



## **MDG Technology For UPDM User Guide**

*Welcome to the MDG Technology For UPDM*



© Copyright 2010 Sparx Systems Pty Ltd

# MDG Technology for UPDM User Guide

## Introduction

---

*MDG Technology for UPDM brings the Unified Profile for DoDAF and MODAF (UPDM) into Enterprise Architect modeling.*

# MDG Technology for UPDM User Guide

© 2010 Sparx Systems Pty Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: April 2010

## **Publisher**

*Sparx Systems*

## **Managing Editor**

*Geoffrey Sparks*

## **Technical Editor**

*Neil Capey*

## **Special thanks to:**

*All the people who have contributed suggestions, examples, bug reports and assistance in the development of MDG Technology for UPDM. The task of developing and maintaining this tool has been greatly enhanced by their contribution.*

# Table of Contents

Foreword	1
<b>Welcome</b>	<b>2</b>
Copyright Notice .....	2
Software Product License Agreement .....	3
Acknowledgement of Trademarks .....	6
Support .....	6
System Requirements .....	6
<b>Getting Started</b>	<b>8</b>
<b>Using UPDM</b>	<b>9</b>
UPDM Add-In and Technology .....	9
UPDM Add-In Menu .....	11
Model Wizard .....	12
UPDM Diagram Types .....	13
UPDM Toolboxes .....	13
UPDM Stereotypes .....	15
Abstract Stereotypes .....	51
Quicklinks .....	54
Tagged Values .....	55
Model Validation .....	56
Model Validation Rules .....	57
Relationship Matrix .....	64
Model Views .....	65
UPDM Framework Diagram .....	66
Glossary .....	67
Using Enterprise Architect Elements .....	68
Example Model .....	68
<b>Index</b>	<b>70</b>

# Foreword

MDG Technology for UPDM brings the Unified Profile for DoDAF and MODAF (UPDM) into Enterprise Architect modeling.

## 1 Welcome



Welcome to the *MDG Technology for UPDM - Enterprise Architect MDG Add-In, Version 1.0.3*.

This Add-in provides a UML profile that extends the capability of Enterprise Architect to enable 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.

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: <http://www.omg.org/spec/UPDM/>.

### Getting Started

For instructions on how to begin using the MDG Technology for UPDM, see [Getting Started](#)<sup>[8]</sup>.

#### See Also

- [Copyright Notice](#)<sup>[2]</sup>
- [Acknowledgement of Trademarks](#)<sup>[6]</sup>
- [License Agreement](#)<sup>[3]</sup>
- [Support](#)<sup>[6]</sup>
- [System Requirements](#)<sup>[6]</sup>

## 1.1 Copyright Notice

### Copyright © 2010 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 [license agreement](#)<sup>[3]</sup> 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.**  
7 Curtis St,  
Creswick, Victoria 3363,  
AUSTRALIA

Phone: +61 (3) 5345 1140  
Fax: +61 (3) 5345 1104

Support Email: [support@sparxsystems.com](mailto:support@sparxsystems.com)

Sales Email: [sales@sparxsystems.com](mailto:sales@sparxsystems.com)

Website: <http://www.sparxsystems.com>

## 1.2 Software Product License Agreement

### **SOFTWARE PRODUCT LICENSE AGREEMENT**

MDG Technology for UPDM, version 1.0. Enterprise Architect MDG Add-In.  
Copyright (C) 2010 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 SYSTEMS ("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 return the unused SOFTWARE PRODUCT to the place of purchase for a full refund.

The copyright in the SOFTWARE PRODUCT and its documentation is owned by Sparx Systems Pty Ltd A.C.N 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 the SOFTWARE PRODUCT indicates your acceptance of this EULA and warranty.

### **DEFINITIONS**

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

- "ACADEMIC EDITION" means an edition of the SOFTWARE PRODUCT purchased for educational purposes at an academic discount price.
- "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" means the edition of the SOFTWARE PRODUCT, which is available for purchase from the web site: <<http://www.sparxsystems.com/updm/purchase.html>>, following the 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 <<http://www.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.



Windows, Windows NT, Windows ME, Windows 2000, Windows XP, Windows Vista and Windows 7 are trademarks of Microsoft.

### GOVERNING LAW

This agreement shall be construed in accordance with the laws of the Commonwealth of AUSTRALIA, in the state of Victoria. **MDG Technology for UPDM - Enterprise Architect MDG Add-In**, Version 1.0.3.

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

**IMPORTANT-READ CAREFULLY:** This End User Licence 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 and modify 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 Licence Agreement, unless the contrary intention appears:

- "EULA" means this End User Licence 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.
- "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.

### GRANT OF LICENSE

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

- a) 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.
- b) 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.
- c) To make copies of the SOFTWARE PRODUCT for backup, archival and instructional purposes.

### EVALUATION LICENSE

The Trial Version of MDG Technology for UPDM is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use this software for evaluation purposes without charge for a period of 30 days.

Upon expiration of the 30 days, the SOFTWARE PRODUCT must be removed from the computer. Unregistered use of MDG Technology for Unified DoDAF-MODAF 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 this software after the 30 day evaluation period a license must be purchased (as described at [http://www.sparxsystems.com/ea\\_purchase.htm](http://www.sparxsystems.com/ea_purchase.htm)). Upon payment of the license fee, YOU will be sent details on where to download the registered edition of MDG Technology for UPDM and will be provided with a suitable software 'key' by email.

### ADDITIONAL RIGHTS AND LIMITATIONS

YOU hereby undertake not to sell or sub-licence the SOFTWARE PRODUCT other than as expressly authorized by this EULA.

**NO WARRANTY.** The SOFTWARE PRODUCT is provided "AS IS", without warranty of any kind, and SPARX expressly disclaims all warranties and/or conditions with respect to the SOFTWARE PRODUCT, either express, implied or statutory, including, but not limited to, the implied warranties and/or conditions of merchantability, of satisfactory quality, of fitness for a particular purpose, of accuracy, of quiet enjoyment, and of non-infringement of third party rights.

## LIMITATION

Under no circumstances shall SPARX be liable for any incidental, special, indirect or consequential damages arising out of or relating to this license or YOUR use, reproduction, modification, distribution of the SOFTWARE PRODUCT, or any portion thereof, whether under a theory of contract, warranty, strict liability or otherwise, even if the copyright holder has been advised of the possibility of such damages and notwithstanding the failure of essential purpose of any remedy.

## 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.

Windows, Windows NT®, Windows ME, Windows XP, Windows Vista, Windows 2000 and Windows 2003 are trademarks of Microsoft®.

## GOVERNING LAW

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

## 1.3 Acknowledgement of Trademarks

### Trademarks of Microsoft

- Microsoft®
- Windows®

### Trademarks of the OMG

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

## 1.4 Support

Technical support for MDG Technology for UPDM is available to registered users of Enterprise Architect. Responses to support queries are sent by email. Sparx Systems endeavors to provide a rapid response to all product-related questions or concerns.

Registered users can lodge a support request, by visiting:  
[http://www.sparxsystems.com/registered/reg\\_support.html](http://www.sparxsystems.com/registered/reg_support.html).

Trial users can contact Sparx Systems with questions regarding their evaluation at:  
[support@sparxsystems.com](mailto:support@sparxsystems.com).

An online user forum is also available for your questions and perusal, at  
<http://www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi>.

## 1.5 System Requirements

MDG Technology for UPDM runs under the following environments:

### Operating Systems

- Windows XP Professional
- Windows XP Home
- Windows XP Media Edition
- Windows XP Tablet Edition
- Windows 2000 Professional (SP3 or later)

- Windows Vista (32 bit)

### **Enterprise Architect Versions**

- Enterprise Architect Professional Version 8.0.855 (or later)
- Enterprise Architect Corporate Version 8.0.855 (or later)

#### **Note:**

While the MDG Technology for UPDM isn't specifically blocked from executing with the Enterprise Architect Desktop Edition, be aware that the Desktop Edition cannot make use of some of the capabilities of the product; e.g. Model Validation.

## 2 Getting Started

To install the MDG Technology for UPDM Add-In, first close down any open instances of Enterprise Architect, then double-click the installer and follow the instructions in the installation wizard. Once the Add-In has been installed, start Enterprise Architect; the Add-In should be fully enabled and ready to use.

If you are reinstalling over a previous installation, the MDG Technology might not be enabled when you start Enterprise Architect. In this case:

- Select the **Add-Ins | Manage Add-Ins** menu option and, on the **Manage Add-Ins** dialog, select the **Load on Startup** checkbox against UPDM. Click on the **OK** button and restart Enterprise Architect.

If that does not enable the Add-In:

- Select the **Settings | MDG Technologies** menu option and, on the **MDG Technologies** dialog, select the **Enabled** checkbox against UPDM. Click on the **OK** button and restart Enterprise Architect.

## 3 Using UPDM

UPDM is the Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF). UPDM is an Object Management Group (OMG) initiative; the specification is available from the OMG website at: <http://www.omg.org/spec/UPDM/>.

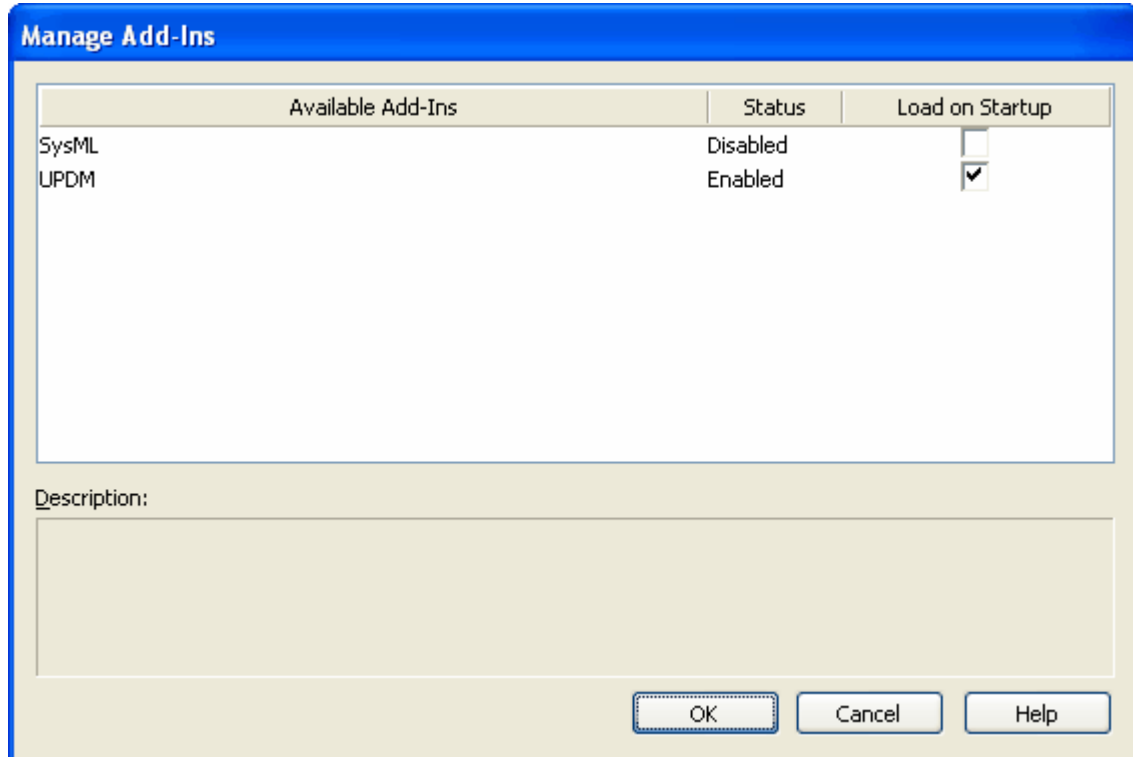
The MDG Technology for UPDM is an Add-In that enables you to use Enterprise Architect for UPDM modeling. The Add-In provides:

- The UPDM Profile (wrapped in an [MDG Technology](#) <sup>[9]</sup>), which defines the stereotyped UML elements that are used for UPDM modeling.
- [Add-in menu](#) <sup>[11]</sup> that enables you to access the Add-In functionality from Enterprise Architect's main menu.
- [Custom diagram types](#) <sup>[13]</sup> for each UPDM view.
- [Custom Toolbox pages](#) <sup>[13]</sup> for each UPDM diagram, which give easy access to the elements used on that diagram type.
- [Model wizard](#) <sup>[12]</sup> that can be used to import a template package for each UPDM view that provides a brief description of the view and what might be expected of the modeler.
- [Quicklinks](#) <sup>[54]</sup> for stereotyped elements that guide you towards creating correct relationships between elements.
- [Model Validation](#) <sup>[56]</sup> rules that enable you to check your models for correctness.
- [Relationship Matrix](#) <sup>[64]</sup> profiles for showing the relationships between elements.
- [Model Views](#) <sup>[65]</sup> that help you navigate your model quickly to find specific diagram more easily.
- [Glossary](#) <sup>[67]</sup> import, with items describing each UPDM stereotype for easy reference.
- Custom compartments and callout notation. (**Not yet implemented**).
- [Tagged Values](#) <sup>[55]</sup> that enable you enter metadata specific to UPDM elements.
- [Example Model](#) <sup>[68]</sup> that illustrates a typical UPDM problem and its solution, implemented using Enterprise Architect.

### 3.1 UPDM Add-In and Technology

#### Manage Add-Ins

After installing the Add-In, start Enterprise Architect. Select the **Add-Ins | Manage Add-Ins** menu option. The **Manage Add-Ins** dialog displays, listing the currently installed Add-Ins.



If the UPDM Add-In is marked as **Disabled**, tick the **Load on Startup** checkbox and restart Enterprise Architect.

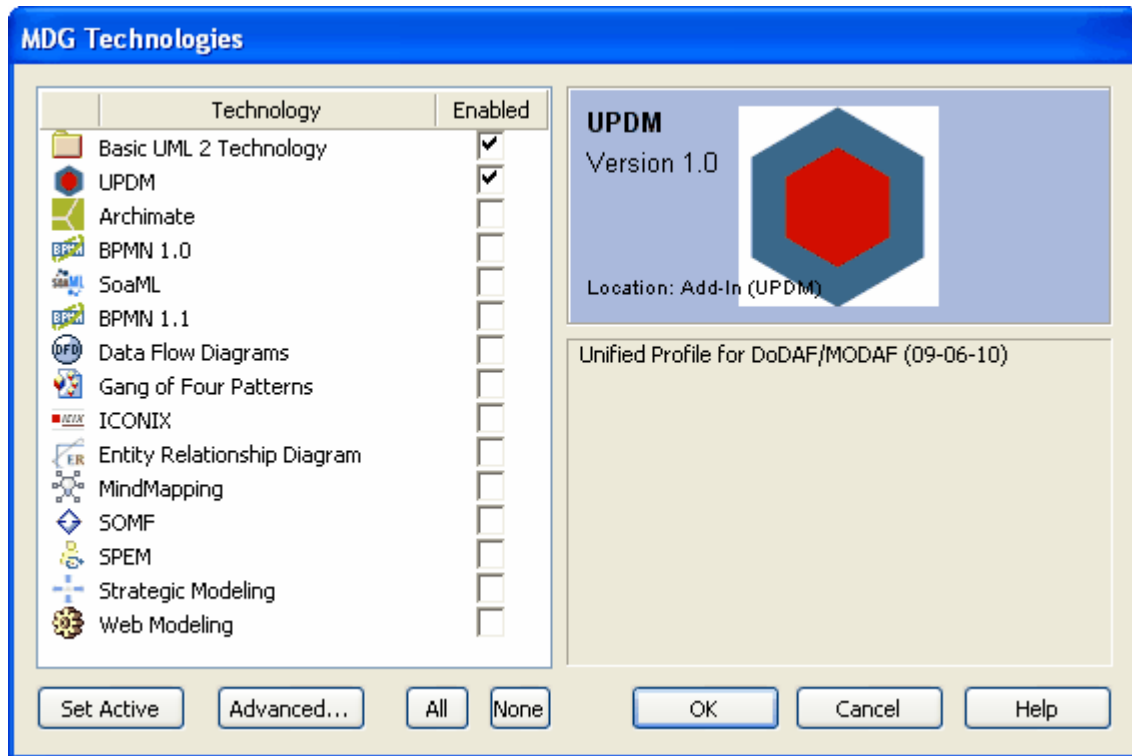
### UPDM Add-In

To display the UPDM Add-In menu, select the Enterprise Architect **Add-Ins | UPDM** menu option. The menu and the commands it offers are described in the [UPDM Add-In Menu](#) topic.

### UPDM Technology

Enterprise Architect makes use of MDG Technologies to extend its capabilities in specific domains. The UPDM Technology is automatically loaded by the UPDM Add-In if the Add-in is enabled.

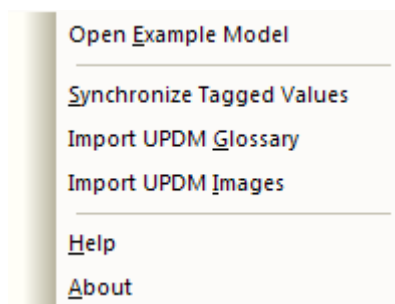
As well as enabling the Add-In, you must enable the MDG Technology. This happens automatically the first time you run Enterprise Architect after installing the UPDM Add-In, but thereafter you can enable and disable the MDG Technology as required, and its state is persisted to the next time you run Enterprise Architect. To enable or disable an MDG Technology, select the **Settings | MDG Technologies** menu option. The **MDG Technologies** dialog displays.



Either set or clear the checkbox next to **UPDM** and click on the **OK** button. You must restart Enterprise Architect for changes to take effect. You might also want to select the UPDM Technology and click on the **Set Active** button. Making a technology active opens all **Toolbox** pages that have been defined as *Common*. The UPDM Technology has a common **Toolbox** page named **All UPDM Stereotypes**, which lists all stereotypes in alphabetical order; this can be very useful for finding stereotypes if you don't know which View they are commonly used in.

### 3.2 UPDM Add-In Menu

You can perform certain tasks on your UPDM model using the **UPDM Technology** menu, which you display by selecting the **Add-Ins | UPDM** menu option in Enterprise Architect.

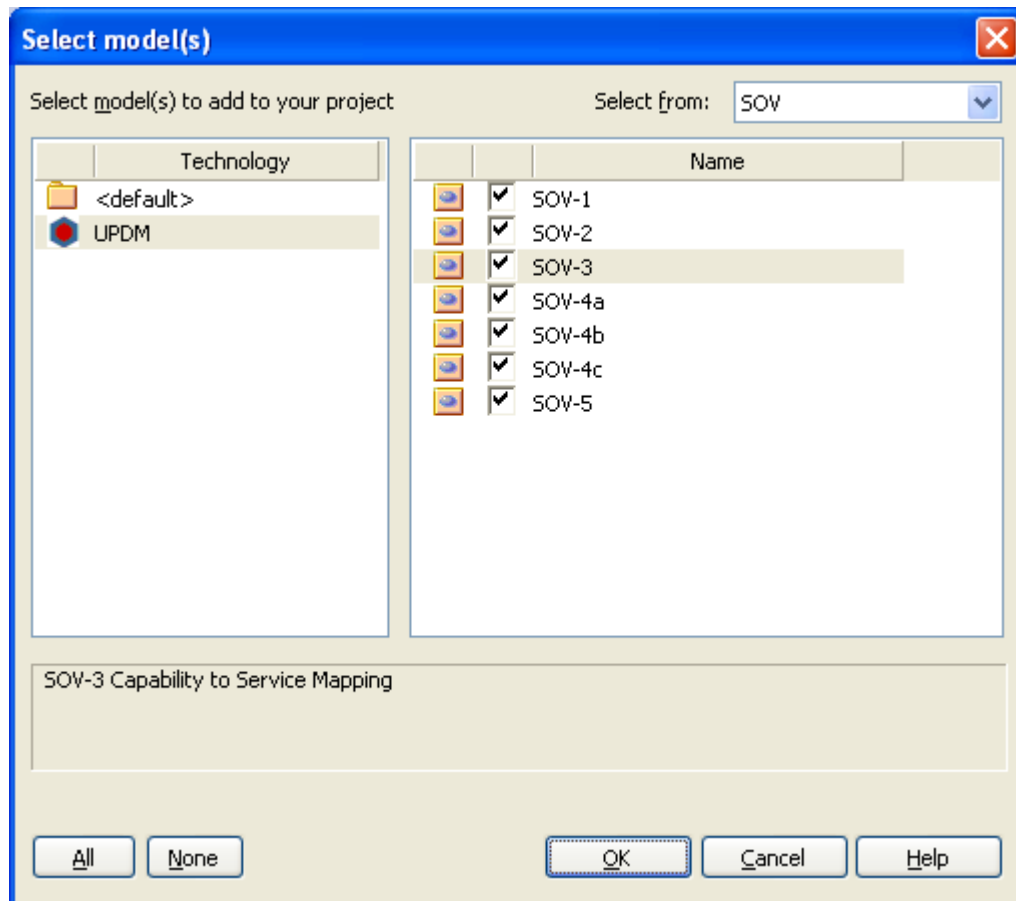


Command	Use to
Open Example Model	Open the UPDM example model; for more details, see the <a href="#">Example Model</a> <sup>[68]</sup> topic.
Synchronize Tagged Values	Add missing Tagged Values to all elements in the model that require them; for more details, see the <a href="#">Tagged Values</a> <sup>[55]</sup> topic.
Import UPDM Glossary	Import UPDM information into the Enterprise Architect Glossary; for more details, see the <a href="#">Glossary</a> <sup>[67]</sup> topic.

Command	Use to
<b>Import UPDM Images</b>	Import the alternate images (as used in the <a href="#">UPDM Framework Diagram</a> <sup>66</sup> and the <a href="#">Example Model</a> <sup>68</sup> ) into the current model. You may use these images to decorate your own models (select a diagram object, right-click and <b>Appearance   Select Alternate Image</b> ) or you may prefer to design your own.
<b>Help</b>	Open this Help file.
<b>About</b>	Show the version of the MDG Technology for UPDM that you are using. The version number has the format <b>1.0.001</b> , where <i>1.0</i> is the version of the UPDM specification that is supported, and <i>001</i> is the incremental build number.

### 3.3 Model Wizard

To use the Model Wizard to load a UPDM template, right-click on a Package in the Enterprise Architect **Project Browser** and select the **Add | Add a New Model using Wizard** context menu option. The **Select Models** dialog displays.



Option	Use to
<b>Technology</b>	Select UPDM to access the UPDM model templates.
<b>Select From</b>	With the UPDM Technology selected, select between showing all templates for all views, or just the templates for the AV, AcV, OV, SV, SOV, StV or TV views.
<b>Name</b>	Select from the available templates by ticking the checkbox against each named template to import; click on a name to show a brief description in the text box at the bottom of the dialog.

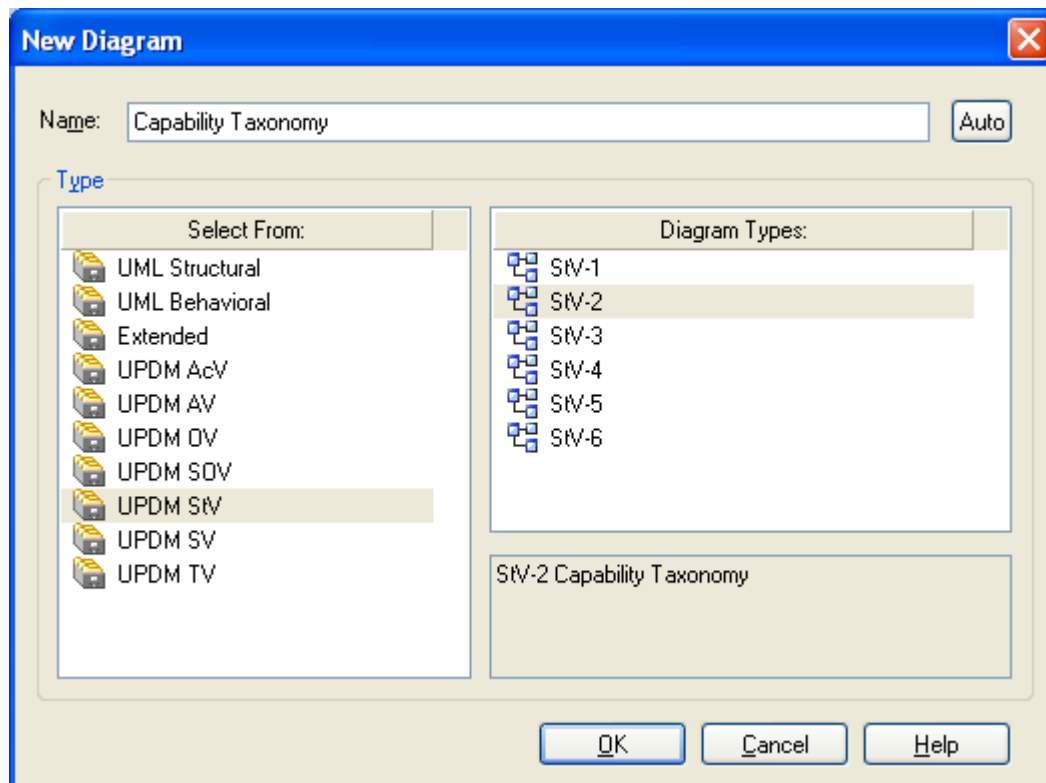


Option	Use to
All	Select all of the listed templates.
None	Clear all templates selected.
OK	Load the templates for your project.
Cancel	Abort the creation of templates.
Help	Display this Help topic for the dialog.

### 3.4 UPDM Diagram Types

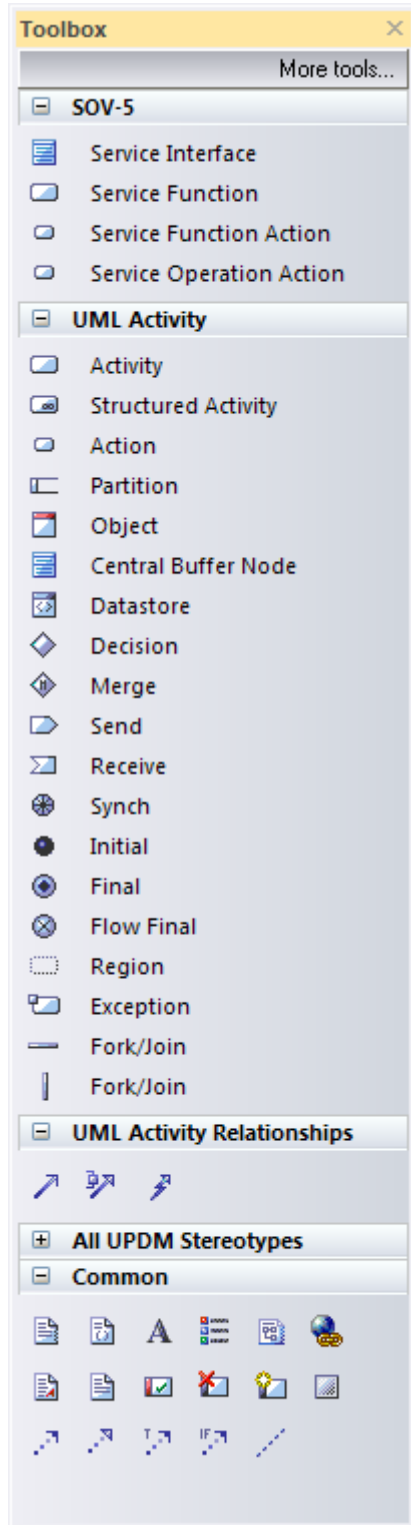
The MDG Technology for UPDM introduces a number of custom diagram types into Enterprise Architect. These are, for the most part, extended UML diagrams. On opening a UPDM diagram, Enterprise Architect automatically opens the appropriate UPDM toolbox pages for the view.

You can use the UPDM diagrams that are created by the [Model Wizard](#)<sup>12</sup>, or create a new UPDM diagram.



### 3.5 UPDM Toolboxes

When you open a diagram, Enterprise Architect opens the **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, as well as the **UML Activity** and **UML Activity Relationships** pages.



In addition, the **Common** elements and relationships page is always available, regardless of which diagram is open.

If you hide the default toolboxes and want to get them back, simply switch to the **Start Page** and back to the current diagram, and all the default toolboxes for the current diagram type are re-opened.

### All UPDM Stereotypes

For your convenience, a toolbox page is provided that includes every stereotype in the UPDM profile, in

alphabetical order. If you cannot remember which context-sensitive toolbox page a stereotype appeared in, just go to the [All UPDM Stereotypes](#) toolbox page instead. To make this page available at all times, either:

- Select the **Settings | MDG Technologies** menu option, select **UPDM Technology** in the table, and click on the **Set Active** button, or
- Select **UPDM Technology** from the list box on the **Default Tools** toolbar.

### 3.5.1 UPDM Stereotypes

#### ActualMeasurementSet

**Description:** A set or collection of measurements.

**Used in:** AV-3, OV-3, SV-6, SV-7.

**Extensions:** Object.

**Constraints:** Classifier must be a MeasurementSet.

**Usage:** Press **[Ctrl]+drag** a *MeasurementSet* from the **Project Browser** to create an instance, or drop an ActualMeasurementSet from the 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

**Description:** An actual specific organization, an instance of an organization Class.

**Used in:** AcV-1, OV-4, StV-5, TV-1, TV-2.

**Extensions:** Object.

**Generalizations:** ActualOrganizationalResource.

**Constraints:**

- Classifier must be an Organization.

**Usage:**

- Press **[Ctrl]+drag** an Organization from the **Project Browser** to create an instance, or drop an ActualOrganization from the 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

**Description:** 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

**Description:** A named individual that fulfills an ActualPost.

**Used in:** OV-4.

**Extensions:** Object.

**Constraints:**

- Classifier must be a Person.

**Usage:**

- Press **[Ctrl]**+drag a Person from the **Project Browser** to create an instance, or drop an ActualPerson from the toolbox and press **[Ctrl]**+**[L]** to set classifier.
- Can be client of a FillsPost dependency to an ActualPost.

### ActualPost

**Description:** An actual, specific post, an instance of the Post Class.

**Used in:** AcV-1, OV-4, StV-5.

**Extensions:** Object.

**Generalizations:** ActualOrganizationalResource.

**Constraints:**

- Classifier must be a Post.

**Usage:**

- Press **[Ctrl]**+drag a Post from the **Project Browser** to create an instance, or drop an ActualPost from the toolbox and press **[Ctrl]**+**[L]** to set the classifier.
- 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.
- Can be supplier of a FillsPost dependency from an ActualPerson.

### ActualProject

**Description:** A time-limited endeavour to create a specific set of products or services.

**Used in:** AcV-1, AcV-2, StV-3, StV-5, SV-8.

**Extensions:** Object.

**Constraints:**

- Classifier must be a Project.

**Usage:**

- Press **[Ctrl]**+drag a Project from the **Project Browser** to create an instance, or drop an ActualProject from the 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, DeployedMilestone).

### ActualProjectMilestone

**Description:** An event in a project by which progress is measured. See also: IncrementMilestone, OutOfServiceMilestone, NoLongerUsedMilestone, DeployedMilestone.

**Used in:** AcV-2, StV-3, StV-5, SV-8.

**Extensions:** Object.

**Specializations:** IncrementMilestone, OutOfServiceMilestone, NoLongerUsedMilestone, DeployedMilestone.

**Constraints:**

- Classifier must be a ProjectMilestone.

**Usage Can::**

- Have a set of associated Resource.
- Be client/supplier of a MilestoneSequence.

## Alias

**Description:** A comment used to define an alternative name for an element.

**Used in:** AV-2.

**Extensions:** Note.

**Constraints:**

- AnnotatedElement must be a UPDMElement.

**Usage:**

- Just drag a quicklink NoteLink from the Alias to the annotated element.

## Arbitrary Relationship

**Description:** 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.

**Usage:**

- Drag a quicklink from a ConceptRole.

## ArchitecturalDescription

**Description:** A specification of a system of systems at a technical level, which also provides the business context.

**Used in:** AV-1.

**Extensions:** Package.

**Usage:** Can:

- Have a DefinesArchitecture realization to an EnterprisePhase.
- Have an ArchitecturalReference dependency to another ArchitecturalDescription.
- Be annotated with an ArchitectureMetadata note.

## ArchitecturalReference

**Description:** Asserts that one architectural description refers to another.

**Used in:** AV-1.

**Extensions:** Dependency.

**Constraints:**

- Client and Supplier must both be stereotyped ArchitecturalDescription.

**Usage:**

- Drag a quicklink from an ArchitecturalDescription.

## ArchitectureMetadata

**Description:** Information on architectural description.

**Used in:** AV-1.

**Extensions:** Note.

**Generalizations:** Metadata.

**Constraints:**

- AnnotatedElement must be an ArchitecturalDescription.

**Usage:**

- Drag a quicklink from an ArchitecturalDescription.

## Capability

**Description:** 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, SV-9.

**Extensions:** Class.

**Generalizations:** SubjectOfForecast.

**Usage:**

- 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

**Description:** 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, AcV-2.

**Extensions:** Class.

**Generalizations:** Resource, ConceptItem, Performer, ResourceInteractionItem, SubjectOfResourceConstraint, SubjectOfForecast, SystemsElement, SubjectOfResourceStateMachine, ResourceInteractionItem.

**Specializations:** SystemsNode.

**Usage:** 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, OperationalActivity) (from Performer).

## Climate

**Description:** A type of weather condition, or combination of weather conditions, in which a Performer performs.

**Used in:** StV-2.

**Extensions:** Class.

**Generalizations:** EnvironmentalType.

**Usage:**

- Can be the type of an EnvironmentProperty.

## Commands

**Description:** Asserts that one OrganizationalResource commands another.

**Used in:** OV-4, SV-1, SV-10c.

**Extensions:** InformationFlow.

**Generalizations:** ResourceInteraction.

**Constraints:**

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

**Usage:**

- Conveys a DataElement.

## CompatibleWith

**Description:** 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, PhysicalLocation).

**Usage:**

- Drag a quicklink from a Node.

## Competence

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

**Used in:** OV-4, SV-1, SV-9.

**Extensions:** Class.

**Generalizations:** SubjectOfForecast.

**Usage:** Can be:

- Supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast).
- Supplier of a ProvidesCompetence dependency.
- Supplier of a RequiresCompetence dependency.

## ConceptRole

**Description:** 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.

**Usage:**

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

**ConfigurationExchange**

**Description:** CapabilityConfigurations that are exchanged between Nodes.

**Used in:** OV-2, OV-3, OV-6c.

**Extensions:** InformationFlow.

**Generalizations:** OperationalExchange.

**Constraints:**

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

**Usage:**

- Conveys a CapabilityConfiguration.

**Controls**

**Description:** A type of ResourceInteraction where one Resource controls another.

**Used in:** SV-1, SV-10c.

**Extensions:** InformationFlow.

**Generalizations:** ResourceInteraction.

**Constraints:**

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

**Usage:**

- Conveys a DataElement.

**DataElement**

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

**Used in:** OV-4, SV-1, SV-2, SV-4, SV-6, SV-10a, SV-10b, SV-11

**Extensions:** Class.

**Generalizations:** SubjectOfResourceConstraint, ResourceInteractionItem, SystemsElement, SubjectOfResourceStateMachine.

**Usage:**

- 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**

**Description:** A DoDAF alias for ResourceInteraction.

**Extensions:** InformationFlow.

**Generalizations:** ResourceInteraction, SystemsElement.

**Usage:**

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



## DefinesArchitecture

**Description:** Establishes a relationship between ArchitecturalDescription and EnterprisePhase.

**Used in:** AV-1.

**Extensions:** Realization.

**Constraints:**

- Client is an ArchitecturalDescription.
- Supplier is an EnterprisePhase.

**Usage:**

- Drag a quicklink from an ArchitecturalDescription.

## Definition

**Description:** A definition of an element in the architecture.

**Used in:** AV-2.

**Extensions:** Note.

**Constraints:**

- Annotated Element is a UPDMElement.

**Usage:**

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

## DeployedMilestone

**Description:** 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).

**Usage:**

- can have a set of associated ("usedBy") ActualOrganizationalResource (ActualOrganization, ActualPost).
- can have a set of associated Resource (from ActualProjectMilestone).
- can be client/supplier of a MilestoneSequence (from ActualProjectMilestone).

## EnduringTask

**Description:** A type of behavior recognised by an enterprise as being essential to achieving its goals - i.e. a strategic specification of what the enterprise does.

**Used in:** StV-1.

**Extensions:** Class

**Usage:**

- Target of association from EnterprisePhase.

## Energy

**Description:** Energy to be exchanged between Nodes.

**Used in:** OV-2, OV-3, OV-5, SV-1, SV-4, SV-6.

**Extensions:** Class.

**Generalizations:** ResourceInteractionItem, OperationalExchangeItem.

**Usage:**

- Conveyed on a EnergyExchange information flow.

### EnergyExchange

**Description:** A relationship specifying the need to exchange energy between nodes.

**Used in:** OV-2, OV-3, OV-6c.

**Extensions:** InformationFlow.

**Generalizations:** OperationalExchange, OperationalElement.

**Constraints:**

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

**Usage:**

- Conveys a class stereotyped Energy.

### EnterpriseGoal

**Description:** A specific required objective of the enterprise that the architecture represents.

**Used in:** StV-1.

**Extensions:** Class.

**Usage:**

- has an association to one EnterprisePhase.

### EnterprisePhase

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

**Used in:** AV-1, StV-1, StV-2, StV-5, SV-9.

**Extensions:** Class.

**Specializations:** WholeLifeEnterprise.

**Usage:**

- 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

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

**Used in:** StV-1.

**Extensions:** Class.

**Usage:**

- has an association to one EnterprisePhase.

### EntityAttribute

**Description:** A defined property of an EntityItem.

**Used in:** OV-7, SV-11.

**Extensions:** Attribute.

**Usage:**

- is owned by an EntityItem.

## EntityItem

**Description:** A definition (type) of an item of interest.

**Used in:** OV-7, SV-11.

**Extensions:** Class.

**Constraints:**

- owned attributes must be stereotyped EntityAttribute.

**Usage:**

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

## EntityRelationship

**Description:** Asserts that there is a relationship between two EntityItem.

**Used in:** OV-7, SV-11.

**Extensions:** Association.

**Constraints:**

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

## Environment

**Description:** A definition of the conditions in which the Enterprise exists or functions.

**Used in:** AV-1, StV-2.

**Extensions:** Class.

**Constraints:**

- Owned attributes must be EnvironmentProperty.

## EnvironmentProperty

**Description:** Asserts that an Environment has one or more properties. These may be Climate, Location or LightCondition.

**Used in:** StV-2.

**Extensions:** Attribute.

**Constraints:**

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

**Usage:**

- Owned by an Environment element.

## Equipment

**Description:** 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, Post).
- Type must be a ResourceArtifact.

**Usage:**

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

### ExhibitsCapability

**Description:** 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

**Description:** Assertion that a service interface exposes a capability.

**Extensions:** Dependency.

**Constraints:**

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

### ExternalIndividual

**Description:** An individual defined by an external ontology.

**Used in:** AV-2.

**Extensions:** Object

**Usage:**

- can be the supplier of a SameAs dependency.

### ExternalNode

**Description:** Operational node that is external to the architecture.

**Used in:** OV-2.

**Extensions:** Class.

**Generalizations:** Node, Performer.

**Usage:**

- can own a RequestPoint port. (from Node)
- can own a ServicePoint port. (from Node)
- can be client of an ExhibitsCapability dependency to a Capability. (from Node)
- can have a Performs dependency to a PerformedActivity (Function, OperationalActivity) (from Performer).
- can have a CompatibleWith dependency to a ReferredLocation (PhysicalLocation, Location) (from Node).

## ExternalType

**Description:** A type defined by an external ontology.

**Used in:** AV-2.

**Extensions:** Class.

**Usage:**

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

## FieldedCapability

**Description:** A deployed and fully realized instance of a capability.

**Used in:** SV-2.

**Extensions:** Object.

**Constraints:**

- its classifier must be a CapabilityConfiguration.

## FillsPost

**Description:** Asserts that ActualPerson fills an ActualPost.

**Used in:** OV-4.

**Extensions:** Dependency.

**Constraints:**

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

## Forecast

**Description:** 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, Software)
- Client and Supplier must be the same specialization of SubjectOfForecast.

## Function

**Description:** An activity which is specified in context of the resource that performs it.

**Used in:** OV-4, SV-1, SV-4, SV-5, SV-10a.

**Extensions:** Activity.

**Generalizations:** PerformedActivity, SystemsElement, SubjectOfResourceConstraint.

**Constraints:**

- owned parameters are FunctionParameter.

**Usage:**

- can be Supplier of a Performs dependency. (from PerformedActivity)
- can own ServiceOperationAction, FunctionAction, FunctionEdge.
- can be Client of an ImplementsOperational dependency to an OperationalActivity. (from SystemsElement)

- can have an attached ResourceConstraint (from SubjectOfResourceConstraint).

### FunctionAction

**Description:** 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.

**Usage:**

- Ctrl+L to set the function.

### FunctionEdge

**Description:** 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.

**Usage:**

- may realize a ResourceInteraction (right-click > Advanced > Information Flows Realized).

### FunctionParameter

**Description:** 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, Software).

**Usage:**

- owned by a Function.

### HighLevelOperationalConcept

**Description:** A generalized model for operations.

**Used in:** OV-1.

**Extensions:** Class.

**Constraints:**

- owned attributes are ConceptRole.

**Usage:**

- can have a set of described Mission.

### HostedSoftware

**Description:** 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.

**Usage:**

- can have a RequiresCompetence dependency to a Competence. (from ResourceRole)
- can have a set of associations to "used" Functions. (from ResourceRole)

## HumanResource

**Description:** 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, Post).

**Usage:**

- can have a RequiresCompetence dependency to a Competence. (from ResourceRole)
- can have a set of associations to "used" Functions. (from ResourceRole)

## ImplementsOperational

**Description:** 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

**Description:** 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, SV-8.

**Extensions:** Object.

**Generalizations:** ActualProjectMilestone.

**Constraints:**

- classifier must be a ProjectMilestone (from ActualProjectMilestone).

**Usage:**

- 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

**Description:** Information exchanged between nodes.

**Used in:** OV-2, OV-3, OV-5, OV-6a, OV-6b, OV-7.

**Extensions:** Class.

**Generalizations:** OperationalExchangeItem, SubjectOfOperationalConstraint, SubjectOfOperationalStateMachine, OperationalElement.

**Usage:**

- can have a set of associations with (represented by) EntityItem classes.
- can be conveyed on an InformationExchange - right-click > Advanced > Information Items Conveyed.
- can have an attached OperationalConstraint (from SubjectOfOperationalConstraint)
- can own an OperationalStateMachine (from SubjectOfOperationalStateMachine)

## InformationExchange

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

**Used in:** OV-2, OV-3, OV-6c.

**Extensions:** InformationFlow.

**Generalizations:** OperationalExchange.

**Constraints:**

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

## InternalDataModel

**Description:** DoDAF alias for PhysicalDataModel.

**Used in:** SV-11.

**Extensions:** Package.

**Generalizations:** PhysicalDataModel, DataModel.

**Constraints:**

- owns EntityItem elements. (from DataModel)

## KnownResource

**Description:** 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, ResourceArtifact).
- Class must be a NodeParent (Node, LogicalArchitecture) (from NodeChild)

## LightCondition

**Description:** A specification of environmental lighting conditions.

**Used in:** StV-2.

**Extensions:** Class.

**Generalizations:** EnvironmentalType.

**Usage:**

- can be the type of an EnvironmentProperty (from EnvironmentalType).



## Location

**Description:** A general specification of the surroundings/scenario in which an operation may take place. Examples would be "desert", "arctic", "at sea" etc.

**Used in:** OV-1, OV-2.

**Extensions:** Class.

**Generalizations:** ReferredLocation, ConceptItem, EnvironmentalType.

**Usage:**

- can be supplier to a CompatibleWith dependency from a Node (from ReferredLocation).
- can be type of a ConceptRole (from ConceptItem)
- can be the type of an EnvironmentProperty (from EnvironmentalType).

## LogicalArchitecture

**Description:** A composite structure model whose parts are either NodeRoles, ProblemDomains, or KnownResources.

**Used in:** OV-2.

**Extensions:** Class.

**Generalizations:** NodeParent.

**Usage:**

- can own ProblemDomain properties.

## LogicalDataModel

**Description:** 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

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

**Extensions:** Dependency.

**Used in:** StV-6.

**Constraints:**

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

## MaterielExchange

**Description:** Materiel that is exchange between Nodes.

**Used in:** OV-2, OV-3, OV-6c.

**Extensions:** InformationFlow.

**Generalizations:** OperationalExchange.

**Constraints:**

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

**Usage:**

- can convey a ResourceArtifact or Software.

**Measurement**

**Description:** A category of measures.

**Used in:** AV-3, OV-2, SV-7.

**Extensions:** Attribute.

**Specializations:** PerformanceParameter.

**Usage:**

- owned by a MeasurementSet class.

**MeasurementSet**

**Description:** A set or collection of Measurements.

**Used in:** AV-3, OV-3, SV-7.

**Extensions:** Class.

**Constraints:**

- owned attributes must be Measurement.

**Usage:**

- has a set of associations with ("measuredElement") UPDMElement.
- is classifier of ActualMeasurementSet object.

**Metadata**

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

**Used in:** AV-1.

**Extensions:** Note.

**Specializations:** ArchitectureMetadata.

**MilestoneSequence**

**Description:** A relationship between two milestones.

**Used in:** AcV-2, SV-8.

**Extensions:** Dependency.

**Constraints:**

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

**Mission**

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

**Used in:** AV-1, OV-1, OV-6a, OV-6b.

**Extensions:** UseCase.

**Generalizations:** SubjectOfOperationalConstraint, SubjectOfOperationalStateMachine.

**Usage:**

- Fulfilled by an EnterprisePhase.
- can have an attached OperationalConstraint (from SubjectOfOperationalConstraint)
- can own an OperationalStateMachine (from SubjectOfOperationalStateMachine)

## MovementOfPeople

**Description:** MODAF alias for OrganizationalExchange.

**Extensions:** InformationFlow.

**Generalizations:** OrganizationalExchange.

**Usage:**

- Conveys an OrganizationalResource (Organization, Post).

## Needline

**Description:** Documents the requirement to exchange information between nodes.

**Used in:** OV-2, 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).

**Usage:**

- Realizes an OperationalExchange. Create a Needline between the same two elements as an OperationalExchange, then right-click the Needline and "Advanced > Information Flows Realized".

## NoLongerUsedMilestone

**Description:** 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)

**Usage:**

- 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

**Description:** Logical entity that performs operational activities.

**Used in:** OV-1, OV-2, OV-3, OV-5, OV-6a, OV-6b, OV-6c.

**Extensions:** Class.

**Generalizations:** Performer, ConceptItem, NodeParent, SubjectOfOperationalConstraint, SubjectOfOperationalStateMachine, OperationalElement.

**Specializations:** OperationalNode.

**Constraints:**

- owned ports must be NodePort, RequestPoint or ServicePoint.

**Usage:**

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

### NodePort

**Description:** 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, System).

**Usage:**

- owned by a Node.
- can be the ends of a Needline.

### NodeRole

**Description:** Used to link a parent Node to its sub-nodes.

**Used in:** OV-2, OV-3, OV-6c.

**Extensions:** Part.

**Generalizations:** NodeChild.

**Specializations:** ProblemDomain.

**Constraints:**

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

### OperationalActivity

**Description:** 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, SV-5.

**Extensions:** Activity.

**Generalizations:** PerformedActivity, SubjectOfOperationalConstraint, OperationalElement, SubjectOfOperationalStateMachine.

**Specializations:** StandardOperationalActivity.

**Constraints:**

- owned parameters must be OperationalParameter.

**Usage:**

- can be Supplier of a Performs dependency (from PerformedActivity).
- can be Supplier of an OwnsProcess dependency.

- can be the Activity/Behavior of an OperationalActivityAction.
- can be the owner of an OperationalActivityEdge.
- can have an attached OperationalConstraint (from SubjectOfOperationalConstraint)
- can be the Supplier of a SupportsOperationalActivity dependency.
- can own an OperationalStateMachine (from SubjectOfOperationalStateMachine)

### OperationalActivityAction

**Description:** A call behavior action that invokes the activity that needs to be performed.

**Used in:** OV-5.

**Extensions:** CallBehaviorAction.

**Constraints:**

- Activity/Behavior must be an OperationalActivity.

**Usage:**

- can be the Source or Target of an OperationalActivityEdge.

### OperationalActivityEdge

**Description:** 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.

**Usage:**

- can have a set of OperationalExchange (ConfigurationExchange, EnergyExchange, InformationExchange, MaterielExchange, OrganizationalExchange) information flows which it realizes.
- can carry a set of OperationalExchangeItem (Post, Organization, ResourceArtifact, System)

### OperationalConstraint

**Description:** 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, Mission).

### OperationalMessage

**Description:** Message for use in an Operational Event Trace which carries any of the subtypes of OperationalExchange.

**Used in:** OV-6c.

**Extensions:** Message.

**Generalizations:** OperationalElement.

**Usage:**

- can have a set of OperationalExchange (ConfigurationExchange, EnergyExchange, InformationExchange, MaterielExchange, OrganizationalExchange) information flows which it realizes.

## OperationalNode

**Description:** 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.

**Usage:**

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

## OperationalParameter

**Description:** Represents inputs and outputs of an operational activity.

**Used in:** OV-5.

**Extensions:** ActivityParameter.

**Constraints:**

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

**Usage:**

- can be owned by an OperationalActivity.

## OperationalRule

**Description:** A DoDAF alias for OperationalConstraint.

**Extensions:** Constraint.

**Generalizations:** OperationalConstraint.

**Constraints:**

- constrained element must be a SubjectOfOperationalConstraint (OperationalActivity, Node, InformationElement, Mission). (from OperationalConstraint)

## OperationalStateMachine

**Description:** A state machine describing an operational behavior or property.

**Used in:** OV-6b.

**Extensions:** StateMachine.

**Constraints:**

- owner is SubjectOfOperationalStateMachine (Mission, InformationElement, Node).

## Organization

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

**Used in:** OV-4, SV-1, SV-3, SV-9, SV-10a, SV-12.

**Extensions:** Class.

**Generalizations:** OrganizationalResource, Resource, Performer, SubjectOfForecast, SubjectOfResourceConstraint.

**Usage:**

- can be classifier to an ActualOrganization.
- can be source or target of a Commands information flow (from OrganizationalResource).
- can be the owning class of a PostRole.
- can be the class or type of a SubOrganization.
- can be the class of an Equipment (from OrganizationalResource).
- can be conveyed by an OrganizationalExchange (from OrganizationalResource).
- can be the type of a HumanResource (from OrganizationalResource).
- can be the source of a Controls information flow (from OrganizationalResource).
- can have a set of associated milestones, stereotyped ActualProjectMilestone (from Resource).
- can be client of a RealizesCapability realization to a Capability (from Resource).
- can be client of a ProvidesCompetence dependency to a Competence (from Resource).
- can have an attached ResourceConstraint (from Resource, SubjectOfResourceConstraint).
- can be supplier or client of a Forecast dependency (both must have same stereotype) (from SubjectOfForecast).
- can own a ServicePoint (from Resource).
- can own a RequestPoint (from Resource).
- can own a ResourcePort (from Resource).
- can be source and target of a ResourceInteraction (from Resource).
- can own a ServiceOperation (from Resource).
- can be type of a KnownResource (from Resource).
- can be type of a ResourceRole (from Resource).
- can have a Performs dependency to a PerformedActivity (Function, OperationalActivity) (from Performer).

## OrganizationalExchange

**Description:** A relationship specifying flow of people across organizations.

**Used in:** OV-2, OV-3, OV-6c.

**Extensions:** InformationFlow.

**Generalizations:** OperationalExchange.

**Specializations:** MovementOfPeople.

**Constraints:**

- conveyed element must be an OrganizationalResource (Organization, Post).
- Source is a Node (from OperationalExchange)
- Target is a Node (from OperationalExchange)

## OutOfServiceMilestone

**Description:** A project milestone that indicates a project's deliverable is to go out of service.

**Used in:** AcV-2, StV-3, SV-8.

**Extensions:** Object

**Generalizations:** ActualProjectMilestone.

**Constraints:**

- Classifier must be a ProjectMilestone.

**Usage:**

- 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**

**Description:** 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, ActualOrganization).
- Supplier must be an OperationalActivity.

**Part**

**Description:** Usage 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.

**Usage:**

- can have a RequiresCompetence dependency to a Competence. (from ResourceRole)
- can have a set of associations to "used" Functions. (from ResourceRole)

**PerformanceParameter**

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

**Extensions:** Attribute.

**Generalizations:** Measurement.

**Usage:**

- owned by a MeasurementSet class.

**Performs**

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

**Used in:** OV-2, OV-3, OV-4, OV-5, SV-1, SV-4.

**Extensions:** Dependency.

**Constraints:**

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



## Person

**Description:** A type of human being.

**Used in:** OV-4.

**Extensions:** Class.

**Usage:**

- can be Classifier of an ActualPerson.

## PhysicalDataModel

**Description:** An implementable specification of a data structure.

**Used in:** SV-11.

**Extensions:** Package.

**Generalizations:** DataModel.

**Specializations:** InternalDataModel.

**Constraints:**

- owns EntityItem elements. (from DataModel).

## PhysicalLocation

**Description:** Anywhere that can be specified.

**Used in:** OV-1, OV-2.

**Extensions:** Class.

**Generalizations:** ReferredLocation, ConceptItem, EnvironmentalType.

**Usage:**

- can be supplier to a CompatibleWith dependency from a Node (from ReferredLocation).
- can be type of a ConceptRole (from ConceptItem)
- can be the type of an EnvironmentProperty (from EnvironmentalType).

## Platform

**Description:** Usage 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.

**Usage:**

- can have a RequiresCompetence dependency to a Competence. (from ResourceRole)
- can have a set of associations to "used" Functions. (from ResourceRole)

## Post

**Description:** A type of point of contact or responsible person.

**Used in:** OV-4, SV-1, SV-3, SV-9, SV-10a, SV-12.

**Extensions:** Class.

**Generalizations:** OrganizationalResource, Resource, Performer, SubjectOfForecast, SubjectOfResourceConstraint.

**Usage:** 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

**Description:** Asserts that a post exists in an organization.

**Used in:** OV-4, SV-1

**Extensions:** Part.

**Generalizations:** OrganizationRole, ResourceRole.

**Constraints:**

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

**Usage:** Can have a:

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

## ProblemDomain

**Description:** 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

**Description:** Used to define a category of project.

**Used in:** AcV-1.

**Extensions:** Class.

**Usage:** Can:

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

## ProjectMilestone

**Description:** A type of project milestone.

**Used in:** AcV-2.

**Extensions:** Class.

**Constraints:**

- Owned attributes must be ProjectTheme.

**Usage:** Can:

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

## ProjectSequence

**Description:** 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

**Description:** An aspect by which the progress of various projects can be measured.

**Used in:** AcV-2.

**Extensions:** Attribute.

**Constraints:**

- Type must be a ProjectThemeStatus.

**Usage:**

- Owned by ProjectMilestone.

## ProjectThemeStatus

**Description:** Specifies a status for a ProjectTheme.

**Extensions:** Class.

**Usage:**

- The type of a ProjectTheme.

## Protocol

**Description:** A standard for communication.

**Used in:** SV-2, TV-1, TV-2.

**Extensions:** Class.

**Generalizations:** Standard, SubjectOfForecast.

**Usage:** 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

**Description:** Asserts that a protocol uses another protocol.

**Used in:** TV-1, TV-2.

**Extensions:** Attribute.

**Constraints:**

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

### ProvidesCompetence

**Description:** Asserts that a resource provides a competence.

**Used in:** OV-4.

**Extensions:** Dependency.

**Constraints:**

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

### RealizesCapability

**Description:** Asserts that a resource provides a capability.

**Used in:** SOV-3, StV-3, StV-5, SV-1.

**Extensions:** Realization.

**Constraints:**

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

### RequestPoint

**Description:** The mechanism by which a Service communicates.

**Used in:** OV-2, SV-1.

**Extensions:** Port.

**Constraints:**

- Type must be a ServiceInterface.

**Usage:**

- Can be owned by a Node or a Resource.

### RequiresCompetence

**Description:** 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

**Description:** A type of man-made object.

**Used in:** OV-2, OV-3, OV-5, SV-1, SV-3, SV-9, SV-10a, SV-12.

**Extensions:** Class.

**Generalizations:** OperationalExchangeItem, ManufacturedResourceType, Resource, SubjectOfForecast, ResourceInteractionItem, Performer, SubjectOfResourceConstraint.

**Specializations:** System.

**Usage:** 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 ManufacturedResourceType).
- 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).

## ResourceComponent

**Description:** 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.

**Usage:** Can have a:

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

## ResourceConnector

**Description:** 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.

**Usage:** Can:

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

## ResourceConstraint

**Description:** 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, Organization).

## ResourceInteraction

**Description:** Represents data that is exchanged between resources.

**Used in:** OV-4, SOV-4c, SV-1, SV-2, SV-3, SV-4, SV-6, 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, System).
- Source must be a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact, System).
- Target must be a Resource (Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact, System).

**Usage:**

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

## ResourceInterface

**Description:** A contractual agreement between two resources that implement protocols.

**Used in:** OV-4, SV-1, SV-2, SV-3, SV-6.

**Extensions:** Association, Connector.

**Generalizations:** SystemsElement.

**Specializations:** SystemConnector.

**Constraints:**

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

**Usage:**

- Can realize a ResourceInteraction.

### ResourceMessage

**Description:** Message for use in a Resource event trace, implements a ResourceInteraction.

**Used in:** SV-10c.

**Extensions:** Message.

**Generalizations:** SystemsElement.

**Usage:**

- Can have a set of ResourceInteraction that it carries.

### ResourcePort

**Description:** 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, DataElement).

**Usage:**

- Can be owned by a Resource.
- Has an association to a Protocol class it "implements" (from ProtocolImplementation).
- Can be the end role of a ResourceConnector.

### ResourceStateMachine

**Description:** 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

**Description:** 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

**Description:** A property of a ServiceInterface that allows performance, reliability and cost values to be captured.

**Used in:** SOV-1.

**Extensions:** Attribute.

**Usage:**

- Owned by a ServiceInterface.

### ServiceFunction

**Description:** Describes the abstract behavior of ServiceOperations, regardless of the actual implementation.

**Used in:** SOV-5.

**Extensions:** Activity.

**Usage:** Can:

- Be the behavior of a ServiceFunctionAction.
- Be the activity of a ServiceOperationAction.
- Own ServicePoint ports.

### ServiceFunctionAction

**Description:** A call behavior action that invokes the ServiceFunction that needs to be performed.

**Used in:** SOV-5.

**Extensions:** CallBehaviorAction.

**Constraints:**

- Behavior must be a ServiceFunction.

### ServiceInteraction

**Description:** Interaction for a service interface.

**Used in:** SOV-4c.

**Extensions:** Interaction.

### ServiceInterface

**Description:** 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, SOV-5.

**Extensions:** Class.

**Constraints:**

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

**Usage:** 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

**Description:** Message for use in a service interaction specification, implements a resource interaction.

**Used in:** SOV-4c.

**Extensions:** Message.

**Usage:**



- Can carry a set of ResourceInteractions.

## ServiceOperation

**Description:** Provides the access point for invoking the behavior of a provided service.

**Used in:** SOV-2. SOV-5.

**Extensions:** Operation.

**Constraints:**

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

**Usage:** Can:

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

## ServiceOperationAction

**Description:** 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.

**Usage:**

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

## ServicePoint

**Description:** The mechanism by which a service communicates.

**Used in:** OV-2, SV-1, SV-12.

**Extensions:** Port.

**Constraints:**

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

**Usage:**

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

## ServicePolicy

**Description:** A constraint governing the consumers and providers of services.

**Used in:** SOV-4a.

**Extensions:** Constraint.

**Usage:**

- Rule can be owned by a ServiceInterface.

## ServiceStateMachine

**Description:** UPDM artifact that extends UML StateMachine.

**Used in:** SOV-4b.

**Extensions:** StateMachine.

## Software

**Description:** Software needed for the functioning of the system.

**Used in:** OV-2, OV-3, SV-1, SV-3, SV-9, SV-10a, SV-12.

**Extensions:** Class.

**Generalizations:** ManufacturedResourceType, Resource, SubjectOfForecast, ResourceInteractionItem, Performer, SubjectOfResourceConstraint.

**Usage:** Can:

- Be conveyed on a MaterielExchange information flow.
- Be type of HostedSoftware.
- Be the target of a Controls flow (from ManufacturedResourceType).
- 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, OperationalActivity) (from Performer).

## Standard

**Description:** A ratified set of rules which are used to guide and/or constrain any UPDM element.

**Used in:** SV-9, TV-1, TV-2.

**Extensions:** Class.

**Generalizations:** SubjectOfForecast.

**Specializations:** Protocol.

**Usage:**

- 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

**Description:** A comment, attached to a CapabilityConfiguration, which indicates that the annotated CapabilityConfiguration is a standard pattern for re-use in the architecture.

**Used in:** TV1, TV-2.

**Extensions:** Note.

**Constraints:**

- The annotated element must be a CapabilityConfiguration.

## StandardOperationalActivity

**Description:** An OperationalActivity that is a standard procedure and that is doctrinal.

**Used in:** OV-5, StV-6.

**Extensions:** Activity.

**Generalizations:** OperationalActivity, PerformedActivity, SubjectOfOperationalConstraint, OperationalElement, SubjectOfOperationalStateMachine.

**Constraints:**

- Owned parameters must be OperationalParameter (from OperationalActivity).

**Usage:** 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

**Description:** 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.

**Usage:**

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

## StructuralPart

**Description:** 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

**Description:** Asserts that one type of organization is typically the parent of another.

**Used in:** OV-4, SV-1.

**Extensions:** Part.

**Generalizations:** OrganizationRole, ResourceRole.

**Constraints:**

- Type must be an Organization.
- Class must be an Organization.

**Usage:** Can:

- Have a RequiresCompetence dependency to a Competence (from ResourceRole).

- Have a set of associations to 'used' Functions (from ResourceRole).

### SubSystemPart

**Description:** Indicates that a sub system 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).

**Usage:** Can:

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

### SupportsOperationalActivity

**Description:** 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

**Description:** 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, ManufacturedResourceType, Resource, SubjectOfForecast, ResourceInteractionItem, Performer, SubjectOfResourceConstraint.

**Usage:** 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 ManufacturedResourceType).
- 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).

## SystemConnector

**Description:** 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).

**Usage:**

- Can realize a ResourceInteraction (from ResourceInterface).

## SystemFunction

**Description:** A DoDAF alias for Function.

**Extensions:** Activity.

**Generalizations:** Function, PerformedActivity, SystemsElement, SubjectOfResourceConstraint.

**Constraints:**

- Owned parameters are FunctionParameter (from Function).

**Usage:** Can:

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

## SystemFunctionAction

**Description:** A DoDAF alias for FunctionAction.

**Extensions:** CallBehaviorAction.

**Generalizations:** FunctionAction.

**Constraints:**

- Activity is stereotyped Function (from FunctionAction).

**Usage:**

- Press **[Ctrl]+[L]** to set the function (from FunctionAction).

## SystemFunctionEdge

**Description:** A DoDAF alias for FunctionEdge.

**Extensions:** ControlFlow.

**Generalizations:** FunctionEdge, SystemsElement.

**Constraints:**

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

**Usage:**

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

## SystemsNode

**Description:** A DoDAF alias for CapabilityConfiguration.

**Extensions:** Class.

**Generalizations:** CapabilityConfiguration, Resource, ConceptItem, Performer, ResourceInteractionItem, SubjectOfResourceConstraint, SubjectOfForecast, SystemsElement, SubjectOfResourceStateMachine, ResourceInteractionItem.

**Usage:** 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

**Description:** 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, Software) (from Forecast).
- Client and Supplier must be the same specialization of SubjectOfForecast (from Forecast).

## TemporalPart

**Description:** 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

**Description:** 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.

**Usage:** Can:

- Have a RequiresCompetence dependency to a Competence (from ResourceRole).
- Have a set of associations (usedFunctions) to Function (from ResourceRole).

## VisionStatement

**Description:** A high-level textual description of an EnterpriseVision.

**Extensions:** Note.

## WholeLifeEnterprise

**Description:** A purposeful endeavor of any size involving people, organizations and supporting systems.

**Used in:** AV-1, StV-1.

**Extensions:** Class.

**Generalizations:** EnterprisePhase.

**Usage:** 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).

### 3.5.2 Abstract Stereotypes

#### ActualOrganizationalResource

**Description:** An actual organization or post.

**Specializations:** ActualOrganization, ActualPost.

#### ConceptItem

**Description:** An item that might feature in a high level operational concept.

**Specializations:** CapabilityConfiguration, Node, ReferredLocation, Resource.

#### DataModel

**Description:** A structured specification of data, showing classifications of data elements and relationships between them.

**Specializations:** LogicalDataModel, PhysicalDataModel.

#### EnvironmentalType

**Description:** A type of environment.

**Specializations:** LightCondition, Location, PhysicalLocation, Climate.

### ManufacturedResourceType

**Description:** A resource artifact or software.

**Generalizations:** Resource.

**Specializations:** ResourceArtifact, Software.

### NodeChild

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

**Specializations:** NodeRole, ProblemDomain, KnownResource.

### NodeParent

**Description:** Represents the owners/context of composite structure at the operational level.

**Specializations:** Node, ExternalNode, OperationalNode, LogicalArchitecture.

### OperationalElement

**Description:** Elements relating to operational models.

**Specializations:** OperationalActivity, StandardOperationalActivity, OperationalMessage, Node, ExternalNode, OperationalNode, Needline, OperationalExchange, InformationElement, OperationalActivityEdge.

### OperationalExchange

**Description:** Describes the characteristics of an exchanged item, such as the content, format (voice, imagery, text and message format, etc.), throughput requirements, security or classification level, timeliness requirement, and the degree of interoperability.

**Generalizations:** OperationalElement.

**Specializations:** ConfigurationExchange, EnergyExchange, InformationExchange, MaterielExchange, OrganizationalExchange.

### OperationalExchangeltem

**Description:** An item that participates in an operational exchange.

**Specializations:** Post, Organization, ResourceArtifact, System.

### OrganizationalResource

**Description:** Either an organization or a post.

**Generalizations:** Resource, OperationalExchangeltem.

**Specializations:** Post, Organization.

### OrganizationRole

**Description:** Represents properties in an organization that are typed by another organization or a post.

**Generalizations:** ResourceRole.

**Specializations:** SubOrganization, PostRole.

### PerformedActivity

**Description:** A behavior that can be performed by a Performer.

**Specializations:** OperationalActivity, Function.



## Performer

**Description:** A structural element that can perform behaviors (i.e. PerformedActivity)

**Specializations:** Node, Resource.

## ProtocolImplementation

**Description:** An element that implements a specific protocol.

**Specializations:** ResourcePort, ResourceInteraction, Controls, Commands, DataExchange, ResourceConnector.

## ReferredLocation

**Description:** Either an actual location or a type of location (i.e. environment) at/in which operations may be conducted.

**Generalizations:** ConceptItem, EnvironmentalType.

**Specializations:** Location, PhysicalLocation.

## Resource

**Description:** 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

**Description:** Represents the items exchanged between resources through a resource interaction.

**Specializations:** Energy, Resource, DataElement.

## ResourceRole

**Description:** Defines the usage of any resource in the system.

**Specializations:** UsedConfiguration, Equipment, SubOrganization, PostRole, Part, SubSystemPart, HumanResource, ResourceComponent, Platform, HostedSoftware.

## SubjectOfForecast

**Description:** Any element that may be subject to a forecast.

**Specializations:** Standard, Protocol, Capability, Competence, Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact, System.

## SubjectOfOperationalConstraint

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

**Specializations:** OperationalActivity, InformationElement, Node, Mission.

## SubjectOfOperationalStateMachine

**Description:** The element being described by the state machine.

**Specializations:** OperationalActivity, InformationElement, Node, Mission.

## SubjectOfResourceConstraint

**Description:** Anything that may be constrained by a ResourceConstraint.

**Specializations:** Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact, System, DataElement, Function.

### SubjectOfResourceStateMachine

**Description:** The element being described by the state machine.

**Specializations:** Post, Organization, CapabilityConfiguration, SystemsNode, Software, ResourceArtifact, System, DataElement.

### SystemsElement

**Description:** Elements relating to system models.

**Specializations:** Resource, ResourceInteraction, ResourceMessage, ResourceInteraction, DataElement, ResourceStateMachine, FunctionEdge, Function.

### UPDMElement

**Description:** Super type for all UPDM elements. It provides a means of extending UPDM elements in a common way.

**Specializations:** all UPDM stereotypes.

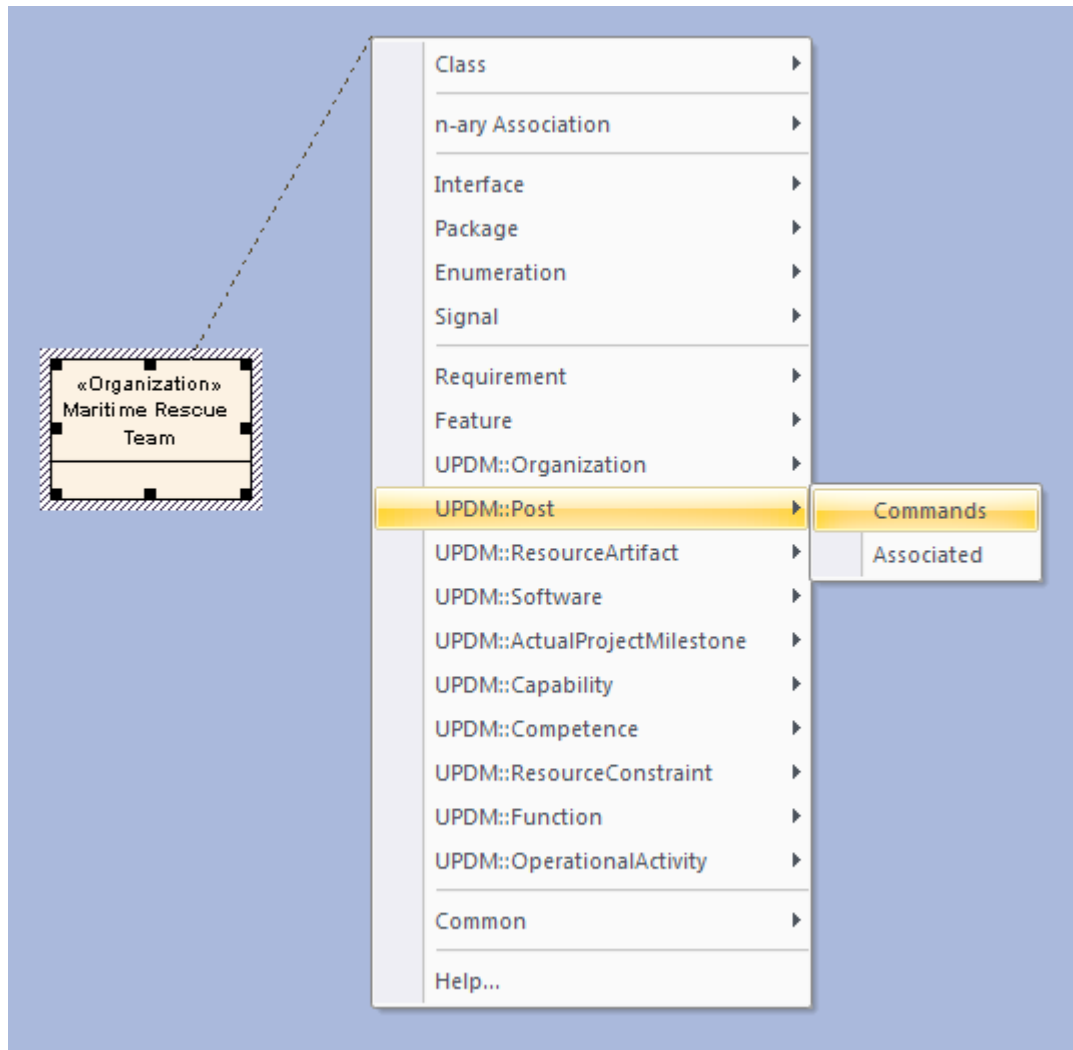
## 3.6 Quicklinks

The MDG Technology for UPDM makes use of Enterprise Architect's *Quicklink* feature to make it quicker and easier to create correct and self-consistent UPDM models.

To see an element's Quicklink pointer, select the element. In the following diagram, the Quicklink pointer is the upwards-pointing arrow off the top-right corner of the element.



Drag the Quicklink pointer away from the element and release it over empty diagram space; the Quicklink context menu displays. This context menu shows all the UPDM elements that would commonly be attached to the element, as in the following diagram.



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

### 3.7 Tagged Values

UPDM is an extension of UML. UML is extended by applying stereotypes to elements and these stereotypes can apply tagged values. The tagged values can provide additional information to that normally associated with a UML element. Since UPDM makes frequent use of tagged values, it is worthwhile keeping the **Tagged Values** window opened and docked at all times.

#### Synchronizing Tagged Values

The list of tagged values owned by an element can get "out of date". Tagged values can be deleted by the user, or a new version of the UML Profile might define new or modified tagged values for an element type. Also, you might apply the stereotype using the stereotype combo box which doesn't add tagged values. If you wish to refresh the list of tagged values for a single element, you can drag and drop the stereotype from the **Toolbox** onto the element and choose "Apply". This only works for diagram objects, not connectors, and it only works for single elements. If you want to refresh the list of tagged values for every element in your model, choose the command **Add-Ins | UPDM | Synchronize Tagged Values**.

#### 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 the MDG Technology for UPDM, this tagged value has been omitted and you are asked to use the standard Enterprise Architect functionality to achieve the same result:

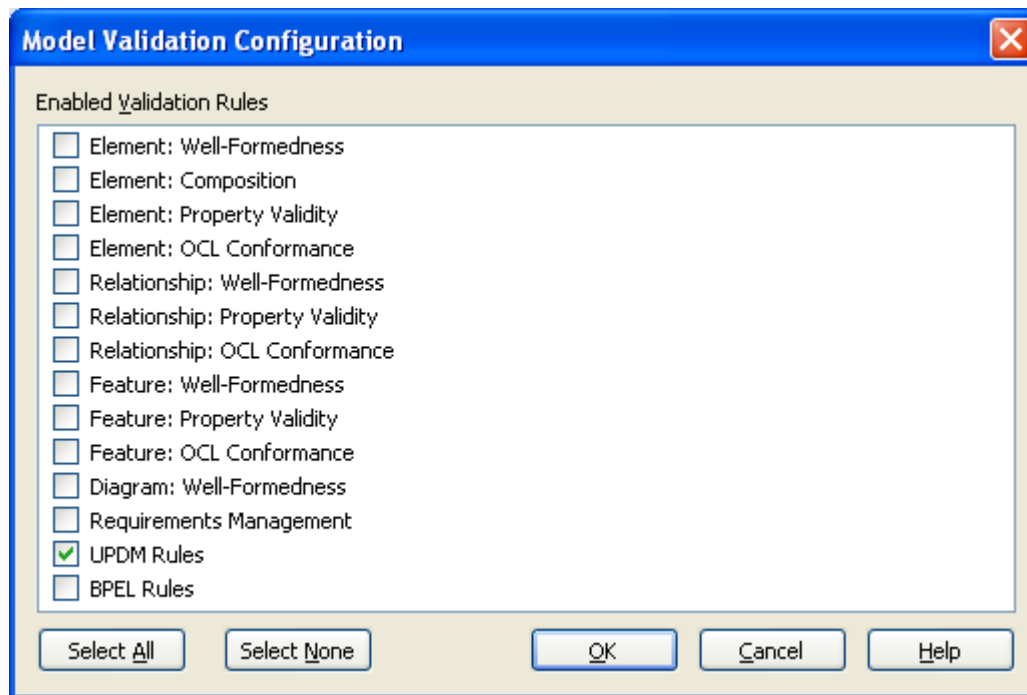
open the Properties dialog for any element, navigate to the Files tab, and enter a web location.

### 3.8 Model Validation

The MDG Technology for UPDM provides model validation of UPDM models, validating and reporting errors against more than 160 different rules.

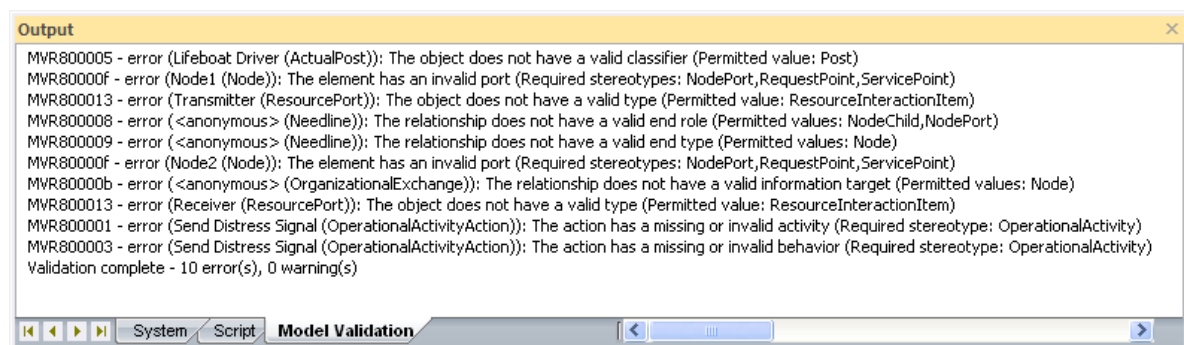
#### Configure Model Validation

Before validating a model, you should first indicate which rules to validate against. Select the **Project | Model Validation | Configure** option from Enterprise Architect's main menu and switch off all of the rules except for the UPDM ones, as below:



#### Perform Model Validation

To perform model validation, simply open a diagram or select a package or select a number of elements and either select the **Project | Model Validation | Validate Selected** menu option or press **[Ctrl]+[Alt]+[V]**. Validation results are displayed in the **Output** window, which is opened if it isn't already displayed. To go to the element that caused a validation error, double-click the error message in the **Output** window.



### 3.8.1 Model Validation Rules

Errors will have an error code similar to "MVRxxnnnn" where "xx" will be "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" will be a hexadecimal number from "0001" to "0013" as indicated below.

#### MVRxx0001 - activity

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

**Description:** This validation rule checks for stereotyped Action elements that they are owned by an Activity with the required stereotype. In the following table, the first column lists the Action stereotypes and the second column lists the Activity stereotypes.

FunctionAction	Function
OperationalActivityAction	OperationalActivity
ServiceOperationAction	Function
ServiceOperationAction	ServiceFunction

**Solution:** The solution to this error is to locate the action in the **Project Browser**, locate an activity with one of the named stereotypes (or their specializations) or create a new one, and drag the action to the activity.

#### MVRxx0002 - annotatedElement

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

**Description:** This validation rule checks for stereotyped Note elements that they are attached (with a NoteLink connector) to an element with the required stereotype. In the following table, the first column lists the Note stereotypes and the second column lists the annotated element stereotypes.

Alias	UPDMElement
ArchitectureMetadata	ArchitecturalDescription
Definition	UPDMElement
StandardConfiguration	CapabilityConfiguration
StereotypeExtension	UPDMElement

**Solution:** The solution to this error is to 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 by deleting the NoteLink connector and creating a new one with the quicklinker.

#### MVRxx0003 - behavior

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

**Description:** This validation rule checks for stereotyped CallBehaviorAction elements that they call a Behavior with the required stereotype. In the following table, the first column lists the Action stereotypes and the second column lists the Behavior stereotypes.

OperationalActivityAction	OperationalActivity
ServiceFunctionAction	ServiceFunction

**Solution:** The solution to this error is to select the action, right-click it and select **Advanced | Set Behavioral Classifier**, or press **Ctrl+L**, and select a behavior element with the named stereotype (or one of its specializations).

#### MVRxx0004 - class

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

**Description:** This validation rule checks for stereotyped Property elements (parts or attributes) that they are

owned by a class with the required stereotype. In the following table, the first column lists the Property stereotypes and the second column lists the 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

**Solution:** The solution to this error is to locate the property in the **Project Browser**, locate a class with one of the named stereotypes (or their specializations) or create a new one, and drag the property to the class.

### MVRxx0005 - classifier

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

**Description:** This validation rule checks for stereotyped instance elements (objects) that they are classified by elements with the required stereotypes. In the following table, the first column lists the Object stereotypes and the second column lists the Classifier stereotypes.

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

**Solution:** The solution to this error is to 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).

### MVRxx0006 - client

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

**Description:** This validation rule checks for stereotyped Dependency or Realization relationships that their client (source) elements have the required stereotypes. In the following table, the first column lists the relationship stereotypes and the second column lists the 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

**Solution:** The solution to this error is to drag the end of the relationship without the arrowhead to a different element, one with the named stereotype (or one of its specializations).

### MVRxx0007 - constrainedElement

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

**Description:** This validation rule checks for stereotyped Constraint elements that they are attached (by a NoteLink) to elements with the required stereotypes. In the following table, the first column lists the Constraint stereotypes and the second column lists the constrained element stereotypes.

OperationalConstraint	SubjectOfOperationalConstraint
ResourceConstraint	SubjectOfResourceConstraint

**Solution:** The solution to this error is to 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 with the quicklinker.

### MVRxx0008 - endRoles

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

**Description:** This validation rule checks for stereotyped Association or Connector relationships that the elements at both ends of the relationship have the required stereotypes. In the following table, the first column lists the relationship stereotypes and the second column lists the end element stereotypes.

Needline	NodeChild
Needline	NodePort

ResourceConnector	ResourcePort
ResourceInterface	ResourceRole

**Solution:** The solution to this error is to drag one or both ends of the relationship to different elements with the named stereotype (or one of its specializations).

### MVRxx0009 - endType

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

**Description:** This validation rule checks for stereotyped Connectors that the elements (Objects or Parts) at both ends of the relationship are typed by the required stereotypes. In the following table, the first column lists the Connector stereotypes and the second column lists the end type stereotypes.

EntityRelationship	EntityItem
Needline	Node
ResourceInterface	Resource

**Solution:** The solution to this error is to drag one or both ends of the relationship to different elements which have types with the named stereotype (or one of its specializations).

### MVRxx000a - informationSource

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

**Description:** This validation rule checks for stereotyped InformationFlow relationships that their source elements have the required stereotypes. In the following table, the first column lists the InformationFlow stereotypes and the second column lists the source element stereotypes.

ActualOrganizationRelationship	ActualOrganizationalResource
Commands	OrganizationalResource
Controls	OrganizationalResource
OperationalExchange	Node
ResourceInteraction	Resource

**Solution:** The solution to this error is to drag the end of the information flow without the arrowhead to a different element, one with the named stereotype (or one of its specializations).

### MVRxx000b - informationTarget

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

**Description:** This validation rule checks for stereotyped InformationFlow relationships that their target elements have the required stereotypes. In the following table, the first column lists the InformationFlow stereotypes and the second column lists the target element stereotypes.

ActualOrganizationRelationship	ActualOrganizationalResource
Commands	OrganizationalResource
Controls	OrganizationalResource
OperationalExchange	Node
ResourceInteraction	Resource

**Solution:** The solution to this error is to drag the end of the information flow with the arrowhead to a different element, one with the named stereotype (or one of its specializations).



### MVRxx000c - ownedAttribute

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

**Description:** This validation rule checks for stereotyped Class elements that any attributes that they own have the required stereotypes. In the following table, the first column lists the Class stereotypes and the second column lists the attribute stereotypes.

EntityItem	EntityAttribute
Environment	EnvironmentProperty
HighLevelOperationalConcept	ConceptRole
MeasurementSet	Measurement
ProjectMilestoneType	ProjectTheme
ServiceInterface	ServiceAttribute

**Solution:** The solution to this error is to replace the attribute with one with the named stereotype (or one of its specializations).

### MVRxx000d - ownedOperation

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

**Description:** This validation rule checks for stereotyped Class elements that any operations that they own have the required stereotypes. In the following table, the first column lists the Class stereotypes and the second column lists the operation stereotypes.

ServiceInterface	ServiceOperation
------------------	------------------

**Solution:** The solution to this error is to replace the operation with one with the named stereotype (or one of its specializations).

### MVRxx000e - ownedParameter

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

**Description:** This validation rule checks for stereotyped Activity elements that any ActivityParameter elements that they own have the required stereotypes. In the following table, the first column lists the Activity stereotypes and the second column lists the ActivityParameter stereotypes.

Function	FunctionParameter
OperationalActivity	OperationalParameter

**Solution:** The solution to this error is to locate the activity parameter in the **Project Browser** and drag and drop it to an element with the appropriate stereotype and/or replace the activity parameter in its current owner with an activity parameter with the named stereotype.

### MVRxx000f - ownedPort

**Error Message:** "The element has an invalid port (Required stereotypes: <stereotypeList>)"

**Description:** This validation rule checks for stereotyped Class elements that any Ports that they own have the required stereotypes. In the following table, the first column lists the Class stereotypes and the second column lists the Port stereotypes.

Node	NodePort
Node	RequestPoint
Node	ServicePoint
Resource	RequestPoint

Resource	ResourcePort
Resource	ServicePoint

**Solution:** The solution to this error is to locate the port in the **Project Browser** and drag and drop it to an element with the appropriate stereotype and/or replace the port in its current owner with a port with one of the named stereotypes.

### MVRxx0010 - source

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

**Description:** This validation rule checks for stereotyped ActivityEdge connectors that their source elements have the required stereotypes. In the following table, the first column lists the ActivityEdge stereotypes and the second column lists the source element stereotypes.

FunctionEdge	ServiceOperationAction
OperationalActivityEdge	OperationalActivityAction

**Solution:** The solution to this error is to drag the end of the relationship without the arrowhead to a different element, one with the named stereotype (or one of its specializations).

### MVRxx0011 - supplier

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

**Description:** This validation rule checks for stereotyped Dependency or Realization relationships that their supplier (target) elements have the required stereotypes. In the following table, the first column lists the relationship stereotypes and the second column lists the 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

**Solution:** The solution to this error is to drag the end of the relationship with the arrowhead to a different element, one with the named stereotype (or one of its specializations).

**MVRxx0012 - target**

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

**Description:** This validation rule checks for stereotyped ActivityEdge connectors that their target elements have the required stereotypes. In the following table, the first column lists the ActivityEdge stereotypes and the second column lists the target element stereotypes.

FunctionEdge	ServiceOperationAction
OperationalActivityEdge	OperationalActivityAction

**Solution:** The solution to this error is to drag the end of the relationship with the arrowhead to a different element, one with the named stereotype (or one of its specializations).

**MVRxx0013 - type**

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

**Description:** This validation rule checks for stereotyped Property elements (parts or attributes) that their type elements have the required stereotypes. In the following table, the first column lists the Property stereotypes and the second column lists the 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

**Solution:** The solution to this error for parts is to select the part, right-click it and select **Advanced | Set**

**Property Type**, or press **Ctrl+L**, and select a type element with the named stereotype (or one of its specializations). The solution to this error for attributes is to open the Properties dialog for the attribute and select a type element with the named stereotype (or one of its specializations) in the appropriate data field.

### 3.9 Relationship Matrix

The Relationship Matrix is a spreadsheet display of relationships between model elements within packages. You select a source package and a target package, the relationship type and direction, and Enterprise Architect identifies all the relationships between source and target elements by highlighting a grid square and displaying an arrow indicating the direction of the relationship.

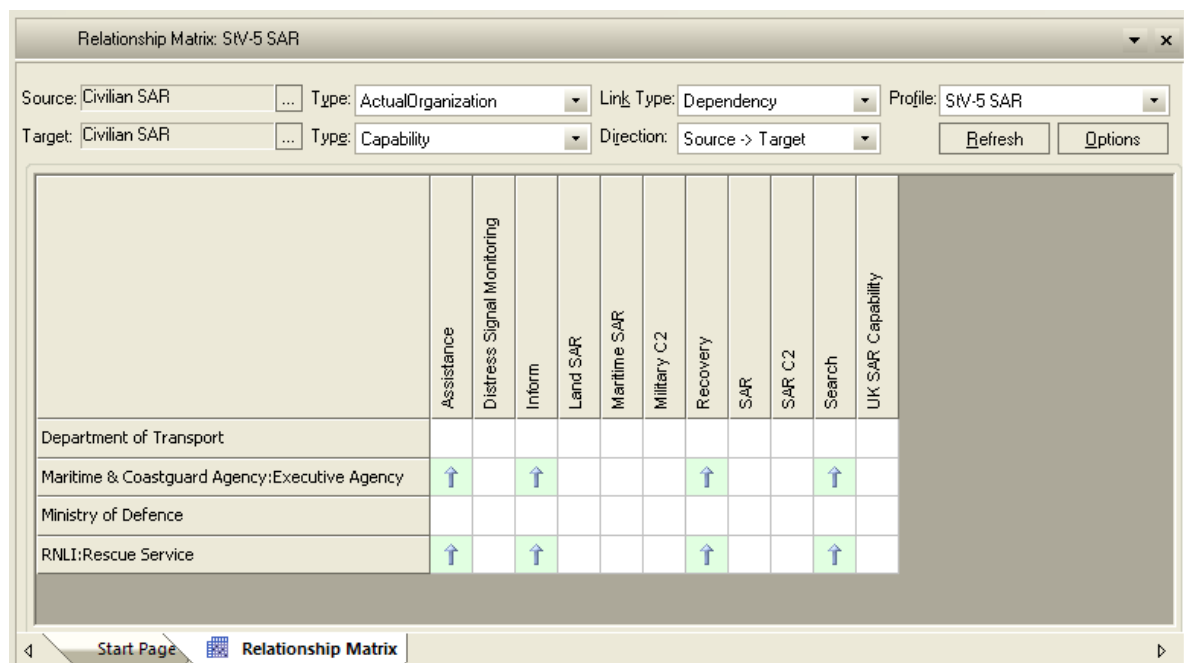
The Relationship Matrix is a convenient method of visualizing relationships quickly and definitively. It also enables you to create, modify and delete relationships between elements with a single mouse click - another quick way to set up complex sets of element relationships with a minimum of effort.

To open the Relationship Matrix you select **View | Relationship Matrix** from the Enterprise Architect main menu.

#### Setting up a Relationship Matrix

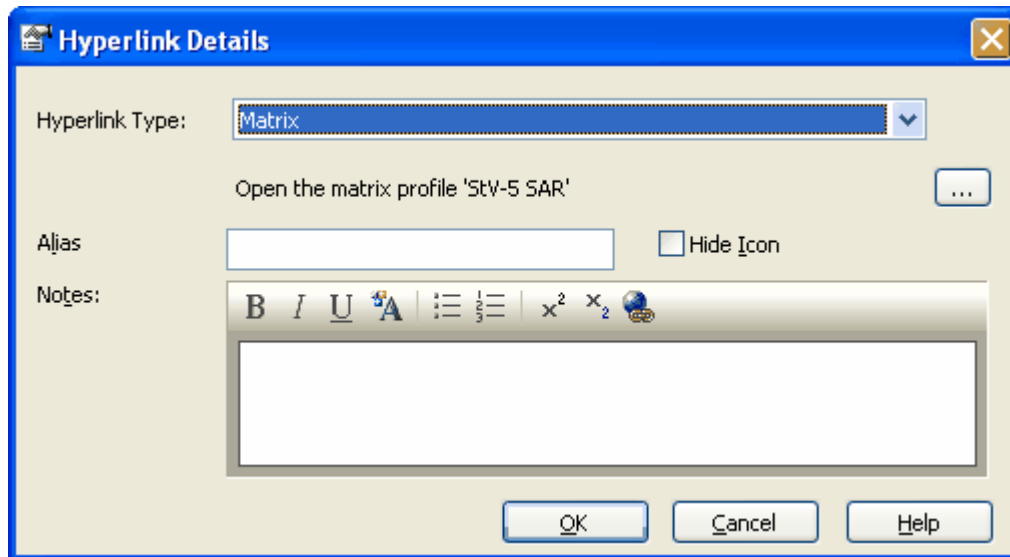
You define the information that is shown in a relationship matrix by filling in the following fields:

- **Source and Target.** Click the ellipsis (...) buttons next to either the Source or Target to choose a root package for the rows or columns respectively. This will limit the elements shown to only those in the chosen sub-tree.
- **Type.** Choose the type of element shown in the rows and columns. Note that you can choose **<All>** to show all elements in the chosen sub-tree.
- **Link Type.** Choose the type of relationship shown in the matrix.
- **Direction.** Choose the direction of relationship shown.



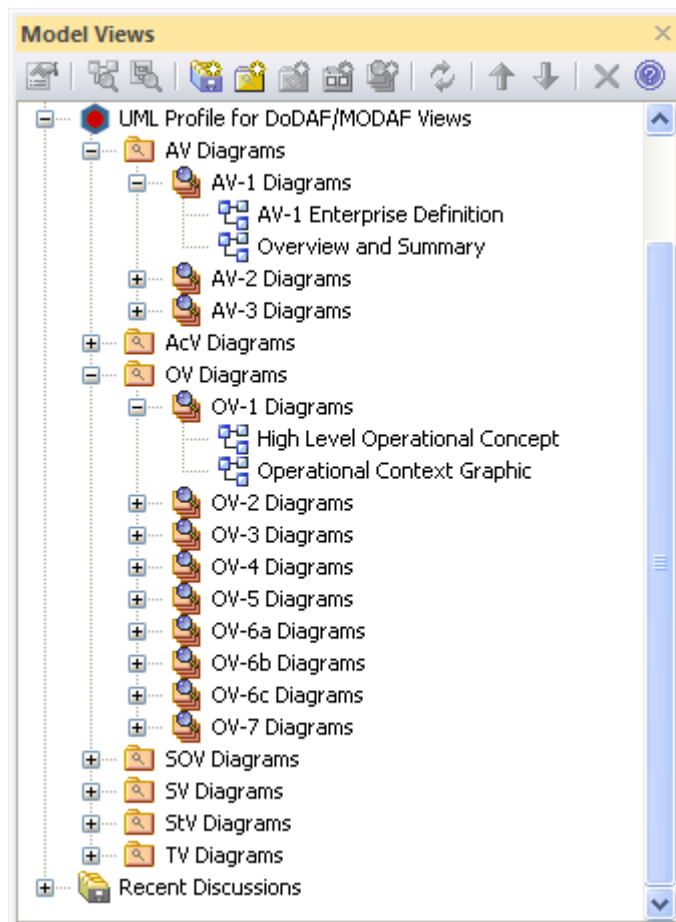
#### Saving a Relationship Matrix

Once you have a matrix configuration you are happy with and want to keep, you can save it by clicking the **Options** button and choosing the **Profiles | Save as New Profile** command. You can then reload the saved profile at any time by selecting it in the Profile listbox, or by creating a hyperlink to it. To create a hyperlink to a saved matrix profile, drop a hyperlink element onto a diagram and open its Properties dialog. Set the Hyperlink Type to **Matrix** and choose the required matrix profile from the list that pops up.



### 3.10 Model Views

The **Model Views** window displays a variety of different views on the model data, providing an alternative to the **Project Browser**. The MDG Technology for UPDM uses this window as a quick and easy method of locating all of your diagrams in the current model.



To open the **Model Views** window, select the **View | Model Views** menu option. Once you locate the required

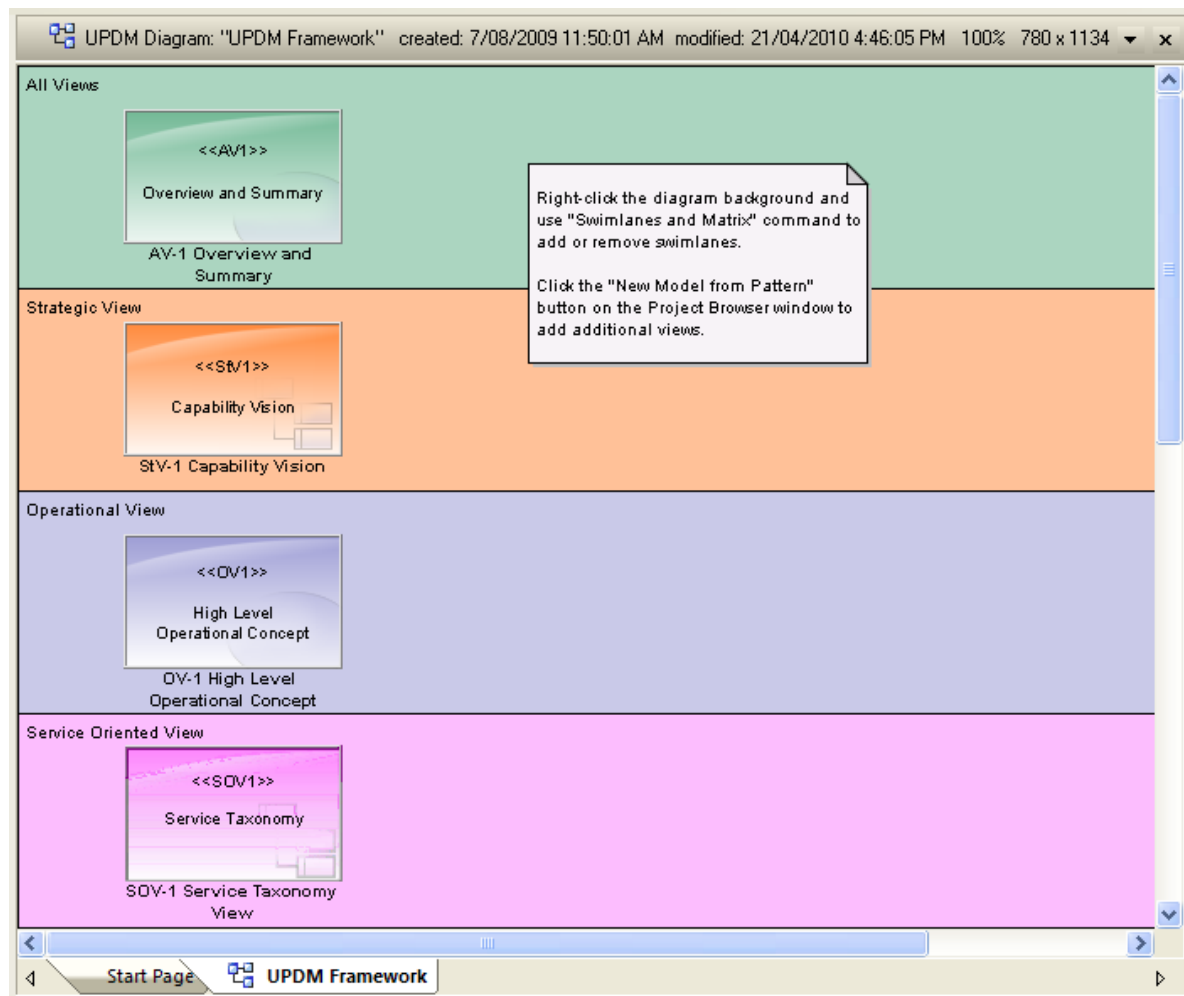
diagram, double-click on it to open it.

Note: This functionality is not available with OpenEdge repositories.

### 3.11 UPDM Framework Diagram

When developing and distributing a model, it is useful to have a single front page diagram which has hyperlinks to all the important information in the model. That is the aim of the UPDM Framework Diagram. It is created with color-coded swimlanes for each set of views, and you may drag and drop Packages (which act as hyperlinks to the documents that they own), Document Artifacts, Hyperlinks pointing to custom SQL queries, relationship matrix profiles, or external files, or any kind of composite element that points to its child diagram.

To create a UPDM Framework Diagram, click the New Model from Pattern button in the **Project Browser** toolbox.



#### Editing Swimlanes

You can add, remove and modify the swimlanes on the framework diagram. From the Enterprise Architect main menu, select **Diagram | Swimlanes and Matrix**. Click the Help button for assistance in using this dialog.

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 alternate image. To load these images into your model, from the main menu use the command **Add-Ins | UPDM | Import UPDM Images**. If you wish to apply your own bitmap image to these elements, you will first need to import your own images into the model using the **Settings | Images** command. Then you can either select the element and press **Ctrl +Shift+W** to add an alternate 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 the following shape script:

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

## 3.12 Glossary

The MDG Technology for UPDM add-in allows you to import descriptions of all UPDM stereotypes into EA's Glossary. This gives you a quick reference to the meaning of each stereotype, and lists the views that the stereotype may appear in and, for abstract stereotypes, lists the concrete stereotypes that inherit from the abstract stereotype.

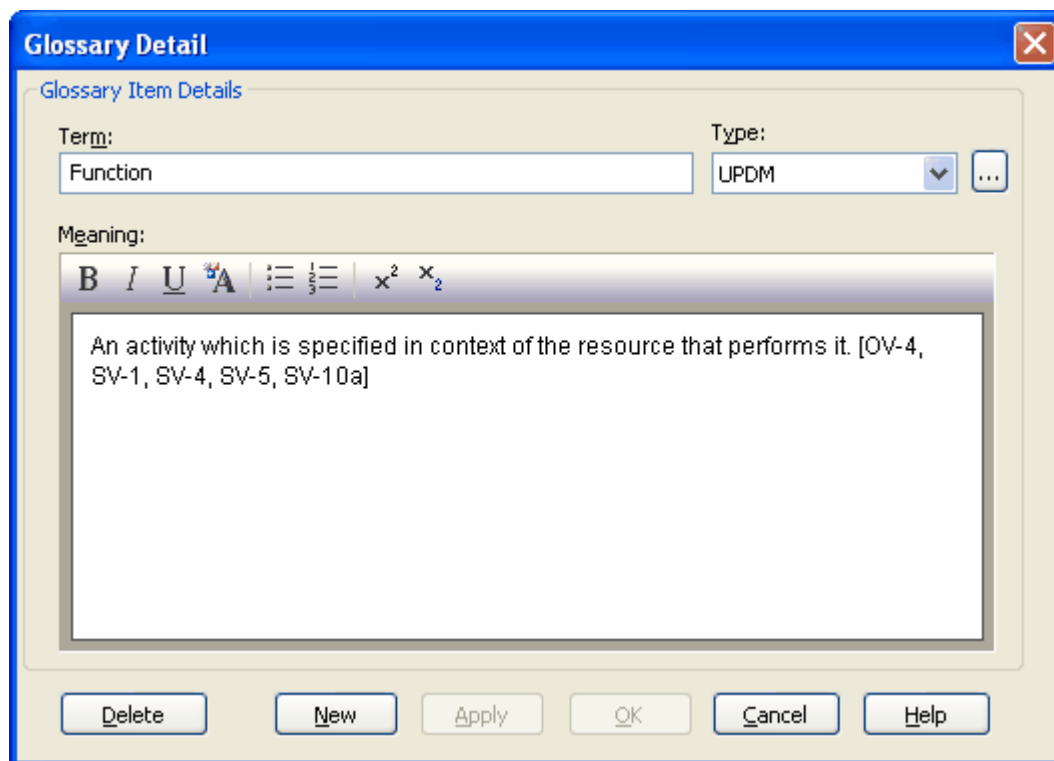
### Import Glossary

The Glossary definitions will need to be imported into each model individually. To do this, use the [Add-Ins | UPDM | Import UPDM Glossary](#) command from EA's main menu.

### Viewing the Glossary

There are a number of ways to view the Glossary.

1. From EA's main menu, use the [View | Other Project Tools | System](#) command to open the **System** docked window and click on the **Project Glossary** tab.
2. From EA's main menu, use the [Project | Documentation | Glossary](#) command to open the **Glossary** dialog.
3. From any Rich Text Format (RTF) notes field, click on any automatic Glossary hyperlink. These will be underlined and colored blue.



## 3.13 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, and CapabilityConfiguration/FieldedCapability.

If you have an element that is the classifier part of one of the above Classifier/Instance pairs, there are two main approaches to creating the instance: you can create the instance first and then set its classifier, or you can create the instance *from* the classifier. How to do that is described below:

1. **Setting classifier of an instance.** To set the classifier for an existing instance element, locate the instance element in a diagram, select it by clicking it once, and then either press Ctrl+L or right-click it and select the **Advanced | Instance Classifier** command. The same command sets the type of a port or part.
2. **Creating an instance from a classifier.** To create an instance from an existing classifier element, locate the classifier element in the Project Browser window and, while holding down the Ctrl key, drag it onto a diagram. When the **Paste Element** dialog appears, select the "Paste as Instance of Element" option. An anonymous instance will be created with the appropriate stereotype. You can select the instance and press F2 to give it a name.

### Setting run state of an object

Where a classifier may own a set of attributes, an instance of that classifier may own a slot for each attribute. The set of assigned values for these slots is known as the run state. You can set the run state of an object by right-clicking it on a diagram and selecting the **Advanced | Set Run State** command or pressing Ctrl+Shift+R.

Note that 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 EA so you won't find UPDM's slot-extending stereotypes in EA's implementation. UPDM Stereotypes that extend Slot are: ActualMeasurement (ActualMeasurementSet), ActualOrganizationRole (ActualOrganization), MeasureOfPerformance (ActualMeasurementSet), ProjectStatus (ActualProjectMilestone).

### Properties

Some stereotypes in UPDM have been 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 will be given the choice of creating an attribute, a part, or a port. These are all different representation of the UML Property metaclass, and 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, open the association's properties dialog by double-clicking it, go to either the Source Role or Target Role tab, click the "..." button next to the Stereotype field, select UPDM from the Profile field on the **Stereotype for Association** dialog, and choose all stereotypes that apply.

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.

## 3.14 Example Model

To open the UPDM Example Model, use the command **Add-Ins | UPDM | Open Example Model** from Enterprise Architect's main menu. The example shows the common civilian maritime search and rescue operation scenario of a yacht in distress. It is based on the Search and Rescue (SAR) example given in the UPDM specification document which was designed to illustrate how UPDM can support DoDAF and MODAF, providing a model that illustrates a sample of DoDAF and MODAF views addressing a given typical problem space.



## Framework Diagram

On opening the Example Model, you will be presented with the model's default diagram which is a [UPDM Framework Diagram](#)<sup>[66]</sup>. This diagram has colored swimlanes for each of All Views (AV), Strategic View (StV), Operational View (OV), Service Oriented View (SOV), Systems View (SV) and Acquisition View (AcV). In each swimlane are elements that act as hyperlinks to individual views. You may double-click on each element to go to its view. The elements represent the following:

- Most of the views are package elements and double-clicking one will open the package's child diagram.
- The element named "AV-1 Overview and Summary Information" is a document artifact element and double-clicking it opens the document in EA's document editor.
- The StV-5 and SOV-3 elements are hyperlinks to Relationship Matrix profiles and double-clicking them opens the [Relationship Matrix](#)<sup>[64]</sup>.
- Finally, the OV-3 element is a hyperlink to a custom SQL query, and double-clicking it executes the query and presents the results in a tabular form.

# Index

## - C -

Compiled 7 April 2010 2

## - D -

Diagrams  
MDG Technology For DoDAF-MODAF 13

## - M -

MDG Technology For DoDAF-MODAF  
 Acknowledgement Of Trademarks 6  
 Add-In Menu 11  
 Copyright Notice 2  
 Diagram Types 13  
 Getting Started 8  
 license Agreement 3  
 Model Validation 56  
 Model Views 65  
 Model Wizard 12  
 Support 6  
 System Requirements 6  
 Toolbox Pages 13  
 Using DoDAF-MODAF 9  
 MDG Technology For Unified Profile For DoDAF And  
 MODAF 2  
 MDG Technology For UPDM  
 Welcome 2  
 Menu  
 MDG Technology For DoDAF-MODAF 11  
 Model Validation  
 MDG Technology For DoDAF-MODAF 56  
 Model Views  
 MDG Technology For DoDAF-MODAF 65  
 Model Wizard  
 MDG Technology For DoDAF-MODAF 12

## - Q -

Quicklinks  
 And MDG Technology For DoDAF-MODAF 54  
 Create DoDAF-MODAF Connectors 54  
 Create DoDAF-MODAF Elements 54

## - S -

Support  
 For MDG Technology For DoDAF-MODAF 6  
 System Requirements  
 For MDG Technology For DoDAF-MODAF 6

## - T -

Toolbox Pages  
 And MDG Technology For DoDAF-MODAF 13



# MDG Technology for UPDM User Guide

[www.sparxsystems.com](http://www.sparxsystems.com)