ENTERPRISE ARCHITECT

User Guide Series

Unified Profile for DoDAF/MODAF (UPDM)

Author: Sparx Systems
Date: 10/11/2023
Version: 16.1

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

Unified Profile for DoDAF/MODAF (UPDM) 3
Brief Introduction 4
UPDM Support 5
UPDM System Requirements 6
Licensing Copyright and Trademarks 7
MDG Technology for UPDM Copyright Notice 8
MDG Technology for UPDM Software Product License Agreement 9
Acknowledgement of Trademarks - UPDM 12
Using UPDM 13
Getting Started with UPDM 14
Model Wizard in UPDM 15
UPDM Extensions Menu 16
UPDM Framework Diagram 17
UPDM Diagram Types 19
UPDM Toolbox Pages 20
UPDM Stereotypes 22
Abstract Stereotypes 72
Quicklinks 78
Tagged Values for UPDM 79
Model Views in UPDM 80
Glossary 82
Using Enterprise Architect Elements 83
Model Validation in UPDM 85
Model Validation Rules 86

Unified Profile for DoDAF/MODAF (UPDM)

UPDM (Unified Profile for DoDAF-MODAF) provides a UML profile that extends the capability of Enterprise Architect to provide a standard approach for modeling systems and Enterprise Architectures in support of DoDAF and MODAF.

DoDAF is the abbreviation of Department of Defense Architecture Framework (USA); MODAF is the abbreviation of Ministry of Defence Architecture Framework (UK).

Discussion

The topics described here provide an introduction to, and procedural explanation of, using UPDM in Enterprise Architect.

Section	Content
Welcome	This section provides an introduction to UPDM, and explains the support available for the Framework and the system requirements for its use.
Licensing Copyright and Trademarks	These topics contain the formal documentation defining the use of the MDG Technology for UPDM with Enterprise Architect.
Using UPDM	Work with UPDM, learning about the model structure, templates, diagram types and more.
Model Validation	Learn how to develop and configure model validation for UPDM.

Brief Introduction

Welcome to the UPDM 2.0 profile in Sparx Systems Enterprise Architect.

This UML profile extends the capability of Enterprise Architect to support the creation of Unified Profile for DoDAF and MODAF (UPDM) architecture models. DoDAF is the abbreviation of Department of Defense Architecture Framework (USA); MODAF is the abbreviation of Ministry of Defence Architecture Framework (UK).

The UPDM profile provides a standard approach for modeling systems and enterprise architectures in support of DoDAF and MODAF. It improves interoperability of architecture data among architecture modeling tools, enhances reuse of architecture data, and improves communication among DoDAF and MODAF stakeholders.

UPDM is already integrated with the Enterprise Architect Ultimate and Unified Editions; an MDG Technology can be purchased separately to be used with the Enterprise Architect Professional or Corporate Editions.

This technology is based on the Unified Profile for DoDAF-MODAF (UPDM) version 1.0. UPDM 1.0 is based on DoDAF version 1.5 and MODAF version 1.2. Full details of the profile, including the latest specification, can be obtained from the Object Management Group (OMG) website.

Getting Started

For instructions on how to use UPDM, see the topics *Getting Started with UPDM* and *Using UPDM*.

UPDM Support

Technical support for the UPDM profile is available to registered users of Enterprise Architect in exactly the same way as for Enterprise Architect itself.

UPDM System Requirements

The UPDM profile version 2.0 runs under these environments:

Operating Systems

- Windows 10
- Windows 8
- Windows 7
- Windows 2008 Server
- Windows 2003 Server
- Windows XP Service Pack 2

Enterprise Architect Versions

- Enterprise Architect Version 9.0 or later

Licensing Copyright and Trademarks

For the Sparx Systems MDG Technology for UPDM, this topic provides the:

- Copyright Notice
- Software Product License Agreement (or End User License Agreement) and
- The acknowledgement of trademarks of other products referenced in the User Interface and documentation

MDG Technology for UPDM Copyright Notice

This legal information concerns the copyright ownership of Enterprise Architect and of any 3rd party tools of code that require a statement of copyright ownership.

Copyright © 2010 - 2022 Sparx Systems Pty. Ltd. All rights reserved.

The software contains proprietary information of Sparx Systems Pty Ltd. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. Please read the product license agreement for full details.

Due to continued product development, this information may change without notice. The information and intellectual property contained herein is confidential between Sparx Systems and the client and remains the exclusive property of Sparx Systems. If you find any problems in the documentation, please report them to us in writing. Sparx Systems does not warrant that this document is error-free. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Sparx Systems. Licensed users are granted the right to print a single hardcopy of the user manual per licensed copy of the software, but may not sell, distribute or otherwise dispose of the hardcopy without written consent of Sparx Systems.

Sparx Systems Pty. Ltd.

99 Albert St,

Creswick, Victoria 3363,

AUSTRALIA

Phone: +61 (3) 5345 1140

Fax: +61 (3) 5345 1104

Support Email: support@sparxsystems.com

Sales Email: sales@sparxsystems.com

Website: sparxsystems.com

MDG Technology for UPDM Software Product License Agreement

This Software Product License Agreement relates to the separately-purchased MDG Technology for UPDM for use with the Corporate and Professional Editions of Sparx Systems Enterprise Architect. Where the MDG Technology for UPDM is integrated with the Ultimate and Unified Editions of Enterprise Architect, it is covered by the Sparx Systems Enterprise Architect Modelling Tool.

MDG Technology for UPDM, Enterprise Architect MDG Add-In, Version 2.0.

Copyright (C) 2010 - 2022 Sparx Systems Pty Ltd. All Rights Reserved

IMPORTANT-READ CAREFULLY: This End User License Agreement ("EULA") is a legal agreement between YOU as Licensee and SPARX for the SOFTWARE PRODUCT identified above. By installing, copying, or otherwise using the SOFTWARE PRODUCT, YOU agree to be bound by the terms of this EULA. If YOU do not agree to the terms of this EULA, promptly delete the unused SOFTWARE PRODUCT.

The copyright in the SOFTWARE PRODUCT and its documentation is owned by Sparx Systems Pty Ltd, A.B.N 38 085 034 546. Subject to the terms of this EULA, YOU are granted a non-exclusive right for the duration of the EULA to use the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

Your use of this software indicates your acceptance of this EULA and warranty.

DEFINITIONS

In this End User License Agreement, unless the contrary intention appears:

- "EULA" means this End User License Agreement
- "SPARX" means Sparx Systems Pty Ltd A.C.N 085 034 546
- "LICENSEE" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA
- "Registered Edition of MDG Technology for UPDM" means the edition of the SOFTWARE PRODUCT, which is available for purchase from the web site: https://sparxsystems.com/updm/purchase.html, following a thirty-day free evaluation period
- "SOFTWARE PRODUCT" or "SOFTWARE" means MDG Technology for UPDM, which includes computer software and associated media and printed materials, and may include online or electronic documentation
- "SUPPORT SERVICES" means email-based support provided by SPARX, including advice on usage of the SOFTWARE PRODUCT, investigation of bugs, fixes, repairs of models, if and when appropriate, and general product support
- "SPARX SUPPORT ENGINEERS" means employees of SPARX who provide on-line support services
- "TRIAL EDITION" means the edition of the SOFTWARE PRODUCT, which is available free of charge for evaluation purposes for a period of thirty (30) days

GRANT OF LICENSE

In accordance with the terms of this EULA, YOU are granted the following rights:

- To install and use one copy of the SOFTWARE PRODUCT, or in its place, any prior version for the same operating system, on a single computer; as the primary user of the computer on which the SOFTWARE PRODUCT is installed, YOU may make a second copy for your exclusive use on either a home or portable computer
- To store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT over an internal network; if YOU wish to increase the number of users

entitled to concurrently access the SOFTWARE PRODUCT, YOU must notify SPARX and agree to pay an additional fee

- To make copies of the SOFTWARE PRODUCT for backup and archival purposes only

EVALUATION LICENSE

The TRIAL EDITION is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use the SOFTWARE PRODUCT for evaluation purposes without charge for a period of thirty (30) days.

Upon expiration of the thirty (30) days, the Software Product must be removed from the computer. Unregistered use of the SOFTWARE PRODUCT after the 30-day evaluation period is in violation of Australian, U.S. and international copyright laws.

SPARX may extend the evaluation period on request and at their discretion.

If YOU choose to use the SOFTWARE PRODUCT after the 30-day evaluation period, a license must be purchased (as described at https://sparxsystems.com/updm/purchase.html). Upon payment of the license fee, YOU will be sent details on where to download the registered edition of the software product and will be provided with a suitable software 'key' by email.

ADDITIONAL RIGHTS AND LIMITATIONS

YOU hereby undertake not to sell, rent, lease, translate, adapt, vary, modify, decompile, disassemble, reverse engineer, create derivative works of, modify, sub-license, loan or distribute the SOFTWARE PRODUCT other than as expressly authorized by this EULA.

YOU further undertake not to reproduce or distribute license key-codes except under the express and written permission of SPARX.

If the Software Product purchased is an ACADEMIC EDITION, YOU acknowledge that the license is limited to use in an educational context, either for self-education or use in a registered teaching institution. The ACADEMIC EDITION may not be used to produce commercial software products or be used in a commercial environment, without the express written permission of SPARX.

ASSIGNMENT

YOU may only assign all your rights and obligations under this EULA to another party if YOU supply to the transferee a copy of this EULA and all other documentation including proof of ownership. Your license is then terminated.

TERMINATION

Without prejudice to any other rights, SPARX may terminate this EULA if YOU fail to comply with the terms and conditions. Upon termination YOU or YOUR representative shall destroy all copies of the SOFTWARE PRODUCT and all of its component parts or otherwise return or dispose of such material in the manner directed by SPARX.

WARRANTIES AND LIABILITY

WARRANTIES

SPARX warrants that:

- The SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and
- Any SUPPORT SERVICES provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX SUPPORT ENGINEERS will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

EXCLUSIONS

To the maximum extent permitted by law, SPARX excludes, for itself and for any supplier of software incorporated in the SOFTWARE PRODUCT, all liability for all claims, expenses, losses, damages and costs made against or incurred or suffered by YOU directly or indirectly (including without limitation lost costs, profits and data) arising out of:

- YOUR use or misuse of the SOFTWARE PRODUCT;
- YOUR inability to use or obtain access to the SOFTWARE PRODUCT;
- Negligence of SPARX or its employees, contractors or agents, or of any supplier of software incorporated in the SOFTWARE PRODUCT, in connection with the performance of SPARX's obligations under this EULA; or
- Termination of this EULA by either party for any reason.

LIMITATION

The SOFTWARE PRODUCT and any documentation are provided "AS IS" and all warranties, whether express, implied, statutory or otherwise, relating in any way to the subject matter of this EULA or to this EULA generally, including without limitation, warranties as to: quality; fitness; merchantability; correctness; accuracy; reliability; correspondence with any description or sample, meeting your or any other requirements; uninterrupted use; compliance with any relevant legislation; and being error or virus free are excluded. Where any legislation implies in this EULA any term, and that legislation avoids or prohibits provisions in a contract excluding or modifying such a term, such term shall be deemed to be included in this EULA. However, the liability of SPARX for any breach of such term shall, if permitted by legislation, be limited, at SPARX's option to any one or more of the following upon return of the SOFTWARE PRODUCT and a copy of the receipt:

- If the breach relates to the SOFTWARE PRODUCT:
- The replacement of the SOFTWARE PRODUCT, or the supply of an equivalent SOFTWARE PRODUCT;
- The repair of such SOFTWARE PRODUCT, or the payment of the cost of replacing the SOFTWARE PRODUCT, or of acquiring an equivalent SOFTWARE PRODUCT; or
- The payment of the cost of having the SOFTWARE PRODUCT repaired.
- If the breach relates to services in relation to the SOFTWARE PRODUCT:
- The supplying of the services again; or
- The payment of the cost of having the services supplied again.

TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and Licenses.

GOVERNING LAW

This agreement shall be construed in accordance with the laws of the Commonwealth of AUSTRALIA, in the state of Victoria.

Acknowledgement of Trademarks - UPDM

Trademarks of Microsoft

- Microsoft®
- Windows®

Trademarks of the OMG

- OMG™
- Object Management Group™
- UML™
- Unified Modeling Language™

Using UPDM

UPDM is the Unified Profile for the Department of Defense Architecture Framework (DoDAF) and Ministry of Defence Architecture Framework (MODAF). UPDM is an Object Management Group (OMG) initiative; the specification is available from the OMG website.

You can perform UPDM modeling within Enterprise Architect, using these facilities:

- The UPDM Profile, which defines the stereotyped UML elements that are used for UPDM modeling
- Custom diagram types for each UPDM view
- Custom Diagram Toolbox pages for each UPDM diagram type, which give easy access to the elements used on diagrams of that type
- Options within the Model Wizard (Start Page 'Create from Pattern' tab) that can be used to import a template Package for each UPDM view and that provide a brief description of the view and what might be expected of the modeler
- Quicklinks for stereotyped elements that guide you towards creating correct relationships between elements
- Model Validation rules that you can apply to check your models for correctness
- Relationship Matrix profiles for showing the relationships between elements
- Model Views that help you navigate your model quickly to find specific diagram more easily
- A Glossary import, with items describing each UPDM stereotype for easy reference
- Tagged Values that you can use to enter metadata specific to UPDM elements
- An Example Model that illustrates a typical UPDM problem and its solution, implemented using Enterprise Architect

Getting Started with UPDM

When you install the Unified or Ultimate Edition of Enterprise Architect, the UPDM profile is fully enabled and ready to use.

If you have the Corporate or Professional Edition of Enterprise Architect, you can purchase and install an MDG Technology for UPDM separately; once you have entered the registration key for the MDG Technology for UPDM, it is automatically available in and integrated with Enterprise Architect, as for the Unified and Ultimate Editions.

Access the MDG Technology

1. Create a new Enterprise Architect project file, and click on the top-level Package.
2. Select the 'Design > Package > Model Wizard' option.
3. In the Start Page 'Create from Pattern' tab (Model Wizard), select the 'Systems Engineering > UPDM' Perspective and the 'UPDM Frameworks' Pattern Group; select either the 'DoDAF Framework' Pattern or the 'MODAF Framework' Pattern.
4. Click on the Create Model(s) button.

A new base DoDAF or MODAF model is created in the Browser window.

Model Wizard in UPDM

You can create UPDM models within your project using templates selected from the Enterprise Architect Model Wizard (Start Page 'Create from Pattern' tab).

Access

Ribbon	Design > Package > Model Wizard
Context Menu	Browser window | Right-click on Package | Add a Model using Wizard > Model Patterns
Keyboard Shortcuts	Ctrl+Shift+M

Notes

- In the Model Wizard, click on the <perspective name> button and select 'System Engineering > UPDM'
- Expand the 'UPDM Frameworks' group or one of the 'DoDAF' or 'MODAF' groups, and click on the required Pattern in that group
- Click on the Create Model(s) button to generate the corresponding UPDM model structures in your project

UPDM Extensions Menu

You can perform various tasks on your UPDM model using the UPDM Technology menu.

Access

Ribbon	Specialize > Technologies > UPDM 2.0
Context Menu	Right-click on Package, diagram or element | Specialize | UPDM 2.0

Options

Option	Action
Synchronize Tagged Values	Add missing Tagged Values to all elements in the model that require them.
Import Glossary	Import UPDM information into the Enterprise Architect Glossary.
Import Images	Import the alternative images (as used in the UPDM Framework diagram) into the current model. You can use these images to decorate your own models (select a diagram object, right-click and select 'Appearance | Select Alternate Image') or you can design your own.
Help	Display this Help topic.
About	Show the version of the MDG Technology for UPDM that you are using. The version number has the format 1.0.001, where 1.0 is the version of the UPDM specification that is supported, and 001 is the incremental build number.

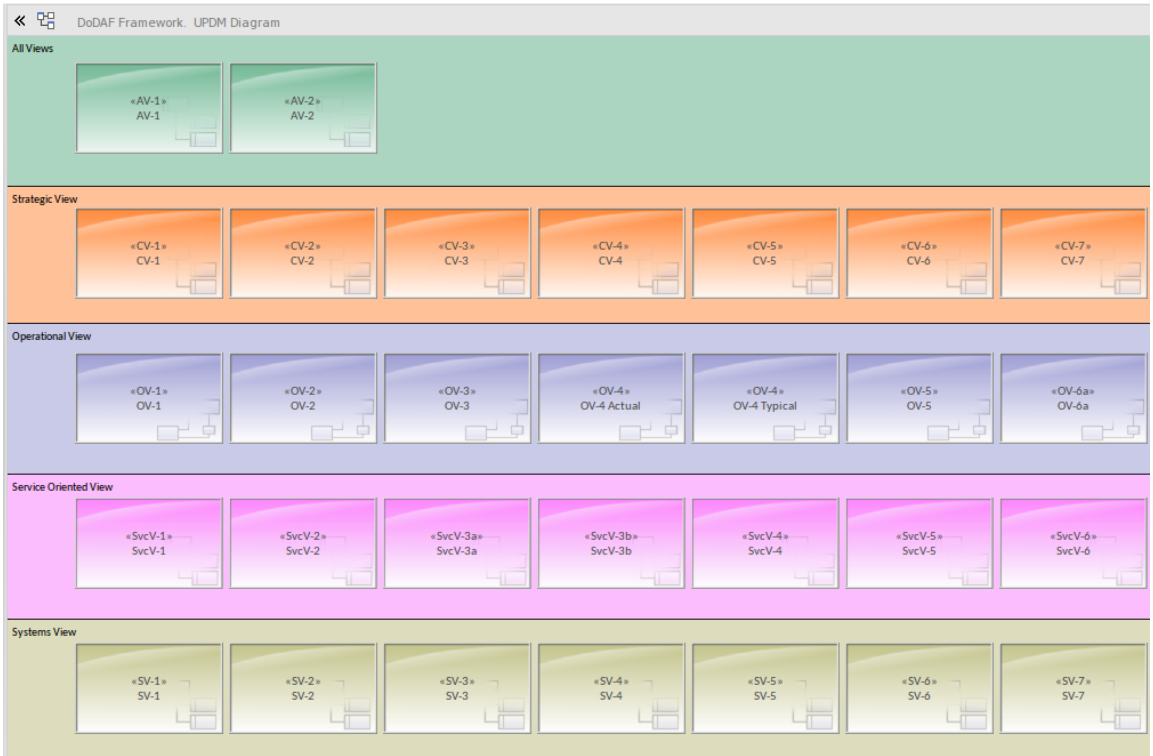
UPDM Framework Diagram

When developing and distributing a model, it is useful to have a single front page diagram that has hyperlinks to all the important information in the model. That is the aim of the two UPDM Framework diagrams (one for DoDAF, one for MODAF), which are created with color-coded swimlanes for each set of views. You can drag and drop on to these diagrams:

- Packages, which act as hyperlinks to the documents that they own
- Document Artifacts
- Any kind of composite element that points to its child diagram
- Hyperlinks pointing to custom SQL queries, Relationship Matrix profiles or external files

Create a UPDM Framework Diagram

1. In the Start Page 'Create from Pattern' tab (Model Wizard), click on the <perspective name> button and select 'System Engineering > UPDM'.
2. Expand the 'UPDM Frameworks' group and click on the required Pattern, either 'DODAF Framework' or 'MODAF Framework'.
3. Click on the Create Model(s) button to generate the corresponding UPDM model structures in your project.



Editing Swimlanes

You can add, remove and modify the swimlanes on the Framework diagram. Select 'Design > Diagram > Manage > Swimlanes'.

To change the width of swimlanes, use the mouse to drag their boundaries.

Changing Appearances

Each Package, document and hyperlink on the Framework diagram has an alternative image. To load these images into your model, select the 'Settings > Reference Data > Images' option.

If you want to apply your own bitmap images to the UPDM elements, you must first import the images into the model, also using the 'Settings > Reference Data > Images' option. Then you can either select the element and press Ctrl+Shift+W to add an alternative image to the element, or you can apply your own stereotype to apply a Shape Script to the element. For example, you might define a stereotype with this Shape Script:

```
shape main
{
	v_align="center";
	h_align="center";
	defSize(90,70);
	image("myBitMap.bmp",0,0,100,100);
	printWrapped("#name#");
}
```

UPDM Diagram Types

UPDM introduces a number of custom diagram types into Enterprise Architect. These are, for the most part, extended UML diagrams. On opening a UPDM diagram, Enterprise Architect automatically opens the appropriate UPDM Diagram Toolbox pages for the diagram type.

You can use the UPDM diagrams that are generated by the Model Wizard (Start Page 'Create from Pattern' tab), or create a new UPDM diagram.

Access

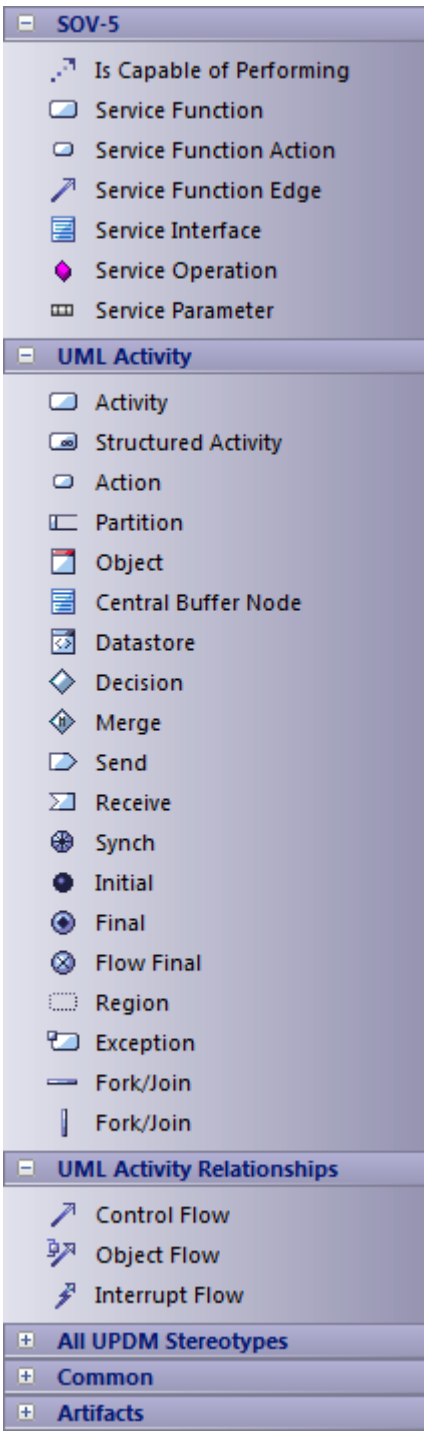
Ribbon	Design > Diagram > Add Diagram
Context Menu	Browser window | Right-click on Package | Add Diagram

Notes

- On the 'New Diagram' dialog, select 'UPDM' in the 'Select From' panel and the appropriate diagram type in the 'Diagram Types' panel
- Click on the OK button to open the Diagram View with the empty diagram displayed

UPDM Toolbox Pages

When you open a diagram, Enterprise Architect opens the Diagram Toolbox pages that are most useful for that particular diagram type. For a UPDM diagram, Enterprise Architect opens the Toolbox pages that contain elements and relationships appropriate to the particular View that the diagram is part of, as well the pages for the diagram type. For example, if you open an SOV-5 Activity diagram, Enterprise Architect opens the 'SOV-5 Elements' page, the 'UML Activity' page and the 'UML Activity Relationships' page.



In addition, the 'Common Elements' and 'Common Relationships' pages and various global task pages of the Diagram Toolbox are always available, regardless of which diagram is open.

If you hide the default Toolbox pages and want to get them back, simply switch to the Start Page and back to the current

diagram, and all the default Toolbox pages for the current diagram type are re-opened.

All UPDM Stereotypes

For your convenience, a Diagram Toolbox page is provided that includes every stereotype in the UPDM profile, listed in alphabetical order. If you cannot remember which context-sensitive Toolbox page a stereotype appears in, just go to the 'All UPDM Stereotypes' Toolbox page instead. To make this page available at all times, either:

- Select the 'Specialize > Technologies > Manage Technology' ribbon option, select 'UPDM Technology' in the table, and click on the Set Active button, or
- Select 'UPDM 2.0' from the list box on the Default Tools toolbar

UPDM Stereotypes

ActualMeasurementSet

A set or collection of measurements; used in AV-3, OV-3, SV-6 and SV-7.

Extensions:

- Object

Constraints:

- Classifier must be a MeasurementSet

Use:

- Press Ctrl and drag a MeasurementSet element from the Browser window to create an instance, or drop an ActualMeasurementSet from the Diagram Toolbox and press Ctrl+L to set the classifier; set the Run State and enter actual values for each of the classifier's attributes

ActualOrganization

An actual specific organization as an instance of an organization Class; used in AcV-1, OV-4, StV-5, TV-1 and TV-2.

Extensions:

- Object

Generalizations:

- ActualOrganizationalResource

Constraints:

- Classifier must be an Organization

Use:

- Press Ctrl and drag an Organization from the Browser window to create an instance, or drop an ActualOrganization from the Diagram Toolbox and press Ctrl+L to set the classifier
- Can have a set of 'ratifiedStandards' (Standard)
- Can be 'responsibleFor' a set of ActualProject
- Can be client and/or supplier of an ActualOrganizationRelationship
- Can be client of an OwnsProcess dependency to an OperationalActivity

ActualOrganizationRelationship

A relationship between two actual organizational resources (organizations or posts); used in OV-4.

Extensions:

- InformationFlow

Constraints:

- Supplier must be an ActualOrganizationalResource (ActualOrganization or ActualPost)
- Client must be an ActualOrganizationalResource (ActualOrganization or ActualPost)
- Realizes a ResourceInteraction

ActualPerson

A named individual that fulfills an ActualPost; used in OV-4.

Extensions:

- Object

Constraints:

- Classifier must be a Person

Use:

- Press Ctrl and drag a Person from the Browser window to create an instance, or drop an ActualPerson from the Diagram Toolbox and press Ctrl+L to set the classifier
- Can be a client of a FillsPost dependency to an ActualPost

ActualPost

An actual, specific post, as an instance of the Post Class; used in AcV-1, OV-4 and StV-5.

Extensions:

- Object

Generalizations:

- ActualOrganizationalResource

Constraints:

- Classifier must be a Post

Use:

- Press Ctrl and drag a Post from the Browser window to create an instance, or drop an ActualPost from the Diagram Toolbox and press Ctrl+L to set the classifier
- Can be responsible for a set of ActualProject
- Can be client and/or supplier of an ActualOrganizationRelationship
- Can be client of an OwnsProcess dependency to an OperationalActivity
- Can be supplier of a FillsPost dependency from an ActualPerson

ActualProject

A time-limited attempt to create a specific set of products or services; used in AcV-1, AcV-2, StV-3, StV-5 and SV-8.

Extensions:

- Object

Constraints:

- Classifier must be a Project

Use:

- Press Ctrl and drag a Project from the Browser window to create an instance, or drop an ActualProject from the Diagram Toolbox and press Ctrl+L to set the classifier
- Can have Aggregations to or from another ActualProject
- Can have a set of 'ownedMilestones' (type ActualProjectMilestone, including IncrementMilestone,

OutOfServiceMilestone, NoLongerUsedMilestone and DeployedMilestone)

ActualProjectMilestone

An event in a project by which progress is measured; used in AcV-2, StV-3, StV-5 and SV-8.

See also: IncrementMilestone, OutOfServiceMilestone, NoLongerUsedMilestone and DeployedMilestone.

Extensions:

- Object

Specializations:

- IncrementMilestone
- OutOfServiceMilestone
- NoLongerUsedMilestone
- DeployedMilestone

Constraints:

- Classifier must be a ProjectMilestone

Use:

- Can have a set of associated Resource
- Can be client/supplier of a MilestoneSequence

Alias

A comment used to define an alternative name for an element; used in AV-2.

Extensions:

- Note

Constraints:

- AnnotatedElement must be a UPDMElement

Use:

- Just drag a Quicklink NoteLink from the Alias to the annotated element

Arbitrary Relationship

Represents a visual indication of a connection used in high level operational concept diagrams. The connections are purely visual and cannot be related to any architectural semantics; used in OV-1.

Extensions:

- Dependency

Constraints:

- Client and Supplier must both be stereotyped ConceptRole

Use:

- Drag a Quicklink from a ConceptRole

ArchitecturalDescription

A specification of a system of systems at a technical level, which also provides the business context; used in AV-1.

Extensions:

- Package

Use:

- Can have a DefinesArchitecture Realization to an EnterprisePhase
- Can have an ArchitecturalReference Dependency to another ArchitecturalDescription
- Can be annotated with an ArchitectureMetadata note

ArchitecturalReference

Asserts that one architectural description refers to another; used in AV-1.

Extensions:

- Dependency

Constraints:

- Client and Supplier must both be stereotyped ArchitecturalDescription

Use:

- Drag a Quicklink from an ArchitecturalDescription.

ArchitectureMetadata

Information on architectural description; used in AV-1.

Extensions:

- Note

Generalizations:

- Metadata

Constraints:

- AnnotatedElement must be an ArchitecturalDescription

Use:

- Drag a quicklink from an ArchitecturalDescription

Capability

A high-level specification of the enterprise's ability; used in AV-1, OV-2, SOV-3, StV-1, StV-2, StV-3, StV-4, StV-5, StV-6, SV-1 and SV-9.

Extensions:

- Class

Generalizations:

- SubjectOfForecast

Use:

- Can have a set of associated environment conditions stereotyped Environment
- Capabilities can be composed of Capabilities (Composite aggregation)
- Capabilities can be dependent on Capabilities (Dependency)
- Capabilities can sub-class Capabilities (Generalization)
- Can be supplier or client of a Forecast (both must be same stereotype) (from SubjectOfForecast)

CapabilityConfiguration

A set of physical and human resources (and their interactions) configured to provide a capability; used in OV-1, OV-2, OV-3, StV-3, StV-5, SV-1, SV-3, SV-9, SV-10a, SV-12, TV-1, TV-2 and AcV-2.

Extensions:

- Class

Generalizations:

- Resource
- ConceptItem
- Performer
- ResourceInteractionItem
- SubjectOfResourceConstraint
- SubjectOfForecast
- SystemsElement
- SubjectOfResourceStateMachine
- ResourceInteractionItem

Specializations:

- SystemsNode

Use: Can:

- Have a set of associated deployed milestones, stereotyped DeployedMilestone
- Have an optional associated no longer used milestone, stereotyped NoLongerUsedMilestone
- Have a set of associated increment milestones, stereotyped IncrementMilestone
- Have an optional associated out of service milestone, stereotyped OutOfServiceMilestone
- Be annotated by a StandardConfiguration note
- Be the type of a ConceptRole (from ConceptItem)
- Have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource)
- Be the client of a RealizesCapability Realization to a Capability (from Resource)
- Be the client of a ProvidesCompetence Dependency to a Competence (from Resource)
- Have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint)
- Be the supplier or client of a Forecast Dependency (both must have same stereotype) (from SubjectOfForecast)
- Own a ServicePoint (from Resource)
- Own a RequestPoint (from Resource)
- Own a ResourcePort (from Resource)
- Be the source and target of a ResourceInteraction (from Resource)
- Own a ServiceOperation (from Resource)
- Be the type of a KnownResource (from Resource)

- Be the type of a ResourceRole (from Resource)
- Have a Performs Dependency to a PerformedActivity (Function or OperationalActivity) (from Performer)

Climate

A type of weather condition, or combination of weather conditions, in which a Performer performs; used in StV-2.

Extensions:

- Class

Generalizations:

- EnvironmentalType

Use:

- Can be the type of an EnvironmentProperty

Commands

Asserts that one OrganizationalResource commands another; used in OV-4, SV-1 and SV-10c.

Extensions:

- InformationFlow

Generalizations:

- ResourceInteraction

Constraints:

- Source must be an OrganizationalResource
- Target must be an OrganizationalResource

Use:

- Conveys a DataElement

CompatibleWith

Relates a node to a location to assert that the operational node must be situated at that location; used in OV-2.

Extensions:

- Dependency

Constraints:

- Client is a Node
- Supplier is a ReferredLocation (Location or PhysicalLocation)

Use:

- Drag a Quicklink from a Node

Competence

A specific set of abilities defined by knowledge, skills and attitude; used in OV-4, SV-1 and SV-9.

Extensions:

- Class

Generalizations:

- SubjectOfForecast

Use: Can be:

- The supplier or client of a Forecast Dependency (both must have same stereotype) (from SubjectOfForecast)
- The supplier of a ProvidesCompetence Dependency
- The supplier of a RequiresCompetence Dependency

ConceptRole

A relationship that asserts that a ConceptItem forms part of the high level operational concept; used in OV-1.

Extensions:

- Part

Constraints:

- Type is a ConceptItem

Use:

- Owned by a HighLevelOperationalConcept
- Can be supplier and client of an ArbitraryRelationship dependency

ConfigurationExchange

CapabilityConfigurations that are exchanged between Nodes; used in OV-2, OV-3 and OV-6c.

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Constraints:

- Source is a Node (from OperationalExchange)
- Target is a Node (from OperationalExchange)

Use:

- Conveys a CapabilityConfiguration

Controls

A type of ResourceInteraction where one Resource controls another; used in SV-1 and SV-10c.

Extensions:

- InformationFlow

Generalizations:

- ResourceInteraction

Constraints:

- Source is an OrganizationalResource (Organization or Post)
- Target is a ManufacturedResourceType (ResourceArtifact or Software)

Use:

- Conveys a DataElement

DataElement

A formalized representation of data that is managed by or exchanged between systems; used in OV-4, SV-1, SV-2, SV-4, SV-6, SV-10a, SV-10b and SV-11.

Extensions:

- Class

Generalizations:

- SubjectOfResourceConstraint
- ResourceInteractionItem
- SystemsElement
- SubjectOfResourceStateMachine

Use:

- Can have an attached ResourceConstraint (from SubjectOfResourceConstraint)
- Can have a set of associated defined EntityItems
- Can be conveyed on a Controls or Commands information flow

DataExchange

A DoDAF alias for ResourceInteraction.

Extensions:

- InformationFlow

Generalizations:

- ResourceInteraction
- SystemsElement

Use:

- Conveys ResourceInteractionItem (Energy, Post, Organization, CapabilityConfiguration, Software, ResourceArtifact, or DataElement)

DefinesArchitecture

Establishes a relationship between ArchitecturalDescription and EnterprisePhase; used in AV-1.

Extensions:

- Realization

Constraints:

- Client is an ArchitecturalDescription

- Supplier is an EnterprisePhase

Use:

- Drag a Quicklink from an ArchitecturalDescription

Definition

A definition of an element in the architecture; used in AV-2.

Extensions:

- Note

Constraints:

- Annotated Element is a UPDMElement

Use:

- Drop from toolbox and drag a NoteLink to any UPDM element

DeployedMilestone

Asserts that an ActualOrganizationResource started to use, or is slated to start using, a CapabilityConfiguration from a specific point in time; used in StV-5.

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone (from ActualProjectMilestone)

Use: Can:

- Have a set of associated (usedBy) ActualOrganizationalResource (ActualOrganization or ActualPost)
- Have a set of associated Resource (from ActualProjectMilestone)
- Be client/supplier of a MilestoneSequence (from ActualProjectMilestone)

EnduringTask

A type of behavior recognized by an enterprise as being essential to achieving its goals - that is, a strategic specification of what the enterprise does; used in StV-1.

Extensions:

- Class

Use:

- Target of association from EnterprisePhase

Energy

Energy to be exchanged between Nodes; used in OV-2, OV-3, OV-5, SV-1, SV-4 and SV-6.

Extensions:

- Class

Generalizations:

- ResourceInteractionItem
- OperationalExchangeItem

Use:

- Conveyed on an EnergyExchange information flow

EnergyExchange

A relationship specifying the need to exchange energy between nodes; used in OV-2, OV-3 and OV-6c.

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange
- OperationalElement

Constraints:

- Source is a Node (from OperationalExchange)
- Target is a Node (from OperationalExchange)

Use:

- Conveys a Class stereotyped Energy

EnterpriseGoal

A specific required objective of the enterprise that the architecture represents; used in StV-1.

Extensions:

- Class

Use:

- Has an association to one EnterprisePhase

EnterprisePhase

A specific, required objective of the enterprise that the architecture represents; used in AV-1, StV-1, StV-2, StV-5 and SV-9.

Extensions:

- Class

Specializations:

- WholeLifeEnterprise

Use:

- Can have a set of associations (statementTasks) to EnduringTask Class

- Can have a set of associations (exhibits) to Capability Class
- Can have a set of associations (inhabits) to Environment Class
- Can have a set of associations (goals) with EnterpriseGoal Class
- Can have a set of associations (visions) with EnterpriseVision Class
- Can be the type of a StructuralPart or TemporalPart
- Fulfills a Mission Use Case
- Can be Supplier of a DefinesArchitecture Realization

EnterpriseVision

The overall aims of an enterprise over a given period of time; used in StV-1.

Extensions:

- Class

Use:

- Has an association to one EnterprisePhase

EntityAttribute

A defined property of an EntityItem; used in OV-7 and SV-11.

Extensions:

- Attribute

Use:

- Is owned by an EntityItem

EntityItem

A definition (type) of an item of interest; used in OV-7 and SV-11.

Extensions:

- Class

Constraints:

- Owned attributes must be stereotyped EntityAttribute

Use: Can:

- Be owned by a DataModel
- Be the end type of an EntityRelationship
- Have a set of associated (definedBy) DataElement
- Have a set of associated (represents) InformationElement
- Be conveyed on a Commands or Controls information flow

EntityRelationship

Asserts that there is a relationship between two EntityItems; used in OV-7 and SV-11.

Extensions:

- Association

Constraints:

- The types of any object at either end must be stereotyped EntityItem

Environment

A definition of the conditions in which the Enterprise exists or functions; used in AV-1 and StV-2.

Extensions:

- Class

Constraints:

- Owned attributes must be EnvironmentProperty

EnvironmentProperty

Asserts that an Environment has one or more properties such as Climate, Location or LightCondition; used in StV-2.

Extensions:

- Attribute

Constraints:

- Type must be an EnvironmentalType (LightCondition, Location, PhysicalLocation or Climate)

Use:

- Owned by an Environment element

Equipment

A physical resource that is used to accomplish a task or function in a system or an environment; used in SV-1.

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

- Class must be an OrganizationResource (Organization or Post)
- Type must be a ResourceArtifact

Use:

- Can have a RequiresCompetence dependency to a Competence (from ResourceRole)
- Can have a set of associations (usedFunctions) to Function (from ResourceRole)

ExhibitsCapability

Assertion that a Node is required to have a Capability; used in OV-2.

Extensions:

- Dependency

Constraints:

- Client must be a Node
- Supplier must be a Capability

Expose

Assertion that a service interface exposes a capability.

Extensions:

- Dependency

Constraints:

- Client must be a ServiceInterface
- Supplier must be a Capability

ExternalIndividual

An individual defined by an external ontology; used in AV-2.

Extensions:

- Object

Use:

- Can be the supplier of a SameAs dependency

ExternalNode

Operational node that is external to the architecture; used in OV-2.

Extensions:

- Class

Generalizations:

- Node
- Performer

Use: Can:

- Own a RequestPoint Port (from Node)
- Own a ServicePoint Port (from Node)
- Be client of an ExhibitsCapability dependency to a Capability (from Node)
- Have a Performs dependency to a PerformedActivity (Function or OperationalActivity) (from Performer)
- Have a CompatibleWith dependency to a ReferredLocation (PhysicalLocation or Location) (from Node)

ExternalType

A type defined by an external ontology; used in AV-2.

Extensions:

- Class

Use:

- Can be the Supplier of a SameAs dependency
- Any UPDM element can have a Generalization to an ExternalType

FieldedCapability

A deployed and fully realized instance of a capability; used in SV-2.

Extensions:

- Object

Constraints:

- Its classifier must be a CapabilityConfiguration

FillsPost

Asserts that ActualPerson fills an ActualPost; used in OV-4.

Extensions:

- Dependency

Constraints:

- Client must be an ActualPerson
- Supplier must be an ActualPost

Forecast

The actual or predicted status of a system at a project milestone; used in SV-9.

Extensions:

- Dependency

Specializations:

- TechnologyForecast

Constraints:

- Client and Supplier are both SubjectOfForecast (Standard, Competence, Capability, CapabilityConfiguration, Organization, Post, ResourceArtifact or Software)
- Client and Supplier must be the same specialization of SubjectOfForecast

Function

An activity that is specified in context of the resource that performs it; used in OV-4, SV-1, SV-4, SV-5 and SV-10a.

Extensions:

- Activity

Generalizations:

- PerformedActivity
- SystemsElement
- SubjectOfResourceConstraint

Constraints:

- Owned parameters are FunctionParameter

Use: Can:

- Be Supplier of a Performs dependency (from PerformedActivity)
- Own ServiceOperationAction, FunctionAction and FunctionEdge
- Be Client of an ImplementsOperational dependency to an OperationalActivity (from SystemsElement)
- Have an attached ResourceConstraint (from SubjectOfResourceConstraint)

FunctionAction

A call behavior action that invokes the function that needs to be performed; used in SV-4.

Extensions:

- Action (Call Behavior)

Specializations:

- SystemFunctionAction

Constraints:

- Activity is stereotyped Function

Use:

- Ctrl+L to set the function

FunctionEdge

Models the flow of control/objects through a function; used in SV-4.

Extensions:

- ControlFlow

Generalizations:

- SystemsElement

Specializations:

- SystemFunctionEdge

Constraints:

- Source must be a ServiceOperationAction
- Target must be a ServiceOperationAction

Use:

- Can realize a ResourceInteraction (Right-click | Advanced | Information Flows Realized)

FunctionParameter

Represents inputs and outputs of a Function; used in SV-4.

Extensions:

- ActivityParameter

Constraints:

- Type must be a ResourceInteractionItem (Energy, DataElement, CapabilityConfiguration, Organization, Post, ResourceArtifact or Software)

Use:

- Owned by a Function

HighLevelOperationalConcept

A generalized model for operations; used in OV-1.

Extensions:

- Class

Constraints:

- Owned attributes are ConceptRole

Use:

- Can have a set of described Mission

HostedSoftware

Asserts that software is hosted on a ResourceArtifact; used in SV-1.

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

- Owning Class must be a ResourceArtifact
- Type must be a Software

Use: Can:

- Have a RequiresCompetence dependency to a Competence (from ResourceRole)
- Have a set of associations to 'used' Functions (from ResourceRole)

HumanResource

The role of a Post or Organization in a CapabilityConfiguration; used in SV-1.

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

- Owning Class must be a CapabilityConfiguration
- Type must be an OrganizationalResource (Organization or Post)

Use: Can:

- Have a RequiresCompetence dependency to a Competence (from ResourceRole)
- Have a set of associations to 'used' Functions (from ResourceRole)

ImplementsOperational

Relationship between a system element and the operational element that it implements; used in SV-5.

Extensions:

- Dependency

Constraints:

- Client must be a SystemsElement (Function)
- Supplier must be an OperationalElement (OperationalActivity)

IncrementMilestone

An ActualProjectMilestone that indicates the point in time at which a project is predicted to deliver or has delivered a Capability; used in AcV-2, StV-3 and SV-8.

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone (from ActualProjectMilestone)

Use:

- Can be the supplier or client of a MilestoneSequence dependency (from ActualProjectMilestone)
- Can have a set of associated Resource (from ActualProjectMilestone)
- Has a set of associations with CapabilityConfiguration

InformationElement

Information exchanged between nodes; used in OV-2, OV-3, OV-5, OV-6a, OV-6b and OV-7.

Extensions:

- Class

Generalizations:

- OperationalExchangeItem

- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine
- OperationalElement

Use: Can:

- Have a set of associations with (represented by) EntityItem Classes
- Be conveyed on an InformationExchange - right-click > Advanced > Information Items Conveyed
- Have an attached OperationalConstraint (from SubjectOfOperationalConstraint)
- Own an OperationalStateMachine (from SubjectOfOperationalStateMachine)

InformationExchange

A relationship specifying the need to exchange information between nodes; used in OV-2, OV-3 and OV-6c.

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Constraints:

- Conveys an InformationElement
- Source is a Node (from OperationalExchange)
- Target is a Node (from OperationalExchange)

InternalDataModel

DoDAF alias for PhysicalDataModel; used in SV-11.

Extensions:

- Package

Generalizations:

- PhysicalDataModel
- DataModel

Constraints:

- Owns EntityItem elements (from DataModel)

KnownResource

Asserts that a known resource plays a part in the architecture; used in OV-2.

Extensions:

- Part

Generalizations:

- NodeChild

Constraints:

- Type must be a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software or ResourceArtifact)
- Class must be a NodeParent (Node or LogicalArchitecture) (from NodeChild)

LightCondition

A specification of environmental lighting conditions; used in StV-2.

Extensions:

- Class

Generalizations:

- EnvironmentalType

Use:

- Can be the type of an EnvironmentProperty (from EnvironmentalType)

Location

A general specification of the surroundings/scenario in which an operation might take place. Examples include 'desert', 'arctic', 'at sea'; used in OV-1 and OV-2.

Extensions:

- Class

Generalizations:

- ReferredLocation
- ConceptItem
- EnvironmentalType

Use: Can be:

- Supplier to a CompatibleWith dependency from a Node (from ReferredLocation)
- Type of a ConceptRole (from ConceptItem)
- The type of an EnvironmentProperty (from EnvironmentalType)

LogicalArchitecture

A composite structure model whose parts are either NodeRoles, ProblemDomains, or KnownResources; used in OV-2.

Extensions:

- Class

Generalizations:

- NodeParent

Use:

- Can own ProblemDomain properties

LogicalDataModel

A specification of business information requirements as a formal data structure; used in OV-7.

Extensions:

- Package

Generalizations:

- DataModel

Constraints:

- Owns EntityItem elements (from DataModel)

MapsToCapability

Asserts that a StandardOperationalActivity is in some way part of a capability; used in StV-6.

Extensions:

- Dependency

Constraints:

- Client must be a StandardOperationalActivity
- Supplier must be a Capability

MaterielExchange

Materiel that is exchanged between Nodes; used in OV-2, OV-3 and OV-6c.

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Constraints:

- Source is a Node (from OperationalExchange)
- Target is a Node (from OperationalExchange)

Use:

- Can convey a ResourceArtifact or Software

Measurement

A category of measures; used in AV-3, OV-2 and SV-7.

Extensions:

- Attribute

Specializations:

- PerformanceParameter

Use:

- Owned by a MeasurementSet Class

MeasurementSet

A set or collection of Measurements; used in AV-3, OV-3 and SV-7.

Extensions:

- Class

Constraints:

- Owned attributes must be Measurement

Use:

- Has a set of associations with (measuredElement) UPDMElement
- Is classifier of ActualMeasurementSet object

Metadata

Annotation that can be applied to any element in the architecture; used in AV-1.

Extensions:

- Note

Specializations:

- ArchitectureMetadata

MilestoneSequence

A relationship between two milestones; used in AcV-2 and SV-8.

Extensions:

- Dependency

Constraints:

- Client must be an ActualProjectMilestone
- Supplier must be an ActualProjectMilestone

Mission

A purpose to which a person, organization, or autonomous system is tasked; used in AV-1, OV-1, OV-6a and OV-6b.

Extensions:

- UseCase

Generalizations:

- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine

Use:

- Fulfilled by an EnterprisePhase
- Can have an attached OperationalConstraint (from SubjectOfOperationalConstraint)

- Can own an OperationalStateMachine (from SubjectOfOperationalStateMachine)

MovementOfPeople

MODAF alias for OrganizationalExchange.

Extensions:

- InformationFlow

Generalizations:

- OrganizationalExchange

Use:

- Conveys an OrganizationalResource (Organization or Post)

Needline

Documents the requirement to exchange information between nodes; used in OV-2 and OV-3.

Extensions:

- Association
- Connector

Generalizations:

- OperationalElement

Constraints:

- End Types must be Node
- End Roles must be NodePort
- End Roles must be NodeChild (NodeRole, ProblemDomain, KnownResource)

Use:

- Realizes an OperationalExchange - create a Needline between the same two elements as an OperationalExchange, then right-click on the Needline and select 'Advanced > Information Flows Realized'

NoLongerUsedMilestone

Asserts that an ActualOrganizationResource ceased to use - or is slated to cease using - a CapabilityConfiguration from a specific point in time; used in StV-5.

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone (from ActualProjectMilestone)

Use:

- Has set of associations to 'noLongerUsedBy' ActualOrganizationalResource (ActualOrganization or ActualPost) objects

- Can have a set of associated Resource (from ActualProjectMilestone)
- Can be client/supplier of a MilestoneSequence (from ActualProjectMilestone)
- Has a set of associations with 'configuration' CapabilityConfiguration Classes

Node

Logical entity that performs operational activities; used in OV-1, OV-2, OV-3, OV-5, OV-6a, OV-6b and OV-6c.

Extensions:

- Class

Generalizations:

- Performer
- ConceptItem
- NodeParent
- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine
- OperationalElement

Specializations:

- OperationalNode

Constraints:

- Owned ports must be NodePort, RequestPoint or ServicePoint

Use: Can:

- Have a Performs dependency to a PerformedActivity (Function or OperationalActivity) (from Performer)
- Be the Client of a CompatibleWith dependency to a ReferredLocation (Location or PhysicalLocation)
- Be the type of a ConceptRole (from ConceptItem)
- Own a RequestPoint port
- Own a ServicePoint port
- Be client of an ExhibitsCapability dependency to a Capability
- Own NodeChild (NodeRole, KnownResource, ProblemDomain) (from NodeParent)
- Be source and target of an OperationalExchange (ConfigurationExchange, EnergyExchange, InformationExchange, MaterielExchange or OrganizationalExchange) information flow
- Be the end type of a Needline association
- Have an attached OperationalConstraint (from SubjectOfOperationalConstraint)
- Own an OperationalStateMachine (from SubjectOfOperationalStateMachine)
- Be the type of a NodeRole
- Own ServiceOperations

NodePort

A property of a Node that specifies a distinct interaction point between the node and its environment or between the node and its internal parts.

Extensions:

- Port

Constraints:

- Type must be an OperationalExchangeItem (Post, Organization, ResourceArtifact or System)

Use:

- Owned by a Node
- Can be the ends of a Needline

NodeRole

Used to link a parent Node to its sub-nodes; used in OV-2, OV-3 and OV-6c.

Extensions:

- Part

Generalizations:

- NodeChild

Specializations:

- ProblemDomain

Constraints:

- Class must be a Node
- Type must be a Node

OperationalActivity

A logical process, specified independently of how the process is carried out; used in OV-2, OV-3, OV-4, OV-5, OV-6a, OV-6b and SV-5.

Extensions:

- Activity

Generalizations:

- PerformedActivity
- SubjectOfOperationalConstraint
- OperationalElement
- SubjectOfOperationalStateMachine

Specializations:

- StandardOperationalActivity

Constraints:

- Owned parameters must be OperationalParameter

Use: Can:

- Be Supplier of a Performs dependency (from PerformedActivity)
- Be Supplier of an OwnsProcess dependency
- Be the Activity/Behavior of an OperationalActivityAction
- Be the owner of an OperationalActivityEdge
- Have an attached OperationalConstraint (from SubjectOfOperationalConstraint)

- Be the Supplier of a SupportsOperationalActivity dependency
- Own an OperationalStateMachine (from SubjectOfOperationalStateMachine)

OperationalActivityAction

A call behavior action that invokes the activity to be performed; used in OV-5.

Extensions:

- CallBehaviorAction

Constraints:

- Activity/Behavior must be an OperationalActivity

Use:

- Can be the Source or Target of an OperationalActivityEdge

OperationalActivityEdge

Models the flow of control/objects through an OperationalActivity; used in OV-5.

Extensions:

- ControlFlow

Generalizations:

- OperationalElement

Constraints:

- Must be owned by an OperationalActivity
- Source must be an OperationalActivityAction
- Target must be an OperationalActivityAction

Use: Can:

- Have a set of OperationalExchange (ConfigurationExchange, EnergyExchange, InformationExchange, MaterielExchange or OrganizationalExchange) information flows that it realizes
- Carry a set of OperationalExchangeItem (Post, Organization, ResourceArtifact or System)

OperationalConstraint

A rule governing an operational behavior or property; used in OV-6a.

Extensions:

- Constraint

Specializations:

- OperationalRule

Constraints:

- Constrained element must be a SubjectOfOperationalConstraint (OperationalActivity, Node, InformationElement or Mission)

OperationalMessage

Message for use in an Operational Event Trace, which carries any of the subtypes of OperationalExchange; used in OV-6c.

Extensions:

- Message

Generalizations:

- OperationalElement

Use:

- Can have a set of OperationalExchange (ConfigurationExchange, EnergyExchange, InformationExchange, MaterialExchange or OrganizationalExchange) information flows that it realizes

OperationalNode

An element of the operational architecture that produces, consumes, or processes information.

Extensions:

- Class

Generalizations:

- Node

Constraints:

- Owned ports must be NodePort, RequestPoint or ServicePoint

Use: Can:

- Have a Performs dependency to a PerformedActivity (Function, OperationalActivity) (from Performer)
- Be the Client of a CompatibleWith dependency to a ReferredLocation (Location or PhysicalLocation)
- Be the type of a ConceptRole (from ConceptItem)
- Own a RequestPoint port
- Own a ServicePoint port
- Be client of an ExhibitsCapability dependency to a Capability
- Own NodeChild (NodeRole, KnownResource, ProblemDomain) (from NodeParent)
- Be source and target of an OperationalExchange (ConfigurationExchange, EnergyExchange, InformationExchange, MaterielExchange or OrganizationalExchange) information flow
- Be the end type of a Needline association
- Have an attached OperationalConstraint (from SubjectOfOperationalConstraint)
- Own an OperationalStateMachine (from SubjectOfOperationalStateMachine)
- Be type of a NodeRole
- Own ServiceOperations

OperationalParameter

Represents inputs and outputs of an operational activity; used in OV-5.

Extensions:

- ActivityParameter

Constraints:

- Type must be an OperationalExchangeItem (Post, Organization, ResourceArtifact or System)

Use:

- Can be owned by an OperationalActivity

OperationalRule

A DoDAF alias for OperationalConstraint.

Extensions:

- Constraint

Generalizations:

- OperationalConstraint

Constraints:

- Constrained element must be a SubjectOfOperationalConstraint (OperationalActivity, Node, InformationElement or Mission) (from OperationalConstraint)

OperationalStateMachine

A StateMachine describing an operational behavior or property; used in OV-6b.

Extensions:

- StateMachine

Constraints:

- Owner is SubjectOfOperationalStateMachine (Mission, InformationElement or Node)

Organization

A group of persons, associated for a particular purpose; used in OV-4, SV-1, SV-3, SV-9, SV-10a and SV-12.

Extensions:

- Class

Generalizations:

- OrganizationalResource
- Resource, Performer
- SubjectOfForecast
- SubjectOfResourceConstraint

Use: Can:

- Be classifier to an ActualOrganization
- Be source or target of a Commands information flow (from OrganizationalResource)
- Be the owning Class of a PostRole
- Be the Class or type of a SubOrganization
- Be the Class of an Equipment (from OrganizationalResource)
- Be conveyed by an OrganizationalExchange (from OrganizationalResource)

- Be the type of a HumanResource (from OrganizationalResource)
- Be the source of a Controls information flow (from OrganizationalResource)
- Have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource)
- Be the client of a RealizesCapability realization to a Capability (from Resource)
- Be the client of a ProvidesCompetence dependency to a Competence (from Resource)
- Have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint)
- Be supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast)
- Own a ServicePoint (from Resource)
- Own a RequestPoint (from Resource)
- Own a ResourcePort (from Resource)
- Be source and target of a ResourceInteraction (from Resource)
- Own a ServiceOperation (from Resource)
- Be type of a KnownResource (from Resource)
- Be type of a ResourceRole (from Resource)
- Have a Performs dependency to a PerformedActivity (Function or OperationalActivity) (from Performer)

OrganizationalExchange

A relationship specifying flow of people across organizations; used in OV-2, OV-3 and OV-6c.

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Specializations:

- MovementOfPeople

Constraints:

- Conveyed element must be an OrganizationalResource (Organization or Post)
- Source is a Node (from OperationalExchange)
- Target is a Node (from OperationalExchange)

OutOfServiceMilestone

A project milestone that indicates a project's deliverable is to go out of service; used in AcV-2, StV-3 and SV-8.

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone

Use:

- Has a set of association ('configuration') with CapabilityConfiguration

- Can have a set of associated Resource (from ActualProjectMilestone)
- Can be client/supplier of a MilestoneSequence (from ActualProjectMilestone)

OwnsProcess

A relationship that asserts that an ActualOrganizationalResource has responsibility for an OperationalActivity; used in OV-4.

Extensions:

- Dependency

Constraints:

- Client must be an ActualOrganizationalResource (ActualPost or ActualOrganization)
- Supplier must be an OperationalActivity

Part

Use of a ResourceArtifact as a part of another ResourceArtifact; used in SV-1.

Extensions:

- Part

Generalizations:

- ResourceRole

Specializations:

- SubSystemPart

Constraints:

- Class must be a ResourceArtifact
- Type must be a ResourceArtifact

Use: Can have:

- A RequiresCompetence dependency to a Competence (from ResourceRole)
- A set of associations to 'used' Functions (from ResourceRole)

PerformanceParameter

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

Extensions:

- Attribute

Generalizations:

- Measurement

Use:

- Owned by a MeasurementSet class

Performs

Links a Performer to the behavior that it can perform; used in OV-2, OV-3, OV-4, OV-5, SV-1 and SV-4.

Extensions:

- Dependency

Constraints:

- Client must be a Performer (Node, ExternalNode, OperationalNode, Post, Organization, CapabilityConfiguration, SystemsNode, Software or ResourceArtifact)
- Supplier must be a PerformedActivity (OperationalActivity or Function)

Person

A type of human being; used in OV-4.

Extensions:

- Class

Use:

- Can be Classifier of an ActualPerson

PhysicalDataModel

An implementable specification of a data structure; used in SV-11.

Extensions:

- Package

Generalizations:

- DataModel

Specializations:

- InternalDataModel

Constraints:

- Owns EntityItem elements (from DataModel)

PhysicalLocation

Anywhere that can be specified; used in OV-1 and OV-2.

Extensions:

- Class

Generalizations:

- ReferredLocation
- ConceptItem
- EnvironmentalType

Use: Can be:

- Supplier to a CompatibleWith dependency from a Node (from ReferredLocation)
- Type of a ConceptRole (from ConceptItem)

- The type of an EnvironmentProperty (from EnvironmentalType)

Platform

Use of an artifact as a platform in a particular ResourceConfiguration; used in SV-1.

Extensions:

- Part

Generalizations:

- ResourceComponent
- ResourceRole

Constraints:

- Class must be a CapabilityConfiguration
- Type must be a ResourceArtifact

Use:

- Can have a RequiresCompetence dependency to a Competence (from ResourceRole)
- Can have a set of associations to 'used' Functions (from ResourceRole)

Post

A type of point of contact or responsible person; used in OV-4, SV-1, SV-3, SV-9, SV-10a and SV-12.

Extensions:

- Class

Generalizations:

- OrganizationalResource
- Resource
- Performer
- SubjectOfForecast
- SubjectOfResourceConstraint

Use: Can:

- Be Classifier of an ActualPost
- Be the Type of a PostRole
- Be source or target of a Commands information flow (from OrganizationalResource)
- Be the Class of an Equipment (from OrganizationalResource)
- Be conveyed by an OrganizationalExchange (from OrganizationalResource)
- Be the type of a HumanResource (from OrganizationalResource)
- Be the source of a Controls information flow (from OrganizationalResource)
- Have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource)
- Be client of a RealizesCapability realization to a Capability (from Resource)
- Be client of a ProvidesCompetence dependency to a Competence (from Resource)
- Have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint)
- Be supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast)

- Own a ServicePoint (from Resource)
- Own a RequestPoint (from Resource)
- Own a ResourcePort (from Resource)
- Be source and target of a ResourceInteraction (from Resource)
- Own a ServiceOperation (from Resource)
- Be type of a KnownResource (from Resource)
- Be type of a ResourceRole (from Resource)
- Have a Performs dependency to a PerformedActivity (Function, OperationalActivity) (from Performer)

PostRole

Asserts that a post exists in an organization; used in OV-4 and SV-1.

Extensions:

- Part

Generalizations:

- OrganizationRole
- ResourceRole

Constraints:

- Class must be an Organization
- Type must be a Post

Use: Can have a:

- RequiresCompetence dependency to a Competence (from ResourceRole)
- Set of associations to 'used' Functions (from ResourceRole)

ProblemDomain

The boundary containing those Nodes that can be realized by functional resources; used in OV-2.

Extensions:

- Part

Generalizations:

- NodeRole
- NodeChild

Constraints:

- Class must be a LogicalArchitecture
- Type must be a Node (from NodeRole)

Project

Used to define a category of project; used in AcV-1.

Extensions:

- Class

Use: Can:

- Be classifier of an ActualProject
- Have an association to a ProjectMilestone Class

ProjectMilestone

A type of project milestone; used in AcV-2.

Extensions:

- Class

Constraints:

- Owned attributes must be ProjectTheme

Use: Can:

- Be classifier of an ActualProjectMilestone
- Have an association from a Project Class

ProjectSequence

Asserts that one ActualProject follows on from another; used in AcV-2.

Extensions:

- Dependency

Constraints:

- Client must be an ActualProject
- Supplier must be an ActualProject

ProjectTheme

An aspect by which the progress of various projects can be measured; used in AcV-2.

Extensions:

- Attribute

Constraints:

- Type must be a ProjectThemeStatus

Use:

- Owned by ProjectMilestone

ProjectThemeStatus

Specifies a status for a ProjectTheme.

Extensions:

- Class

Use:

- The type of a ProjectTheme

Protocol

A standard for communication; used in SV-2, TV-1 and TV-2.

Extensions:

- Class

Generalizations:

- Standard
- SubjectOfForecast

Use: Can:

- Have a set of associations with ('ratifiedBy') ActualOrganization objects (from Standard)
- Have ProtocolLayers
- Be the type of ProtocolLayers
- Be the client and supplier of a Forecast dependency

ProtocolLayer

Asserts that a protocol uses another protocol; used in TV-1 and TV-2.

Extensions:

- Attribute

Constraints:

- Owning Class must be a Protocol
- Type must be a Protocol

ProvidesCompetence

Asserts that a resource provides a competence; used in OV-4.

Extensions:

- Dependency

Constraints:

- Client must be a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software or ResourceArtifact)
- Supplier must be a Competence

RealizesCapability

Asserts that a resource provides a capability; used in SOV-3, StV-3, StV-5 and SV-1.

Extensions:

- Realization

Constraints:

- Client must be a Resource or a ServiceInterface
- Supplier must be a Capability

RequestPoint

The mechanism by which a Service communicates; used in OV-2 and SV-1.

Extensions:

- Port

Constraints:

- Type must be a ServiceInterface

Use:

- Can be owned by a Node or a Resource

RequiresCompetence

Asserts that a role requires a competence; used in SV-1.

Extensions:

- Dependency

Constraints:

- Client must be a ResourceRole
- Supplier must be a Competence

ResourceArtifact

A type of man-made object; used in OV-2, OV-3, OV-5, SV-1, SV-3, SV-9, SV-10a and SV-12.

Extensions:

- Class

Generalizations:

- OperationalExchangeItem
- ManfacturedResourceType
- Resource
- SubjectOfForecast
- ResourceInteractionItem
- Performer
- SubjectOfResourceConstraint

Specializations:

- System

Use: Can:

- Be conveyed by a MaterielExchange
- Be the type of an OperationalParameter (from OperationalExchangeItem)
- Own HostedSoftware
- Be the Class and type of a Part
- Be the type of a ResourceComponent
- Be the type of an Equipment
- Be the target of a Controls flow (from ManfacturedResourceType)
- Have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource)
- Be client of a RealizesCapability realization to a Capability (from Resource)
- Be client of a ProvidesCompetence dependency to a Competence (from Resource)
- Have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint)
- Be supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast)
- Own a ServicePoint (from Resource)
- Own a RequestPoint (from Resource)
- Own a ResourcePort (from Resource)
- Be source and target of a ResourceInteraction (from Resource)
- Own a ServiceOperation (from Resource)
- Be type of a KnownResource (from Resource)
- Be type of a ResourceRole (from Resource)
- Have a Performs dependency to a PerformedActivity (Function or OperationalActivity) (from Performer)

ResourceComponent

A well-defined resource that is used by a CapabilityConfiguration to accomplish a capability; used in SV-1.

Extensions:

- Part

Generalizations:

- ResourceRole

Specializations:

- Platform

Constraints:

- Type must be a ResourceArtifact
- Owning Class must be a CapabilityConfiguration

Use: Can have a:

- RequiresCompetence dependency to a Competence (from ResourceRole)
- Set of associations to 'used' Functions (from ResourceRole)

ResourceConnector

A physical connection between two resources that implements protocols through which the source resource can transmit items to the destination resource; used in SV-2.

Extensions:

- Connector

Generalizations:

- ProtocolImplementation

Constraints:

- End roles must be ResourcePort

Use: Can:

- Have a set of ResourceInterface that it realizes
- Realize a ResourceInteraction

ResourceConstraint

Specifies the set of rules that govern the structural or functional aspects of the system; used in SV-10a.

Extensions:

- Constraint

Constraints:

- Constrained element must be a SubjectOfResourceConstraint (DataElement, Function, SystemFunction, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact, System, Post or Organization)

ResourceInteraction

Represents data that is exchanged between resources; used in OV-4, SOV-4c, SV-1, SV-2, SV-3, SV-4, SV-6 and SV-10c.

Extensions:

- InformationFlow

Generalizations:

- SystemsElement
- ProtocolImplementation

Specializations:

- Controls
- Commands
- DataExchange

Constraints:

- Realizing connector is a ResourceInterface
- Realizing activity edge is a FunctionEdge
- Conveyed elements must be ResourceInteractionItem (DataElement, Energy, Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact or System)
- Source must be a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact or System)
- Target must be a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact or System)

Use:

- Can realize an OperationalExchange (OrganizationalExchange, InformationExchange, EnergyExchange or MaterielExchange)
- Can realize an ActualOrganizationRelationship
- Has an association to ('implements') a Protocol (from ProtocolImplementation)

ResourceInterface

A contractual agreement between two resources that implement protocols; used in OV-4, SV-1, SV-2, SV-3 and SV-6.

Extensions:

- Association
- Connector

Generalizations:

- SystemsElement

Specializations:

- SystemConnector

Constraints:

- End roles must be ResourceRole
- End types must be Resource

Use:

- Can realize a ResourceInteraction

ResourceMessage

Message for use in a Resource event trace, implements a ResourceInteraction; used in SV-10c.

Extensions:

- Message

Generalizations:

- SystemsElement

Use:

- Can have a set of ResourceInteraction that it carries

ResourcePort

An interaction point for a resource through which it can interact with the outside environment; used in SV-2.

Extensions:

- Port

Generalizations:

- ProtocolImplementation

Constraints:

- Type must be a ResourceInteractionItem (Energy, Post, Organization, CapabilityConfiguration, Software, ResourceArtifact or DataElement)

Use:

- Can be owned by a Resource
- Has an association to a Protocol Class that it 'implements' (from ProtocolImplementation)
- Can be the end role of a ResourceConnector

ResourceStateMachine

UPDM artifact that extends a UML StateMachine applied to Resources; used in SV-10b.

Extensions:

- StateMachine

Generalizations:

- SystemsElement

Constraints:

- Owner must be SubjectOfResourceStateMachine (Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact, System or DataElement)

SameAs

Asserts that two elements refer to the same real-world thing; used in AV-2.

Extensions:

- Dependency

Constraints:

- Client must be a UPDMElement
- Supplier must be an ExternalIndividual or ExternalType

ServiceAttribute

A property of a ServiceInterface that allows performance, reliability and cost values to be captured; used in SOV-1.

Extensions:

- Attribute

Use:

- Owned by a ServiceInterface

ServiceFunction

Describes the abstract behavior of ServiceOperations, regardless of the actual implementation; used in SOV-5.

Extensions:

- Activity

Use: Can:

- Be the behavior of a ServiceFunctionAction

- Be the activity of a ServiceOperationAction
- Own ServicePoint ports

ServiceFunctionAction

A call behavior action that invokes the ServiceFunction to be performed; used in SOV-5.

Extensions:

- CallBehaviorAction

Constraints:

- Behavior must be a ServiceFunction

ServiceInteraction

Interaction for a service interface; used in SOV-4c.

Extensions:

- Interaction

ServiceInterface

A contractual agreement between two resources that implement protocols through which the source service interacts with the destination resource; used in SOV-1, SOV-2, SOV-3, SOV-4a, SOV-4b, SOV-4c and SOV-5.

Extensions:

- Class

Constraints:

- Owned attributes must be ServiceAttribute
- Owned operations must be ServiceOperation

Use: Can:

- Be client of a SupportsOperationalActivity dependency to an OperationalActivity
- Be client of a RealizesCapability realization to a Capability
- Own ServicePolicy
- Have one association to a ServiceStateMachine
- Have one association to a ServiceInteraction
- Be type of a RequestPoint or ServicePoint port
- Be dependent on another ServiceInterface
- Be client of an Expose dependency to a Capability

ServiceMessage

Message for use in a service interaction specification, implements a resource interaction; used in SOV-4c.

Extensions:

- Message

Use:

- Can carry a set of ResourceInteractions

ServiceOperation

Provides the access point for invoking the behavior of a provided service; used in SOV-2 and SOV-5.

Extensions:

- Operation

Constraints:

- Owner must be a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact or System)
- Owner must be a Node

Use: Can:

- Have an association to a (concreteBehavior) Function
- Be owned by a ServiceInterface
- Be the operation of a ServiceOperationAction
- Have an association to an (abstractBehavior) ServiceFunction

ServiceOperationAction

A call action that represents a Resource or ServiceFunction invoking a ServiceOperation; used in SOV-5.

Extensions:

- CallOperationAction

Constraints:

- Activity must be a ServiceFunction
- Activity must be a Function
- Operation must be a ServiceOperation

Use:

- Can be the Source and Target of a FunctionEdge control flow

ServicePoint

The mechanism by which a service communicates; used in OV-2, SV-1 and SV-12.

Extensions:

- Port

Constraints:

- Type must be a ServiceInterface
- Owned behavior is a ServiceFunction

Use:

- Can be owned by a Node or a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software,

ResourceArtifact or System)

ServicePolicy

A constraint governing the consumers and providers of services; used in SOV-4a.

Extensions:

- Constraint

Use:

- Rule can be owned by a ServiceInterface

ServiceStateMachine

UPDM artifact that extends UML StateMachine; used in SOV-4b.

Extensions:

- StateMachine

Software

Software needed for the functioning of the system; used in OV-2, OV-3, SV-1, SV-3, SV-9, SV-10a and SV-12.

Extensions:

- Class

Generalizations:

- ManufacturedResourceType
- Resource
- SubjectOfForecast
- ResourceInteractionItem
- Performer
- SubjectOfResourceConstraint

Use: Can:

- Be conveyed on a MaterielExchange information flow
- Be type of HostedSoftware
- Be the target of a Controls flow (from ManfacturedResourceType)
- Have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource)
- Be client of a RealizesCapability realization to a Capability (from Resource)
- Be client of a ProvidesCompetence dependency to a Competence (from Resource)
- Have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint)
- Be supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast)
- Own a ServicePoint (from Resource)
- Own a RequestPoint (from Resource)
- Own a ResourcePort (from Resource)

- Be source and target of a ResourceInteraction (from Resource)
- Own a ServiceOperation (from Resource)
- Be a type of a KnownResource (from Resource)
- Be a type of a ResourceRole (from Resource)
- Have a Performs dependency to a PerformedActivity (Function or OperationalActivity) (from Performer)

Standard

A ratified set of rules that are used to guide and/or constrain any UPDM element; used in SV-9, TV-1 and TV-2.

Extensions:

- Class

Generalizations:

- SubjectOfForecast

Specializations:

- Protocol

Use:

- Any UPDMElement can have a 'conformsTo' association to a Standard
- Can have an association (ratifiedBy) with an ActualOrganization
- Can be supplier or client of a Forecast (both must be same stereotype) (from SubjectOfForecast)

StandardConfiguration

A comment, attached to a CapabilityConfiguration, indicating that the annotated CapabilityConfiguration is a standard Pattern for re-use in the architecture; used in TV1 and TV-2.

Extensions:

- Note

Constraints:

- The annotated element must be a CapabilityConfiguration

StandardOperationalActivity

An OperationalActivity that is a standard procedure and that is doctrinal; used in OV-5 and StV-6.

Extensions:

- Activity

Generalizations:

- OperationalActivity
- PerformedActivity
- SubjectOfOperationalConstraint
- OperationalElement
- SubjectOfOperationalStateMachine

Constraints:

- Owned parameters must be OperationalParameter (from OperationalActivity)

Use: Can:

- Be Client of a MapsToCapability dependency to a Capability Class
- Be Supplier of a Performs dependency (from PerformedActivity)
- Be Supplier of an OwnsProcess dependency (from OperationalActivity)
- Be the Activity/Behavior of an OperationalActivityAction (from OperationalActivity)
- Be the owner of an OperationalActivityEdge (from OperationalActivity)
- Have an attached OperationalConstraint (from SubjectOfOperationalConstraint)
- Be the Supplier of a SupportsOperationalActivity dependency (from OperationalActivity)
- Own an OperationalStateMachine (from SubjectOfOperationalStateMachine)

StereotypeExtension

Defines an additional stereotype used in the architecture that is not defined in this metamodel; used in AV-2.

Extensions:

- Note

Constraints:

- Annotated element must be a UPDMElement

Use:

- Can have a set of associations (ontologyReference) to ExternalType

StructuralPart

Describes a structural part of an EnterprisePhase; used in AV-1.

Extensions:

- Part

Constraints:

- Type must be an EnterprisePhase
- Class must be an EnterprisePhase

SubOrganization

Asserts that one type of organization is typically the parent of another; used in OV-4 and SV-1.

Extensions:

- Part

Generalizations:

- OrganizationRole
- ResourceRole

Constraints:

- Type must be an Organization

- Class must be an Organization

Use: Can:

- Have a RequiresCompetence dependency to a Competence (from ResourceRole)
- Have a set of associations to 'used' Functions (from ResourceRole)

SubSystemPart

Indicates that a subsystem is part of another system; used in SV-1.

Extensions:

- Part

Generalizations:

- Part
- ResourceRole

Constraints:

- Class must be a ResourceArtifact (from Part)
- Type must be a ResourceArtifact (from Part)

Use: Can:

- Have a RequiresCompetence dependency to a Competence (from ResourceRole)
- Have a set of associations to 'used' Functions (from ResourceRole)

SupportsOperationalActivity

An assertion that a Service in some way contributes or assists in the execution of an OperationalActivity.

Extensions:

- Dependency

Constraints:

- Client must be a ServiceInterface
- Supplier must be an OperationalActivity

System

Any organized assembly of resources and procedures united and regulated by interaction of interdependence to accomplish a set of specific functions.

Extensions:

- Class

Generalizations:

- ResourceArtifact
- OperationalExchangeItem
- ManfacturedResourceType
- Resource
- SubjectOfForecast

- ResourceInteractionItem
- Performer
- SubjectOfResourceConstraint

Use: Can:

- Be conveyed by a MaterielExchange (from ResourceArtifact)
- Be the type of an OperationalParameter (from OperationalExchangeItem)
- Own HostedSoftware (from ResourceArtifact)
- Be the Class and type of a Part (from ResourceArtifact)
- Be the type of a ResourceComponent (from ResourceArtifact)
- Be the type of an Equipment (from ResourceArtifact)
- Be the target of a Controls flow (from ManfacturedResourceType)
- Have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource)
- Be client of a RealizesCapability realization to a Capability (from Resource)
- Be client of a ProvidesCompetence dependency to a Competence (from Resource)
- Have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint)
- Be supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast)
- Own a ServicePoint (from Resource)
- Own a RequestPoint (from Resource)
- Own a ResourcePort (from Resource)
- Be source and target of a ResourceInteraction (from Resource)
- Own a ServiceOperation (from Resource)
- Be type of a KnownResource (from Resource)
- Be type of a ResourceRole (from Resource)
- Have a Performs dependency to a PerformedActivity (Function or OperationalActivity) (from Performer)

SystemConnector

A link between two systems.

Extensions:

- Association
- Connector

Generalizations:

- ResourceInterface
- SystemsElement

Specializations:

- SystemConnector

Constraints:

- End roles must be ResourceRole (from ResourceInterface)
- End types must be Resource (from ResourceInterface)

Use:

- Can realize a ResourceInteraction (from ResourceInterface)

SystemFunction

A DoDAF alias for Function.

Extensions:

- Activity

Generalizations:

- Function
- PerformedActivity
- SystemsElement
- SubjectOfResourceConstraint

Constraints:

- Owned parameters are FunctionParameter (from Function)

Use: Can:

- Be Supplier of a Performs dependency (from PerformedActivity)
- Own ServiceOperationAction, FunctionAction or FunctionEdge (from Function)
- Be Client of an ImplementsOperational dependency to an OperationalActivity (from SystemsElement)
- Have an attached ResourceConstraint (from SubjectOfResourceConstraint)

SystemFunctionAction

A DoDAF alias for FunctionAction.

Extensions:

- CallBehaviorAction

Generalizations:

- FunctionAction

Constraints:

- Activity is stereotyped Function (from FunctionAction)

Use:

- Press Ctrl+L to set the function (from FunctionAction)

SystemFunctionEdge

An alias for FunctionEdge.

Extensions:

- A DoDAF ControlFlow

Generalizations:

- FunctionEdge
- SystemsElement

Constraints:

- Source must be a ServiceOperationAction (from FunctionEdge)
- Target must be a ServiceOperationAction (from FunctionEdge)

Use:

- Can realize a ResourceInteraction (right-click, Advanced > Information Flows Realized) (from FunctionEdge)

SystemsNode

A DoDAF alias for CapabilityConfiguration.

Extensions:

- Class

Generalizations:

- CapabilityConfiguration
- Resource, ConceptItem
- Performer
- ResourceInteractionItem
- SubjectOfResourceConstraint
- SubjectOfForecast
- SystemsElement
- SubjectOfResourceStateMachine
- ResourceInteractionItem

Use: Can:

- Have a set of associated deployed milestones, stereotyped DeployedMilestone (from CapabilityConfiguration)
- Have an optional associated no longer used milestone, stereotyped NoLongerUsedMilestone (from CapabilityConfiguration)
- Have a set of associated increment milestones, stereotyped IncrementMilestone (from CapabilityConfiguration)
- Have an optional associated out of service milestone, stereotyped OutOfServiceMilestone (from CapabilityConfiguration)
- Be annotated by a StandardConfiguration note (from CapabilityConfiguration)
- Be the type of a ConceptRole (from ConceptItem)
- Have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource)
- Be client of a RealizesCapability realization to a Capability (from Resource)
- Be client of a ProvidesCompetence dependency to a Competence (from Resource)
- Have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint)
- Be supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast)
- Own a ServicePoint (from Resource)
- Own a RequestPoint (from Resource)
- Own a ResourcePort (from Resource)
- Be source and target of a ResourceInteraction (from Resource)
- Own a ServiceOperation (from Resource)
- Be the type of a KnownResource (from Resource)
- Be the type of a ResourceRole (from Resource)
- Have a Performs dependency to a PerformedActivity (Function, OperationalActivity) (from Performer)

TechnologyForecast

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

Extensions:

- Forecast
- Dependency

Constraints:

- Client and Supplier are both SubjectOfForecast (Standard, Competence, Capability, CapabilityConfiguration, Organization, Post, ResourceArtifact or Software) (from Forecast)
- Client and Supplier must be the same specialization of SubjectOfForecast (from Forecast)

TemporalPart

EnterprisePhase elements that have a time-based nature; used in AV-1.

Extensions:

- Part

Constraints:

- Type must be an EnterprisePhase
- Class must be an EnterprisePhase

UsedConfiguration

The use of a CapabilityConfiguration in another CapabilityConfiguration; used in SV-1.

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

- Type must be a CapabilityConfiguration
- Class must be a CapabilityConfiguration

Use: Can:

- Have a RequiresCompetence Dependency to a Competence (from ResourceRole)
- Have a set of Associations (usedFunctions) to Function (from ResourceRole)

VisionStatement

A high-level textual description of an EnterpriseVision.

Extensions:

- Note

WholeLifeEnterprise

A purposeful endeavor of any size involving people, organizations and supporting systems; used in AV-1 and StV-1.

Extensions:

- Class

Generalizations:

- EnterprisePhase

Use: Can:

- Have a set of Associations (statementTasks) to EnduringTask Class (from EnterprisePhase)
- Have a set of Associations (exhibits) to Capability Class (from EnterprisePhase)
- Have a set of Associations (inhabits) to Environment Class (from EnterprisePhase)
- Have a set of Associations (goals) with EnterpriseGoal Class (from EnterprisePhase)
- Have a set of Associations (visions) with EnterpriseVision Class (from EnterprisePhase)
- Be the type of a StructuralPart or TemporalPart (from EnterprisePhase)
- Fulfill a Mission Use Case (from EnterprisePhase)
- Be Supplier of a DefinesArchitecture Realization (from EnterprisePhase)

Abstract Stereotypes

Stereotype Specializations

Stereotype	Description
ActualOrganizationalResource	An actual organization or post. Specializations: • ActualOrganization • ActualPost
ConceptItem	An item that might feature in a high level operational concept. Specializations: • CapabilityConfiguration • Node • ReferredLocation • Resource
DataModel	A structured specification of data, showing classifications of data elements and the relationships between them. Specializations: • LogicalDataModel • PhysicalDataModel
EnvironmentalType	A type of environment. Specializations: • LightCondition • Location • PhysicalLocation • Climate
ManufacturedResourceType	A resource artifact or software. Generalizations: • Resource Specializations: • ResourceArtifact • Software
NodeChild	An abstract element used for supporting the composite structuring of operational elements such as Nodes and LogicalArchitectures. Specializations: • NodeRole • ProblemDomain • KnownResource

NodeParent	Represents the owners/context of composite structure at the operational level. Specializations: • Node • ExternalNode • OperationalNode • LogicalArchitecture
OperationalElement	Elements relating to operational models. Specializations: • OperationalActivity • StandardOperationalActivity • OperationalMessage • Node • ExternalNode • OperationalNode • Needline • OperationalExchange • InformationElement • OperationalActivityEdge
OperationalExchange	Describes the characteristics of an exchanged item, such as the content, format (voice, imagery, text and message format), throughput requirements, security or classification level, timeliness requirement, and the degree of interoperability. Generalizations: • OperationalElement Specializations: • ConfigurationExchange • EnergyExchange • InformationExchange • MaterielExchange • OrganizationalExchange
OperationalExchangeItem	An item that participates in an operational exchange. Specializations: • Post • Organization • ResourceArtifact • System
OrganizationalResource	Either an organization or a post. Generalizations: • Resource • OperationalExchangeItem Specializations: • Post • Organization

OrganizationRole	Represents properties in an organization that are typed by another organization or a post. Generalizations: • ResourceRole Specializations: • SubOrganization • PostRole
PerformedActivity	A behavior that can be performed by a Performer. Specializations: • OperationalActivity • Function
Performer	A structural element that can perform behaviors (such as PerformedActivity) Specializations: • Node • Resource
ProtocolImplementation	An element that implements a specific protocol. Specializations: • ResourcePort • ResourceInteraction • Controls • Commands • DataExchange • ResourceConnector
ReferredLocation	Either an actual location or a type of location (that is, environment) at/in which operations can be conducted. Generalizations: • ConceptItem • EnvironmentalType Specializations: • Location • PhysicalLocation
Resource	A physical asset, organizational resource or functional resource that can contribute towards fulfilling a capability. Generalizations: • SystemsElement • SubjectOfResourceStateMachine • ResourceInteractionItem • Performer • SubjectOfResourceConstraint • ConceptItem • SubjectOfForecast

	Specializations: • Post • Organization • CapabilityConfiguration • SystemsNode • Software • ResourceArtifact • System
ResourceInteractionItem	Represents the items exchanged between resources through a resource interaction. Specializations: • Energy • Resource • DataElement
ResourceRole	Defines the usage of any resource in the system. Specializations: • UsedConfiguration • Equipment • SubOrganization • PostRole • Part • SubSystemPart • HumanResource • ResourceComponent • Platform • HostedSoftware
SubjectOfForecast	Any element that can be subject to a forecast. Specializations: • Standard • Protocol • Capability • Competence • Post • Organization • CapabilityConfiguration • SystemsNode • Software • ResourceArtifact • System
SubjectOfOperationalConstraint	An element of the architecture that can be subject to an OperationalConstraint or OperationalStateDescription. Specializations: • OperationalActivity

	• InformationElement • Node • Mission
SubjectOfOperationalStateMachine	The element being described by the StateMachine. Specializations: • OperationalActivity • InformationElement • Node • Mission
SubjectOfResourceConstraint	Anything that can be constrained by a ResourceConstraint. Specializations: • Post • Organization • CapabilityConfiguration • SystemsNode • Software • ResourceArtifact • System • DataElement • Function
SubjectOfResourceStateMachine	The element being described by the StateMachine. Specializations: • Post • Organization • CapabilityConfiguration • SystemsNode • Software • ResourceArtifact • System • DataElement
SystemsElement	Elements relating to system models. Specializations: • Resource • ResourceInteraction • ResourceMessage • ResourceInteraction • DataElement • ResourceStateMachine • FunctionEdge • Function
UPDMElement	A super type for all UPDM elements, providing a means of extending UPDM

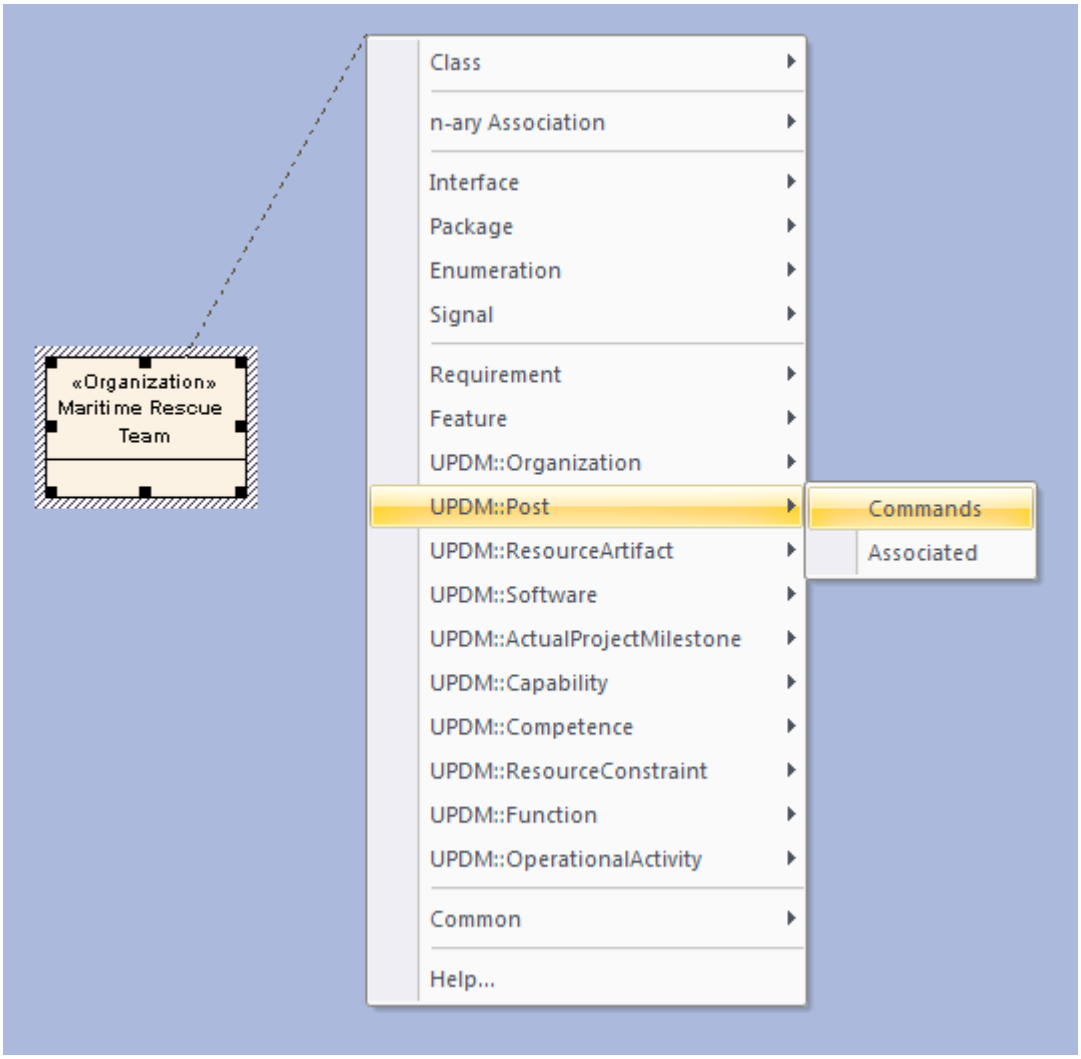
	elements in a common way. Specializations: • All UPDM stereotypes

Quicklinks

The UPDM profile makes use of Enterprise Architect's 'Quicklink' feature to make it quicker and easier to create correct and consistent UPDM models.

When you select an element, the Quicklink arrow displays next to the top-right corner of the element.

Drag the arrow away from the element and release it over empty diagram space. The Quicklink context menu displays, listing all the UPDM elements that could commonly be attached to the element, as shown.



Selecting the 'UPDM::Post | Commands' option in the context menu creates a new Post element connected to the Organization element by a Commands relationship.

Tagged Values for UPDM

UPDM is an extension of UML, which is extended by applying stereotypes to elements. The stereotypes in turn apply Tagged Values that provide additional information to that normally associated with a UML element. Since UPDM makes frequent use of Tagged Values, it is recommended to keep the Properties window docked and visible at all times, with the 'UPDM' section expanded.

Synchronize Tagged Values

The list of Tagged Values owned by an element can get out of date. A new version of the UML Profile might define new or modified Tagged Values for an element type, or as user might delete some. Also, you might apply the stereotype using the stereotype combo box, which doesn't add Tagged Values. If you want to refresh the list of Tagged Values for a single element, you can drag and drop the stereotype from the Diagram Toolbox onto the element and select the 'Apply' option. This only works for single diagram objects, and not for connectors.

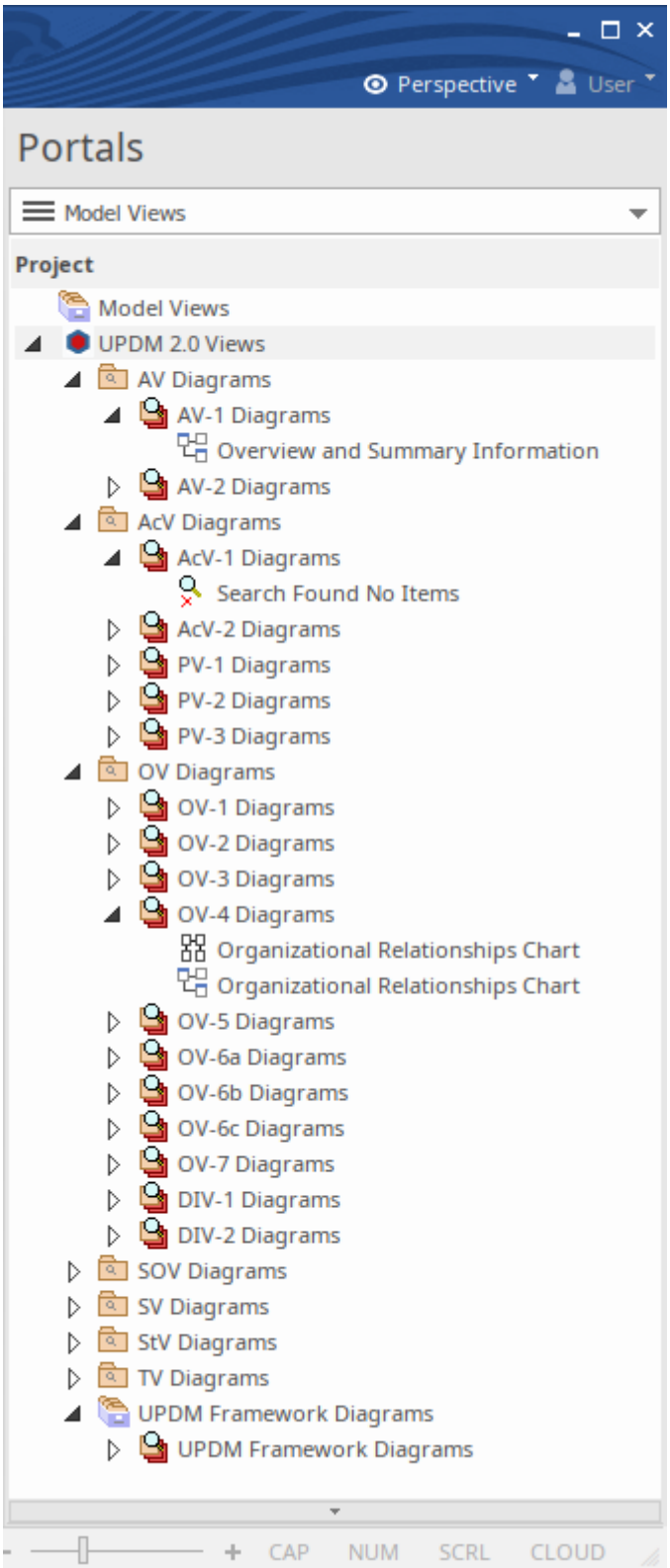
If you want to refresh the list of Tagged Values for every element in your model, select the 'Specialize > Technologies > UPDM > Synchronize Tagged Values' menu option.

The URL/URI Tagged Value

In the UPDM Profile Specification the stereotype «UPDMElement» - from which all profile elements are derived - provides a Tagged Value URL/URI. In Enterprise Architect, this Tagged Value has been omitted and you must use the standard Enterprise Architect functionality to achieve the same result: that is, open the 'Properties' dialog for the element, select the 'Files' tab or page, and type in a web location.

Model Views in UPDM

The 'Model Views' tab of the Focus window displays a variety of different views on the model data, providing an alternative to the Browser window. You can use this tab as a quick and easy method of locating all of your UPDM diagrams in the current model.



To open the 'Model Views' tab, select 'Start > All Windows > Design > Focus > Model Views'. Expand the appropriate

folders and double-click on the required diagram to open it.

Glossary

UPDM provides the ability to import descriptions of all UPDM stereotypes into the Enterprise Architect Glossary. This gives you a quick reference to the meaning of each stereotype, lists the views that the stereotype might appear in and, for abstract stereotypes, lists the concrete stereotypes that inherit from the abstract stereotype.

Import Glossary

You import the Glossary definitions into each model individually. To do this, select the 'Publish > Model Exchange > Import > Other Tools/Formats' ribbon option.

View the Glossary

To view the Glossary, select one of:

- 'Design > Dictionary > Glossary > Glossary View to display the Project Glossary view
- 'Design > Dictionary > Glossary > Edit' to open the 'Glossary' dialog
- In any dialog 'Notes' field, a Glossary hyperlink (underlined and colored blue)

Using Enterprise Architect Elements

Creating an instance from a Class

UPDM has Classifier/Instance pairs where the classifier describes a class of elements and the instance represents a single member of that Class. The Classifier/Instance pairs in UPDM are:

- MeasurementSet/ActualMeasurementSet
- Organization/ActualOrganization
- Person/ActualPerson
- Post/ActualPost
- Project/ActualProject
- ProjectMilestoneType/ActualProjectMilestone
- CapabilityConfiguration/FieldedCapability

If you have an element that is the classifier part of one of these Classifier/Instance pairs, you can choose between two main approaches for creating the instance:

1. Set the classifier of an existing instance - Click on the instance element in a diagram and then either press Ctrl+L or right-click and select 'Advanced | Instance Classifier'; the same command sets the type of a Port or Part.
2. Create an instance from an existing classifier - Press Ctrl while dragging the classifier element from the Browser window onto a diagram. The 'Paste Element' dialog displays; select the 'Paste as Instance of Element' option. An anonymous instance is created with the appropriate stereotype; select the instance, press F2 and give it a name.

Set the run state of an object

Where a classifier can own a set of attributes, an instance of that classifier can own a Slot for each attribute. The set of assigned values for these Slots is known as the run state. To set the run state of an object on a diagram, right-click on it and select 'Features | Set Run State' or press Ctrl+Shift+R.

Some stereotypes are defined by UPDM as extending the Slot metaclass. Each run state attribute represents a Slot, but it is not possible to stereotype Slots in Enterprise Architect, so UPDM's slot-extending stereotypes are not available in Enterprise Architect's implementation. UPDM stereotypes that extend Slot are:

- ActualMeasurement (ActualMeasurementSet)
- ActualOrganizationRole (ActualOrganization)
- MeasureOfPerformance (ActualMeasurementSet)
- ProjectStatus (ActualProjectMilestone)

Properties

Some stereotypes in UPDM are defined as extending the UML Property metaclass. This gives you the choice of a number of different representations for these elements in your model. If you drag one of the properties from the Toolbox onto a classifier element on a diagram, you are prompted to select to create an attribute, a Part, or a Port. These are all different representation of the UML Property metaclass; which one you choose depends on what rendering of the Property you want to see in your model.

Another representation of the UML Property metaclass is the Association End; to apply one of UPDM's Property stereotypes to an Association End:

1. Double-click on the element to display the 'Properties' dialog.

2. Select the 'Roles' tab.
3. Click on the button next to the appropriate 'Stereotype' field.
4. On the 'Stereotype for Association' dialog, select 'UPDM' from the 'Profile' field.
5. Select every stereotype that applies.

Stereotypes that extend Property are:

- ConceptRole
- EntityAttribute
- EnvironmentProperty
- Equipment
- HostedSoftware
- HumanResource
- KnownResource
- Measurement
- NodeRole
- Part
- PerformanceParameter
- Platform
- PostRole
- ProblemDomain
- ProjectTheme
- ProtocolLayer
- ResourceComponent
- ServiceAttribute
- StructuralPart
- SubOrganization
- SubSystemPart
- TemporalPart
- UsedConfiguration

Model Validation in UPDM

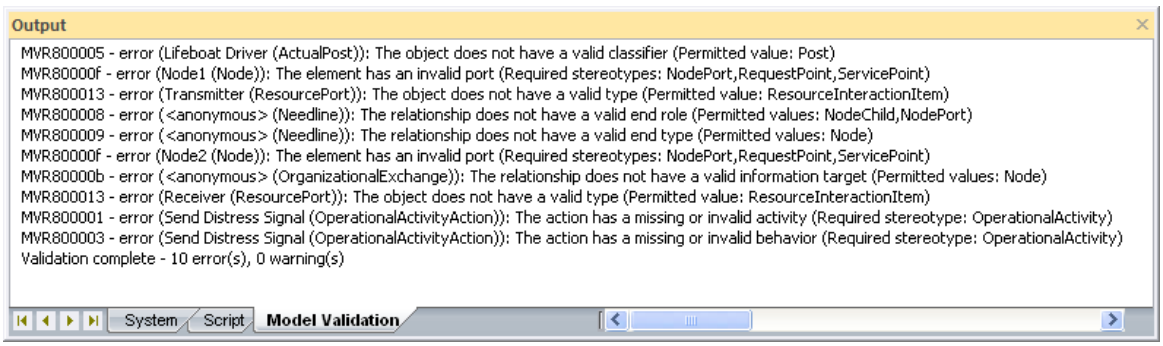
Enterprise Architect supports model validation of UPDM models, validating and reporting errors against more than 160 different rules.

Configure Model Validation

Before being able to validate a model, you first have to select the rules to validate against. Select 'Design >Package > Manage > Validate > Configure Validation Rules' and deselect the checkbox against all validation rules except for the UPDM set.

Perform Model Validation

Open a diagram or select either a Package or a number of elements in the Browser window, then select the 'Design > Package > Manage > Validate > Validate Current Package' ribbon option (or press Ctrl+Alt+V). Validation results are displayed in the System Output window, which is opened if it isn't already displayed. To go to the element that caused a validation error, double-click on the error message in the System Output window.



Model Validation Rules

Errors are indicated by an error code of the format MVRxxnnnn where:

- xx is 80 by default (if the MDG Technology for UPDM is the only Add-In that you have installed) but could be some other number, and
- nnnn is a hexadecimal number from 0001 to 0013 as described here

MVRxx0001 - activity

Error Message: The action has a missing or invalid activity (Required stereotype: <stereotypeList>)

The validation rule checks that stereotyped Action elements are owned by an Activity with the required stereotype.

Solution: Locate the Action in the Browser window, locate an Activity with one of the named stereotypes (or their specializations) or create a new one, and drag the Action to the Activity.

Action Stereotypes	Activity Stereotypes
FunctionAction	Function
OperationalActivityAction	OperationalActivity
ServiceOperationAction	Function
ServiceOperationAction	ServiceFunction

MVRxx0002 - annotatedElement

Error Message: The note has an invalid annotated element (Required stereotype: <stereotype>)

This validation rule checks that stereotyped Note elements are attached (by a NoteLink connector) to an element with the required stereotype.

Solution: Attach the Note to an element with the named stereotype (or one of its specializations). You can do this by either dragging the opposite end of the NoteLink connector, or deleting the NoteLink connector and creating a new one with the Quick Linker.

Note Stereotypes	Annotated Element Stereotypes
Alias	UPDMElement
ArchitectureMetadata	ArchitecturalDescription
Definition	UPDMElement
StandardConfiguration	CapabilityConfiguration
StereotypeExtension	UPDMElement

MVRxx0003 - behavior

Error Message: The action has a missing or invalid behavior (Required stereotype: <stereotype>)

This validation rule checks that stereotyped CallBehaviorAction elements call a Behavior with the required stereotype.

Solution: Right-click on the Action and select Advanced | Set Behavioral Classifier, or press Ctrl+L, and select a behavior element with the named stereotype (or one of its specializations).

Action Stereotypes	Behavior Stereotypes
OperationalActivityAction	OperationalActivity
ServiceFunctionAction	ServiceFunction

MVRxx0004 - class

Error Message: The object does not have a valid owning Class (Permitted values: <stereotypeList>)

This validation rule checks that stereotyped Property elements (Parts or attributes) are owned by a Class with the required stereotype.

Solution: Locate the property in the Browser window, locate a Class with one of the named stereotypes (or their specializations) or create a new one, and drag the property to the Class.

Property Stereotypes	Class Stereotypes
Equipment	OrganizationalResource
HostedSoftware	ResourceArtifact
HumanResource	CapabilityConfiguration
NodeChild	NodeParent
NodeRole	Node
Part	ResourceArtifact
PostRole	Organization
ProblemDomain	LogicalArchitecture
ProtocolLayer	Protocol
ResourceComponent	CapabilityConfiguration
ResourceRole	Resource
StructuralPart	EnterprisePhase
SubOrganization	Organization

TemporalPart	EnterprisePhase
UsedConfiguration	CapabilityConfiguration

MVRxx0005 - classifier

Error Message: The object does not have a valid classifier (Permitted value: <stereotype>)

This validation rule checks that stereotyped instance elements (objects) are classified by elements with the required stereotypes.

Solution: Select the object, right-click it and select Advanced | Instance Classifier, or press Ctrl+L, and select a classifier element with the named stereotype (or one of its specializations).

Object Stereotypes	Classifier Stereotypes
ActualMeasurementSet	MeasurementSet
ActualOrganization	Organization
ActualPerson	Person
ActualPost	Post
ActualProject	Project
ActualProjectMilestone	ProjectMilestoneType
FieldedCapability	CapabilityConfiguration

MVRxx0006 - client

Error Message: The relationship does not have a valid client (Permitted values: <stereotypeList>)

This validation rule checks that, for stereotyped Dependency or Realization relationships, their client (source) elements have the required stereotypes.

Solution: Drag the end of the relationship without the arrowhead to an element with the named stereotype (or one of its specializations).

Relationship Stereotypes	Client Element Stereotypes
ArbitraryRelationship	HighLevelOperationalConcept
ArchitecturalReference	ArchitecturalDescription
CompatibleWith	Node
DefinesArchitecture	ArchitecturalDescription

ExhibitsCapability	Node
Expose	ServiceInterface
FillsPost	ActualPerson
Forecast	SubjectOfForecast
ImplementsOperational	SystemsElement
MapsToCapability	StandardOperationalActivity
MilestoneSequence	ActualProjectMilestone
OwnsProcess	ActualOrganizationalResource
Performs	Performer
ProjectSequence	ActualProject
ProvidesCompetence	Resource
RealizesCapability	Resource
RealizesCapability	ServiceInterface
RequiresCompetence	ResourceRole
SameAs	UPDMElement
SupportsOperationalActivity	ServiceInterface

MVRxx0007 - constrainedElement

Error Message: The constraint has an invalid constrained element (Required stereotype: %s)

This validation rule checks that stereotyped Constraint elements are attached (by a NoteLink) to elements with the required stereotypes.

Solution: Attach the constraint to an element with the named stereotype (or one of its specializations). You can do this by either dragging the opposite end of the NoteLink connector, or by deleting the NoteLink connector and creating a new one using the Quick Linker.

Constraint Stereotypes	Constrained Element Stereotypes
OperationalConstraint	SubjectOfOperationalConstraint
ResourceConstraint	SubjectOfResourceConstraint

MVRxx0008 - endRoles

Error Message: The relationship does not have a valid end role (Permitted values: <stereotypeList>)

This validation rule checks that, for stereotyped Association or Connector relationships, the elements at both ends of the relationship have the required stereotypes.

Solution: Drag one or both ends of the relationship to elements with the named stereotype (or one of its specializations).

Relationship Stereotypes	End Element Stereotypes
Needline	NodeChild
Needline	NodePort
ResourceConnector	ResourcePort
ResourceInterface	ResourceRole

MVRxx0009 - endType

Error Message: The relationship does not have a valid end type (Permitted values: <stereotypeList>)

This validation rule checks that, for stereotyped connectors, the elements (Objects or Parts) at both ends of the relationship are typed by the required stereotypes.

Solution: Drag one or both ends of the relationship to elements that have types with the named stereotype (or one of its specializations).

Connector Stereotypes	End Type Stereotypes
EntityRelationship	EntityItem
Needline	Node
ResourceInterface	Resource

MVRxx000a - informationSource

Error Message: The relationship does not have a valid information source (Permitted values: <stereotypeList>)

This validation rule checks that stereotyped InformationFlow relationship source elements have the required stereotypes.

Solution: Drag the end of the information flow without the arrowhead to an element with the named stereotype (or one of its specializations).

InformationFlow Stereotypes	Source Element Stereotypes
ActualOrganizationRelationship	ActualOrganizationalResource

Commands	OrganizationalResource
Controls	OrganizationalResource
OperationalExchange	Node
ResourceInteraction	Resource

MVRxx000b - informationTarget

Error Message: The relationship does not have a valid information target (Permitted values: <stereotypeList>)

This validation rule checks that stereotyped InformationFlow relationship target elements have the required stereotypes.

Solution: Drag the end of the information flow with the arrowhead to an element with the named stereotype (or one of its specializations).

InformationFlow Stereotypes	Target Element Stereotypes
ActualOrganizationRelationship	ActualOrganizationalResource
Commands	OrganizationalResource
Controls	OrganizationalResource
OperationalExchange	Node
ResourceInteraction	Resource

MVRxx000c - ownedAttribute

Error Message: The element has an invalid attribute (Required stereotype: <stereotype>)

This validation rule checks that, for stereotyped Class elements, any attributes that they own have the required stereotypes.

Solution: Replace the attribute with one with the named stereotype (or one of its specializations).

Class Stereotypes	Attribute Stereotypes
EntityItem	EntityAttribute
Environment	EnvironmentProperty
HighLevelOperationalConcept	ConceptRole
MeasurementSet	Measurement

ProjectMilestoneType	ProjectTheme
ServiceInterface	ServiceAttribute

MVRxx000d - ownedOperation

Error Message: The element has an invalid operation (Required stereotype: %s)

This validation rule checks that, for stereotyped Class elements, any operations that they own have the required stereotypes.

Solution: Replace the operation with one with the named stereotype (or one of its specializations).

Class Stereotype	Operation Stereotype
ServiceInterface	ServiceOperation

MVRxx000e - ownedParameter

Error Message: The element has an invalid activity parameter (Required stereotype: %s)

This validation rule checks that, for stereotyped Activity elements, any ActivityParameter elements that they own have the required stereotypes.

Solution: Locate the ActivityParameter in the Browser window and drag and drop it onto an element with the appropriate stereotype, and/or replace the ActivityParameter in its current owner with an ActivityParameter with the named stereotype.

Activity Stereotypes	ActivityParameter Stereotypes
Function	FunctionParameter
OperationalActivity	OperationalParameter

MVRxx000f - ownedPort

Error Message: The element has an invalid Port (Required stereotypes: <stereotypeList>)

This validation rule checks that, for stereotyped Class elements, any Ports that they own have the required stereotypes.

Solution: Locate the Port in the Browser window and drag and drop it onto an element with the appropriate stereotype, and/or replace the Port in its current owner with a Port with one of the named stereotypes.

Class Stereotypes	Port Stereotypes
Node	NodePort
Node	RequestPoint
Node	ServicePoint

Resource	RequestPoint
Resource	ResourcePort
Resource	ServicePoint

MVRxx0010 - source

Error Message: The relationship does not have a valid source (Permitted values: <stereotypeList>)

This validation rule checks that stereotyped ActivityEdge connector source elements have the required stereotypes.

Solution: Drag the end of the relationship without the arrowhead to an element with the named stereotype (or one of its specializations).

ActivityEdge Stereotypes	Source Element Stereotypes
FunctionEdge	ServiceOperationAction
OperationalActivityEdge	OperationalActivityAction

MVRxx0011 - supplier

Error Message: The relationship does not have a valid supplier (Permitted values: <stereotypeList>)

This validation rule checks that stereotyped Dependency or Realization relationship supplier (target) elements have the required stereotypes.

Solution: Drag the end of the relationship with the arrowhead to an element with the named stereotype (or one of its specializations).

Relationship Stereotypes	Supplier Element Stereotypes
ArbitraryRelationship	HighLevelOperationalConcept
ArchitecturalReference	ArchitecturalDescription
CompatibleWith	ReferredLocation
DefinesArchitecture	EnterprisePhase
ExhibitsCapability	Capability
Expose	Capability
FillsPost	ActualPost
Forecast	SubjectOfForecast
ImplementsOperational	OperationalElement

MapsToCapability	Capability
MilestoneSequence	ActualProjectMilestone
OwnsProcess	OperationalActivity
Performs	PerformedActivity
ProjectSequence	ActualProject
ProvidesCompetence	Competence
RealizesCapability	Capability
RealizesCapability	Competence
RequiresCompetence	ExternalIndividual
SameAs	ExternalType
SupportsOperationalActivity	OperationalActivity

MVRxx0012 - target

Error Message: The relationship does not have a valid target (Permitted values: <stereotypeList>)

This validation rule checks that stereotyped ActivityEdge connector target elements have the required stereotypes.

Solution: Drag the end of the relationship with the arrowhead to an element with the named stereotype (or one of its specializations).

ActivityEdge Stereotypes	Target Element Stereotypes
FunctionEdge	ServiceOperationAction
OperationalActivityEdge	OperationalActivityAction

MVRxx0013 - type

Error Message: The object does not have a valid type (Permitted value: <stereotype>)

This validation rule checks that stereotyped Property elements (Parts or attributes) have type elements with the required stereotypes.

Solution: For Parts, right-click on the Part and select 'Advanced | Set Property Type', or press Ctrl+L, and select a type element with the named stereotype (or one of its specializations). For attributes, open the Features window for the attribute and select a type element with the named stereotype (or one of its specializations) in the 'Type' field.

Property Stereotypes	Type Element Stereotypes

ConceptRole	ConceptItem
EnvironmentProperty	EnvironmentalType
Equipment	ResourceArtifact
FunctionParameter	ResourceInteractionItem
HostedSoftware	Software
HumanResource	OrganizationalResource
KnownResource	Resource
NodePort	OperationalExchangeItem
NodeRole	Node
OperationalParameter	OperationalExchangeItem
Part	ResourceArtifact
PostRole	Post
ProjectTheme	ProjectThemeStatus
ProtocolLayer	Protocol
RequestPoint	ServiceInterface
ResourceComponent	ResourceArtifact
ResourcePort	ResourceInteractionItem
ServicePoint	ServiceInterface
StructuralPart	EnterprisePhase
SubOrganization	Organization
TemporalPart	EnterprisePhase
UsedConfiguration	CapabilityConfiguration