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
means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

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

CORBA®, CORBA logos®, FIBO®, Financial Industry Business Ontology®, FINANCIAL INSTRUMENT GLOBAL IDENTIFIER®, IIOP®, IMM®, Model Driven Architecture®, MDA®, Object Management Group®, OMG®, OMG Logo®, SoaML®, SOAAML®, SysML®, UAF®, Unified Modeling Language®, UML®, UML Cube Logo®, VSIPL®, and XMI® are registered trademarks of the Object Management Group, Inc.

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

**PREFACE** ...................................................................................................................................................... 1

1 INTRODUCTION.......................................................................................................................................................... 3
  1.1 OVERVIEW ........................................................................................................................................................... 3

2 ADDITIONAL INFORMATION........................................................................................................................................... 4
  2.1 LANGUAGE ARCHITECTURE ................................................................................................................................ 4
  2.2 CORE PRINCIPLES ................................................................................................................................................. 4
  2.3 REPRESENTING STEREOTYPE CONSTRAINTS ......................................................................................................... 4
      2.3.1 Metaconstraint dependency .......................................................................................................................... 4
      2.3.2 Metarelationship dependency ....................................................................................................................... 5
      2.3.3 Stereotyped relationship dependency ............................................................................................................ 7

3 UAF STEREOTYPES.................................................................................................................................................. 9
  3.1 UAF ................................................................................................................................................................. 9
      3.1.1 UAF::Dictionary ............................................................................................................................................ 9
      3.1.2 UAF::Parameters ....................................................................................................................................... 11
      3.1.3 UAF::Metadata ......................................................................................................................................... 25
      3.1.4 UAF::Strategic .......................................................................................................................................... 38
      3.1.5 UAF::Operational .................................................................................................................................... 52
      3.1.6 UAF::Services ......................................................................................................................................... 73
      3.1.7 UAF::Personnel ....................................................................................................................................... 83
      3.1.8 UAF::Resources ..................................................................................................................................... 91
      3.1.9 UAF::Security ........................................................................................................................................... 116
      3.1.10 UAF::Projects ....................................................................................................................................... 132
      3.1.11 UAF::Standards ..................................................................................................................................... 144
      3.1.12 UAF::Actual Resources ......................................................................................................................... 146
      3.1.13 UAF::Summary and Overview ............................................................................................................. 158

4 UAF VIEW SPECIFICATIONS....................................................................................................................................... 166
  4.1 VIEW SPECIFICATIONS .................................................................................................................................... 166
      4.1.1 View Specifications::Strategic .................................................................................................................. 166
      4.1.2 View Specifications::Operational ........................................................................................................... 173
      4.1.3 View Specifications::Services .................................................................................................................. 182
      4.1.4 View Specifications::Personnel ................................................................................................................ 190
      4.1.5 View Specifications::Resources ................................................................................................................ 203
      4.1.6 View Specifications::Security .................................................................................................................. 213
      4.1.7 View Specifications::Projects .................................................................................................................. 221
      4.1.8 View Specifications::Standards ............................................................................................................. 227
      4.1.9 View Specifications::Actual Resources ................................................................................................. 230
      4.1.10 View Specifications::Dictionary ........................................................................................................... 233
      4.1.11 View Specifications::Requirements ...................................................................................................... 233
      4.1.12 View Specifications::Summary & Overview ......................................................................................... 234
      4.1.13 View Specifications::Information ........................................................................................................ 235
      4.1.14 View Specifications::Parameters ......................................................................................................... 236

5 MEASUREMENT LIBRARY ...................................................................................................................................... 240

iv

Unified Architecture Framework Profile (UAFP) Version 1.1
## TABLE OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:1</td>
<td>MapsToCapability Stereotype</td>
<td>5</td>
</tr>
<tr>
<td>2:2</td>
<td>Connector Extension</td>
<td>5</td>
</tr>
<tr>
<td>2:3</td>
<td>Capabilities Generalization</td>
<td>6</td>
</tr>
<tr>
<td>2:4</td>
<td>Visualizing «metarelationship» dependency</td>
<td>6</td>
</tr>
<tr>
<td>2:5</td>
<td>Use of the AchievedEffect «stereotyped relationship» dependency</td>
<td>7</td>
</tr>
<tr>
<td>3:1</td>
<td>Alias</td>
<td>9</td>
</tr>
<tr>
<td>3:2</td>
<td>Definition</td>
<td>10</td>
</tr>
<tr>
<td>3:3</td>
<td>SameAs</td>
<td>10</td>
</tr>
<tr>
<td>3:4</td>
<td>ActualCondition</td>
<td>11</td>
</tr>
<tr>
<td>3:5</td>
<td>ActualEnvironment</td>
<td>12</td>
</tr>
<tr>
<td>3:6</td>
<td>ActualLocation</td>
<td>12</td>
</tr>
<tr>
<td>3:7</td>
<td>ActualMeasurement</td>
<td>13</td>
</tr>
<tr>
<td>3:8</td>
<td>ActualMeasurementSet</td>
<td>14</td>
</tr>
<tr>
<td>3:9</td>
<td>ActualPropertySet</td>
<td>15</td>
</tr>
<tr>
<td>3:10</td>
<td>Condition</td>
<td>16</td>
</tr>
<tr>
<td>3:11</td>
<td>Environment</td>
<td>16</td>
</tr>
<tr>
<td>3:12</td>
<td>EnvironmentProperty</td>
<td>17</td>
</tr>
<tr>
<td>3:13</td>
<td>GeoPoliticalExtentType</td>
<td>18</td>
</tr>
<tr>
<td>3:14</td>
<td>Location</td>
<td>19</td>
</tr>
<tr>
<td>3:15</td>
<td>LocationHolder</td>
<td>20</td>
</tr>
<tr>
<td>3:16</td>
<td>MeasurableElement</td>
<td>22</td>
</tr>
<tr>
<td>3:17</td>
<td>Measurement</td>
<td>23</td>
</tr>
<tr>
<td>3:18</td>
<td>MeasurementSet</td>
<td>24</td>
</tr>
<tr>
<td>3:19</td>
<td>PropertySet</td>
<td>25</td>
</tr>
<tr>
<td>3:20</td>
<td>ActualState</td>
<td>26</td>
</tr>
<tr>
<td>3:21</td>
<td>ISO8601DateTime</td>
<td>26</td>
</tr>
<tr>
<td>3:22</td>
<td>Exchange</td>
<td>27</td>
</tr>
<tr>
<td>3:23</td>
<td>Resource</td>
<td>27</td>
</tr>
<tr>
<td>3:24</td>
<td>Activity</td>
<td>28</td>
</tr>
<tr>
<td>3:25</td>
<td>CapableElement</td>
<td>29</td>
</tr>
<tr>
<td>3:26</td>
<td>IsCapableToPerform</td>
<td>29</td>
</tr>
<tr>
<td>3:27</td>
<td>PerformsInContext</td>
<td>31</td>
</tr>
<tr>
<td>3:28</td>
<td>ArchitectureMetadata</td>
<td>32</td>
</tr>
<tr>
<td>3:29</td>
<td>DataModel</td>
<td>32</td>
</tr>
<tr>
<td>3:30</td>
<td>Information</td>
<td>33</td>
</tr>
<tr>
<td>3:31</td>
<td>Metadata</td>
<td>34</td>
</tr>
<tr>
<td>3:32</td>
<td>Rule</td>
<td>35</td>
</tr>
<tr>
<td>3:33</td>
<td>ArchitecturalReference</td>
<td>36</td>
</tr>
<tr>
<td>3:34</td>
<td>Implements</td>
<td>37</td>
</tr>
<tr>
<td>3:35</td>
<td>ActualEnterprisePhase</td>
<td>39</td>
</tr>
<tr>
<td>3:36</td>
<td>Capability</td>
<td>40</td>
</tr>
<tr>
<td>3:37</td>
<td>EnterpriseGoal</td>
<td>41</td>
</tr>
<tr>
<td>3:38</td>
<td>EnterprisePhase</td>
<td>42</td>
</tr>
<tr>
<td>3:39</td>
<td>EnterpriseVision</td>
<td>42</td>
</tr>
</tbody>
</table>
Figure 4:56 - Project Structure...............................................................................................222
Figure 4:57 - Project Connectivity.........................................................................................223
Figure 4:58 - Project Processes .............................................................................................224
Figure 4:59 - Project Roadmap .........................................................................................225
Figure 4:60 - Project Traceability .......................................................................................226
Figure 4:61 - Standards Taxonomy .......................................................................................227
Figure 4:62 - Standards Structure .......................................................................................228
Figure 4:63 - Standards Roadmap .......................................................................................229
Figure 4:64 - Standards Traceability ...................................................................................230
Figure 4:65 - Actual Resources Structure ............................................................................231
Figure 4:66 - Actual Resources Connectivity .......................................................................232
Figure 4:67 - Actual Resources Traceability ......................................................................232
Figure 4:68 - Dictionary .....................................................................................................233
Figure 4:69 - Requirements ...............................................................................................234
Figure 4:70 - Summary & Overview ....................................................................................234
Figure 4:71 - Information Model ........................................................................................235
Figure 4:72 - Parameters: Environment ..............................................................................236
Figure 4:73 - Parameters: Measurements ..........................................................................238
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
- **Helvetica/Arial - 10 pt. Bold:** OMG Interface Definition Language (OMG IDL) and syntax elements.
- **Courier - 10 pt. Bold:** Programming language elements.
- **Helvetica/Arial - 10 pt:** Exceptions

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

Issues
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 Introduction

1.1 Overview

This document is a normative supplement to the UAF DMM document (c4i/19-06-16).

This document specifies a UAF profile to enable practitioners to express architectural model elements and organize them in a set of domains, model kinds, and view specification (specified in the UAF DMM) that support the specific needs of end users in defense and commercial industry.

UAFP 1.1 defines a set of stereotypes and model elements and relationships to satisfy the requirements of the UPDM 3.0 RFP and the UAF DMM. The profile specification documents the language architecture in terms of UML profiling mechanism.

A number of UAFP stereotypes inherit from SysML stereotypes where reuse of SysML semantics is necessary. The reusable portions of the SysML specification are not included directly in the specification but are made explicit through the stereotype inheritance.
2 Additional Information

2.1 Language Architecture
The UAFP 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 document. This document describes the language architecture in terms of the UML 2.5.1 and SysML 1.5 parts that are reused and the defined UML 2.5.1 extensions; and specifies how to implement UAFP. This clause explains design principles and how they are applied to define the UAFP language architecture.

2.2 Core Principles
The fundamental design principles for UAFP are:

- **Requirements-driven**: UAFP is intended to satisfy the requirements of the UPDM 3.0 RFP Mandatory Requirements.
- **UAF Domain Metamodel (DMM) driven**: The DMM served as a foundation for profile development.
- **Reuse of existing specifications**: UAFP reuses UML/SysML wherever practical to satisfy the requirements of the UAFP 3.0 RFP and leverage features from both UML and SysML to provide a robust modeling capability. Consequently, UAFP is intended to be relatively easy to implement for vendors who support UML 2.x and SysML 1.x.
- **Compliance levels**: UAFP has a single compliance level based upon a combination of the reuse of UML and SysML elements. It is expected that the views that are created as result of this profile have frames that reflect the underlying SysML diagram type that is used as the basis for the view. It also expected that the graphical notation used to display elements within those views correspond to the standard SysML graphical notation of the SysML/UML metaclass that the stereotype extends.
- **Interoperability**: UAFP inherits the XMI interchange capability from UML. The UAFP specification reuses a subset of UML 2.5.1 and provides additional extensions needed to address requirements in the UPDM 3.0 RFP Mandatory Requirements.

2.3 Representing Stereotype Constraints
The UAF Profile uses an enhanced standard notation to represent metaconstraints graphically in the UAF profile diagrams to improve readability of the UAF Profile specification and overcome limitations of being unable to visualize constraints diagrammatically in UML.

The metaconstraints appears in the UAFP specification diagrams for visualization purposes only, however the representation in the XMI is as a UML constraint, specified in structured English. These constraints are implementable in a tool, by OCL for example.

A simple UML profile defines these metaconstraints.

The following subsections detail the metaconstraint profile definition within the UAF profile.

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

A sample of the «metaconstraint» dependency is a diagram for stereotype extending the Dependency metaclass.
MapsToCapability is a UAFP stereotype that extends Abstraction (a type of Dependency in UML). The constraint on this stereotype is that its client end must be stereotyped by an Activity (which is abstract) and its supplier end must be stereotyped by a Capability. But as it is not possible to show this constraint graphically the diagram does not communicate the needed information. We then use the "metaconstraint" dependency to visualize the constraint.
With the metaconstraint dependency added to the diagram (see Figure 2:1) which shows that MapsToCapability is a stereotype extending the Abstraction metaclass, that inherits the properties of a MeasurableElement and is used for modeling a relationship between an Activity (or its specializations) and a Capability (or its specializations). A Dependency stereotyped MapsToCapability must have its values for the client property stereotyped as an Activity, and its values for the supplier property must be stereotyped Capability.

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

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

2.3.2 Metarelationship dependency

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

For example: A Capability may depend on other Capabilities or be subtype of a Capability, but this concept cannot be visualized on the diagram:
Figure 2:3 – Capabilities Generalization
We are using the «metarelationship» dependency to visualize the dependency and the generalization concept.

Figure 2:4 – Visualizing «metarelationship»
This diagram should be read as follows:
Capability may have other Capabilities related to it, using the UML Dependency metaclass and it may have sub types of Capabilities related to it, using the the UML Generalization metaclass.
The «metarelationship» dependency will appear only in the specification diagrams, but not the profile XMI.

2.3.3 Stereotyped relationship dependency
Although the «metarelationship» dependency creates a good way to show the constrained ends of the stereotyped relationship, it also creates some overhead when showing the relationship between two stereotypes.
For example, Figure 2:2 below shows that elements of subtype Achiever have a stereotyped relationship called AchievedEffect with elements of type ActualState.

Figure 2:2 – Use of the AchievedEffect «stereotyped relationship» dependency
This page intentionally left blank.
3 UAF Stereotypes

UAFP imports the entire SysML profile. This is intended to provide more seamless integration with system modeling using SysML and to be able to fully leverage the capabilities of SysML in UAFP. An example of this is the integration of Requirements into the UAFP and also the use of Parametric Diagrams and integration of elements based upon instance specifications to allow the assessment of measures within an architecture developed using UAFP.

3.1 UAF

UAF is top level profile root.

3.1.1 UAF::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.

Alias

Package: Dictionary

isAbstract: No

Generalization: MeasurableElement

Extension: Comment

Description

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

![Diagram of Alias model](image)

Figure 3:1 - Alias

Attributes

nameOwner : String[*]  Someone or something that uses this alternative name.

Constraints

[1] Alias.annotatedElement  Value for the annotatedElement metaproperty must be stereotyped by the specialization of «UAFElement».

Definition

Package: Dictionary

isAbstract: No

Generalization: MeasurableElement
**Extension**: Comment

**Description**

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

![Diagram](image)

**Figure 3:2 - Definition**

**Attributes**

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

**Constraints**

1. Definition.annotatedElement  
   Value for the annotatedElement metaproperty must be stereotyped by the specialization of «UAEElement».

**SameAs**

**Package**: Dictionary

**isAbstract**: No

**Generalization**: MeasurableElement

**Extension**: Dependency

**Description**

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

![Diagram](image)

**Figure 3:3 - SameAs**

**Constraints**

1. SameAs.client  
   Values for the client metaproperty must be stereotyped by the specialization of «UAEElement».
[2] SameAs.supplier Values for the supplier metaproperty must be stereotyped by the specialization of «UAFElement».

3.1.2 UAF::Parameters

ActualCondition

Package: Parameters
isAbstract: No
Generalization: ActualPropertySet
Extension: InstanceSpecification

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

![Diagram of ActualCondition class diagram]

Figure 3:4 - ActualCondition
Constraints
[1] ActualCondition.classifier Value for the classifier metaproperty has to be stereotyped «Condition» or its specializations.

ActualEnvironment

Package: Parameters
isAbstract: No
Generalization: ActualCondition
Extension: InstanceSpecification

Description
Actual circumstances of an Environment.
ActualEnvironment

Constraints

[1] ActualEnvironment.classifier Value for the classifier metaproperty has to be stereotyped «Environment» or its specializations.

ActualLocation

Package: Parameters
isAbstract: No
Generalization: ActualCondition
Extension: InstanceSpecification

Description
A physical location, for example using text to provide an address, Geo-coordinates, etc.

Figure 3:6 – 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[1]    Boolean that indicates if the ActualLocation address is embedded in the ActualLocation name. By default = false.

Associations

locationKind : LocationKind[1]    Enumerated value describing the kind of ActualLocation.

Constraints

[1] ActualLocation.classifier    Classifier metaproperty value must be stereotyped «Location» or its specializations.

**ActualMeasurement**

Package: Parameters

isAbstract: No

Generalization: ActualState

Extension: Slot

Description

An actual value that is applied to a Measurement.

**Figure 3:7 – ActualMeasurement**

Associations


Constraints


[1] ActualMeasurement.definingFeature Value for the definingFeature metaproperty must be stereotyped «Measurement» or its specializations.

[2] ActualMeasurement.owningInstance Value for the owningInstance metaproperty must be stereotyped «ActualPropertySet» or its specializations.

**ActualMeasurementKind**

**Package:** Parameters  
isAbstract: No

**Description**

Enumeration of the possible kinds of ActualMeasurement. Its enumeration literals are:

- Actual - Indicates that the ActualMeasurement associated with the ActualMeasurementKind is a realworld value.
- Required - Indicates that the ActualMeasurement associated with the ActualMeasurementKind is a value that is expected to be achieved.
- Estimate - Indicates that the ActualMeasurement associated with the ActualMeasurementKind is an estimate of a realworld value.

**ActualMeasurementSet**

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

**Description**

A set of ActualMeasurements.

---

**Figure 3:8 - ActualMeasurementSet**

**Associations**

- appliesFor : MeasurableElement[*]  
  Relates the ActualMeasurementSet to the elements that are being measured.

**Constraints**

- [1] ActualMeasurementSet.classifier Classifier metaproperty value must be stereotyped «MeasurementSet» or its specializations.
ActualPropertySet

Package: Parameters
isAbstract: No
Generalization: ActualState
Extension: InstanceSpecification

Description
A set or collection of Actual properties.

Constraints
[1] ActualPropertySet.classifier Value for the classifier metaproperty must be stereotyped by the specialization of «PropertySet».

Condition
Package: Parameters
isAbstract: No
Generalization: PropertySet, ValueType
Extension: DataType

Description
A type that defines the Location, Environment and/or GeoPoliticalExtent.
Environment
Package: Parameters
isAbstract: No
Generalization: Condition
Extension: DataType

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.

EnvironmentKind
Package: Parameters
isAbstract: No

Figure 3:10 - Condition

Figure 3:11 - Environment
Associations

EnvironmentKind
Package: Parameters
isAbstract: No
Description

Enumeration of the possible kinds of Environment. Its enumeration literals are:

- **TerrainType** - Indicates that the Environment associated with EnvironmentKind captures a kind of terrain used to describe the terrain state of an environment at a particular time (e.g., muddy, frozen ground, deep snow, etc.).
- **WeatherConditions** - Indicates that the Environment associated with EnvironmentKind captures a kind of weather condition (e.g., Typhoon, Hurricane, Very Hot, Humid etc.).
- **LightConditions** - Indicates that the Environment associated with EnvironmentKind captures a kind of light condition (e.g., broad daylight, dusk, moonlit, etc.).
- **CBRNEnvironment** - Indicates that the Environment associated with EnvironmentKind is of a Chemical, Biological, Radiological or Nuclear (CBRN) kind.
- **SituationType** - Indicates that the Environment associated with EnvironmentKind captures a kind of situation used to describe the types and levels of threat (e.g., Corrosive, Fire, Smoke, Peaceful etc.).

**EnvironmentProperty**

**Package:** Parameters

**isAbstract:** No

**Generalization:** MeasurableElement

**Extension:** Property

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.

![EnvironmentProperty Diagram](image)

**Figure 3:12 - EnvironmentProperty**

**Constraints**

1. **EnvironmentalProperty.class**
   Value for the class metaproperty must be stereotyped «Environment» or its specializations.

2. **EnvironmentalProperty.type**
   Value for the type property must be stereotyped «Condition» or its specializations.

**GeoPoliticalExtentType**

**Package:** Parameters

**isAbstract:** No

**Generalization:** ResourceExchangeItem, OperationalExchangeItem, Condition

**Extension:** DataType

Description

A type of geospatial extent whose boundaries are defined by declaration or agreement by political parties.
Attributes

customKind : String[0..1]  Captures the kind of GeoPoliticalExtentType if the GeoPoliticalExtentTypeKind has been set to "OtherType".

Associations

kind : GeoPoliticalExtentTypeKind[1]  Captures the kind of GeoPoliticalExtentType.

**GeoPoliticalExtentTypeKind**

**Package:** Parameters

**isAbstract:** No

**Description**

Enumeration of the possible kinds of GeoPoliticalExtentType. Its enumeration literals are:

- GeoFeatureType - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type of object that encompasses meteorological, geographic, and control features mission significance.
- RegionOfCountryType - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type of large, usually continuous segment of a political state, nation or its territory.
- CountryType - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type of political state, nation or its territory.
- RegionOfWorldType - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type of large, usually continuous segment of a surface or space; area.
- **FacilityType** - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type of a real property entity consisting of underlying land and one or more of the following: a building, a structure (including linear structures), a utility system, or pavement.
- **SiteType** - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type of Physical (geographic) location that is or was owned by, leased to, or otherwise possessed. Each site is assigned to a single installation. A site may exist in one of three forms: (1) Land only, where there are no facilities present and where the land consists of either a single land parcel or two or more contiguous land parcels. (2) Facility or facilities only, where the underlying land is neither owned nor controlled by the government. A stand-alone facility can be a site. If a facility is not a stand-alone facility, it must be assigned to a site. (3) Land and all the facilities thereon, where the land consists of either a single land parcel or two or more contiguous land parcels.
- **InstallationType** - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type of base, camp, post, station, yard, center, or other activity, including leased facilities, without regard to the duration of operational control. An installation may include one or more sites.
- **OtherType** - Indicates that the GeoPoliticalExtentType associated with the GeoPoliticalExtentTypeKind is a type not covered by the standard GeoPoliticalExtentTypeKinds.

### Location

**Package**: Parameters

**isAbstract**: No

**Generalization**: ConceptItem, Condition

**Extension**: DataType

**Description**

A specification of the generic area in which a LocationHolder is required to be located.

**Figure 3:14 - Location**

**Attributes**

- **customKind** : String [0..1]  Captures the kind of Location if the LocationTypeKind has been set to "OtherType".

**Associations**

- **kind** : LocationTypeKind[1]  Captures the kind of Location.
**LocationHolder**

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

Description  
Abstract grouping used to define elements that are allowed to be associated with a Location.

**Figure 3:15 – LocationHolder**

**Associations**

- **physicalLocation**: ActualLocation[0..*]  
  Relates a LocationHolder (i.e. OperationalPerformer, OperationalRole, ResourceRole etc.) to its ActualLocation.

- **requiredEnvironment**: ActualEnvironment[0..*]  
  Relates a LocationHolder (i.e. OperationalPerformer, OperationalRole, ResourceRole etc.) to the Environment in which it is required to perform/be used.

**LocationKind**

**Package:** Parameters  
**isAbstract:** No

Description  
Enumeration of the possible kinds of location applicable to an ActualLocation. Its enumeration literals are:

- **SolidVolume** - Indicates that the ActualLocation associated with the LocationKind is the amount of space occupied by a three-dimensional object of definite shape; not liquid or gaseous.
- **Surface** - Indicates that the ActualLocation associated with the LocationKind is a portion of space having length and breadth but no thickness or regards to time.
- **Line** - Indicates that the ActualLocation associated with the LocationKind is a geometric figure formed by a point moving along a fixed direction and the reverse direction.
- **Point** - Indicates that the ActualLocation associated with the LocationKind is a unidimensional Individual.
• GeoStationaryPoint - Indicates that the ActualLocation associated with the LocationKind is a unidimensional Individual.
• PlanarSurface - Indicates that the ActualLocation associated with the LocationKind is a two-dimensional portion of space.
• PolygonArea - Indicates that the ActualLocation associated with the LocationKind is a space enclosed by a polygon.
• RectangularArea - Indicates that the ActualLocation associated with the LocationKind is a space enclosed by a rectangle.
• ElipticalArea - Indicates that the ActualLocation associated with the LocationKind is a space enclosed by an ellipse.
• CircularArea - Indicates that the ActualLocation associated with the LocationKind is a space enclosed by a circle.
• Other - Indicates that the ActualLocation associated with the LocationKind is a LocationKind that is not on the enumerated list.

**LocationTypeKind**

**Package:** Parameters

**isAbstract:** No

**Description**

Enumeration of the possible kinds of location type that are applicable to a Location. Its enumeration literals are:

• OtherType - Indicates that the Location associated with the LocationTypeKind describes a type of is a LocationKindType that is not on the enumerated list.
• SolidVolumeType - Indicates that the Location associated with the LocationTypeKind describes a type of amount of space occupied by a three-dimensional object of definite shape; not liquid or gaseous.
• SurfaceType - Indicates that the Location associated with the LocationTypeKind describes a type of portion of space having length and breadth but no thickness or regards to time.
• LineType - Indicates that the Location associated with the LocationTypeKind describes a type of geometric figure formed by a point moving along a fixed direction and the reverse direction.
• PointType - Indicates that the Location associated with the LocationTypeKind describes a type of unidimensional Individual.
• GeoStationaryPointType - Indicates that the Location associated with the LocationTypeKind describes a type of unidimensional Individual.
• PlanarSurfaceType - Indicates that the Location associated with the LocationTypeKind describes a type of is a two-dimensional portion of space.
• PolygonAreaType - Indicates that the Location associated with the LocationTypeKind describes a type of space enclosed by a polygon.
• RectangularAreaType - Indicates that the Location associated with the LocationTypeKind describes a type of space enclosed by a rectangle.
• ElipticalAreaType - Indicates that the Location associated with the LocationTypeKind describes a type of space enclosed by an ellipse.
• CircularAreaType - Indicates that the Location associated with the LocationTypeKind describes a type of space enclosed by a circle.

**MeasurableElement**

**Package:** Parameters

**isAbstract:** Yes

**Generalization:** UAFElement

**Extension:** Element
Description
Abstract grouping for elements that can be measured by applying MeasurementSets to them.

**Figure 3:16 – MeasurableElement**

**Associations**
- `actualMeasurementSet : ActualMeasurementSet[*]` Relates the MeasurableElement to the ActualMeasurementSet that provides its ActualMeasurements.
- `measurementSet : MeasurementSet[*]` Relates the MeasurableElement to the MeasurementSet that provides its Measurements by which it can be measured.

**Measurement**
**Package:** Parameters
isAbstract: No

**Generalization:** MeasurableElement

**Extension:** Property

**Description**

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

![Diagram](image)

**Figure 3:17 – Measurement**

**Associations**

environmentalContext : ActualCondition[0..1]  
Relates the Measurement to the Condition (which provides the environmentalContext) under which the Measurement is expected to be taken.

**Constraints**

[1] Measurement.class  
Value for the class metaproperty must be stereotyped by the specialization of «PropertySet».

**MeasurementSet**

**Package:** Parameters  
isAbstract: No

**Generalization:** PropertySet, ValueType

**Extension:** DataType

**Description**

A collection of Measurements.
Figure 3:18 – MeasurementSet

Associations
   appliesFor : MeasurableElement[*]  Relates the MeasurementSet to the MeasurableElement that it is applicable to.

**PropertySet**

**Package:** Parameters  
isAbstract: Yes

**Generalization:** UAFElement  
**Extension:** Element

**Description**

An abstract grouping of architectural elements that can own Measurements.
Figure 3:19 - PropertySet

### 3.1.3 UAF::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.

### UAF::Metadata::Taxonomy

Contains the elements that contribute to the Metadata Taxonomy Viewpoint.

**ActualState**

**Package:** Taxonomy
**isAbstract:** Yes
**Generalization:** UAFElement
**Extension:** Element

**Description**

Abstract element that applies temporal extent to a set of elements realized as Instance Specifications.
Figure 3:20 – ActualState

Attributes

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

ISO8601DateTime

**Package:** Taxonomy

**isAbstract:** No

**Generalization:** UAFEElement

**Extension:** LiteralString

Description

A date and time specified in the ISO8601 date-time format including timezone designator (TZD): YYYY-MM-DDThh:mm:ssTZD.

Figure 3:21 - ISO8601DateTime

UAF::Metadata::Connectivity

Contains the elements that contribute to the Metadata Connectivity Viewpoint.

**Exchange**

**Package:** Connectivity

**isAbstract:** Yes

**Generalization:** MeasurableElement, ItemFlow, SubjectOfSecurityConstraint
Extension: InformationFlow

Description
Abstract grouping for OperationalExchanges and ResourceExchanges that exchange Resources.

Figure 3:22 - Exchange

Resource
Package: Connectivity
isAbstract: Yes
Generalization: PropertySet
Extension: Element

Description
Abstract element grouping for all elements that can be conveyed by an Exchange.

Figure 3:23 - Resource

UAF::Metadata::Processes
Contains the elements that contribute to the Metadata Processes Viewpoint.

Activity
Package: Processes
isAbstract: Yes

**Generalization:** MeasurableElement

**Extension:** Activity

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

![Activity Diagram]

**Figure 3:24 – Activity**

**Associations**
- `activityPerformableUnderCondition : ActualCondition[*]` The environment under which an activity is performed.

**CapabilityElement**

**Package:** Processes

isAbstract: Yes

**Generalization:** UAFelement

**Extension:** Element

**Description**
An abstract type that represents a structural element that can exhibit capabilities.
Figure 3:25 - CapableElement

**IsCapableToPerform**

**Package:** Processes

**isAbstract:** No

**Generalization:** MeasurableElement, Allocate

**Extension:** Abstraction

**Description**

An Abstraction relationship defining the traceability between the structural elements to the Activities that they can perform.

Figure 3:26 – IsCapableToPerform
Constraints

[1] IsCapableOfPerforming.client
In case of value for IsCapableToPerform.supplier is stereotyped:
  a. «OperationalActivity» or its specializations, values for the client metaproperty must be stereotyped by any of specializations of «OperationalAgent»,
  b. «ServiceFunction» or its specializations, values for the client metaproperty must be stereotyped «ServiceSpecification» or its specializations,
  c. «Function» or its specializations, except for «ProjectActivity», values for the client metaproperty must be stereotyped by any of specializations of «ResourcePerformer»,
  d. «ProjectActivity» or its specializations, values for the client metaproperty must be stereotyped by any of specializations of «Project».

[2] IsCapableOfPerforming.supplier
In case of value for IsCapableToPerform.client is stereotyped:
  a. by a specialization of «OperationalAgent», values for the supplier metaproperty must be stereotyped «OperationalActivity» or its specializations,
  b. «ServiceSpecification» or its specializations, values for the supplier metaproperty must be stereotyped «ServiceFunction» or its specializations,
  c. by a specialization of «ResourcePerformer», values for the supplier metaproperty must be stereotyped «Function» or its specializations, except for «ProjectActivity»,
  d. by a specialization of «Project», values for the supplier metaproperty must be stereotyped «ProjectActivity» or its specializations.

PerformsInContext

Package: Processes
isAbstract: No
Generalization: MeasurableElement, Allocate
Extension: Abstraction

Description
An abstraction relationship that relates an OperationalAction to a 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.
Figure 3.27 – PerformsInContext

Constraints

[1] PerformsInContext.client
   In case of value for PerformsInContext.supplier is stereotyped:
   a. «OperationalActivityAction» or its specializations, values for the client metaproperty must be stereotyped «OperationalRole» or its specializations,
   b. «ServiceFunctionAction» or its specializations, values for the client metaproperty must be stereotyped «ServiceSpecificationRole» or its specializations,
   c. «FunctionAction» or its specializations, except for «ProjectActivityAction», values for the client metaproperty must be stereotyped «ResourceRole» or its specializations.
   d. «ProjectActivityAction» or its specializations, values for the client metaproperty must be stereotyped «ProjectRole» or its specializations.

[2] PerformsInContext.supplier
   In case of value for PerformsInContext.client is stereotyped:
   a. «OperationalRole» or its specializations, values for the supplier metaproperty must be stereotyped «OperationalActivityAction» or its specializations,
   b. «ServiceSpecificationRole» or its specializations, values for the supplier metaproperty must be stereotyped «ServiceFunctionAction» or its specializations,
   c. «ResourceRole» or its specializations, values for the supplier metaproperty must be stereotyped «FunctionAction» or its specializations.

UAF::Metadata::Information

Contains the elements that contribute to the Metadata Information Viewpoint.

ArchitectureMetadata

Package: Information
isAbstract: No
Generalization: Metadata
**Extension:** Comment

**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.

![Diagram of ArchitectureMetadata](image)

**Figure 3:28 – ArchitectureMetadata**

**Constraints**

[1] ArchitectureMetadata.annotatedElement Value for the annotatedElement metaproperty must be stereotyped «ArchitecturalDescription» or its specializations.

**DataModel**

**Package:** Information

**isAbstract:** No

**Generalization:** SubjectOfOperationalConstraint, SubjectOfResourceConstraint

**Extension:** Package

**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).

![Diagram of DataModel](image)

**Figure 3:29 – DataModel**

**Associations**

kind : DataModelKind[] Captures the kind of DataModel being represented, Conceptual, Logical or Physical.

**DataModelKind**

**Package:** Information
Description

Enumeration of the possible kinds of DataModel. Its enumeration literals are:

- **Conceptual** - Indicates that the DataModel associated with the DataModelKind is a conceptual DataModel that defines the required high-level data concepts and their relationships.
- **Logical** - Indicates that the DataModel associated with the DataModelKind is a logical data model that allows analysis of an architecture’s data definition aspect, without consideration of implementation specific or product specific issues. It details the conceptual data model.
- **Physical** - Indicates that the DataModel associated with the DataModelKind is a physical data model that is an implementable specification of a data structure. A physical data model realizes a logical data model, taking into account implementation restrictions and performance issues while still enforcing the constraints, relationships and typing of the logical data model.

**Information**

**Package**: Information

**isAbstract**: No

**Generalization**: MeasurableElement

**Extension**: Comment

Description

A comment that describes the state of an item of interest in any medium or form -- and is communicated or received.

![Diagram of Information](image)

**Figure 3:30 – Information**

**Associations**


**Constraints**

- [1] Information.annotatedElement  Value for the annotatedElement metaproperty must be stereotyped by a specialization of «UAFElememt».
**InformationKind**

**Package:** Information  
**isAbstract:** No

**Description**

Enumeration of the possible kinds of Information. Its enumeration literals are:

- **Information** - Indicates that the Information associated with the InformationKind describes the state of a something of interest that is materialized -- in any medium or form -- and communicated or received.
- **DomainInformation** - Indicates that the Information associated with the InformationKind describes information within the scope or domain of the architecture.
- **PositionReferenceFrame** - Indicates that the Information associated with the InformationKind describes an arbitrary set of axes with reference to which the position or motion of something is described or physical laws are formulated.
- **PedigreeInformation** - Indicates that the Information associated with the InformationKind describes information pedigree.
- **Data** - Indicates that the Information associated with the InformationKind describes the representation of information in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means. Examples could be whole models, packages, entities, attributes, classes, domain values, enumeration values, records, tables, rows, columns, and fields.

**Metadata**

**Package:** Information  
**isAbstract:** No

**Generalization:** MeasurableElement  
**Extension:** Comment

**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.

![Figure 3:31 – Metadata](image-url)
Attributes

category : String[0..1]  Defines the category of a Metadata element example: http://purl.org/dc/terms/abstract.
dublinCoreElement : 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.

Constraints

[1] Metadata.annotatedElement  Value for the annotatedElement metaproperty must be stereotyped by a specialization of «UAFElement».

**UAF::Metadata::Constraints**

Contains the elements that contribute to the Metadata Constraints Viewpoint.

**Rule**

**Package:** Constraints

**isAbstract:** Yes

**Generalization:** MeasurableElement

**Extension:** Constraint

**Description**

An abstract grouping for all types of constraint (i.e. an OperationalConstraint could detail the rules of accountancy best practice).

**Figure 3:32 – Rule**

**Associations**

ruleKind : RuleKind[1]  Captures the kind of Rule that is being described.

**RuleKind**

**Package:** Constraints

**isAbstract:** No
Description

Enumeration of the possible kinds of Rules applicable to constraints. Its enumeration literals are:

- **StructuralAssertion** - Indicates that the Rule associated with the RuleKind is a statement that details that something of importance either exists as a concept of interest or exists in relationship to another thing of interest.
- **ActionAssertion** - Indicates that the Rule associated with the RuleKind is a statement that concerns some dynamic aspect.
- **Derivation** - Indicates that the Rule associated with the RuleKind is a statement that details a Rule derived from another Rule.
- **Contract** - Indicates that the Rule associated with the RuleKind is a statement that details a consent among parties regarding the terms and conditions of activities that said parties participate in.
- **Constraint** - Indicates that the Rule associated with the RuleKind is a statement that details a limitation, e.g. business rule, restraint, operational limitation.
- **Guidance** - Indicates that the Rule associated with the RuleKind is a statement that details an authoritative statement intended to lead or steer the execution of actions.
- **SecurityPolicy** - Indicates that the Rule associated with the RuleKind is a statement that details a constraint that specifies policy for information handling, physical security, encryption, etc.
- **Caveat** - Indicates that the Rule associated with the RuleKind is a statement that details alternate conditions under which the rule is not valid.

**UAF::Metadata::Traceability**

Contains the elements that contribute to the Metadata Traceability Viewpoint.

**ArchitecturalReference**

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

**Description**

A dependency relationship that specifies that one architectural description refers to another.

![Diagram of ArchitecturalReference](image)

**Figure 3:33 – ArchitecturalReference**

**Constraints**

1. **ArchitecturalReference.client**  
   Value for the client metaproperty must be stereotyped «ArchitecturalDescription» or its specializations.

2. **ArchitecturalReference.supplier**  
   Value for the supplier metaproperty must be stereotyped «ArchitecturalDescription» or its specializations.
Implements
Package: Traceability
isAbstract: No
Generalization: Allocate, MeasurableElement
Extension: Abstraction

Description
An abstraction relationship 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.

Figure 3:34 – Implements
Constraints

[1] Implements.client  
In case of value for Implements.supplier is stereotyped:
  a. by any of specializations of «OperationalAgent», values for the client metaproperty must be stereotyped by any of specializations of «ResourcePerformer»,
  b. «OperationalActivity» or its specializations, values for the client metaproperty must be stereotyped «Function» or its specializations,
  c. «ServiceFunction» or its specializations, values for the client metaproperty must be stereotyped «Function» or its specializations,
  d. «ServiceInterface» or its specializations, values for the client metaproperty must be stereotyped «ResourceInterface» or its specializations,
  e. «OperationalInterface» or its specializations, values for the client metaproperty must be stereotyped «ResourceInterface» or its specializations,
  f. «OperationalConnector» or its specializations, values for the client metaproperty must be stereotyped «ResourceConnector» or its specializations,
  g. «OperationalExchange» or its specializations, values for the client metaproperty must be stereotyped «ResourceExchange» or its specializations,
  h. «OperationalRole» or its specializations, values for the client metaproperty must be stereotyped «ResourceRole» or its specializations,
  i. «ActualEnduringTask» or its specializations, values for the client metaproperty must be stereotyped «OperationalActivity» or its specializations,
  j. «InformationElement» or its specializations, values for the client metaproperty must be stereotyped «DataElement» or its specializations.

[2] Implements.supplier  
In case of value for Implements.client is stereotyped:
  a. by any of specializations of «ResourcePerformer», values for the supplier metaproperty must be stereotyped by any of specializations of «OperationalAgent»,
  b. «Function» or its specializations, values for the supplier metaproperty must be stereotyped «OperationalActivity», «ServiceFunction» or their specializations,
  c. «ResourceInterface» or its specializations, values for the supplier metaproperty must be stereotyped «ServiceInterface», «OperationalInterface», or their specializations,
  d. «ResourceConnector» or its specializations, values for the supplier metaproperty must be stereotyped «OperationalConnector», «ResourceConnector» or their specializations,
  e. «ResourceExchange» or its specializations, values for the supplier metaproperty must be stereotyped «OperationalExchange» or its specializations,
  f. «ResourceRole» or its specializations, values for the supplier metaproperty must be stereotyped «OperationalRole» or its specializations,
  g. «OperationalActivity» or its specializations, values for the supplier metaproperty must be stereotyped «ActualEnduringTask» or its specializations,
  h. «DataElement» or its specializations, values for the supplier metaproperty must be stereotyped «InformationElement» or its specializations.

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

UAF::Strategic::Taxonomy
Contains the elements that contribute to the Strategic Taxonomy Viewpoint.

ActualEnterprisePhase
Package: Taxonomy
isAbstract: No
Generalization: ActualPropertySet, CapableElement, Achiever
Extension: InstanceSpecification

Description
A phase of an actual enterprise endeavor.

Figure 3:35 – ActualEnterprisePhase

Associations
concern : Concern[*]
goal : EnterpriseGoal[*] The Goal towards which this Phase is directed and is in support of.
operationalArchitectureOfEnterprisePhase : OperationalArchitecture[*] Relates an ActualEnterprisePhase to its relevant OperationalArchitecture.
statementTask : ActualEnduringTask[*] Relates the ActualEnterprisePhase to the ActualEnduringTasks that are intended to be implemented during that phase.
vision : EnterpriseVision[*] The Vision towards which this Phase is directed and is in support of.
Constraints

[1] ActualEnterprisePhase.classifier  Value for the classifier metaproperty must be stereotyped by «EnterprisePhase» or its specializations.

[2] ActualEnterprisePhase.start/endDate  Must fall within the start and end dates of the enclosing ActualEnterprisePhase having this ActualEnterprisePhase set as a value for a slot.

Capability

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, Desirer, Block
Extension: Class

Description

An enterprise's ability to Achieve a DesiredEffect realized through a combination of ways and means (e.g., CapabilityConfigurations) along with specified measures.

EnterpriseGoal

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, Requirement
Extension: Class
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. https://www.omg.org/spec/BMM/1.3/

**Figure 3:37 – EnterpriseGoal**

**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.

**Associations**

- **enterprisePhase : ActualEnterprisePhase[*]** Relates the EnterpriseGoal to the ActualEnterprisePhase in which the EnterpriseGoal is attained.

**EnterprisePhase**

- **Package:** Taxonomy
- **isAbstract:** No
- **Generalization:** PropertySet, Block
- **Extension:** Class

Description

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

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, Block
Extension: Class

Description
A Vision describes the future state of the enterprise, without regard to how it is to be achieved. https://www.omg.org/spec/BMM/1.3/
Associations

enterprisePhase : ActualEnterprisePhase[0..1]  
Relates the EnterpriseVision to the ActualEnterprisePhase in which the EnterpriseVision is expected to be realized.

statement : VisionStatement[*]  
A description of the Vision.

VisionStatement

Package: Taxonomy
isAbstract: No
Generalization: MeasurableElement
Extension: Comment

Description
A type of comment that describes the future state of the enterprise, without regard to how it is to be achieved.

https://www.omg.org/spec/BMM/1.3/

Figure 3:40 – VisionStatement

Constraints

[1] VisionStatement.ownedAttribute  
Values for annotatedElement metaproperty must be stereotyped «EnterpriseVision» or its specializations.

WholeLifeEnterprise

Package: Taxonomy
isAbstract: No
Generalization: EnterprisePhase
Extension: Class

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

Contains the elements that contribute to the Strategic Structure Viewpoint.

CapabilityRole

Package: Structure
isAbstract: No
Generalization: MeasurableElement, Desirer
Extension: Property

Description

Property of a Capability typed by another Capability, enabling whole-part relationships and structures.

StructuralPart

Package: Structure
isAbstract: No
Generalization: MeasurableElement
Extension: Property
Description
Usage of an EnterprisePhase in the context of another EnterprisePhase. It asserts that one EnterprisePhase is a spatial part of another. Creates a whole-part relationship that represents the structure of the EnterprisePhase.

Figure 3:43 – StructuralPart

Constraints
[1] StructuralPart.class Value for class metaproperty must be stereotyped «EnterprisePhase» or its specializations.
[2] StructuralPart.type Value for type metaproperty must be stereotyped «EnterprisePhase» or its specializations.

TemporalPart

Package: Structure
isAbstract: No
Generalization: MeasurableElement
Extension: Property

Description
Usage of an EnterprisePhase in the context of another EnterprisePhase. It asserts that one EnterprisePhase is a spatial part of another. Creates a whole-part relationship that represents the temporal structure of the EnterprisePhase.

Figure 3:44 - TemporalPart

Constraints
[1] TemporalPart.class Value for class metaproperty must be stereotyped «EnterprisePhase» or its specializations.
[2] TemporalPart.type Value for type metaproperty must be stereotyped «EnterprisePhase» or its specializations.

UAF::Strategic::Processes

Contains the elements that contribute to the Strategic Processes Viewpoint.
**ActualEnduringTask**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** [CapableElement, ActualPropertySet]  
**Extension:** InstanceSpecification

**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.

**Constraints**

[1] ActualEnduringTask.classifier  
Value for the classifier metaproperty must be stereotyped by «EnduringTask» or its specializations.

**CapabilityForTask**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** [MeasurableElement, Allocate]  
**Extension:** Abstraction

**Description**

An abstraction relationship that asserts that a Capability is required in order for an Enterprise to conduct a phase of an EnduringTask.
Constraints
[1] CapabilityForTask.client Value for the client metaproperty must be stereotyped «Capability» or its specializations.
[2] CapabilityForTask.supplier Value for the supplier metaproperty must be stereotyped «ActualEnduringTask» or its specializations.

EnduringTask
Package: Processes
isAbstract: No
Generalization: PropertySet, Block
Extension: Class

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.

AchievedEffect
Package: States
isAbstract: No
Generalization: MeasurableElement
Extension: Dependency
Description

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

Figure 3:48 – AchievedEffect

Associations

desiredEffect : DesiredEffect[*]  Relates the effect that is achieved with the originally expected DesiredEffect. Providing a means of comparison, between the expectation of the desirer and the actual result.

Constraints

[1] AchievedEffect.client  Value for the client metaproperty must be stereotyped by the specialization of «Achiever».

[2] AchievedEffect.supplier  Value for the supplier metaproperty must be stereotyped by the specialization of «ActualState».

Achiever

Package: States

isAbstract: Yes

Generalization: UAFElement

Extension: InstanceSpecification

Description

An ActualResource, ActualProject or ActualEnterprisePhase that can deliver a DesiredEffect.
Figure 3:49 - Achiever

**DesiredEffect**

Package: States
isAbstract: No

**Generalization**: MeasurableElement

**Extension**: Dependency

**Description**

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

Figure 3:50 – DesiredEffect

**Associations**

- achievedEffect : AchievedEffect[*]

**Constraints**

1. DesiredEffect.client Value for the client metaproperty must be stereotyped a specialization of «Desirer».
2. DesiredEffect.supplier Value for the supplier metaproperty must be stereotyped a specialization of «ActualState».
**Desirer**
Package: States
isAbstract: Yes

**Generalization:** UAFElement

**Extension:** Element

Description
Abstract element used to group architecture elements that might desire a particular effect.

**Figure 3:51 - Desirer**

**UAF::Strategic::Traceability**
Contains the elements that contribute to the Strategic Traceability Viewpoint.

**Exhibits**

**Package:** Traceability
isAbstract: No

**Generalization:** MeasurableElement, Allocate

**Extension:** Abstraction

Description
An abstraction relationship that exists between a CapableElement and a Capability that it meets under specific environmental conditions.
Associations

environmentalConditions : Environment[*]  Defines the environmental conditions constraining the way that a Capability is exhibited.

Constraints

[1] Exhibits.client  Value for the client metaproperty must be stereotyped a specialization of «CapableElement».
[2] Exhibits.supplier  Value for the supplier metaproperty must be stereotyped «Capability».

MapsToCapability

Package: Traceability

isAbstract: No

Generalization: MeasurableElement, Allocate

Extension: Abstraction

Description

An Abstraction relationship denoting that an Activity contributes to providing a Capability.

Constraints

[1] MapsToCapability.client  Value for the client metaproperty must be stereotyped a specialization of «Activity».
[2] MapsToCapability.supplier  Value for the supplier metaproperty must be stereotyped «Capability».
**OrganizationInEnterprise**

**Package:** Traceability  
**isAbstract:** No  
**Generalization:** MeasurableElement, Allocate  
**Extension:** Abstraction

**Description**

An abstraction relationship relating an ActualOrganization to an ActualEnterprisePhase to denote that the ActualOrganization plays a role or is a stakeholder in an ActualEnterprisePhase.

**Constraints**

[1] OrganizationInEnterprise.client Value for the client metaproperty must be stereotyped «ActualOrganization» or its specializations.

[2] OrganizationInEnterprise.supplier Value for the supplier metaproperty must be stereotyped «ActualEnterprisePhase» or its specializations.

### 3.1.5 UAF::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.

**UAF::Operational::Taxonomy**

Contains the elements that contribute to the Operational Taxonomy Viewpoint.

**ArbitraryConnector**

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

**Description**

Represents a visual indication of a connection used in high level operational concept diagrams.
Figure 3.55 – ArbitraryConnector

Constraints

[1] ArbitraryConnector.client The value for client metaproperty has to be stereotyped «ConceptRole» or its specializations.

[2] ArbitraryConnector.supplier The value for supplier metaproperty has to be stereotyped «ConceptRole» or its specializations.

ConceptItem

Package: Taxonomy
isAbstract: Yes
Generalization: UAFElement
Extension: Element

Description

Abstract, an item which may feature in a HighLevelOperationalConcept.

Figure 3.56 - ConceptItem

ConceptRole

Package: Taxonomy
isAbstract: No
Generalization: MeasurableElement
Extension: Property
Description

Usage of a ConceptItem in the context of a HighLevelOperationalConcept.

Constraints

[1] ConceptRole.class  Value for the class metaproperty must be stereotyped «HighLevelOperationalConcept» or its specializations.

[2] ConceptRole.type  Value for the type metaproperty must be stereotyped by a specialization of «ConceptItem».

HighLevelOperationalConcept

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, Block
Extension: Class

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.
**UAF::Operational::Structure**
Contains the elements that contribute to the Operational Structure Viewpoint.

**KnownResource**
**Package:** Structure
**isAbstract:** No
**Generalization:** OperationalPerformer, ResourcePerformer
**Extension:** Class

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

![Diagram of KnownResource](image)

**Figure 3:59 - KnownResource**

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

**Description**
An abstract type grouping OperationalArchitecture and OperationalPerformer.
Figure 3:60 – OperationalAgent

Constraints

[1] OperationalAgent.isCapableToPerform
   Is capable of performing only «OperationalActivity» elements or its specializations.

[2] OperationalAgent.ownedOperation
   Values for the ownedOperation metaproperty must be stereotyped «OperationalMethod» or its specializations.

[3] OperationalAgent.ownedPort
   Values for the ownedPort metaproperty must be stereotyped «OperationalPort» or its specializations.

OperationalArchitecture

Package: Structure
isAbstract: No
Generalization: OperationalAgent, Architecture
Extension: Class

Description

An element used to denote a model of the Architecture, described from the Operational perspective.
**OperationalMethod**

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

**Description**

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

---

Unified Architecture Framework Profile (UAFP) Version 1.1
Constraints

[1] OperationalMethod.method  Value for the method metaproperty must be stereotyped «OperationalActivity» or its specializations.

[2] OperationalMethod.ownedParameter  The values for the ownedParameter metaproperty must be stereotyped «OperationalParameter» or its specializations.

**OperationalParameter**

**Package:** Structure

**isAbstract:** No

**Generalization:** MeasurableElement

**Extension:** Parameter

**Description**

An element that represents inputs and outputs of an OperationalActivity. It is typed by an OperationalExchangeItem.

![Figure 3:63 – OperationalParameter](image)

Constraints

[1] OperationalParameter.type  Value for the type metaproperty must be stereotyped by specialization of «OperationalExchangeItem».

**OperationalPerformer**

**Package:** Structure

**isAbstract:** No

**Generalization:** OperationalAgent

**Extension:** Class

**Description**

A logical agent that IsCapableToPerform OperationalActivities which produce, consume and process Resources.
Figure 3:64 - OperationalPerformer

**OperationalPort**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** MeasurableElement, ProxyPort  
**Extension:** Port

**Description**

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

Figure 3:65 – OperationalPort

**Constraints**

1. OperationalPort.class  
   Value for class metaproperty must be stereotyped «OperationalAgent» or its specializations.

2. OperationalPort.type  
   Value for type metaproperty must be stereotyped «OperationalInterface» or its specializations.
**OperationalRole**

**Package:** Structure  
isAbstract: No  
**Generalization:** MeasurableElement, LocationHolder, AssetRole  
**Extension:** Property

**Description**

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

**Figure 3:66 – OperationalRole**

Constraints

1. OperationalRole.class Value for class metaproperty must be stereotyped by a specialization of «OperationalAgent».
2. OperationalRole.type Value for type metaproperty must be stereotyped by a specialization of «OperationalAgent».

**ProblemDomain**

**Package:** Structure  
isAbstract: No  
**Generalization:** OperationalRole  
**Extension:** Property

**Description**

A property associated with an OperationalArchitecture, used to specify the scope of the problem.
Constraints

[1] ProblemDomain.class  Value for the class metaproperty must be stereotyped «OperationalArchitecture» or its specializations.

[2] ProblemDomain.type  Value for the type metaproperty must be stereotyped «OperationalPerformer» or its specializations.

**UAF::Operational::Connectivity**

Contains the elements that contribute to the Operational Connectivity Viewpoint.

**OperationalConnector**

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

Description

A Connector that goes between OperationalRoles representing a need to exchange Resources. It can carry a number of OperationalExchanges.
Figure 3:68 – OperationalConnector

Constraints

[1] OperationalConnector.end  The value for the role metaproperty for the owned ConnectorEnd must be stereotype «OperationalRole»/«OperationalPort» or its specializations.

OperationalExchange

Package: Connectivity

isAbstract: No

Generalization: Exchange, SubjectOfOperationalConstraint

Extension: InformationFlow

Description

Asserts that a flow can exist between OperationalPerformers (i.e. flows of information, people, materiel, or energy).
Figure 3:69 – OperationalExchange

Attributes

trustLevel : Real[0..1] Captures the directional arbitrary level of trust related to an OperationalExchange between two OperationalPerformers.

Associations

exchangeKind : OperationalExchangeKind[0..1] Captures the kind of Resource being exchanged.

Constraints

[1] OperationalExchange.conveyed In case of OperationalExchange.operationalExchangeKind:
= InformationExchange, the conveyed element must be stereotyped «InformationElement» or its specializations,
= MaterielExchange, the conveyed element must be stereotyped «ResourceArtifact» or its specializations,
= EnergyExchange, the conveyed element must be stereotyped «NaturalResource» or its specializations,
= OrganizationalExchange, the conveyed element must be stereotyped «OrganizationalResource» or its specializations,
= ConfigurationExchange, the conveyed element must be stereotyped «CapabilityConfiguration» or its specializations, or
= GeoPoliticalExtentExchange, the conveyed element must be stereotyped «GeoPoliticalExtentType» or its specializations.

[2] OperationalExchange.informationSource Value for informationSource metaproperty has to be stereotyped «OperationalPerformer» or its specializations.

[3] OperationalExchange.informationTarget Value for informationTarget metaproperty has to be stereotyped «OperationalPerformer» or its specializations.

[4] OperationalExchange.realizingActivityEdge Value for realizingActivityEdge metaproperty has to be stereotyped by any specialization of «OperationalActivityEdge».

[5] OperationalExchange.realizingConnector Value for realizingConnector metaproperty has to be stereotyped «OperationalConnector» or its specializations.

[6] OperationalExchange.realizingMessage Value for realizingMessage metaproperty has to be stereotyped «OperationalMessage» or its specializations.
OperationalExchangeItem

Package: Connectivity
isAbstract: Yes
Generalization: Resource

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

Figure 3:70 – OperationalExchangeItem

Associations
activity : OperationalActivity[*] A collection of OperationalActivities that consume and/or produce the OperationalExchangeItem internally.

OperationalExchangeKind

Package: Connectivity
isAbstract: No

Description
Enumeration of the possible kinds of operational exchange applicable to an OperationalExchange. Its enumeration literals are:

- MaterielExchange - Indicates that the OperationalExchange associated with the OperationalExchangeKind is a logical flow of materiel (artifacts) between Functions.
- OrganizationalExchange - Indicates that the OperationalExchange associated with the OperationalExchangeKind is a logical flow where human resources (PostTypes, RoleTypes) flow between OperationalPerformers.
- EnergyExchange - Indicates that the OperationalExchange associated with the OperationalExchangeKind is a logical flow where energy is flowed from one OperationalPerformer to another.
- InformationExchange - Indicates that the OperationalExchange associated with the OperationalExchangeKind is a logical flow where information is flowed from one OperationalPerformer to another.
- ConfigurationExchange - Indicates that the OperationalExchange associated with the OperationalExchangeKind is a logical flow where CapabilityConfigurations flow from one OperationalPerformer to another.
- GeoPoliticalExtentExchange - Indicates that the OperationalExchange associated with the OperationalExchangeKind is a logical flow where GeoPoliticalExtentTypes (i.e. Borders) flow from one place to another.

**OperationalInterface**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** PropertySet, InterfaceBlock  
**Extension:** Class

**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 3:71 – OperationalInterface**

**Constraints**

[1] OperationalInterface.ownedOperation  
Values for the ownedOperation metaproperty must be stereotyped «OperationalMethod» or its specializations.

**OperationalSignal**

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

An OperationalSignal is a specification of a kind of communication between operational performers in which a reaction is asynchronously triggered in the receiver without a reply.

Figure 3:72 - OperationalSignal

OperationalSignalProperty

Package: Connectivity
isAbstract: No
Generalization: MeasurableElement
Extension: Property

Description

A property of an OperationalSignal typed by OperationalExchangeItem. It enables OperationalExchangeItem e.g. InformationElement to be passed as arguments of the OperationalSignal.

Figure 3:73 – OperationalSignalProperty

Constraints

[1] OperationalSignalProperty.class Value for class metaproperty must be stereotyped «OperationalSignal» or its specializations.

[2] OperationalSignalProperty.type Value for type metaproperty must be stereotyped by a specialization of «OperationalExchangeItem».

UAF::Operational::Processes

Contains the elements that contribute to the Operational Processes Viewpoint.

OperationalActivity

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

**Figure 3:74 – OperationalActivity**

**Associations**

affectedResource : OperationalExchangeItem[*]  A collection of OperationalExchangeItems consumed and produced internally within the OperationalActivity.

**Constraints**

[1] OperationalActivity.ownedParameter  The values for the ownedParameter metaproperty must be stereotyped «OperationalParameter» or its specializations.

**OperationalActivityAction**

**Package**: Processes

**isAbstract**: No

**Generalization**: MeasurableElement

**Extension**: CallBehaviorAction

**Description**

A call of an OperationalActivity in the context of another OperationalActivity.
Figure 3:75 – OperationalActivityAction

Constraints

[1] OperationalActivityAction.activity Value for the activity metaproperty must be stereotyped «OperationalActivity» or its specializations.

[2] OperationalActivityAction.behavior Value for activity metaproperty must be stereotyped «OperationalActivity» or its specializations.

OperationalActivityEdge

Package: Processes
isAbstract: Yes

Generalization: MeasurableElement

Extension: ActivityEdge

Description
Abstract grouping for OperationalControlFlow and OperationalObjectFlow.

Figure 3:76 – OperationalActivityEdge

Constraints

[1] OperationalActivityEdge.owner «OperationalActivityEdge» must be owned directly or indirectly by «OperationalActivity» or its specializations.

OperationalControlFlow

Package: Processes
isAbstract: No
**Generalization:** OperationalActivityEdge

**Extension:** ControlFlow

**Description**

An ActivityEdge that shows the flow of control between OperationalActivityActions.

![Diagram of OperationalControlFlow](image)

**Constraints**

1. OperationalControlFlow.source
   - Value for the source metaproperty must be stereotyped «OperationalActivityAction» or its specializations.
2. OperationalControlFlow.target
   - Value for the target metaproperty must be stereotyped «OperationalActivityAction» or its specializations.

**OperationalObjectFlow**

**Package:** Processes

**isAbstract:** No

**Generalization:** OperationalActivityEdge

**Extension:** ObjectFlow

**Description**

An ActivityEdge that shows the flow of Resources (objects/information) between OperationalActivityActions.

![Diagram of OperationalObjectFlow](image)

**StandardOperationalActivity**

**Package:** Processes

**isAbstract:** No

**Generalization:** OperationalActivity
**Extension**: Activity

**Description**
A sub-type of OperationalActivity that is a standard operating procedure.

![Diagram of StandardOperationalActivity](image)

**Figure 3:79 - StandardOperationalActivity**

**UAF::Operational::States**
Contains the elements that contribute to the Operational States Viewpoint.

**OperationalStateDescription**
Package: States
isAbstract: No
**Generalization**: MeasurableElement
**Extension**: StateMachine

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

![Diagram of OperationalStateDescription](image)

**Figure 3:80 – OperationalStateDescription**

**Constraints**
[1] OperationalStateDescription.owner Values for the owner metaproperty must be stereotyped with specializations of «OperationalAgent».

**UAF::Operational::Interaction Scenarios**
Contains the elements that contribute to the Operational Interaction Scenarios Viewpoint.

**OperationalMessage**
Package: Interaction Scenarios
isAbstract: No
**Generalization**: MeasurableElement
**Extension**: Message
Description
Message for use in an operational interaction scenario which carries any of the subtypes of OperationalExchange.

Figure 3:81 – OperationalMessage

Constraints
[1] OperationalMessage.receiveEvent.event.operation Values for the receiveEvent.event.operation metaproperty must be stereotyped with «OperationalMethod» or its specializations.

UAF::Operational::Information
Contains the elements that contribute to the Operational Information Viewpoint.

InformationElement
Package: Information
isAbstract: No
Generalization: OperationalAsset, OperationalExchangeItem, SubjectOfOperationalConstraint
Extension: Class

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).
Figure 3:82 – InformationElement

Constraints
[1] InformationElement.owner Values for the owner metaproperty must be stereotyped «DataModel» or its specializations.

**UAF::Operational::Constraints**

Contains the elements that contribute to the Operational Constraints Viewpoint.

**OperationalConstraint**

Package: Constraints
isAbstract: No
Generalization: Rule
Extension: Constraint

Description
A Rule governing an operational architecture element i.e. OperationalPerformer, OperationalActivity, InformationElement etc.

Figure 3:83 – OperationalConstraint

Constraints
[1] OperationalConstraint.constrainedElement Value for the constrainedElement metaproperty must be stereotyped by any specialization of «SubjectOfOperationalConstraint».
SubjectOfOperationalConstraint

Package: Constraints
isAbstract: Yes
Generalization: UAFElement
Extension: Element

Description
An abstract grouping of elements that can be the subject of an OperationalConstraint.

Figure 3:84 - SubjectOfOperationalConstraint

3.1.6 UAF::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.

UAF::Services::Taxonomy

Contains the elements that contribute to the Services Taxonomy Viewpoint.

ServiceSpecification

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, VersionedElement, CapableElement, Block
Extension: Class

Description
The specification of a set of functionalities provided by one element for the use of others.
Figure 3:85 - ServiceSpecification

UAF::Services::Structure

Contains the elements that contribute to the Services Structure Viewpoint.

**ServiceMethod**

Package: Structure  
isAbstract: No  
Generalization: MeasurableElement  
Extension: Operation

**Description**

A behavioral feature of a ServiceSpecification whose behavior is specified in a ServiceFunction.

Figure 3:86 – ServiceMethod

**Constraints**

[1] ServiceMethod.method Value for the method metaproperty must be stereotyped «ServiceFunction» or its specializations.
[2] ServiceMethod.ownedParameter  The values for the ownedParameter metaproperty must be stereotyped «ServiceParameter» or its specializations.

[3] ServiceMethod.owner  The values for the owner metaproperty must be stereotyped «ServiceSpecification» or its specializations.

**ServiceParameter**

**Package:** Structure

isAbstract: No

**Generalization:** MeasurableElement

**Extension:** Parameter

**Description**

An element that represents inputs and outputs of a ServiceFunction, represents inputs and outputs of a ServiceSpecification.

---

**Figure 3:87 – ServiceParameter**

**Constraints**

[1] ServiceParameter.type  The values for the type metaproperty must be stereotyped a specialization of «OperationalExchangeItem».

**ServicePort**

**Package:** Structure

isAbstract: No

**Generalization:** ProxyPort, MeasurableElement

**Extension:** Port

**Description**

An interaction point for a ServiceSpecification through which it can interact with the outside environment and which is defined by a ServiceInterface.
Constraints

[1] ServicePort.class Value for the class metaproperty must be stereotyped «ServiceSpecification» or its specializations.

[2] ServicePort.type Value for the type metaproperty must be stereotyped «ServiceInterface» or its specializations.

**ServiceSpecificationRole**

**Package:** Structure

**isAbstract:** No

**Generalization:** MeasurableElement

**Extension:** Property

**Description**

Figure 3:89 – ServiceSpecificationRole

Constraints
[1] ServiceSpecificationRole.class Value for the class metaproperty must be stereotyped «ServiceSpecification» or its specializations.
[2] ServiceSpecificationRole.type Value for the type metaproperty must be stereotyped «ServiceSpecification» or its specializations.

UAF::Services::Connectivity
Contains the elements that contribute to the Services Connectivity Viewpoint.

ServiceConnector
**Package:** Connectivity
**isAbstract:** No
**Generalization:** MeasurableElement
**Extension:** Connector

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

Constraints
[1] ServiceConnector.end The value for the role metaproperty for the owned ConnectorEnd must be stereotyped «ServicePort», «ServiceSpecificationRole» or their specializations.

ServiceInterface

Package: Connectivity
isAbstract: No
Generalization: PropertySet, InterfaceBlock
Extension: Class

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

Figure 3:91 – ServiceInterface

Constraints
[1] ServiceInterface.ownedOperation Values for the ownedOperation metaproperty must be stereotyped «ServiceMethod» or its specializations.

UAF::Services::Processes

Contains the elements that contribute to the Services Processes Viewpoint.
**ServiceFunction**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** Activity  
**Extension:** Activity

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

**Figure 3:92 – ServiceFunction**

**Constraints**
[1] ServiceFunction.ownedParameter The values for the ownedParameter metaproperty must be stereotyped «ServiceParameter».

**ServiceFunctionAction**

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

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

[1] ServiceFunctionAction.activity  Value for the behavior metaproperty must be stereotyped «ServiceFunction» or its specializations.


**UAF::Services::States**

Contains the elements that contribute to the Services States Viewpoint.

**ServiceStateDescription**

**Package:** States  
**isAbstract:** No  
**Generalization:** MeasurableElement  
**Extension:** StateMachine

**Description**

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

Figure 3:94 – ServiceStateDescription

Constraints

[1] ServiceStateMachine.owner  Values for the owner metaproperty must be stereotyped «ServiceSpecification» or its specializations.
**UAF::Services::Interaction Scenarios**
Contains the elements that contribute to the Services Interaction Scenarios Viewpoint.

**ServiceMessage**

- **Package:** Interaction Scenarios
- **isAbstract:** No
- **Generalization:** MeasurableElement
- **Extension:** Message

**Description**
Message for use in a Service Event-Trace.

**Figure 3:95 – ServiceMessage**

**Constraints**

[1] ServiceMessage.receiveEvent.event.operation

Values for the receiveEvent.event.operation metaproperty must be stereotyped with «ServiceMethod» or its specializations.

**UAF::Services::Constraints**
Contains the elements that contribute to the Services Constraints Viewpoint.

**ServicePolicy**

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

**Description**
A constraint governing the use of one or more ServiceSpecifications.
Constraints

[1] ServicePolicy.constrainedElement Values for constrainedElement metaproperty must be stereotyped «ServiceSpecification» or its specializations.

**UAF::Services::Traceability**

Contains the elements that contribute to the Services Traceability Viewpoint.

**Consumes**

**Package:** Traceability  
**isAbstract:** No  
**Generalization:** Allocate, MeasurableElement  
**Extension:** Abstraction

**Description**

An abstraction relationship that asserts that a service in someway contributes or assists in the execution of an OperationalActivity.

Constraints

[1] Consumes.client Value for the client metaproperty must be stereotyped «OperationalActivity» or its specializations.

[2] Consumes.supplier Value for the supplier metaproperty must be stereotyped «ServiceSpecification» or its specializations.
3.1.7 **UAF::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).

**UAF::Personnel::Taxonomy**

Contains the elements that contribute to the Personnel Taxonomy Viewpoint.

**Organization**

*Package:* Taxonomy
*isAbstract:* No
*Generalization:* [OrganizationalResource](#)
*Extension:* Class

Description

A group of OrganizationalResources (Persons, Posts, Organizations and Responsibilities) associated for a particular purpose.

![Diagram](image)

**Figure 3:98 - Organization**

**OrganizationalResource**

*Package:* Taxonomy
*isAbstract:* Yes
*Generalization:* [PhysicalResource](#), [Stakeholder](#)
*Extension:* Class

Description

An abstract element grouping for Organization, Person, Post and Responsibility.
Figure 3:99 – OrganizationalResource

**Person**

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

**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
Extension: Class

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

![Diagram of Post]

Figure 3:101 - Post

Responsibility
Package: Taxonomy
isAbstract: No
Generalization: OrganizationalResource
Extension: Class

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

![Diagram of Responsibility]

Figure 3:102 - Responsibility

UAF::Personnel::Connectivity
Contains the elements that contribute to the Personnel Connectivity Viewpoint.

Command
Package: Connectivity
isAbstract: No
Generalization: ResourceExchange
Extension: InformationFlow
Description

A type of ResourceExchange that asserts that one OrganizationalResource commands another.

![Figure 3:103 – Command](image)

Constraints

1. Command.conveyed
   Value for the conveyed metaproperty must be stereotyped «DataElement» or its specializations.

2. Command.informationSource
   Value for the informationSource metaproperty must be stereotyped by the specialization of «OrganizationalResource».

3. Command.informationTarget
   Value for the informationTarget metaproperty must be stereotyped by the specialization of «OrganizationalResource».

Control

Package: Connectivity
isAbstract: No
Generalization: ResourceExchange
Extension: InformationFlow

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 3:104 – Control](image)

Constraints
[1] Control.conveyed Value for the conveyed metaproperty must be stereotyped «DataElement» or its specializations.
[2] Control.informationSource Value for the informationSource metaproperty must be stereotyped by the specialization of «PhysicalResource».
[3] Control.informationTarget Value for the informationTarget metaproperty must be stereotyped by the specialization of «PhysicalResource» or its specializations.

**UAF::Personnel::Processes**
Contains the elements that contribute to the Personnel Processes Viewpoint.

**CompetenceToConduct**

*Package:* Processes  
*isAbstract:* No  
*Generalization:* MeasurableElement, Allocate  
*Extension:* Abstraction

**Description**
An abstraction relationship used to associate a Function with a specific set of Competencies needed to conduct the Function.

![Diagram of CompetenceToConduct](image)

**Figure 3:105 – CompetenceToConduct**

**Constraints**
[1] CompetenceToConduct.client Value for the client metaproperty must be stereotyped «Function» or its specializations.
[2] CompetenceToConduct.supplier Value for the supplier metaproperty must be stereotyped «Competence» or its specializations.

**UAF::Personnel::Constraints**
Contains the elements that contribute to the Personnel Constraints Viewpoint.

**Competence**

*Package:* Constraints  
*isAbstract:* No  
*Generalization:* SubjectOfForecast, PropertySet, Block  
*Extension:* Class
Description

A specific set of abilities defined by knowledge, skills and aptitude.

Figure 3:106 - Competence

**CompetenceForRole**

**Package:** Constraints
**isAbstract:** No

**Generalization:** MeasurableElement, Allocate

**Extension:** Abstraction

Description

An abstraction relationship used to associate an organizational role with a specific set of required competencies.

Figure 3:107 – CompetenceForRole

Constraints

[1] CompetenceForRole.client Value for the client metaproperty must be stereotyped «ResourceRole» or its specializations.

[2] CompetenceForRole.supplier Value for the supplier metaproperty must be stereotyped «Competence» or its specializations.
**RequiresCompetence**

**Package:** Constraints

isAbstract: No

**Generalization:** MeasurableElement, Allocate

**Extension:** Abstraction

**Description**

An abstraction relationship that asserts that an ActualOrganizationalResource is required to have a specific set of Competencies.

![Diagram of RequiresCompetence](image)

**Figure 3:108 – RequiresCompetence**

**Constraints**

1. RequiresCompetence.client Value for the client metaproperty must be stereotyped a specialization of «OrganizationalResource».

2. RequiresCompetence.supplier Value for the supplier metaproperty must be stereotyped «Competence» or its specializations.

**UAF::Personnel::Traceability**

Contains the elements that contribute to the Personnel Traceability Viewpoint.

**ResponsibleFor**

**Package:** Traceability

isAbstract: No

**Generalization:** MeasurableElement, Allocate

**Extension:** Abstraction

**Description**

An abstraction relationship between an ActualResponsibleResource and an ActualResponsibility or ActualProject. It defines the duties that the ActualResponsibleResource is ResponsibleFor.
Attributes

endDate : ISO8601DateTime [0..1]
End date of an ActualResponsibleResource being ResponsibleFor and ActualProject or ActualResponsibility.

startDate : ISO8601DateTime [0..1]
Start date of an ActualResponsibleResource being ResponsibleFor and ActualProject or ActualResponsibility.

Associations

responsibleRoleKind : ResponsibleRoleKind[1]
Captures the kind of role (Manager or ResponsibleOwner) responsible for the ActualProject or ActualResponsibility.

Constraints

[1] ResponsibleFor.client
Value for the client metaproperty must be stereotyped by the specialization of «ActualResponsibleResource».

[2] ResponsibleFor.supplier
Value for the supplier metaproperty must be stereotyped «ActualProject», «ActualResponsibility», «ActualProjectMilestone» or their specializations.

**ResponsibleRoleKind**

**Package**: Traceability

**isAbstract**: No

**Description**

Enumeration of the possible kinds of ResponsibleFor relationship. Its enumeration literals are:

- Manager - Indicates that the ResourceInteraction associated with the ResourceInteractionKind is an implementation of logical flow.
- ResponsibleOwner - Indicates that the ResourceInteraction associated with the ResourceInteractionKind is an implementation of logical flow.
3.1.8 UAF::Resources

Stakeholders: Systems Engineers, Resource Owners, Implementers, Solution Providers, IT Architects. 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.

UAF::Resources::Taxonomy

Contains the elements that contribute to the Resources Taxonomy Viewpoint.

CapabilityConfiguration

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture
Extension: Class

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

![Figure 3:110 – CapabilityConfiguration](image)

Associations

- doctrine : StandardOperationalActivity[*] Represents the doctrinal line of development of the Capability.

NaturalResource

Package: Taxonomy
isAbstract: No
Generalization: PhysicalResource
Extension: Class

Description
Type of physical resource that occurs in nature such as oil, water, gas or coal.
Figure 3:111 - NaturalResource

**PhysicalResource**

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

**Description**

An abstract grouping that defines physical resources (i.e. OrganizationalResource, ResourceArtifact and NaturalResource).

Figure 3:112 - PhysicalResource

**ResourceArchitecture**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ResourcePerformer, Architecture  
**Extension:** Class

**Description**

An element used to denote a model of the Architecture, described from the ResourcePerformer perspective.
Figure 3:113 - ResourceArchitecture

**ResourceArtifact**

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

**Description**

A type of man-made object that contains no human beings (i.e., satellite, radio, petrol, gasoline, etc.).

Figure 3:114 - ResourceArtifact

**ResourcePerformer**

**Package:** Taxonomy  
**isAbstract:** Yes  
**Generalization:** ResourceAsset, ResourceExchangeItem, SubjectOfResourceConstraint, VersionedElement, CapableElement, SubjectOfForecast, OperationalExchangeItem, Desirer  
**Extension:** Class

Unified Architecture Framework Profile (UAFP) Version 1.1
Description

An abstract grouping of elements that can perform Functions.

Figure 3:115 – ResourcePerformer

Attributes

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

Associations

- milestone : ProjectMilestone[*] Relates ResourcePerformer to ProjectMilestones that affect it.

Constraints

1. ResourcePerformer.isCapableOfPerforming Is capable of performing only «Function» elements or its specializations.
2. ResourcePerformer.ownedOperation Values for the ownedOperation metaproperty must be stereotyped «ResourceMethod» or its specializations.
3. ResourcePerformer.ownedPort Values for the ownedPort metaproperty must be stereotyped «ResourcePort» or its specializations.

Software

Package: Taxonomy

isAbstract: No

Generalization: ResourceArtifact

Extension: Class

Description

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

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture
Extension: Class

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).

UAF::Resources::Structure
Contains the elements that contribute to the Resources Structure Viewpoint.

ResourceMethod

Package: Structure
isAbstract: No
Generalization: MeasurableElement
Extension: Operation

Description
A behavioral feature of a ResourcePerformer whose behavior is specified in a Function.
Figure 3:118 - ResourceMethod

Constraints

1. ResourceMethod.method
   Value for the method metaproperty must be stereotyped «Function» or its specializations.

2. ResourceMethod.ownedParameter
   The values for the ownedParameter metaproperty must be stereotyped «ResourceParameter».

ResourceParameter

Package: Structure
isAbstract: No
Generalization: MeasurableElement
Extension: Parameter

Description

An element that represents inputs and outputs of a Function. It is typed by a ResourceInteractionItem.
Figure 3:119 - ResourceParameter

Constraints

[1] ResourceParameter.type Value for the type metaproperty must be stereotyped with a specialization of «ResourceInteractionItem».

ResourcePort

Package: Structure
isAbstract: No
Generalization: ProxyPort, MeasurableElement, ProtocolImplementation
Extension: Port

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

[1] ResourcePort.type  Value for the type metaproperty must be stereotyped «ResourcePort» or its specializations.
[2] ResourcePort.class  Value for the class metaproperty must be stereotyped by the specialization of «ResourcePerformer».

ResourceRole

Package: Structure
isAbstract: No
Generalization: LocationHolder, SubjectOfResourceConstraint, MeasurableElement, AssetRole
Extension: Property

Description

Figure 3:121 – ResourceRole

Associations
roleKind : RoleKind[1]  Captures the kind of role a Resource can play.

Constraints
[1] ResourceRole.type  Value for the type metaproperty must be stereotyped by the specialization of «ResourcePerformer».
[2] ResourceRole.class  Value for the class metaproperty must be stereotyped by the specialization of «ResourcePerformer».
**RoleKind**

**Package:** Structure

**isAbstract:** No

**Description**

Enumeration of the possible kinds of roles that a ResourceRole may play in the context of a ResourcePerformer. Its enumeration literals are:

- **Part** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of a ResourcePerformer that is used as a part of another ResourcePerformer.
- **Component** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of Software that is used in the context of a ResourcePerformer.
- **UsedConfiguration** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of existing CapabilityConfiguration that is used in the context of a ResourcePerformer.
- **HumanResource** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of human resource that is used in the context of a ResourcePerformer.
- **Platform** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of a ResourcePerformer that represents a platform (e.g. vessel, aircraft, etc.) that is used in the context of a SystemsResource.
- **System** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of assembly of ResourcePerformers that is used in the context of another ResourcePerformer.
- **SubOrganization** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of Organization that is typically the parent of another - e.g. a squadron may be part of a batallion, that is used in the context of a ResourcePerformer.
- **PostRole** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of Post that is used in the context of a ResourcePerformer.
- **ResponsibilityRole** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of Responsibility associated with a role that is used in the context of a ResourcePerformer.
- **Equipment** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of man-made resource that is used to accomplish a task or function in the context of a ResourcePerformer.
- **SubSystemPart** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of subsystem (represented as a ResourcePerformers) is is part of another ResourcePerformer.
- **UsedPhysicalArchitecture** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of existing PhysicalArchitecture that is used in the context of a ResourcePerformer.
- **HostedSoftware** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of software that is used in the context of a ResourcePerformer.
- **ArtifactComponent** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of non human resource that is used as a component in the context of a ResourcePerformer.
- **NaturalResourceComponent** - Indicates that the ResourceRole associated with the ResourceRoleKind is a kind of natural resource that is used as a component in the context of a ResourcePerformer.
- **Other** - Indicates that the ResourceRole associated with the ResourceRoleKind is another kind of RoleKind that is not on the enumerated list.

**UAF::Resources::Connectivity**

Contains the elements that contribute to the Resources Connectivity Viewpoint.

**ResourceConnector**

**Package:** Connectivity

**isAbstract:** No

**Generalization:** MeasurableElement, ProtocolImplementation

**Extension:** Connector
Description

A channel for exchange between two ResourceRoles.

Associations

boundaryCondition : Environment[*]  Relates a ResourceConnector to the extremes of the Environment in which it is required to be made available.

Constraints

[1] ResourceConnector.end  The value for the role metaproperty for the owned ConnectorEnd must be stereotype «ResourcePort», «ResourceRole» or their specializations.

**ResourceExchange**

**Package:** Connectivity

**isAbstract:** No

**Generalization:** [Exchange]

**Extension:** InformationFlow
Description
Asserts that a flow can exist between ResourcePerformers (i.e., flows of data, people, materiel, or energy).

Figure 3:123 – ResourceExchange

Associations
exchangeKind : ResourceExchangeKind[] Captures the kind of ResourceExchange.

Constraints
= ResourceCommunication, the conveyed element must be stereotyped «DataElement» or its specializations,
= ResourceMovement, the conveyed element must be stereotyped by the specialization of «PhysicalResource»,
= ResourceEnergyFlow, the conveyed element must be stereotyped «NaturalResource» or its specializations,
= GeoPoliticalExtentExchange, the conveyed element must be stereotyped «GeoPoliticalExtentType» or its specializations.

[2] ResourceInteraction.informationSource Value for the informationSource metaproperty must be stereotyped by the specialization of «ResourcePerformer».

[3] ResourceInteraction.informationTarget Value for the informationTarget metaproperty must be stereotyped by the specialization of «ResourcePerformer».

[4] ResourceInteraction.realizingActivityEdge Value for the realizingActivityEdge metaproperty must be stereotyped by the specialization of «FunctionEdge».


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

Figure 3:124 – ResourceExchangeItem

Associations

ResourceExchangeKind
Package: Connectivity
isAbstract: No
Description
Enumeration of the possible kinds of resource exchange applicable to a ResourceExchange. Its enumeration literals are:

- ResourceCommunication - Indicates that the ResourceInteraction associated with the ResourceInteractionKind is an implementation of logical flow of data between Resources.
- ResourceMovement - Indicates that the ResourceInteraction associated with the ResourceInteractionKind is an implementation of logical flow of Resources between Resources.
- ResourceEnergyFlow - Indicates that the ResourceInteraction associated with the ResourceInteractionKind is an implementation of logical flow of natural resources between Resources.
- GeoPoliticalExtentExchange - Indicates that the ResourceInteraction associated with the ResourceInteractionKind is an implementation of logical flow where GeoPoliticalExtents (i.e., Borders) flow from one place to another.

ResourceInterface
Package: Connectivity
isAbstract: No
Generalization: PropertySet, InterfaceBlock
Extension: Class
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.
Constraints

[1] ResourceInterface.ownedOperation Values for ownedOperation metaproperty must be stereotyped «ResourceMethod» or its specializations.

**ResourceSignal**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** ResourceExchangeItem  
**Extension:** Signal

**Description**

A ResourceSignal is a specification of a kind of communication between resources (ResourcePerformers) in which a reaction is asynchronously triggered in the receiver without a reply.

**ResourceSignalProperty**

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

**Description**

A property of a ResourceSignal typed by ResourceExchangeItem. It enables ResourceExchangeItem e.g., DataElement to be passed as arguments of the ResourceSignal.
Figure 3:127 – ResourceSignalProperty

Constraints

[1] ResourceSignalProperty.class Value for class metaproperty must be stereotyped «ResourceSignal» or its specializations.


UAF::Resources::Processes

Contains the elements that contribute to the Resources Processes Viewpoint.

Function

Package: Processes
isAbstract: No

Generalization: Activity, SubjectOfResourceConstraint
Extension: Activity

Description

An Activity which is specified in the context to the ResourcePerformer (human or machine) that IsCapableToPerform it.
Figure 3:128 – Function

Associations


Constraints

[1] Function.ownedParameter  The values for the ownedParameter metaproperty must be stereotyped «ResourceParameter» or its specializations.

FunctionAction

Package: Processes

isAbstract: No

Generalization: MeasurableElement

Extension: CallBehaviorAction

Description

A call of a Function indicating that the Function is performed by a ResourceRole in a specific context.
Constraints

[1] FunctionAction.activity Value for the activity metaproperty must be stereotyped «Function» or its specializations.

[2] FunctionAction.behavior Value for the behavior metaproperty must be stereotyped «Function» or its specializations.

**FunctionControlFlow**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** FunctionEdge  
**Extension:** ControlFlow

**Description**

An ActivityEdge that shows the flow of control between FunctionActions.
Constraints

[1] FunctionControlFlow.source  Value for the source metaproperty must be stereotyped «FunctionAction» or its specializations.


**FunctionEdge**

**Package**: Processes  
isAbstract: Yes  
**Generalization**: MeasurableElement  
**Extension**: ActivityEdge

Description

Abstract grouping for FunctionControlFlow and FunctionObjectFlow.

![Diagram](image)

**Figure 3:131 – FunctionEdge**

Constraints

[1] FunctionEdge.owner  «FunctionEdge» must be owned directly or indirectly by «Function» or its specializations.

**FunctionObjectFlow**

**Package**: Processes  
isAbstract: No  
**Generalization**: FunctionEdge  
**Extension**: ObjectFlow

Description

An ActivityEdge that shows the flow of Resources (objects/data) between FunctionActions.
UAF::Resources::States
Contains the elements that contribute to the Resources States Viewpoint.

ResourceStateDescription
Package: States
isAbstract: No
Generalization: MeasurableElement
Extension: StateMachine

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

Constraints
[1] ResourceStateDescription.owner Values for the owner metaproperty must be stereotyped with the specialization of «ResourcePerformer».

UAF::Resources::Interaction Scenarios
Contains the elements that contribute to the Resources Interaction Scenarios Viewpoint.

ResourceMessage
Package: Interaction Scenarios
isAbstract: No
Generalization: MeasurableElement
Extension: Message

Description
Message for use in a Resource Event-Trace which carries any of the subtypes of ResourceExchange.
Figure 3:134 – ResourceMessage

Constraints

**UAF::Resources::Information**

Contains the elements that contribute to the Resources Information Viewpoint.

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

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

Figure 3:135 – DataElement
Constraints

[1] DataElement.owner  Values for the owner metaproperty must be stereotyped «DataModel» or its specializations.

**UAF::Resources::Constraints**

Contains the elements that contribute to the Resources Constraints Viewpoint.

**ResourceConstraint**

**Package**: Constraints  
**isAbstract**: No  
**Generalization**: Rule  
**Extension**: Constraint

Description

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

Figure 3:136 – ResourceConstraint

Constraints

[1] ResourceConstraint.constrainedElement  Value for the constrainedElement metaproperty must be stereotyped by the specialization of «SubjectOfResourceConstraint».

**SubjectOfResourceConstraint**

**Package**: Constraints  
**isAbstract**: Yes  
**Generalization**: UAFElement  
**Extension**: Element

Description

An abstract grouping of elements that can be the subject of a ResourceConstraint.
UAF::Resources::Roadmap

Contains the elements that contribute to the Resources Roadmap Viewpoint.

**Forecast**

**Package:** Roadmap

**isAbstract:** No

**Generalization:** MeasurableElement

**Extension:** Dependency

**Description**

A dependency relationship 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 3:138 – Forecast**

**Associations**

forecastPeriod : ActualEnterprisePhase[0..1]  
Relates the SubjectOfForecast to the ActualEnterprisePhase in which the SubjectOfForecast is expected to be provided.

**Constraints**

[1] Forecast.client  
Value for the client metaproperty must be stereotyped by the specialization of «SubjectOfForecast». 
Values for the client and supplier metaproperties must be stereotyped by the same specialization of «SubjectOfForecast» (e.g. «Software» to «Software», «Standard» to «Standard», etc).

Value for the supplier property must be stereotyped by the specialization of «SubjectOfForecast».

**SubjectOfForecast**

**Package:** Roadmap  
**isAbstract:** Yes  
**Generalization:** UAFElement  
**Extension:** Class  

**Description**  
An abstract grouping of elements that can be the subject of a Forecast.

**Technology**

**Package:** Roadmap  
**isAbstract:** No  
**Generalization:** ResourceArtifact  
**Extension:** Class  

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

**Package:** Roadmap  
**isAbstract:** Yes  
**Generalization:** UAFEElement  
**Extension:** Class

**Description**

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

**Associations**

- `versionReleasedAtMilestone : ActualProjectMilestone[*]`  
  Relates a VersionedElement to the ActualProjectMilestone. It indicates the ActualProjectMilestone at which the VersionedElement is released.

- `versionWithdrawnAtMilestone : ActualProjectMilestone[*]`  
  Relates a VersionedElement to the ActualProjectMilestone. It indicates the ActualProjectMilestone at which the VersionedElement is withdrawn.

**VersionOfConfiguration**

**Package:** Roadmap  
**isAbstract:** No  
**Generalization:** MeasurableElement  
**Extension:** Property
Description

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

Figure 3:142 – VersionOfConfiguration

Constraints

[1] VersionOfConfiguration.class  Value for the class metaproperty must be stereotyped «WholeLifeConfiguration» or its specializations.

[2] VersionOfConfiguration.type  Value for the type metaproperty must be stereotyped by the specialization of «VersionedElement».

VersionSuccession

Package: Roadmap
isAbstract: No
Generalization: MeasurableElement
Extension: Dependency

Description

A dependency relationship between two VersionOfConfigurations that denotes that one VersionOfConfiguration follows from another.

Figure 3:143 – VersionSuccession
Constraints


**WholeLifeConfiguration**

**Package:** Roadmap  
**isAbstract:** No  
**Generalization:** PropertySet, Block  
**Extension:** Class

**Description**  
A set of VersionedElements.

![Class Diagram](image)

**Figure 3:144 – WholeLifeConfiguration**

**Associations**  

**WholeLifeConfigurationKind**

**Package:** Roadmap  
**isAbstract:** No  
**Description**  
Enumeration of the possible kinds of WholeLifeConfiguration. Its enumeration literals are:

- Service - Indicates that the WholeLifeConfiguration associated with the WholeLifeConfigurationKind is the master specification from which Services are versioned.
- ResourcePerformer - Indicates that the WholeLifeConfiguration associated with the WholeLifeConfigurationKind is the master specification from which ResourcePerformers are versioned.
- OrganizationalResource - Indicates that the WholeLifeConfiguration associated with the WholeLifeConfigurationKind is the master specification from which OrganizationalResources are versioned.

**UAF::Resources::Traceability**

Contains the elements that contribute to the Resources Traceability Viewpoint.
**ProtocolImplementation**

**Package:** Traceability  
**isAbstract:** Yes  
**Generalization:** UAFEElement  
**Extension:** Element

**Description**  
An abstract grouping of architectural elements that can implement Protocols.

![Diagram of ProtocolImplementation]

**Figure 3:145 – ProtocolImplementation**

**Associations**

`implements : Protocol[*]`  
Relates the ResourceConnector and ResourcePort to the Protocols that they can implement.

### 3.1.9 UAF::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.

**UAF::Security::Taxonomy**

Contains the elements that contribute to the Security Taxonomy Viewpoint.

**Asset**

**Package:** Taxonomy  
**isAbstract:** Yes  
**Generalization:** ConceptItem, PropertySet, LocationHolder, SubjectOfSecurityConstraint, Block  
**Extension:** Class

**Description**  
Asset as applied to Security views, an abstract element that indicates the types of elements that can be considered as a subject for security analysis.
Figure 3:146 – Asset

Associations

categoryCategorizesAsset : MeasurementSet[0..1] Enables association of an Asset to the set of security related measurements (MeasurementSet).

**OperationalAsset**

**Package:** Taxonomy

**isAbstract:** Yes

**Generalization:** Asset

**Extension:** Class

**Description**

An abstract element used to group the elements of OperationalAgent and InformationElement allowing them to own InformationRoles.
OperationalAsset

Package: Taxonomy
isAbstract: No
Generalization: OperationalArchitecture
Extension: Class

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

ResourceAsset

Package: Taxonomy
isAbstract: Yes
Generalization: Asset
Extension: Class

Description
An abstract element used to group the elements of ResourcePerformer and DataElement allowing them to own DataRoles.
Figure 3:149 - ResourceAsset

**ResourceMitigation**

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

**Description**
A set of ResourcePerformers intended to address against specific risks.

Figure 3:150 - ResourceMitigation

**SecurityEnclave**

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

**Description**
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.
**Figure 3:151 - SecurityEnclave**

**UAF::Security::Structure**
Contains the elements that contribute to the Security Structure Viewpoint.

**AssetRole**

**Package:** Structure  
**isAbstract:** Yes  
**Generalization:** MeasurableElement, SubjectOfSecurityConstraint  
**Extension:** Property

**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.

**Figure 3:152 - AssetRole**

**DataRole**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** AssetRole  
**Extension:** Property
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 3:153 - DataRole

InformationRole

Package: Structure
isAbstract: No
Generalization: AssetRole
Extension: Property

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.

Figure 3:154 – InformationRole

Constraints

[1] SecurityProperty.class Value for the class metaproperty must be stereotyped by the specialization of «Asset».
[2] SecurityProperty.type In case of value for the class metaproperty is stereotyped:
   a. by any of specializations of «OperationalAgent», values for the type metaproperty must be stereotyped «InformationElement» or its specializations,
   b. by any of specializations of «ResourcePerformer», values for the type metaproperty must be stereotyped «DataElement» or its specializations,
   c. «InformationElement», values for the type metaproperty must be stereotyped «InformationElement» or its specializations,
d. «DataElement», values for the type metaproperty must be stereotyped «DataElement» or its specializations.

**UAF::Security::Processes**

Contains the elements that contribute to the Security Processes Viewpoint.

**EnhancedSecurityControl**

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

**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.

**Figure 3:155 - EnhancedSecurityControl**

**Enhances**

**Package**: Processes  
**isAbstract**: No  
**Generalization**: DeriveReqt, MeasurableElement  
**Extension**: Abstraction, Connector

**Description**

A dependency relationship relating the EnhancedSecurityControl to a SecurityControl.

**Figure 3:156 – Enhances**
Constraints

[1] Enhances.client Value for the client metaproperty must be stereotyped «EnhancedSecurityControl» or its specializations.

[2] Enhances.supplier Value for the supplier metaproperty must be stereotyped «SecurityControl» or its specializations.

**Protects**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** [MeasurableElement](#)  
**Extension:** Dependency

**Description**

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

![Diagram of Protects](image)

**Figure 3:157 – Protects**

Constraints

[1] Protects.client Value for the client metaproperty must be stereotyped «SecurityControl» or its specializations.  
[2] Protects.supplier Value for the supplier metaproperty must be stereotyped by the specialization of «Asset».

**ProtectsInContext**

**Package:** Processes  
**isAbstract:** No  
**Generalization:** [MeasurableElement](#)  
**Extension:** Dependency

**Description**

A dependency relationship 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.
Figure 3:158 – ProtectsInContext

Constraints
[1] ProtectsInContext.client  Value for the client metaproperty must be stereotyped «SecurityControlAction» or its specializations.

SecurityProcess

Package: Processes
isAbstract: No
Generalization: OperationalActivity, Function, SubjectOfSecurityConstraint
Extension: Activity

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

**Package:** Processes

**isAbstract:** No

**Generalization:** OperationalActivityAction, FunctionAction

**Extension:** CallBehaviorAction

**Description**


![Diagram](image)

**Figure 3:160 – SecurityProcessAction**

**Constraints**

[1] SecurityControlAction.behavior  Value for behavior metaproperty must be stereotyped «SecurityControl» or its specializations.

**UAF::Security::Constraints**

Contains the elements that contribute to the Security Constraints Viewpoint.

**ActualRisk**

**Package:** Constraints

**isAbstract:** No

**Generalization:** ActualPropertySet

**Extension:** InstanceSpecification

**Description**

Figure 3:161 – ActualRisk

Associations

actualRiskOwner : ActualResponsibleResource[0..1]  Enables association of an ActualRisk to an actual organizational role that is responsible for executing the actual mitigation.

affectedActualResource : ActualResource[0..1]  Asserts that an ActualRisk is applicable to an ActualResource.

Risk

Package: Constraints

isAbstract: No

Generalization: PropertySet, Block

Extension: Class

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.
SecurityConstraint

Package: Constraints
isAbstract: No
Generalization: Rule
Extension: Constraint

Description
A type of rule that captures a formal statement to define security laws, regulations, guidelines, and policy.

Constraints

**SecurityControl**

**Package:** Constraints  
**isAbstract:** No  
**Generalization:** Requirement, MeasurableElement  
**Extension:** Class  

**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].

**Figure 3:164 – SecurityControl**

**Associations**

mitigatingActualResource : ActualResource[*]  
Relates an actual mitigation (an ActualResource for mitigating a Risk) to an ActualRisk.

**SecurityControlFamily**

**Package:** Constraints  
**isAbstract:** No  
**Generalization:** SecurityControl  
**Extension:** Class  

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.

SubjectOfSecurityConstraint
Package: Constraints
isAbstract: Yes
Generalization: UAFElement
Extension: Element

Description
An abstract grouping of elements that can be the subject of a SecurityConstraint.

UAF::Security::Traceability
Contains the elements that contribute to the Security Traceability Viewpoint.

Affects
Package: Traceability
isAbstract: No
Generalization: MeasurableElement
Extension: Dependency
Description
A dependency that asserts that a Risk is applicable to an Asset.

Figure 3:167 – Affects

Constraints
[1] Affects.client Value for the client metaproperty must be stereotyped «Risk» or its specializations.
[2] Affects.supplier Value for the supplier metaproperty must be stereotyped «Asset» or its specializations.

AffectsInContext
Package: Traceability
isAbstract: No
Generalization: MeasurableElement
Extension: Dependency

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

Figure 3:168 – AffectsInContext

Constraints
[1] AffectsInContext.client Value for the client metaproperty must be stereotyped «Risk» or its specializations.
[2] AffectsInContext.supplier Value for the supplier metaproperty must be stereotyped «AssetRole» or its specializations.

Mitigates
Package: Traceability
A dependency 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).

Figure 3:169 – Mitigates

Constraints
[1] Mitigates.client Value for the client metaproperty must be stereotyped «SecurityControl» or its specializations.
[2] Mitigates.supplier Value for the supplier metaproperty must be stereotyped «Risk» or its specializations.

OwnsRisk

Description
An abstraction relating a Risk to an organizational resource that is responsible for executing the risk mitigation.

Figure 3:170 – OwnsRisk

Constraints
[1] OwnsRisk.client Value for the client metaproperty must be stereotyped «OrganizationalResource» or its specializations.
[2] OwnsRisk.supplier Value for the supplier metaproperty must be stereotyped «Risk» or its specializations.

**OwnsRiskInContext**

**Package:** Traceability  
**isAbstract:** No  
**Generalization:** MeasurableElement, Allocate  
**Extension:** Abstraction

**Description**
An abstraction relating a Risk to an organizational role that is responsible for executing the risk mitigation in the specific context or configuration.

**Figure 3:171 – OwnsRiskInContext**

**Constraints**
[1] OwnsRiskInContext.client Value for the client metaproperty must be stereotyped «ResourceRole» or its specializations.
[2] OwnsRiskInContext.supplier Value for the supplier metaproperty must be stereotyped «Risk» or its specializations.

### 3.1.10 UAF::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.

**UAF::Projects::Taxonomy**
Contains the elements that contribute to the Project Taxonomy Viewpoint.

**ActualMilestoneKind**

**Package:** Taxonomy  
**isAbstract:** No

**Description**
Enumeration of the possible kinds of ActualProjectMilestone. Its enumeration literals are:
- InService - Indicates that the ActualProjectMilestone associated with the ActualMilestoneKind is when the configuration goes into service.
• Deployed - Indicates that the ActualProjectMilestone associated with the ActualMilestoneKind is a configuration deployment milestone.
• NoLongerUsed - Indicates that the ActualProjectMilestone associated with the ActualMilestoneKind is when the deployed configuration is no longer used.
• OutOfService - Indicates that the ActualProjectMilestone associated with the ActualMilestoneKind is when the in service configuration goes out of service.
• Other - Indicates that the ActualProjectMilestone associated with the ActualMilestoneKind is not one of the standard ActualMilestoneKinds.

**Project**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** OrganizationalResource, Block  
**Extension:** Class

**Description**

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

![Diagram of Project](image)

**Figure 3:172 - Project**

**ProjectKind**

**Package:** Taxonomy  
**isAbstract:** No

**Description**

Enumeration of the possible kinds of project applicable to an ActualProject. Its enumeration literals are:

- Programme - Indicates that the ActualProject associated with the ProjectKind is an undertaking that is a temporary, flexible organization created to co-ordinate, direct and oversee the implementation of a set of related Projects and Tasks in order to deliver outcomes and benefits related to the organization’s strategic
objectives. A programme is likely to have a lifespan of several years. During a programme lifecycle, projects are initiated, executed, and closed. Programmes provide an umbrella under which these projects can be co-ordinated. The programme integrates the projects so that it can deliver an outcome greater than the sum of its parts.

- **Portfolio** - Indicates that the ActualProject associated with the ProjectKind is an undertaking comprised of the Projects and Programmes that are the totality of an organisation's investment (or segment thereof) in the changes required to achieve its strategic objectives.
- **Project** - Indicates that the ActualProject associated with the ProjectKind is an undertaking that is a time-limited endeavor to create a specific set of products or services.
- **PersonnelDevelopment** - Indicates that the ActualProject associated with the ProjectKind is an undertaking that relates to the training and enablement of personnel to enable them help achieve the organisations objectives.

**ProjectMilestone**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** PropertySet, Block  
**Extension:** Class  

**Description**  
A type of event in a Project by which progress is measured.

![Diagram of ProjectMilestone](image)

**Associations**

- `resource : ResourcePerformer[*]` Relates a ProjectMilestone to the Resources that can be affected by the milestone. It is used to describe aspects of the lifecycle of a Resource.

**Constraints**

- `[1] ProjectMilestone.ownedAttribute` All of the «ProjectThemes», owned by a «ProjectMilestone», must be typed by the same «StatusIndicators» or its specializations.

**UAF::Projects::Structure**

Contains the elements that contribute to the Project Structure Viewpoint.

**ProjectMilestoneRole**

**Package:** Structure  
**isAbstract:** No  
**Generalization:** MeasurableElement  
**Extension:** Property
Description
The role played by a ProjectMilestone in the context of a Project.

Figure 3:174 – ProjectMilestoneRole

Constraints
[1] ProjectMilestoneRole.class  Value for the class metaproperty must be stereotyped «Project» or its specializations.
[2] ProjectMilestoneRole.type   Value for the type metaproperty must be stereotyped «ProjectMilestone» or its specializations.

ProjectRole
Package: Structure
isAbstract: No
Generalization: ResourceRole
Extension: Property

Description
Usage of a Project in the context of another Project. Creates a whole-part relationship.

Figure 3:175 – ProjectRole
Constraints

[1] ProjectRole.class Value for the class metaproperty must be stereotyped «Project» or its specializations.
[2] ProjectRole.type Value for the type metaproperty must be stereotyped «Project» or its specializations.

**ProjectStatus**

**Package:** Structure

isAbstract: No

**Generalization:** [UAFElement]

**Extension:** Slot

**Description**

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

![Diagram](image)

**Figure 3:176 – ProjectStatus**

**Constraints**

[1] ProjectStatus.definingFeature Value for the DefiningFeature metaproperty must be stereotyped «ProjectTheme» or its specializations.

**ProjectTheme**

**Package:** Structure

isAbstract: No

**Generalization:** [MeasurableElement]

**Extension:** Property

**Description**

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

[1] ProjectTheme.class Value for the class metaproperty must be stereotyped «ProjectMilestone» or its specializations.

[2] ProjectTheme.type Value for the type metaproperty must be stereotyped «StatusIndicators» or its specializations.

**StatusIndicators**

**Package:** Structure

isAbstract: No

**Generalization:** MeasurableElement, ValueType

**Extension:** Enumeration

**Description**

An enumerated type that specifies a status for a ProjectTheme.

**Figure 3:178 - StatusIndicators**

**UAF::Projects::Connectivity**

Contains the elements that contribute to the Project Connectivity Viewpoint.

**MilestoneDependency**

**Package:** Connectivity

isAbstract: No

**Generalization:** MeasurableElement
**Extension**: Dependency

**Description**

A dependency relationship between two ActualProjectMilestones that denotes one ActualProjectMilestone follows from another.

**Figure 3:179 – MilestoneDependency**

**Constraints**

1. MilestoneDependency.client
   - Value for the client metaproperty must be stereotyped «ActualProjectMilestone» or its specializations.

2. MilestoneSequence.supplier
   - Value for the supplier metaproperty must be stereotyped «ActualProjectMilestone» or its specializations.

**ProjectSequence**

**Package**: Connectivity

**isAbstract**: No

**Generalization**: MeasurableElement

**Extension**: Dependency

**Description**

A dependency relationship between two ActualProjects that denotes one ActualProject cannot start before the previous ActualProject is finished.
Constraints

[1] ProjectSequence.client Value for the client metaproperty must be stereotyped «ActualProject» or its specializations.

[2] ProjectSequence.supplier Value for the supplier metaproperty must be stereotyped «ActualProject» or its specializations.

**UAF::Projects::Processes**

Contains the elements that contribute to the Project Processes Viewpoint.

**ProjectActivity**

**Package:** Processes

isAbstract: No

**Generalization:** Function

**Extension:** Activity

**Description**

An activity carried out during a project.

**Figure 3:181 - ProjectActivity**

**ProjectActivityAction**

**Package:** Processes

isAbstract: No

**Generalization:** FunctionAction

**Extension:** CallBehaviorAction
Description

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

Figure 3:182 – ProjectActivityAction

Constraints

[1] FunctionAction.behavior Value for the behavior metaproperty must be stereotyped «ProjectActivity» or its specializations.
[2] ProjectActivityAction.activity Value for the activity metaproperty must be stereotyped «ProjectActivity» or its specializations.

UAF::Projects::Roadmap

Contains the elements that contribute to the Project Roadmap Viewpoint.

ActualProject

Package: Roadmap
isAbstract: No
Generalization: ActualOrganizationalResource, Achiever
Extension: InstanceSpecification

Description

A time-limited endeavor to provide a specific set of ActualResources that meet specific Capability needs.
Figure 3:183 – ActualProject

Associations

- **ownedMilestone**: ActualProjectMilestone\[*\]  
  Relates the ActualProjectMilestones to the relevant ActualProject.
- **projectKind**: ProjectKind\[1\]  
  Enumerated value describing the kind of ActualProject.

Constraints

1. **ActualProject.classifier**  
   Value for the classifier metaproperty must be stereotyped «Project» or its specializations.
2. **ActualProject.slot**  
   Value for the slot metaproperty must be stereotyped «ActualProjectRole», «ActualProjectMilestoneRole», or their specializations.

### ActualProjectMilestone

**Package**: Roadmap
**isAbstract**: No
**Generalization**: ActualPropertySet
**Extension**: InstanceSpecification

**Description**

An event with a start date in an ActualProject from which progress is measured.
Figure 3:184 – ActualProjectMilestone

Attributes

- **endDate**: ISO8601DateTime[0]  
  End time for this ActualProjectMilestone.

Associations

- **actualResource**: ActualResource[*]  
  Relates an ActualProjectMilestone to the ActualResources that are affected by the milestone. It is used to describe aspects of the lifecycle of an ActualResource.

- **kind**: ActualMilestoneKind[1]  
  Enumerated value describing the kind of ActualProjectMilestone.

- **versionReleased**: VersionedElement[*]  
- **versionWithdrawn**: VersionedElement[*]

Constraints

- [1] ActualProjectMilestone.classifier  
  Value for the classifier metaproperty must be stereotyped «ProjectMilestone» or its specializations.

**ActualProjectMilestoneRole**

**Package**: Roadmap

**isAbstract**: No

**Generalization**: ActualState

**Extension**: Slot

**Description**

An ActualProjectMilestone that is applied to a ProjectMilestoneRole.
**Constraints**

1. ActualProjectMilestoneRole.definingFeature
   - Value for the definingFeature metaproperty has to be stereotyped «ProjectMilestoneRole» or its specializations.

2. ActualProjectMilestoneRole.owningInstance
   - Value for the owningInstance metaproperty has to be stereotyped «ActualProject» or its specializations.

3. ActualProjectMilestoneRole.value.instance
   - Value for the value.instance metaproperty has to be stereotyped «ActualProjectMilestone» or its specializations.

**ActualProjectRole**

**Package:** Roadmap

**isAbstract:** No

**Generalization:** ActualState

**Extension:** Slot

**Description**

An ActualProject that is applied to a ProjectRole.

---

**Figure 3:186 – ActualProjectRole**

**Description**

An ActualProject that is applied to a ProjectRole.

---
3.1.11 UAF::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.

UAF::Standards::Taxonomy

Contains the elements that contribute to the Standards Taxonomy Viewpoint.

Protocol

Package: Taxonomy
isAbstract: No
Generalization: Standard
Extension: Class

Description

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

Figure 3:187 - Protocol
**ProtocolStack**

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

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

![Figure 3:188 - ProtocolStack](image)

**Standard**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** SubjectOfForecast, PropertySet, Block  
**Extension:** Class  

**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.

![Figure 3:189 – Standard](image)
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.

Associations

- ratifiedBy : ActualOrganization[*] Relates a Standard to the ActualOrganization that ratified the Standard.

**UAF::Standards::Structure**

Contains the elements that contribute to the Standards Structure Viewpoint.

**ProtocolLayer**

**Package:** Structure

**isAbstract:** No

**Generalization:** MeasurableElement

**Extension:** Property

**Description**


![ProtocolLayer Diagram](image)

**Figure 3:190 – ProtocolLayer**

**Constraints**

1. ProtocolLayer.class Value for the class metaproperty must be stereotyped «Protocol» or its specializations.
2. ProtocolLayer.type Value for the type metaproperty must be stereotyped «Protocol» or its specializations.

**3.1.12 UAF::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.

**UAF::Actual Resources::Taxonomy**

Contains the elements that contribute to the Actual Resources Taxonomy Viewpoint.

**ActualOrganization**

**Package:** Taxonomy

**isAbstract:** No
**Generalization:** ActualResponsibleResource

**Extension:** InstanceSpecification

**Description**

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

**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.

**Associations**

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

**Constraints**

1. ActualOrganization.classifier  Classifier metaproperty value must be stereotyped «Organization» or its specializations.
2. ActualOrganization.slot  Slot metaproperty value must be stereotyped «ActualOrganizationRole» or its specializations.

**ActualOrganizationalResource**

**Package:** Taxonomy

**isAbstract:** Yes

**Generalization:** Stakeholder, ActualResource

**Extension:** InstanceSpecification
Description

Abstract element for an ActualOrganization, ActualPerson or ActualPost.

**Figure 3:192 - ActualOrganizationalResource**

**ActualPerson**

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

Description

An individual human being.

**Figure 3:193 – ActualPerson**

Constraints

[1] ActualPerson.classifier  
Value for the classifier metaproperty has to be stereotyped «Person» or its specializations.
**ActualPost**

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

**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.

![Diagram](image)

**Figure 3:194 – ActualPost**

**Constraints**

[1] ActualPost.classifier  
Classifier metaproperty value must be stereotyped «Post» or its specializations.

**ActualResource**

**Package:** Taxonomy  
**isAbstract:** No  
**Generalization:** ActualPropertySet, SubjectOfResourceConstraint, Achiever, CapableElement  
**Extension:** InstanceSpecification

**Description**

A fully-realized ResourcePerformer.
Figure 3.195 – ActualResource

Associations

actualCondition : ActualCondition[0..*]  Relates the ActualResource to the ActualStates of an environment or location describing its situation

milestone : ActualProjectMilestone[*]  Relates an ActualResource to the ActualProjectMilestones. It is used to describe aspects of the lifecycle of an ActualResource.

Constraints


ActualResponsibility

Package: Taxonomy
isAbstract: No
Generalization: ActualOrganizationalResource
Extension: InstanceSpecification

Description

The duty required of a Person or Organization.
Figure 3:196 – ActualResponsibility

Constraints

ActualResponsibleResource

Package: Taxonomy
isAbstract: Yes

Generalization: ActualOrganizationalResource

Extension: InstanceSpecification

Description
An abstract grouping of responsible OrganizationalResources.

Figure 3:197 - ActualResponsibleResource
FieldedCapability

Package: Taxonomy
isAbstract: No

Generalization: ActualResource
Extension: InstanceSpecification

Description
An actual, fully-realized capability. A FieldedCapability is typed by a CapabilityConfiguration.

![Figure 3:198 – FieldedCapability](image)

Constraints
[1] FieldedCapability.classifier Value for the classifier metaproperty must be stereotyped «CapabilityConfiguration» or its specializations.

UAF::Actual Resources::Structure
Contains the elements that contribute to the Actual Resources Structure Viewpoint.

ActualOrganizationRole

Package: Structure
isAbstract: No

Generalization: ActualOrganizationRole
Extension: Slot

Description
An ActualOrganizationResource that is applied to a ResourceRole.
Figure 3:199 – ActualOrganizationRole

Constraints

[1] ActualOrganizationRole.owningInstance  Value for owningInstance metaproperty has to be stereotyped «ActualOrganization» or its specializations.

ActualResourceRole

Package: Structure
isAbstract: No
Generalization: UAFElement
Extension: Slot

Description
An instance of a ResourcePerformer.

Figure 3:200 – ActualResourceRole

Constraints

[1] ActualResourceRole.definingFeature  Value for definingFeature metaproperty has to be stereotyped «ResourceRole» or its specializations.
ActualResourceRole.owningInstance Value for owningInstance metaproperty has to be stereotyped «ActualResource» or its specializations.

**UAF::Actual Resources::Connectivity**

Contains the elements that contribute to the Actual Resources Connectivity Viewpoint.

**ActualResourceRelationship**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** UAFElement, ItemFlow  
**Extension:** InformationFlow

**Description**

An abstract element that details the ActualOrganizationalResources that are able to carry out an ActualResponsibility.

![ActualResourceRelationship Diagram](image)

**Figure 3:201 – ActualResourceRelationship**

**Constraints**

1. ActualResourceRelationship.informationSource Value for informationSource metaproperty must be stereotyped «ActualResource» or its specializations.
2. ActualResourceRelationship.informationTarget Value for informationTarget metaproperty must be stereotyped «ActualResource» or its specializations.
3. ActualResourceRelationship.realizes Value for realizes metaproperty must be stereotyped «ResourceExchange» or its specializations.

**FillsPost**

**Package:** Connectivity  
**isAbstract:** No  
**Generalization:** MeasurableElement, Allocate  
**Extension:** Abstraction

**Description**

A dependency relationship that asserts that an ActualPerson fills an ActualPost.
Figure 3:202 – FillsPost

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.

Constraints
1. FillsPost.client Value for the client metaproperty must be stereotyped by «ActualPerson» or its specializations.
2. FillsPost.supplier Value for the supplier metaproperty must be stereotyped by «ActualPost» or its specializations.

UAF::Actual Resources::Constraints
Contains the elements that contribute to the Actual Resources Constraints Viewpoint.

ActualService
Package: Constraints
isAbstract: No
Generalization: ActualMeasurementSet, CapableElement
Extension: InstanceSpecification

Description
An instance of a ServiceSpecification.

Figure 3:203 – ActualService
Constraints

[1] ActualService.classifier  Value for the classifier metaproperty must be stereotyped by «ServiceSpecification» or its specializations.

ProvidedServiceLevel

Package: Constraints
isAbstract: No
Generalization: ActualService
Extension: InstanceSpecification

Description

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

![ProvidedServiceLevel Diagram](image)

Figure 3:204 - ProvidedServiceLevel

ProvidesCompetence

Package: Constraints
isAbstract: No
Generalization: MeasurableElement
Extension: Dependency

Description

A dependency relationship that asserts that an ActualOrganizationalResource provides a specific set of Competencies.

![ProvidesCompetence Diagram](image)

Figure 3:205 – ProvidesCompetence

Constraints

[1] ProvidesCompetence.client  Value for the client metaproperty must be stereotyped by a specialization of «ActualOrganizationalResource». 

Unified Architecture Framework Profile (UAFP) Version 1.1  156
RequiredServiceLevel

Package: Constraints
isAbstract: No
Generalization: ActualService
Extension: InstanceSpecification

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

UAF::Actual Resources::Traceability

Contains the elements that contribute to the Actual Resources Traceability Viewpoint.

OwnsProcess

Package: Traceability
isAbstract: No
Generalization: MeasurableElement, Allocate
Extension: Abstraction

Description
A dependency relationship denoting that an ActualOrganizationResource owns an OperationalActivity.
Constraints

[1] OwnsProcess.client Value for the client metaproperty must be stereotyped «ActualOrganizationalResource» or its specializations.

[2] OwnsProcess.supplier Value for the supplier metaproperty must be stereotyped «OperationalActivity» or its specializations.

3.1.13 UAF::Summary and 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.

ArchitecturalDescription

Package: Summary and Overview
isAbstract: No
Generalization: MeasurableElement
Extension: Package

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] = true  
  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.

Associations
architectureFramework : String[1] Indicates the type of framework used.
view : View[*] Indicates which views are used in the ArchitecturalDescription.
viewpoint : Viewpoint[*]

Architecture
Package: Summary and Overview
isAbstract: Yes
Generalization: UAFEElement
Extension: Class

Description
An abstract type that represents a generic architecture. Subtypes are OperationalArchitecture and PhysicalArchitecture.

![Diagram of Architecture](image)

Figure 3:209 – Architecture

Associations
describedBy : ArchitecturalDescription[*] The description of an Architecture.

Concern
Package: Summary and Overview
isAbstract: No
Generalization: PropertySet, Block
Extension: Class

Description
Interest in an EnterprisePhase (EnterprisePhase is synonym for System in ISO 42010) relevant to one or more of its stakeholders.
Figure 3:210 – Concern

Associations
enterprisePhase : ActualEnterprisePhase[*]  Relates a Concern to the ActualEnterprisePhase that addresses that concern (ActualEnterprisePhase is synonym for System in ISO 42010).

Stakeholder

Package: Summary and Overview
isAbstract: Yes
Generalization: UAFElement
Extension: Element

Description
individual, team, organization, or classes thereof, having an interest in an EnterprisePhase [ISO/IEC/IEEE 42010:2011].

Figure 3:211 – Stakeholder

Associations
stakeholderConcern : Concern[*]  Relates a Stakeholder to a Concern.
**UAFEElement**

**Package:** Summary and Overview  
**isAbstract:** Yes  
**Extension:** Element

**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.

![Diagram of UAFEElement](image)

**Figure 3:212 – UAFEElement**

**Attributes**

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

**Associations**

- **conformsTo : Standard[*]**  
  Relates a UAFEElement to the Standard that the UAFEElement is conforming to.

**View**

**Package:** Summary and Overview  
**isAbstract:** No  
**Generalization:** PropertySet, View  
**Extension:** Class
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 and Overview

isAbstract: No

**Generalization:** PropertySet, Viewpoint

**Extension:** Class

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)].
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.

Associations

care_of : Concern[*]  Relates the Viewpoint to the Concerns that the Viewpoint addresses.
stakeholder : Stakeholder[*]  Relates the Viewpoint to the Stakeholders whose Concerns are being addressed by the Viewpoint.
This page intentionally left blank.
4 UAF View Specifications

This paragraph is intended as normative guidance for developers and users as to what UAF stereotypes and metaconstraints are applicable for each of the UAF view specifications.

4.1 View Specifications

This section describes the normative stereotypes and metaconstraints needed to define UAF view specifications.

4.1.1 View Specifications::Strategic

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

View Specifications::Strategic::Taxonomy

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

Figure 4:1 - Strategic Taxonomy

Elements
• Capability
• CapabilityRole

View Specifications::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 4:2 - Strategic Structure

Elements
- ActualEnterprisePhase
- ActualOrganization
- Capability
- CapableElement
- EnterprisePhase
- StructuralPart
- TemporalPart
- WholeLifeEnterprise

View Specifications::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 4.3 - Strategic Connectivity

Elements
- Capability
- CapabilityRole

View Specifications::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 4:4 - Strategic States

Elements
- Achiever
- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResource
- ActualResponsibleResource
- ActualState
- Capability
- CapabilityConfiguration
- Desirer
- FieldedCapability
- KnownResource
- NaturalResource
- OperationalAgent
- OperationalArchitecture
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Project
- ResourceArchitecture
- ResourceArtifact
• **ResourceMitigation**
• **ResourcePerformer**

**View Specifications::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 4:5 - Strategic Constraints
```

Elements
• **Capability**
• **Measurement**
• **PropertySet**

**View Specifications::Strategic::Roadmap**

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 4.6 - Strategic Roadmap: Deployment

Elements
- ActualEnterprisePhase
- ActualMilestoneKind
- ActualOrganization
- ActualPerson
- ActualPost
- ActualProject
- ActualProjectMilestone
- ActualResource
- ActualResponsibleResource
- Capability
- CapabilityConfiguration
- CapableElement
- EnterprisePhase
- ResourceArchitecture
- ResourcePerformer

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 4:7 - Strategic Roadmap: Phasing

Elements
- ActualMilestoneKind
- ActualProject
- ActualProjectMilestone
- ActualResource
- Capability
- CapabilityConfiguration
- CapableElement
- FieldedCapability
- ResourceArchitecture
- ResourcePerformer

View Specifications::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 4:8 - Strategic Traceability

Elements
- Activity
- ActualEnduringTask
- Capability
- OperationalActivity
- StandardOperationalActivity

4.1.2 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

Stakeholders: Business Architects, Systems Engineers, Enterprise Architects, Owners responsible for Operational Agents.
Concerns: OperationalAgent types.
Definition: shows the taxonomy types of OperationalAgents.
Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.
Figure 4:9 - Operational Taxonomy

Elements
- ArbitraryConnector
- Asset
- CapabilityConfiguration
- ConceptItem
- ConceptRole
- HighLevelOperationalConcept
- KnownResource
- Location
- NaturalResource
- OperationalAgent
- OperationalAsset
- OperationalPerformer
- Organization
- OrganizationalResource
- PhysicalResource
- Post
View Specifications::Operational::Structure

Stakeholders: Business Architects, Systems Engineers, Enterprise Architects, Owners responsible for Operational Agents.

Concerns: identifies the operational exchange requirements between Operational Performers.

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 4:10 - Operational Structure

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

**View Specifications::Operational::Connectivity**

Stakeholders: Systems Engineers, Architects, Solution Providers.

Concerns: capture the interfaces between OperationalPerformers.

Definition: summarizes logical exchanges between OperationalPerformers 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 4:11 - Operational Connectivity**

Elements:
- **CapabilityConfiguration**
- **Exchange**
- **GeoPoliticalExtentType**
- **InformationElement**
- **MeasurableElement**
• MeasurementSet
• NaturalResource
• OperationalActivity
• OperationalActivityAction
• OperationalActivityEdge
• OperationalAgent
• OperationalConnector
• OperationalControlFlow
• OperationalExchange
• OperationalExchangeItem
• OperationalExchangeKind
• OperationalInterface
• OperationalObjectFlow
• OperationalPerformer
• OperationalPort
• OperationalRole
• OperationalSignal
• Organization
• OrganizationalResource
• Person
• PhysicalResource
• Post
• ResourceArchitecture
• ResourceArtifact
• ResourcePerformer
• Software
• Technology

View Specifications::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, BPMN Process Diagram.
Figure 4:12 - Operational Processes

Elements
- Activity
- ActualCondition
- ActualEnduringTask
- ActualMeasurementSet
- ActualService
- EnduringTask
- MeasurableElement
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalControlFlow
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod
- OperationalObjectFlow
- OperationalParameter
- OperationalPerformer
- OperationalRole
• **RequiredServiceLevel**
• **ServiceSpecification**
• **StandardOperationalActivity**

**View Specifications::Operational::States**
Stakeholders: Systems Engineers, Software Engineers.
Concerns: capture state-based behavior of an operational OperationalPerformer.
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.

```
«stereotype»
OperationalStateDescription
    [StateMachine]
```

Figure 4:13 - Operational States

Elements
• **OperationalAgent**
• **OperationalStateDescription**

**View Specifications::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.
Figure 4:14 - Operational Interaction Scenarios

Elements

- **Activity**
- **OperationalActivity**
- **OperationalAgent**
- **OperationalExchange**
- **OperationalMessage**
- **OperationalMethod**
- **OperationalPerformer**
- **OperationalRole**

**View Specifications::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.
Figure 4:15 - Operational Constraints

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

View Specifications::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.
4.1.3 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**


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 4:17 - Services Taxonomy

Elements

- ActualMeasurement
- ActualMeasurementSet
- ActualPropertySet
- ActualService
- Measurement
- PropertySet
- ProvidedServiceLevel
- RequiredServiceLevel
- ServicePolicy
- ServiceSpecification

View Specifications::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 4:18 - Services Structure

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

View Specifications::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 methods, signal receptions, and/or flow properties, to ensure compatibility and reusability of services.
Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram, tabular format.
Figure 4:19 - Services Connectivity

Elements
- ActualMeasurementSet
- ActualService
- MeasurableElement
- ProvidedServiceLevel
- RequiredServiceLevel
- ServiceConnector
- ServiceInterface
- ServicePort
- ServiceSpecification
- ServiceSpecificationRole

View Specifications::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, BPMN Process Diagram, SysML Block Definition Diagram.
V. View Specifications::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 4:21 - Services States

Elements
- ServiceSpecification
- ServiceStateDescription

V. View Specifications::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.
View Specifications::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.
Elements
- **Rule**
- **ServicePolicy**
- **ServiceSpecification**

**View Specifications::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 4:24 - Services Roadmap

Elements
- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- ServiceSpecification
- VersionedElement
- VersionOfConfiguration
- VersionSuccession
- WholeLifeConfiguration
- WholeLifeConfigurationKind

View Specifications::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 4:25 - Services Traceability**

Elements
- **ActualService**
- **Capability**
- **OperationalActivity**
- **ServiceSpecification**

**4.1.4 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**

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.
Figure 4:26 - Personnel Taxonomy

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

View Specifications::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 4:27 - Personnel Structure

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

View Specifications::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 4:28 - Personnel Connectivity

Elements

- Command
- Control
- DataElement
- Environment
- Exchange
- Function
- MeasurableElement
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Rationale
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceExchangeKind
View Specifications::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.

Figure 4:29 - Personnel Processes

Elements
- Activity
- ActualCondition
- DataElement
- Function
- FunctionAction
- FunctionControlFlow
- FunctionEdge
- FunctionObjectFlow
- Organization
View Specifications::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.

Figure 4:30 - Personnel States

Elements
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourceExchange
- ResourceExchangeItem
- ResourceParameter
- ResourcePerformer
- ResourceRole

View Specifications::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 4:31 - Personnel Interaction Scenarios

Elements

- Activity
- Function
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourceExchange
- ResourceExchangeItem
- ResourceMessage
- ResourceMethod
- ResourcePerformer
- ResourceRole

View Specifications::Personnel::Constraints

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 4:32 - Personnel Constraints: Competence

Elements
- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResponsibility
- ActualResponsibleResource
- Competence
- Function
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- ResourceRole
- Responsibility

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 4:33 - Personnel Constraints: Drivers

Elements

- Activity
- ActualCondition
- ActualMeasurement
- ActualMeasurementSet
- ActualPropertySet
- Function
- MeasurableElement
- Measurement
- MeasurementSet
- OrganizationalResource
- PhysicalResource
- PropertySet
- ResourceConstraint
- ResourcePerformer
- Rule
- RuleKind
- SubjectOfResourceConstraint

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 4.34 - Personnel Constraints: Performance

Elements
- Activity
- ActualCondition
- ActualMeasurement
- ActualMeasurementSet
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualState
- DesiredEffect
- Desirer
- Function
- MeasurableElement
- Measurement
- MeasurementSet
- OrganizationalResource
- Person
- PhysicalResource
**View Specifications::Personnel::Roadmap**


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 4:35 - Personnel Roadmap: Availability**

Elements

- ActualMeasurement
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualProject
- ActualProjectMilestone
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- Asset
- Measurement
- OrganizationalResource
- Person
• PhysicalResource
• Post
• Project
• ProjectKind
• ProjectMilestone
• ProjectMilestoneRole
• PropertySet
• ResourceAsset
• ResourcePerformer

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 4:36 - Personnel Roadmap: Evolution

Elements
• ActualProject
• ActualProjectMilestone
• MilestoneDependency
• Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- VersionedElement
- VersionOfConfiguration
- VersionSuccession
- WholeLifeConfiguration
- WholeLifeConfigurationKind

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.

**Figure 4:37 - Personnel Roadmap: Forecast**

Elements
- ActualEnterprisePhase
- Competence
- Forecast
- Organization
- OrganizationalResource
- PhysicalResource
- Post
- ResourcePerformer
- SubjectOfForecast

**View Specifications::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.

Figure 4.38 - Personnel Traceability

Elements
- Function
- OperationalActivity
- ServiceFunction

4.1.5 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

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 4:39 - Resources Taxonomy

Elements
- Asset
- CapabilityConfiguration
- Measurement
- NaturalResource
- OperationalAgent
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PropertySet
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourceExchange
- ResourceMitigation
- ResourcePerformer
- ResourceRole
- Responsibility
- Software
- System
- Technology

View Specifications::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 4:40 - Resources Structure

Elements
- CapabilityConfiguration
- NaturalResource
- OperationalInterface
- PhysicalResource
- Protocol
- ProtocolImplementation
- ProtocolLayer
- ProtocolStack
- ResourceArchitecture
- ResourceArtifact
- ResourceConnector
- ResourceExchange
- ResourceInterface
- ResourceMitigation
- ResourcePerformer
- ResourcePort
**View Specifications::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 4:41 - Resources Connectivity**

Elements
- **CapabilityConfiguration**
- **DataElement**
- **Exchange**
- **Function**
- **FunctionAction**
- **GeoPoliticalExtentType**
- **MeasurableElement**
- **NaturalResource**
- **OperationalExchange**
- **Organization**
- **OrganizationalResource**
View Specifications::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 4:42 - Resources Processes
Elements
- Activity
- ActualCondition
- DataElement
- Function
- FunctionAction
- FunctionControlFlow
- FunctionEdge
- FunctionObjectFlow
- OperationalActivity
- PhysicalResource
- ResourceArchitecture
- ResourceExchange
- ResourceExchangeItem
- ResourceParameter
- ResourcePerformer
- ResourceRole
- ServiceFunction

**View Specifications::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.

![StateMachine](ResourcePerformer)\[Class\] \(\text{«metaconstraint} \)\[umlRole = "owner"\] ResourceStateDescription\[StateMachine\]

**Figure 4:43 - Resources States**

Elements
- ResourcePerformer
- ResourceStateDescription

**View Specifications::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 4:44 - Resources Interaction Scenarios

Elements
- Activity
- Function
- ResourceExchange
- ResourceExchangeItem
- ResourceMessage
- ResourceMethod
- ResourcePerformer
- ResourceRole

View Specifications::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 4:45 - Resources Constraints

Elements

- ActualResource
- CapabilityConfiguration
- DataElement
- Function
- NaturalResource
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourceArchitecture
- ResourceArtifact
- ResourceConstraint
- ResourcePerformer
- ResourceRole
- Responsibility
- Rule
- RuleKind
- Software
- SubjectOfResourceConstraint
- Technology

View Specifications::Resources::Roadmap

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 4:46 - Resources Roadmap: Evolution

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

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 4:47 - Resources Roadmap: Forecast

Elements
- ActualEnterprisePhase
- CapabilityConfiguration
- Competence
- Forecast
- Organization
- OrganizationalResource
- PhysicalResource
- Post
- ResourceArchitecture
- ResourceArtifact
- ResourcePerformer
- Software
- Standard
- SubjectOfForecast
- Technology

View Specifications::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.
4.1.6 View Specifications::Security

View Specifications::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 4:49 - Security Taxonomy

Elements
- ActualLocation
- Asset
- CapabilityConfiguration
- DataElement
- InformationElement
- KnownResource
- LocationHolder
- MeasurableElement
- Measurement
- MeasurementSet
- NaturalResource
- OperationalAgent
- OperationalAsset
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PropertySet

«stereotype» ActualLocation [InstanceSpecification]
«stereotype» LocationHolder [Element]
«stereotype» Asset [Class]
«stereotype» CapabilityConfiguration [Class]
«stereotype» KnownResource [Class]
«stereotype» OperationalAgent [Class]
«stereotype» OperationalPerformer [Class]
«stereotype» ResourceArchitecture [Class]
«stereotype» PhysicalResource [Class]
«stereotype» Responsibility [Class]
«stereotype» Person [Class]
«stereotype» Post [Class]
«stereotype» Organization [Class]
«stereotype» Software [Class]
«stereotype» Technology [Class]
View Specifications::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 4:50 - Security Structure

Elements
- Asset
- DataElement
- DataRole
- InformationElement
- InformationRole
View Specifications::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 4:51 - Security Connectivity
Elements

- Asset
- DataElement
- Exchange
- InformationElement
- MeasurableElement
- Measurement
- MeasurementSet
- OperationalAgent
- OperationalAsset
- OperationalConnector
- OperationalExchange
- OperationalExchangeItem
- OperationalPerformer
- OperationalPort
- OperationalRole
- PropertySet
- ResourceAsset
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceInterface
- ResourcePerformer
- ResourcePort
- ResourceRole
- SubjectOfSecurityConstraint

View Specifications::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.
Figure 4:52 - Security Processes

Elements
- Activity
- Function
- FunctionAction
- MeasurableElement
- Measurement
- MeasurementSet
- OperationalActivity
- OperationalActivityAction
- OperationalAgent
- OperationalRole
- PropertySet
- ResourcePerformer
- ResourceRole
- SecurityProcess
- SecurityProcessAction

View Specifications::Security::Constraints

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 4:53 - Security Constraints

Elements

- ActualMeasurement
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualRisk
- Asset
- AssetRole
- EnhancedSecurityControl
- MeasurableElement
- Measurement
- MeasurementSet
- OperationalAgent
- OperationalRole
- OrganizationalResource
- PropertySet
- ResourcePerformer
- ResourceRole
- Risk
- SecurityConstraint
View Specifications::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 4:54 - Security Traceability
Elements
- Asset
- AssetRole
- OperationalAgent
- OperationalAsset
- OperationalRole
- ResourceAsset
- ResourcePerformer
- ResourceRole
- Risk
- SecurityControl

4.1.7 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
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.

Figure 4:55 - Project Taxonomy
Elements
- Project
- ProjectMilestone
- ProjectMilestoneRole
- ResourcePerformer

View Specifications::Projects::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 4:56 - Project Structure

Elements
- ActualOrganization
- ActualPost
- ActualProject
- ActualProjectRole
- ActualResponsibleResource
- Project
- ProjectKind
- ProjectMilestone
- ProjectMilestoneRole
- ProjectRole
- ProjectTheme
- ResponsibleFor
- ResponsibleRoleKind
- StatusIndicators

View Specifications::Projects::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 4:57 - Project Connectivity

Elements
- ActualProject
- ActualProjectMilestone
- ActualProjectMilestoneRole
- Project
- ProjectMilestone
- ProjectMilestoneRole

View Specifications::Projects::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.
Figure 4:58 - Project Processes

Elements
- ActualProject
- DataElement
- Function
- FunctionAction
- FunctionControlFlow
- FunctionEdge
- FunctionObjectFlow
- Organization
- OrganizationalResource
- PhysicalResource
- Post
- Project
- ProjectActivity
• ProjectActivityAction
• ProjectRole
• ResourceExchange
• ResourceExchangeItem
• ResourcePerformer
• ResourceRole
• ResourceSignal

View Specifications::Projects::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 4:59 - Project Roadmap

Elements
• ActualMilestoneKind
• ActualOrganization
• ActualPost
• ActualProject
• ActualProjectMilestone
• ActualProjectMilestoneRole
View Specifications::Projects::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 4:60 - Project Traceability

Elements
- ActualProject
- ActualProjectMilestone
- ActualResource
- Capability
- CapabilityConfiguration
- ResourceArchitecture
- CapableElement
- ResourcePerformer
4.1.8 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

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 4:61 - Standards Taxonomy

Elements
- ActualOrganization
- Protocol
- ProtocolLayer
- ProtocolStack
- ResourcePerformer
- Standard
- UAFEElement

View Specifications::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 4:62 - Standards Structure

Elements
- Protocol
- ProtocolLayer
- ProtocolStack
- Standard

View Specifications::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 4:63 - Standards Roadmap

Elements
- ActualEnterprisePhase
- Forecast
- Protocol
- Standard
- SubjectOfForecast

View Specifications::Standards::Traceability

Stakeholders: Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects. 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.
4.1.9 View Specifications::Actual Resources

View Specifications::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 4:65 - Actual Resources Structure

Elements
- ActualOrganization
- ActualOrganizationalResource
- ActualOrganizationRole
- ActualPerson
- ActualPost
- ActualResource
- ActualResponsibility
- ActualResponsibleResource
- Competence
- FieldedCapability
- FillsPost
- Organization
- OrganizationalResource
- Person
- Post
- ResponsibleFor

View Specifications::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 4:66 - Actual Resources Connectivity

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

View Specifications::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.

Figure 4:67 - Actual Resources Traceability

Elements
- ActualResource
- Capability
4.1.10 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.
Recommended Implementation: Tabular format, SysML Block Definition Diagram.

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.

Figure 4:68 – Dictionary

Elements
- Alias
- Definition
- Information
- InformationKind
- SameAs
- UAFElement

4.1.11 View Specifications::Requirements

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.
4.1.12 View Specifications::Summary & Overview

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 4:69 – Requirements

Elements
- UAFElement

Figure 4:70 - Summary & Overview
4.1.13 View Specifications::Information

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.

Figure 4:71 - Information Model
Elements

- DataElement
- DataModel
- DataModelKind
- DataRole
- InformationElement
- InformationRole
- OperationalAsset
- ResourceAsset

4.1.14 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 4:72 - Parameters: Environment
Elements

- Activity
- ActualCondition
- ActualEnvironment
- ActualLocation
- ActualPropertySet
- ActualResource
- Asset
- Condition
- Environment
- EnvironmentKind
- EnvironmentProperty
- GeoPoliticalExtentType
- GeoPoliticalExtentTypeKind
- Location
- LocationHolder
- LocationKind
- LocationTypeKind
- OperationalRole
- ResourceAsset
- 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, MOs, 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 4:73 - Parameters: Measurements

Elements
- ActualCondition
- ActualMeasurement
- ActualMeasurementKind
- ActualMeasurementSet
- ActualPropertySet
- ActualService
- ActualState
- Capability
- Competence
- Condition
- EnterprisePhase
- MeasurableElement
- Measurement
- MeasurementSet
- Project
- PropertySet
- ProvidedServiceLevel
- RequiredServiceLevel
- Resource
- ServiceInterface
- ServiceSpecification
5 Measurement Library

A library of Measurement Sets.

BillingItem

**Package:** Class Library

isAbstract: No

**Description**

Properties indicating the assurance of a piece of information.

**Attributes**

- **cost**: Cost[1] - Details the cost of the BillingItem.
- **id**: String[0..1] - Details the unique identifier of the BillingItem.
- **numberOfUses**: Integer[0..1] - Details the numberOfUses of the BillingItem.
- **paymentLocation**: String[0..1] - Details the location where payment should be made of the BillingItem.
- **paymentModality**: PricingType[1] - Details if a payment is based upon Quantity, Time or Use.
- **paymentPeriod**: Periodicity[1] - Details the frequency of a payment period.
- **paymentTimeDuration**: Duration[*] - Details the length of time the payments should be made i.e., 1 year.
- **periodDuration**: Duration[0..1] - Details the time period between payments.
- **quantity**: String[0..1] - Details the number of units to be delivered.
- **unit**: String[0..1] - Details the units used for the BillingItem e.g., 1 gross.

ClassificationAttributes

**Package:** Class Library

isAbstract: No

**Description**

W3C XML Schema for the Intelligence Community Metadata Standard for Information Security Marking (IC-ISM), which is part of the IC standards for Information Assurance.

**Attributes**

- **classificationReason**: String[] - One or more reason indicators or explanatory text describing the basis for an original classification decision.
- **classifiedBy**: String[] - Details The identity, by name or personal identifier, and position title of the original classification authority for a resource.
- **dateOfExemptedSource**: String[] - Details the specific year, month, and day of publication or release of a source document, or the most recent source document, that was itself marked with a declassification constraint. This element is always used in conjunction with typeOfExemptedSource element.
- **declassDate**: String[] - Details a specific year, month, and day upon which the information shall be automatically declassified if not properly exempted from automatic declassification.
- **declassException**: String[] - Details a single indicator describing an exemption to the nominal 25-year point for automatic declassification. This element is used in conjunction with the Declassification Date or Declassification Event.
- **DeclassManualReview**: String[] - Details a true/false indicator that a manual review is required for declassification. Use this attribute to force the appearance of "//MR" in the header and footer marking titles. Use this attribute ONLY when it is necessary to override the business logic applied to classification and control markings in the document to determine whether manual review is required.
derivedFrom : String[]
Details a citation of the authoritative source or reference to multiple sources of the classification markings used in a classified resource.

DisseminationControls : String[]
Details one or more indicators identifying the expansion or limitation on the distribution of information.

FGIsourceOpen : String[]
Details one or more indicators identifying information which qualifies as foreign government information for which the source(s) of the information is not concealed.

FGIsourceProtected : String[]
Details a single indicator that information qualifies as foreign government information for which the source(s) of the information must be concealed. Within protected internal organizational spaces this element may be used to maintain a record of the one or more indicators identifying information which qualifies as foreign government information for which the source(s) of the information must be concealed. Measures must be taken prior to dissemination of the information to conceal the source(s) of the foreign government information.

nonICmarkings : String[]
Details one or more indicators of the expansion or limitation on the distribution of an information resource or portion within the domain of information originating from non-intelligence components.

ownerProducer : String[]
Details one or more indicators identifying the national government or international organization that have purview over the classification marking of an information resource or portion therein. This element is always used in conjunction with the Classification element. Taken together, the two elements specify the classification category and the type of classification (US, non-US, or Joint). Within protected internal organizational spaces this element may include one or more indicators identifying information which qualifies as foreign government information for which the source(s) of the information must be concealed. Measures must be taken prior to dissemination of the information to conceal the source(s) of the foreign government information.

releasableTo : String[]
Details one or more indicators identifying the country or countries and/or international organization(s) to which classified information may be released based on the determination of an originator in accordance with established foreign disclosure procedures. This element is used in conjunction with the Dissemination Controls element.

SARIdentifier : String[]
Details the Authorized Special Access Required (SAR) program digraph(s) or trigraph(s) preceded by "SAR-". Either (a) a single digraph or trigraph or (b) a space-delimited list of digraphs or trigraphs. Example: "SAR-ABC SAR-DEF ..."

SCIControls : String[]
Details one or more indicators identifying sensitive compartmented information control system(s).

typeOfExemptedSource : String[]
Details a declassification marking of a source document that causes the current, derivative document to be exempted from automatic declassification. This element is always used in conjunction with the Date Of Exempted Source element.

Associations
taxonomy : String[]
Details a single indicator of the highest level of classification applicable to an information resource or portion within the domain of classified national security information. The Classification element is always used in conjunction with the Owner Producer element. Taken together, the two elements specify the classification category and the type of classification (US, non-US, or Joint).

CommunicationsLinkProperties

Package: Class Library
isAbstract: No
Description
Properties detailing aspects of Resource Interfaces.
Attributes
  capacity : String[] Details how much information can be passed on the Communications Link.
  infrastructureTechnology : String[] Details the technology to be used to provide the communications infrastructure.

DataElementProperties
Package: Class Library
isAbstract: No
Description
Properties detailing the aspects of a DataElement.
Attributes
  accuracy : String[] Details the accuracy of the data.
  content : String[] Specifies content of the data element (i.e., actual data to be exchanged).
  formatType : String[] Details the format of the data.
  mediaType : String[] Details the media used to transmit the data.
  scope : String[] Details in text a description of the extent or range of the data element content.
  unitOfMeasurement : String[] Details the units of measurement of the data.

Duration
Package: Class Library
isAbstract: No
Description
Properties detailing aspects OperationalActivities.
Attributes
  timeUnit : String[0..1] Details the units of time e.g. second, hour, day.
  value : Integer[0..1] Details the value of the duration.

ExchangeProperties
Package: Class Library
isAbstract: No
Description
Properties detailing aspects of exchange for Operational Exchange and/or Resource Interaction.
Attributes
  accountability : String[*] Details who or what is responsible for the exchange.
  periodicity : String[*] Details the frequency of the exchange.
  size : String[*] Details the size (in KB) of data that be exchanged.
  throughput : String[*] Details how much information can be exchanged.
  timeliness : String[*] Details the allowable time of delay this system data can tolerate and still be relevant to the receiving system.
  transactionType : String[*] Details the type of transactions used by the exchange.

InformationElementProperties
Package: Class Library
isAbstract: No
Description
Predefined additional DoDAF properties for InformationElement.
Attributes
  accuracy : String[*] Details the degree to which the information conforms to actual fact as required by the information producer and consumer.
  content : String[*] Specifies content of the information element (i.e., actual information to be exchanged).
  language : String[*] Details the language used to capture the information.
  scope : String[*] Details in text a description of the extent or range of the information element content.

OperationalActivityProperties
Package: Class Library
isAbstract: No
Description
Properties detailing aspects OperationalActivities.
Attributes
  cost : String[] Details the cost of an activity.

Periodicity
Package: Class Library
isAbstract: No
Description
Enumeration of how often the information exchange occurs; may be an average or a worst-case estimate and may include conditions. Its enumeration literals are:
  • OnceAMonth - Indicates that an event of some sort may occur monthly.
  • OnceAWeek - Indicates that an event of some sort may occur weekly.
  • Anytime - Indicates that an event of some sort may occur at anytime.
  • OnRequest - Indicates that an event of some sort may occur on request.

PricingType
Package: Class Library
isAbstract: No
Description
Enumeration of a unit of measure of a resource. Its enumeration literals are:
  • perTIme - Indicates that the unit of measure of a resource is based on a unit of time.
  • perUse - Indicates that the unit of measure of a resource is based upon how often the resource is used.
  • perQuantity - Indicates that the unit of measure of a resource is based on a quantity.

SecurityControlAssessmentProperties
Package: Class Library
isAbstract: No
Description
Properties detailing aspects of the Assessment and Authorization process.
Attributes
  coverageOfSecurityControlAssessment : String[*] Security controls assessment method that addresses the scope or breadth of the assessment objects included in the assessment (e.g., types of objects to be assessed and the number of objects to be assessed by type).
depthOfSecurityControlAssessment : String[*]
Security controls assessment method that addresses the rigor and level of detail associated with the application of the method.
effectivenessOfSecurityControl : String[*]
Details if security control is satisfactory or not as assessed.

SecurityControlProperties
Package: Class Library
isAbstract: No
Description
Properties detailing aspects of Security Controls.
Attributes
  securityControlApplicability : String[1]
  Details how applicable a security control is to a given security objective.
  securityControlImportance : String[1]
  Details how important a security control is to a given security objective.

SecurityImpactProperties
Package: Class Library
isAbstract: No
Description
Properties detailing aspects of Security Categories.
Attributes
  securityAvailabilityImpact : String[*]
  Details the potential impact on organization or individuals if the information is not available to those who need to access it.
  securityClassification : String[*]
  Details a classification for the exchange.
  securityConfidentialityImpact : String[*]
  Details the potential impact on organization or individuals due to unauthorized disclosure of information.
  securityIntegrityImpact : String[*]
  Details the potential impact on organization or individuals due to modification or destruction of information, and includes ensuring information non-repudiation and authenticity.