isAbstract: No
Generalization: MeasurableElement

Figure 9:41 - CapabilityRoleDependency

Domain MetaModel::Strategic::States

AchievedEffect
Package: States
isAbstract: No
Generalization: MeasurableElement

Description
A tuple that exists between an ActualState (e.g., observed/measured during testing) of an element that attempts to achieve a DesiredEffect and an Achiever.

Figure 9:42 - AchievedEffect

Achiever
Package: States
isAbstract: Yes
Generalization: UAFElement

Description
An ActualResource, ActualProject or ActualEnterprisePhase that can deliver a DesiredEffect.
**DesiredEffect**

**Package:** States  
**isAbstract:** No  
**Generalization:** [MeasurableElement](#)

**Description**

A tuple relating the Desirer (a Capability or OrganizationalResource) to an ActualState.

**Desirer**

**Package:** States  
**isAbstract:** Yes  
**Generalization:** [UAFEElement](#)

**Description**

Abstract type used to group architecture elements that might desire a particular effect.
Domain MetaModel::Strategic::Traceability

CapabilityForTask

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description
A tuple that asserts that a Capability is required in order for an Enterprise to conduct a phase of an EnduringTask.

Figure 9:46 - CapabilityForTask

CapableElement

Package: Traceability
isAbstract: Yes
Generalization: UAFElememt

Description
An abstract type that represents a structural element that can exhibit capabilities.
**Exhibits**

**Package:** Traceability

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple that exists between a CapableElement and a Capability that it meets under specific environmental conditions.

**MapsToCapability**

**Package:** Traceability

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple denoting that an Activity contributes to providing a Capability.
Figure 9:49 - MapsToCapability

**OrganizationInEnterprise**

**Package:** Traceability  
**isAbstract:** No  
**Generalization:** MeasurableElement

**Description**
A tuple relating an ActualOrganization to an ActualEnterprisePhase to denote that the ActualOrganization plays a role or is a stakeholder in an ActualEnterprisePhase.

Figure 9:50 - OrganizationInEnterprise

**9.1.3 Domain MetaModel::Operational**

**Domain MetaModel::Operational::Taxonomy**

**ArbitraryConnector**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** MeasurableElement

**Description**
Represents a visual indication of a connection used in high level operational concept diagrams.
ConceptItem

Package: Taxonomy
isAbstract: Yes
Generalization: UAFElement

Description

Abstract, an item which may feature in a HighLevelOperationalConcept.

HighLevelOperationalConcept

Package: Taxonomy
isAbstract: No
Generalization: PropertySet

Description

Describes the Resources and Locations required to meet an operational scenario from an integrated systems point of view. It is used to communicate overall quantitative and qualitative system characteristics to stakeholders.
Domain MetaModel::Operational::Structure

**KnownResource**

*Package*: Structure  
*isAbstract*: No  
*Generalization*: OperationalPerformer, ResourcePerformer  

*Description*

Asserts that a known ResourcePerformer constrains the implementation of the OperationalPerformer that plays the role in the OperationalArchitecture.

**OperationalAgent**

*Package*: Structure  
*isAbstract*: Yes  
*Generalization*: SubjectOfOperationalConstraint, CapableElement, OperationalAsset, Desirer  

*Description*

An abstract type grouping OperationalArchitecture and OperationalPerformer.
Figure 9:55 - OperationalAgent

**OperationalArchitecture**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** OperationalAgent, Architecture

**Description**

A type used to denote a model of the Architecture, described from the Operational perspective.

Figure 9:56 - OperationalArchitecture

**OperationalMethod**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** ProcessOperation

**Description**

A behavioral feature of an OperationalAgent whose behavior is specified in an OperationalActivity.
**OperationalParameter**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** ProcessParameter

**Description**

A type that represents inputs and outputs of an OperationalActivity. It is typed by an OperationalExchangeItem.

**Figure 9:58 - OperationalParameter**

**OperationalPerformer**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** OperationalAgent

**Description**

A logical entity that IsCapableToPerform OperationalActivities which produce, consume and process Resources.
Figure 9:59 - OperationalPerformer

**OperationalRole**

*Package:* Structure  
*isAbstract:* No  
*Generalization:* LocationHolder, AssetRole, InteractionRole

**Description**

Usage of an OperationalPerformer or OperationalArchitecture in the context of another OperationalPerformer or OperationalArchitecture. Creates a whole-part relationship.

Figure 9:60 - OperationalRole

**ProblemDomain**

*Package:* Structure  
*isAbstract:* No  
*Generalization:* OperationalRole
Description
A property associated with an OperationalArchitecture, used to specify the scope of the problem.

Figure 9:61 - ProblemDomain

Domain MetaModel::Operational::Connectivity

OperationalConnector

Package: Connectivity
isAbstract: No
Generalization: MeasurableElement

Description
A Connector that goes between OperationalRoles representing a need to exchange Resources. It can carry a number of OperationalExchanges.

Figure 9:62 - OperationalConnector

OperationalExchange

Package: Connectivity
isAbstract: No
Generalization: Exchange, SubjectOfOperationalConstraint

Description
Asserts that a flow can exist between OperationalPerformers (i.e., flows of information, people, materiel, or energy).
Figure 9:63 - OperationalExchange
Attributes
trustLevel : Real[0..1]  Captures the directional arbitrary level of trust related to an OperationalExchange between two OperationalPerformers.

OperationalExchangeItem
Package: Connectivity
isAbstract: Yes
Generalization: Resource, SubjectOfSecurityConstraint, ExchangeItem

Description
An abstract grouping for elements that defines the types of elements that can be exchanged between OperationalPerformers and conveyed by an OperationalExchange.

Figure 9:64 - OperationalExchangeItem
**OperationalInterface**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** PropertySet  

**Description**

A declaration that specifies a contract between the OperationalPerformer it is related to, and any other OperationalPerformers it can interact with.

![OperationalInterface Diagram](image)

**Figure 9:65 - OperationalInterface**

**OperationalPort**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** MeasurableElement  

**Description**

An interaction point for an OperationalAgent through which it can interact with the outside environment and which is defined by an OperationalInterface.

![OperationalPort Diagram](image)

**Figure 9:66 - OperationalPort**

**OperationalSignal**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** SubjectOfOperationalConstraint, OperationalExchangeItem
Description

An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities that the OperationalPerformers are capable of performing (see IsCapableToPerform).

**Figure 9:67 - OperationalSignal**

**Domain MetaModel::Operational::Processes**

**OperationalActivity**

*Package:* Processes  
*isAbstract:* No  
*Generalization:* SubjectOfOperationalConstraint, Process

**Description**

An Activity that captures a logical process, specified independently of how the process is carried out.

**Figure 9:68 - OperationalActivity**

**OperationalActivityAction**

*Package:* Processes

**Figure 9:68 - OperationalActivity**
**isAbstract**: No  
**Generalization**: ProcessUsage

**Description**  
A call of an OperationalActivity in the context of another OperationalActivity.

![Figure 9:69 - OperationalActivityAction](image)

**OperationalActivityEdge**  
**Package**: Processes  
**isAbstract**: Yes  
**Generalization**: ProcessEdge

**Description**  
A tuple that shows the flow of Resources (objects/information) between OperationalActivityActions.

![Figure 9:70 - OperationalActivityEdge](image)

**StandardOperationalActivity**  
**Package**: Processes  
**isAbstract**: No  
**Generalization**: OperationalActivity

**Description**  
A sub-type of OperationalActivity that is a standard operating procedure.
**Domain MetaModel::Operational::States**

**OperationalStateDescription**

**Package:** States  
**isAbstract:** No  
**Generalization:** MeasurableElement, StateDescription

**Description**

A state machine describing the behavior of an OperationalPerformer, depicting how the OperationalPerformer responds to various events and the actions.

**Domain MetaModel::Operational::Interaction Scenarios**

**OperationalInteractionScenario**

**Package:** Interaction Scenarios  
**isAbstract:** No  
**Generalization:** InteractionScenario

**Description**

A specification of the interactions between OperationalPerformers in an OperationalArchitecture.
**OperationalMessage**

**Package:** Interaction Scenarios  
**isAbstract:** No  
**Generalization:** InteractionMessage  

**Description**  
Message for use in an OperationalInteractionScenario which carries any of the subtypes of OperationalExchange.

**Domain MetaModel::Operational::Information**

**InformationElement**  
**Package:** Information  
**isAbstract:** No  
**Generalization:** SubjectOfOperationalConstraint, OperationalAsset, OperationalExchangeItem  

**Description**  
An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities that the OperationalPerformers are capable to perform (see IsCapableToPerform).
Domain MetaModel::Operational::Constraints

OperationalConstraint
Package: Constraints
isAbstract: No
Generalization: Rule
Description
A Rule governing an operational architecture element, i.e., OperationalPerformer, OperationalActivity, InformationElement etc.

SubjectOfOperationalConstraint
Package: Constraints
isAbstract: Yes
Generalization: UAFElement
Description
An abstract type grouping element that can be the subject of an OperationalConstraint.

Unified Architecture Framework (UAF) Domain Metamodel Version 1.1
9.1.4 Domain MetaModel::Services

**Stakeholders:** Enterprise Architects, Solution Providers, Systems Engineers, Software Architects, Business Architects.

**Concerns:** specifications of services required to exhibit a Capability.

**Definition:** shows Service Specifications and required and provided service levels of these specifications required to exhibit a Capability or to support an Operational Activity.

**Domain MetaModel::Services::Taxonomy**

**ServiceSpecification**

**Package:** Taxonomy

**isAbstract:** No

**Generalization:** PropertySet, VersionedElement, CapableElement

**Description**

The specification of a set of functionalities provided by one element for the use of others.

**Figure 9:78 - ServiceSpecification**

**ServiceSpecificationGeneralization**

**Package:** Taxonomy

**isAbstract:** No

**Generalization:** PropertySetGeneralization

**Description**

A ServiceSpecificationGeneralization is a taxonomic relationship between a more general ServiceSpecification and a more specific ServiceSpecification.
Domain MetaModel::Services::Structure

ServiceConnector

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description
A channel for exchange between two ServiceSpecifications. Where one acts as the consumer of the other.

ServiceMethod

Package: Structure
isAbstract: No
Generalization: ProcessOperation

Description
A behavioral feature of a ServiceSpecification whose behavior is specified in a ServiceFunction.
**Figure 9:81 - ServiceMethod**

**ServiceParameter**

**Package:** Structure

**isAbstract:** No

**Generalization:** ProcessParameter

**Description**

A type that represents inputs and outputs of a ServiceFunction, represents inputs and outputs of a ServiceSpecification.

**Figure 9:82 - ServiceParameter**

**ServiceSpecificationRole**

**Package:** Structure

**isAbstract:** No

**Generalization:** MeasurableElement, InteractionRole

**Description**

A behavioral feature of a ServiceSpecification whose behavior is specified in a ServiceFunction.
**Domain MetaModel::Services::Connectivity**

**ServiceInterface**

- **Package:** Connectivity
- **isAbstract:** No
- **Generalization:** PropertySet

**Description**

A contract that defines the ServiceMethods and ServiceMessageHandlers that the ServiceSpecification realizes.

**ServicePort**

- **Package:** Connectivity
- **isAbstract:** No
- **Generalization:** MeasurableElement

**Description**

An interaction point for a ServiceSpecification through which it can interact with the outside environment and which is defined by a ServiceInterface.
Domain MetaModel::Services::Processes

ServiceFunction

Package: Processes
isAbstract: No
Generalization: Process

Description
An Activity that describes the abstract behavior of ServiceSpecifications, regardless of the actual implementation.

ServiceFunctionAction

Package: Processes
isAbstract: No
Generalization: ProcessUsage

Description
A call of a ServiceFunction in the context of another ServiceFunction.
**ServiceFunctionEdge**

**Package:** Processes  
**isAbstract:** Yes  
**Generalization:** ProcessEdge

**Description**

A tuple that shows the flow of Resources (objects/information) between OperationalActivityActions.

**Domain MetaModel::Services::States**

**ServiceStateDescription**

**Package:** States  
**isAbstract:** No  
**Generalization:** MeasurableElement, StateDescription

**Description**

A state machine describing the behavior of a ServiceSpecification, depicting how the ServiceSpecification responds to various events and the actions.
Domain MetaModel::Services::Interaction Scenarios

ServiceInteractionScenario

Package: Interaction Scenarios
isAbstract: No
Generalization: InteractionScenario

Description
A specification of the interactions between ServiceSpecifications.

ServiceMessage

Package: Interaction Scenarios
isAbstract: No
Generalization: InteractionMessage

Description
Message for use in a Service Event-Trace.
**Domain MetaModel::Services::Constraints**

**ServicePolicy**

- **Package:** Constraints
- **isAbstract:** No
- **Generalization:** Rule

**Description**

A constraint governing the use of one or more ServiceSpecifications.

**Domain MetaModel::Services::Traceability**

**Consumes**

- **Package:** Traceability
- **isAbstract:** No
- **Generalization:** MeasurableElement

**Description**

A tuple that asserts that an OperationalActivity make use of a service.
**9.1.5 Domain MetaModel::Personnel**

**Stakeholders:** Human resources, Solution Providers, PMs.

**Concerns:** human factors.

**Definition:** aims to clarify the role of Human Factors (HF) when creating architectures in order to facilitate both Human Factors Integration (HFI) and systems engineering (SE).

**Domain MetaModel::Personnel::Taxonomy**

**Organization**

**Package:** Taxonomy

**isAbstract:** No

**Generalization:** OrganizationalResource

**Description**

A group of OrganizationalResources (Persons, Posts, Organizations and Responsibilities) associated for a particular purpose.
Description
An abstract type for Organization, Person, Post and Responsibility.

**Person**
*Package*: Taxonomy
*isAbstract*: No
*Generalization*: OrganizationalResource

Description
A type of a human being used to define the characteristics that need to be described for ActualPersons (e.g., properties such as address, telephone number, nationality, etc.).

**Post**
*Package*: Taxonomy
*isAbstract*: No
*Generalization*: OrganizationalResource

Description
A type of job title or position that a person can fill (e.g., Lawyer, Solution Architect, Machine Operator or Chief Executive Officer).


**Responsibility**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** OrganizationalResource

Description  
The type of duty required of a Person or Organization.

**Domain MetaModel::Personnel::Structure**

**PostRole**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** OrganizationalResource, ResourceRole  

Description  
A usage of a post in the context of another OrganizationalResource. Creates a whole-part relationship.
**SubOrganization**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** [OrganizationalResource](#), [ResourceRole](#)

**Description**

A type of a human being used to define the characteristics that need to be described for ActualPersons (e.g., properties such as address, telephone number, nationality, etc.).

**Domain MetaModel::Personnel::Connectivity**

**Command**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** [ResourceExchange](#)

**Description**

A type of ResourceExchange that asserts that one OrganizationalResource commands another.
Figure 9:101 - Command

**Control**

Package: Connectivity  
isAbstract: No  
Generalization: ResourceExchange

Description  
A type of ResourceExchange that asserts that one PhysicalResource controls another PhysicalResource (i.e., the driver of a vehicle controlling the vehicle speed or direction).

Figure 9:102 - Control

**Domain MetaModel::Personnel::Interaction Scenarios**

ResourceInteractionScenario

Package: Interaction Scenarios  
isAbstract: No  
Generalization: InteractionScenario

Description  
A specification of the interactions between ResourcePerformers in a ResourceArchitecture.
Domain MetaModel::Personnel::Constraints

Competence
Package: Constraints
isAbstract: No
Generalization: PropertySet, SubjectOfForecast
Description
A specific set of abilities defined by knowledge, skills and aptitude.

CompetenceForRole
Package: Constraints
isAbstract: No
Generalization: MeasurableElement
Description
A tuple used to associate an organizational role with a specific set of required competencies.
Figure 9:105 - CompetenceForRole

**RequiresCompetence**

**Package:** Constraints

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple that asserts that an ActualOrganizationalResource is required to have a specific set of Competencies.

Figure 9:106 - RequiresCompetence

**Domain MetaModel::Personnel::Roadmap**

**FillsPost**

**Package:** Roadmap

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple that asserts that an ActualPerson fills an ActualPost.
Attributes

endDate : ISO8601DateTime[0..1]  End date of an ActualPerson filling an ActualPost.
startDate : ISO8601DateTime[0..1]  Start date of an ActualPerson filling an ActualPost.

Domain MetaModel::Personnel::Traceability

CompetenceToConduct

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to associate a Function with a specific set of Competencies needed to conduct the Function.

9.1.6 Domain MetaModel::Resources

Domain MetaModel::Resources::Taxonomy

CapabilityConfiguration

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture

Description

A composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability.
NaturalResource

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** PhysicalResource

**Description**  
Type of physical resource that occurs in nature such as oil, water, gas or coal.

PhysicalResource

**Package:** Taxonomy  
**isAbstract:** Yes  
**Generalization:** ResourcePerformer

**Description**  
An abstract type defining physical resources (i.e., OrganizationalResource, ResourceArtifact and NaturalResource).
**ResourceArchitecture**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ResourcePerformer, Architecture  
**Description**  
A type used to denote a model of the Architecture, described from the ResourcePerformer perspective.

![ResourceArchitecture Diagram](image)

**Figure 9:112 - ResourceArchitecture**

**ResourceArtifact**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** PhysicalResource  
**Description**  
A type of man-made object that contains no human beings (i.e., satellite, radio, petrol, gasoline, etc.).

![ResourceArtifact Diagram](image)

**Figure 9:113 - ResourceArtifact**

**ResourcePerformer**

**Package:** Taxonomy  
**isAbstract:** Yes  
**Generalization:** ResourceExchangeItem, SubjectOfResourceConstraint, OperationalExchangeItem, SubjectOfForecast, CapableElement, Desirer, VersionedElement, ResourceAsset  
**Description**  
An abstract grouping of elements that can perform Functions.
Attributes

isStandardConfiguration : Boolean[] Indicates if the ResourcePerformer is StandardConfiguration, default=false.

Software

Package: Taxonomy
isAbstract: No
Generalization: ResourceArtifact

Description
A sub-type of ResourceArtifact that specifies an executable computer program.

System

Package: Taxonomy
isAbstract: No
**Generalization:** ResourceArchitecture

**Description**

An integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements (INCOSE SE Handbook V4, 2015).

---

**Domain MetaModel::Resources::Structure**

**ResourceMethod**

**Package:** Structure

**isAbstract:** No

**Generalization:** ProcessOperation

**Description**

A behavioral feature of a ResourcePerformer whose behavior is specified in a Function.

---

**ResourceParameter**

**Package:** Structure

**isAbstract:** No

**Generalization:** ProcessParameter

**Description**

A type that represents inputs and outputs of a Function. It is typed by a ResourceInteractionItem.
ResourcePort

**Package:** Structure

**isAbstract:** No

**Generalization:** ProtocolImplementation, MeasurableElement

**Description**

An interaction point for a ResourcePerformer through which it can interact with the outside environment and which is defined by a ResourceInterface.

---

ResourceRole

**Package:** Structure

**isAbstract:** No

**Generalization:** SubjectOfResourceConstraint, LocationHolder, AssetRole, InteractionRole

**Description**

**Figure 9:120 - ResourceRole**

**Domain MetaModel::Resources::Connectivity**

**ResourceConnector**

- **Package:** Connectivity
- **isAbstract:** No
- **Generalization:** ProtocolImplementation, MeasurableElement

**Description**

A channel for exchange between two ResourceRoles.
Figure 9:121 - ResourceConnector

ResourceExchange
Package: Connectivity
isAbstract: No
Generalization: Exchange

Description
Asserts that a flow can exist between ResourcePerformers (i.e., flows of data, people, material, or energy).

Figure 9:122 - ResourceExchange

ResourceExchangeItem
Package: Connectivity
isAbstract: Yes
Generalization: Resource, SubjectOfSecurityConstraint, ExchangeItem
Description
An abstract type grouping elements that defines the types of elements that can be exchanged between ResourcePerformers and conveyed by a ResourceExchange.

ResourceInterface
Package: Connectivity
isAbstract: No
Generalization: PropertySet

Description
A declaration that specifies a contract between the ResourcePerformers it is related to and any other ResourcePerformers it can interact with. It is also intended to be an implementation of a specification of an Interface in the Business and/or Service layer.
ResourceSignal

**Package:** Connectivity

**isAbstract:** No

**Generalization:** ResourceExchangeItem

**Description**

A property of an element representing something in the physical world, expressed in amounts of a unit of measure.

Domain MetaModel::Resources::Processes

**Function**

**Package:** Processes

**isAbstract:** No

**Generalization:** SubjectOfResourceConstraint, Process

**Description**

An Activity which is specified in the context to the ResourcePerformer (human or machine) that IsCapableToPerform it.
Figure 9:126 - Function

**FunctionAction**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** ProcessUsage  
**Description**  
A call of a Function indicating that the Function is performed by a ResourceRole in a specific context.

Figure 9:127 - FunctionAction

**FunctionEdge**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** ProcessEdge  
**Description**  
A tuple that shows the flow of Resources (objects/data) between FunctionActions.
Domain MetaModel::Resources::States

ResourceStateDescription

**Package:** States  
**isAbstract:** No  
**Generalization:** MeasurableElement, StateDescription

**Description**
A state machine describing the behavior of a ResourcePerformer, depicting how the ResourcePerformer responds to various events and the actions.

Figure 9:129 - ResourceStateDescription

Domain MetaModel::Resources::Interaction Scenarios

ResourceMessage

**Package:** Interaction Scenarios  
**isAbstract:** No  
**Generalization:** InteractionMessage

**Description**
Message for use in a Resource Event-Trace which carries any of the subtypes of ResourceExchange.
**Domain MetaModel::Resources::Information**

**DataElement**
- **Package:** Information
- **isAbstract:** No
- **Generalization:** SubjectOfResourceConstraint, ResourceAsset, ResourceExchangeItem

**Description**
A formalized representation of data that is managed by or exchanged between resources.

**DataRole**
- **Package:** Information
- **isAbstract:** No
- **Generalization:** AssetRole
Description
A usage of DataElement that exists in the context of a ResourceAsset. It also allows the representation of the whole-part aggregation of DataElements.

Figure 9:132 - DataRole

Domain MetaModel::Resources::Constraints

ResourceConstraint
Package: Constraints
isAbstract: No
Generalization: Rule

Description
A rule governing the structural or functional aspects of an implementation.

Figure 9:133 - ResourceConstraint

SubjectOfResourceConstraint
Package: Constraints
isAbstract: Yes
Generalization: UAFElement

Description
An abstract type grouping elements that can be the subject of a ResourceConstraint.
Figure 9:134 - SubjectOfResourceConstraint

**Domain MetaModel::Resources::Roadmap**

**Forecast**

**Package:** Roadmap  
**isAbstract:** No  
**Generalization:** MeasurableElement

**Description**  
A tuple that specifies a transition from one Asset, Standard, Competence to another future one. It is related to an ActualEnterprisePhase to give it a temporal context.

Figure 9:135 - Forecast

**SubjectOfForecast**

**Package:** Roadmap  
**isAbstract:** Yes  
**Generalization:** UAFElemetary

**Description**  
An abstract type grouping elements that can be the subject of a Forecast.
Figure 9:136 - SubjectOfForecast

**Technology**

- **Package**: Roadmap
- **isAbstract**: No
- **Generalization**: ResourceArtifact

**Description**

A sub type of ResourceArtifact that indicates a technology domain, i.e., nuclear, mechanical, electronic, mobile telephony etc.

Figure 9:137 - Technology

**VersionedElement**

- **Package**: Roadmap
- **isAbstract**: Yes
- **Generalization**: UAFElement

**Description**

An abstract type grouping ResourcePerformer and ServiceSpecification that allows VersionOfConfiguration to be related to ActualProjectMilestones.
**VersionOfConfiguration**

**Package:** Roadmap

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A property of a WholeLifeConfiguration, used in version control of a VersionedElement. It asserts that a VersionedElement is a version of a WholeLifeConfiguration.

**VersionSuccession**

**Package:** Roadmap

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple between two VersionOfConfigurations that denotes that one VersionOfConfiguration follows from another.
**WholeLifeConfiguration**

**Package:** Roadmap  
**isAbstract:** No  
**Generalization:** [PropertySet](#)

**Description**
A set of VersionedElements.

**Domain MetaModel::Resources::Traceability**

**ProtocolImplementation**

**Package:** Traceability  
**isAbstract:** Yes  
**Generalization:** [UAEElement](#)

**Description**
An abstract type grouping architectural elements that can implement Protocols.
9.1.7 Domain MetaModel::Security

**Stakeholders:** Security Architects, Security Engineers, Systems Engineers, Operational Architects.

**Concerns:** addresses the security constraints and information assurance attributes that exist on exchanges between resources and OperationalPerformers.

**Definition:** illustrates the security assets, security constraints, security controls, families, and measures required to address specific security concerns.

**Domain MetaModel::Security::Taxonomy**

**Asset**

- **Package:** Taxonomy
- **isAbstract:** Yes
- **Generalization:** SubjectOfForecast, ConceptItem, LocationHolder, PropertySet, SubjectOfSecurityConstraint

**Description**

Asset as applied to Security views, an abstract type that indicates the types of elements that can be considered as a subject for security analysis.
Figure 9:143 - Asset

**OperationalAsset**

- **Package:** Taxonomy
- **isAbstract:** Yes
- **Generalization:** [Asset]

Figure 9:144 - OperationalAsset

**OperationalMitigation**

- **Package:** Taxonomy
- **isAbstract:** No
- **Generalization:** [OperationalArchitecture]

**Description**

A set of OperationalPerformers intended to address against specific operational risks.
**ResourceAsset**

**Package:** Taxonomy  
**isAbstract:** Yes  
**Generalization:** Asset

A set of ResourcePerformers intended to address against specific risks.

**SecurityEnclave**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ResourceArchitecture

Collection of information systems connected by one or more internal networks under the control of a single authority and security policy. The systems may be structured by physical proximity or by function, independent of location.
**AssetRole**

**Package:** Structure

**isAbstract:** Yes

**Generalization:** BPMN2Metamodel::ResourceRole, SubjectOfSecurityConstraint, MeasurableElement

**Description**

AssetRole as applied to Security views, an abstract element that indicates the type of elements that can be considered as a subject for security analysis in the particular context.

**InformationRole**

**Package:** Structure

**isAbstract:** No

**Generalization:** AssetRole

**Description**

A usage of InformationElement that exists in the context of an OperationalAsset. It also allows the representation of the whole-part aggregation of InformationElements.
Domain MetaModel::Security::Processes

**EnhancedSecurityControl**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** SecurityControl

**Description**

Statement of security capability to: (i) build in additional but related, functionality to a basic control; and/or (ii) increase the strength of a basic control.

**Enhances**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** MeasurableElement

**Description**

A tuple relating the EnhancedSecurityControl to a SecurityControl.
Figure 9:152 - Enhances

**Protects**

**Package:** Processes

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple that asserts that a SecurityControl is required to protect an Asset.

Figure 9:153 - Protects

**ProtectsInContext**

**Package:** Processes

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple that relates a SecurityControlAction to an OperationalRole, or a ResourceRole. It indicates that SecurityControl is required to protect an Asset in a specific context or configuration.
### SecurityProcess

**Package:** Processes  
**isAbstract:** No  
**Generalization:** OperationalActivity, Function, SubjectOfSecurityConstraint

**Description**

The security-related procedure that satisfies the security control requirement.

### SecurityProcessAction

**Package:** Processes  
**isAbstract:** No  
**Generalization:** OperationalActivityAction, FunctionAction

**Description**

Domain MetaModel::Security::Constraints

**ActualRisk**

**Package:** Constraints  
**isAbstract:** No  
**Generalization:** ActualPropertySet

**Description**

**Caveat**

**Package:** Constraints  
**isAbstract:** No  
**Generalization:** SecurityConstraint

**Description**
A statement that details alternate conditions under which the rule is not valid.
Risk

Package: Constraints
isAbstract: No
Generalization: PropertySet

Description
A statement of the impact of an event on Assets. It represents a constraint on an Asset in terms of adverse effects, with an associated measure. The measure is used to capture the extent to which an entity is threatened by a potential circumstance or event. Risk is typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence.

SecurityAvailability

Package: Constraints
isAbstract: No
Generalization: SecurityMeasurement

Description
Details the potential impact on organization or individuals if the information is not available to those who need to access it.
SecurityCategory

Package: Constraints
isAbstract: No
Generalization: MeasurementSet

Description

The security categories that have been determined for each type of information processed, stored, or transmitted by those information systems. The generalized format for expressing the security category (SC) of an information system is: SC information system = \{(confidentiality, impact), (integrity, impact), (availability, impact)\}.

SecurityClassification

Package: Constraints
isAbstract: No
Generalization: SecurityMeasurement

Description

Details a classification for the exchange.

SecurityClassificationKind

Package: Constraints
isAbstract: No
Generalization: MeasurableElement

Description

A type that defines acceptable values for the security category (SC) of an information system, where the acceptable values for potential impact are low, moderate, or high.
SecurityConstraint

Package: Constraints

isAbstract: No

Generalization: Rule

Description

A type of rule that captures a formal statement to define access control policy language.

SecurityControl

Package: Constraints

isAbstract: No

Generalization: MeasurableElement

Description

The management, operational, and technical control (i.e., safeguard or countermeasure) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information [NIST SP 800-53].
**SecurityControlFamily**

**Package:** Constraints

**isAbstract:** No

**Generalization:** SecurityControl

**Description**

An element that organizes security controls into a family. Each Security Control Family contains security controls related to the general security topic of the family.

**Figure 9:166 – SecurityControlFamily**
**SecurityIntegrity**

**Package:** Constraints

**isAbstract:** No

**Generalization:** SecurityMeasurement

**Description**

Details the potential impact on organization or individuals due to modification or destruction of information, and includes ensuring information non-repudiation and authenticity.

![SecurityIntegrity Diagram]

**Figure 9:167 - SecurityIntegrity**

**SecurityMeasurement**

**Package:** Constraints

**isAbstract:** Yes

**Generalization:** Measurement

**Description**

An abstract type grouping all types of security measurements (e.g., SecurityIntegrity, SecurityAvailability).

![SecurityMeasurement Diagram]

**Figure 9:168 - SecurityMeasurement**

**SubjectOfSecurityConstraint**

**Package:** Constraints

**isAbstract:** Yes

**Generalization:** UAFElement

**Description**

An abstract type grouping elements that can be the subject of a SecurityConstraint.
Domain MetaModel::Security::Traceability

Affects

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description
A tuple that asserts that a Risk is applicable to an Asset.

AffectsInContext

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description
A tuple that asserts that a Risk is applicable to an AssetRole in the specific context or configuration.
**Mitigates**

**Package:** Traceability  
**isAbstract:** No  
**Generalization:** MeasurableElement
Description
A tuple relating a Security Control to a Risk. Mitigation is established to manage risk and could be represented as an overall strategy or through techniques (mitigation configurations) and procedures (SecurityProcesses).

![Diagram of Mitigates relationship](image)

**Figure 9:172 - Mitigates**

**OwnsRisk**

**Package:** Traceability  
**isAbstract:** No  
**Generalization:** MeasurableElement
Description
A tuple relating a Risk to an organizational resource that is responsible for executing the risk mitigation.

![Diagram of OwnsRisk relationship](image)

**Figure 9:173 - OwnsRisk**

**OwnsRiskInContext**

**Package:** Traceability  
**isAbstract:** No  
**Generalization:** MeasurableElement
Description
A tuple relating a Risk to an organizational role that is responsible for executing the risk mitigation in the specific context or configuration.

![Diagram of OwnsRiskInContext relationship](image)
9.1.8 Domain MetaModel::Projects

Domain MetaModel::Projects::Taxonomy

**Project**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** OrganizationalResource

**Description**

A type that describes types of time-limited endeavors that are required to meet one or more Capability needs.

**ProjectMilestone**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** PropertySet

**Description**

A type of event in a Project by which progress is measured.
**Domain MetaModel::Projects::Structure**

**ActualProjectMilestoneRole**

*Package:* Structure  
*isAbstract:* No  
*Generalization:* ActualState

**Description**

An ActualProjectMilestone that is applied to a ProjectMilestoneRole.

**ProjectMilestoneRole**

*Package:* Structure  
*isAbstract:* No  
*Generalization:* MeasurableElement

**Description**

The role played by a ProjectMilestone in the context of a Project.
**Figure 9:178 - ProjectMilestoneRole**

**ProjectStatus**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** ActualState  

**Description**

The status (i.e., level of progress) of a ProjectTheme for an ActualProject at the time of the ActualProjectMilestone.

**Figure 9:179 - ProjectStatus**

**ProjectTheme**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** MeasurableElement  

**Description**

A property of a ProjectMilestone that captures an aspect by which the progress of ActualProjects may be measured.
**StatusIndicators**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** *MeasurableElement*  

**Description**  
An enumerated type that specifies a status for a ProjectTheme.

**Domain MetaModel::Projects::Connectivity**

**MilestoneDependency**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** *MeasurableElement*  

**Description**  
A tuple between two ActualProjectMilestones that denotes one ActualProjectMilestone follows from another.
Figure 9:182 - MilestoneDependency

**Domain MetaModel::Projects::Processes**

**ProjectActivity**
- **Package:** Processes
- **isAbstract:** No
- **Generalization:** Function, Process

**Description**

An activity carried out during a project.

Figure 9:183 - ProjectActivityAction

**ProjectActivityAction**
- **Package:** Processes
- **isAbstract:** No
- **Generalization:** FunctionAction

**Description**

The ProjectActivityAction is defined as a call behavior action that invokes the activity that needs to be performed.
Domain MetaModel::Projects::Interaction Scenarios

ProjectSequence

Package: Interaction Scenarios
isAbstract: No
Generalization: MeasurableElement

Description
A tuple between two ActualProjects that denotes one ActualProject cannot start before the previous ActualProject is finished.

Figure 9:185 - ProjectSequence

Domain MetaModel::Projects::Roadmap

ActualProject

Package: Roadmap
isAbstract: No
Generalization: ActualOrganizationalResource, Achiever

Description
A time-limited endeavor to provide a specific set of ActualResources that meet specific Capability needs.

Figure 9:186 – ActualProject
ActualProjectMilestone

**Package:** Roadmap

**isAbstract:** No

**Generalization:** ActualPropertySet

**Description**

An event with a start date in an ActualProject from which progress is measured.

![ActualProjectMilestone](image)

**Constraints**

[1] unnamed1  startTime=endTime

---

**Domain MetaModel::Projects::Traceability**

**ResponsibleFor**

**Package:** Traceability

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple between an ActualResponsibleResource and an ActualResponsibility or ActualProject. It defines the duties that the ActualResponsibleResource is ResponsibleFor.
9.1.9 Domain MetaModel::Standards

**Stakeholders:** Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.

**Concerns:** technical and non-technical Standards applicable to the architecture.

**Definition:** shows the technical, operational, and business Standards applicable to the architecture. Defines the underlying current and expected Standards.

**Domain MetaModel::Standards::Taxonomy**

**Protocol**

**Package:** Taxonomy

**isAbstract:** No

**Generalization:** [Standard](#)

**Description**

A Standard for communication over a network. Protocols may be composite, represented as a ProtocolStack made up of ProtocolLayers.
**ProtocolStack**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** Protocol  

**Description**

A sub type of Protocol that contains the ProtocolLayers, defining a complete stack.

![Diagram of ProtocolStack](image)

**Figure 9:190 - ProtocolStack**

**Standard**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** SubjectOfForecast, PropertySet  

**Description**

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.

![Diagram of Standard](image)

**Figure 9:191 - Standard**

**Attributes**

- mandatedDate : ISO8601DateTime[0..1]  
  The date when this version of the Standard was published.
- retiredDate : ISO8601DateTime[0..1]  
  The date when this version of the Standard was retired.

**Domain MetaModel::Standards::Structure**

**ProtocolLayer**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** MeasurableElement  

**Description**

Figure 9:192 – ProtocolLayer

9.1.10 Domain MetaModel::Actual Resources

Stakeholders: Solution Providers, Systems Engineers, Business Architects, Human Resources.

Concerns: the analysis, e.g., evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations.

Definition: illustrates the expected or achieved actual resource configurations and actual relationships between them.

Domain MetaModel::Actual Resources::Taxonomy

ActualOrganization

Package: Taxonomy
isAbstract: No
Generalization: ActualResponsibleResource

Description
An actual formal or informal organizational unit, e.g., "Driving and Vehicle Licensing Agency", "UAF team Alpha".

Figure 9:193 - ActualOrganization

Attributes

- serviceType : String[0..1] Service office code or symbol
- shortName : String[0..1] String providing a simplified means of identifying an ActualOrganization, i.e. SoftWareGroup could use SWG as the shortName.
**ActualOrganizationalResource**

**Package:** Taxonomy  
**isAbstract:** Yes  
**Generalization:** ActualResource, Stakeholder

Description

Abstract element for an ActualOrganization, ActualPerson or ActualPost.

![Diagram of ActualOrganizationalResource](image1)

**ActualPerson**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ActualResponsibleResource

Description

An individual human being.

![Diagram of ActualPerson](image2)
**ActualPost**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ActualResponsibleResource  
**Description**  
An actual, specific post, an instance of a Post "type" - e.g., "President of the United States of America." where the Post would be president.

![ActualPost Diagram]

**ActualResource**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ActualPropertySet, SubjectOfResourceConstraint, Achiever, CapableElement  
**Description**  
An individual, fully-realized ResourcePerformer.

![ActualResource Diagram]
**ActualResourceRelationship**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** [UAFElement](#)  

**Description**  
An actual resource flow existing between ActualResources (i.e., flow of data, people, materiel, or energy).

![ActualResourceRelationship Diagram](image)

**Figure 9:198 - ActualResourceRelationship**

**ActualResponsibility**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** [ActualOrganizationalResource](#)  

**Description**  
An actual duty required of a Person or Organization.

![ActualResponsibility Diagram](image)

**Figure 9:199 – ActualResponsibility**
ActualResponsibleResource

**Package:** Taxonomy  
**isAbstract:** Yes  
**Generalization:** ActualOrganizationalResource  
**Description**

An abstract type grouping responsible OrganizationalResources.

![Diagram of ActualResponsibleResource](image)

FieldedCapability

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ActualResource  
**Description**

An individual, fully-realized capability.

![Diagram of FieldedCapability](image)

Domain MetaModel::Actual Resources::Constraints

ActualService

**Package:** Constraints  
**isAbstract:** Yes  
**Generalization:** ActualMeasurementSet, CapableElement  
**Description**

An individual ServiceSpecification.
Figure 9:202 - ActualService

**ProvidedServiceLevel**

**Package:** Constraints

**isAbstract:** No

**Generalization:** ActualService

**Description**

A sub type of ActualService that details a specific service level delivered by the provider.

Figure 9:203 - ProvidedServiceLevel

**ProvidesCompetence**

**Package:** Constraints

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A tuple that asserts that an ActualOrganizationalResource provides a specific set of Competencies.
**RequiredServiceLevel**

**Package:** Constraints  
**isAbstract:** No  
**Generalization:** ActualService  

**Description**  
A sub type of ActualService that details a specific service level required of the provider.

```
ActualService
    \^ 
RequiredServiceLevel
```

**Figure 9:205 - RequiredServiceLevel**

### 9.1.11 Domain MetaModel::Dictionary

**Stakeholders:** Architects, users of the architecture, Capability Owners, Systems Engineers, Solution Providers.  
**Concerns:** Definitions for all the elements in the architecture, libraries of environments and measurements.  
**Definition:** Presents all the elements used in an architecture. Can be used specifically to capture:
  
a. Elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.  
   b. Measurable properties that can be used to support analysis such as KPIs, MoEs, TPIs etc.  
**Recommended Implementation:** Tabular format, SysML Block Definition Diagram.

### Alias

**Package:** Dictionary  
**isAbstract:** No  
**Generalization:** MeasurableElement

**Description**  
A metamodel Artifact used to define an alternative name for an element.

```
MeasurableElement
    \^ 
Alias
      -nameOwner : String [*]
```

**Figure 9:206 – Alias**
Attributes
nameOwner : String[*] Someone or something that uses this alternative name.

**Definition**

**Package**: Dictionary

**isAbstract**: No

**Generalization**: MeasurableElement

**Description**

A comment containing a description of an element in the architecture.

![Diagram](image)

**Figure 9:207 - Definition**

**Attributes**

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

**SameAs**

**Package**: Dictionary

**isAbstract**: No

**Generalization**: MeasurableElement

**Description**

A tuple that asserts that two elements refer to the same real-world thing.

![Diagram](image)

**Figure 9:208 – SameAs**
9.1.12 Domain MetaModel::Summary & Overview

ArchitecturalDescription

Package: Summary & Overview

isAbstract: No

Generalization: MeasurableElement

Description

An Architecture Description is a work product used to express the Architecture of some System Of Interest. It provides executive-level summary information about the architecture description in a consistent form to allow quick reference and comparison between architecture descriptions -- It includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.

Attributes

- approvalAuthority : String[*]  Someone or something that has the authority to approve the ArchitecturalDescription.
- architect : String[*]  Someone responsible for the creation of ArchitecturalDescription.
- assumptionAndConstraint : String[*]  Any assumptions, constraints, and limitations contained in the ArchitecturalDescription, including those affecting deployment, communications performance, information assurance environments, etc.
- creatingOrganization : String[*]  The organization responsible for creating the ArchitecturalDescription.
- dateCompleted : String[0..1]  Date that the ArchitecturalDescription was completed.
- methodologyUsed : String[*]  The methodology used in developing the architecture.
- 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.
status : String[*] Approval status of the architecture.
summaryOfFindings : String[*] Summarizes the findings that have been developed so far. This may be updated several times during the development of the ArchitecturalDescription.
toBe : Boolean[1] Indicates whether the ArchitecturalDescription represents an Architecture that exists or will exist in the future.
toolsUsed : String[*] Identifies any tools used to develop the ArchitecturalDescription as well as file names and formats if appropriate.
version : String[*] Version number of the architecture.

Architecture
Package: Summary & Overview
isAbstract: Yes
Generalization: UAFEelement
Description
An abstract type that represents a generic architecture. Subtypes are OperationalArchitecture and PhysicalArchitecture.

![Architecture Diagram](image)

Figure 9:210 - Architecture

Concern
Package: Summary & Overview
isAbstract: No
Generalization: PropertySet
Description
Interest in an EnterprisePhase (EnterprisePhase is synonym for System in ISO 42010) relevant to one or more of its stakeholders.
Stakeholder

**Package:** Summary & Overview

**isAbstract:** Yes

**Generalization:** UAFElement

**Description**


UAFEElement

**Package:** Summary & Overview

**isAbstract:** Yes

**Description**

Abstract super type for all of the UAF elements. It provides a way for all of the UAF elements to have a common set of properties.
Attributes

URI : String[0..1]  Captures Unique identifier for the element.

**View**

**Package:** Summary & Overview

**isAbstract:** No

**Generalization:** [PropertySet](#)

Description

An architecture view expresses the architecture of the system-of-interest in accordance with an architecture viewpoint (or simply, viewpoint). [ISO/IEC/IEEE 42010:2011(E)].
Viewpoint

Package: Summary & Overview
isAbstract: No
Generalization: PropertySet

Description

An architecture viewpoint frames (to formulate or construct in a particular style or language) one or more concerns. A concern can be framed by more than one viewpoint. [ISO/IEC/IEEE 42010:2011(E)].

![Viewpoint Diagram]

Figure 9:215 - Viewpoint
Attributes
- language : String[*] The languages used to express the Viewpoint.
- method : String[*] The methods employed in the development of the Viewpoint.
- purpose : String[0..1] The purpose of the Viewpoint.

9.1.13 Domain MetaModel::Information

DataModel

Package: Information
isAbstract: No
Generalization: SubjectOfOperationalConstraint, SubjectOfResourceConstraint

Description

A structural specification of data types, showing relationships between them. The type of data captured in the DataModel is described using the enumeration DataModelKind (Conceptual, Logical and Physical).

![DataModel Diagram]

Figure 9:216 - DataModel
9.1.14 Domain MetaModel::Parameters

**ActualCondition**

**Package:** Parameters  
**isAbstract:** No  
**Generalization:** ActualPropertySet  

**Description**

An individual describing an actual situation with respect to circumstances under which an OperationalActivity, Function or ServiceFunction can be performed.

![Diagram of ActualCondition](image)

**ActualEnvironment**

**Package:** Parameters  
**isAbstract:** No  
**Generalization:** ActualCondition  

**Description**

An individual that describes the circumstances of an Environment.

![Diagram of ActualEnvironment](image)
**ActualLocation**

**Package:** Parameters  
**isAbstract:** No  
**Generalization:** [ActualCondition](#)  

**Description**

An individual that describes a physical location, for example using text to provide an address, Geo-coordinates, etc.

*Figure 9:219 - ActualLocation*

Attributes

- **address**: String[0..1]  
  String describing the address of the ActualLocation, i.e. "1600 Pennsylvania avenue", "The White House"

- **customKind**: String[0..1]  
  String describing a location kind that is not in the LocationKind enumerated list

- **locationNamedByAddress**: Boolean[]  
  Boolean that indicates if the ActualLocation address is embedded in the ActualLocation name. By default = false.

**ActualMeasurement**

**Package:** Parameters  
**isAbstract:** No  
**Generalization:** [ActualState](#)  

**Description**

An actual value that is applied to a Measurement.
Figure 9:220 - ActualMeasurement

**ActualMeasurementSet**

**Package**: Parameters

**isAbstract**: No

**Generalization**: ActualPropertySet

**Description**

A set of ActualMeasurements.

Figure 9:221 - ActualMeasurementSet

**ActualPropertySet**

**Package**: Parameters

**isAbstract**: No

**Generalization**: ActualState

**Description**

A set or collection of Actual properties.
Figure 9:222 - ActualPropertySet

**ActualState**

- **Package:** Parameters
- **isAbstract:** Yes
- **Generalization:** UAFElement

**Description**

Abstract element that applies temporal extent to a set of elements realized as Instance Specifications.

Figure 9:223 - ActualState

**Attributes**

- **endDate:** ISO8601DateTime[0..1]  
  End time for all individual elements.
- **startDate:** ISO8601DateTime[0..1]  
  Start time for all individual elements.

**Condition**

- **Package:** Parameters
- **isAbstract:** No
- **Generalization:** PropertySet
Description

A type that defines the Location, Environment and/or GeoPoliticalExtent.

Figure 9:224 - Condition

**Environment**

**Package:** Parameters

**isAbstract:** No

**Generalization:** Condition

Description

A definition of the environmental factors in which something exists or functions. The definition of an Environment element can be further defined using EnvironmentKind.

Figure 9:225 - Environment

**GeoPoliticalExtentType**

**Package:** Parameters

**isAbstract:** No

**Generalization:** Condition, OperationalExchangeItem, ResourceExchangeItem

Description

A type of geospatial extent whose boundaries are defined by declaration or agreement by political parties.
Figure 9:226 - GeoPoliticalExtentType
Attributes
    customKind : String[]  Captures the kind of GeopoliticalExtentType.

ISO8601DateTime
Package: Parameters
isAbstract: No
Generalization: UAFEElement
Description
A date and time specified in the ISO8601 date-time format including timezone designator (TZD): YYYY-MM-DDTh:mm:ssTZD.

Figure 9:227 - ISO8601DateTime

Location
Package: Parameters
isAbstract: No
Generalization: ConceptItem, Condition
Description
A specification of the generic area in which a LocationHolder is required to be located.
Figure 9:228 - Location
Attributes
  customKind : String[0..1]  Captures the kind of Location if the LocationTypeKind has been set to "OtherType".

**LocationHolder**
*Package:* Parameters
*isAbstract:* Yes
*Generalization:* UAFElement

Description
Abstract type, used to group elements that are allowed to be associated with a Location.

Figure 9:229 - LocationHolder

**MeasurableElement**
*Package:* Parameters
*isAbstract:* Yes
*Generalization:* UAFElement

Description
Abstract type, grouping elements that can be measured by applying MeasurementSets to them.
Figure 9:230 - MeasurableElement

**Measurement**

**Package:** Parameters

**isAbstract:** No

**Generalization:** MeasurableElement

**Description**

A property of an element representing something in the physical world, expressed in amounts of a unit of measure.
Figure 9:231 - Measurement

**MeasurementSet**

**Package:** Parameters  
**isAbstract:** No  
**Generalization:** PropertySet

Description
A collection of Measurements.

Figure 9:232 - MeasurementSet

**PropertySet**

**Package:** Parameters  
**isAbstract:** Yes  
**Generalization:** UAFElement

Description
An abstract type grouping architectural elements that can own Measurements.
Figure 9:233 - PropertySet