SPARX SYSTEMS

Enterprise Architect

User Guide Series

# Architecture Frameworks

Which Architecture Frameworks can I use? Sparx Systems Enterprise Architect has built-in support for TOGAF, Zachman Framework, UPDM, Archimate and customized extensions of UML, all important Enterprise Architecture frameworks and modeling languages.

Author: Sparx Systems

Date: 16/01/2019

Version: 1.0

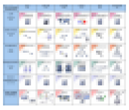
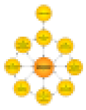
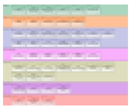
CREATED WITH ENTERPRISE ARCHITECT

# Table of Contents

# Architecture Frameworks

Enterprise Architecture is becoming an important discipline, as organizations need to understand the fundamental aspects of their business in order to keep pace with the global market in a continually evolving world. Enterprise Architect has built-in support for all of the important enterprise architecture frameworks and enterprise modeling languages, allowing an enterprise to be modeled from the business goals and drivers down to Cloud-based infrastructure services.

## Frameworks and Modeling Languages

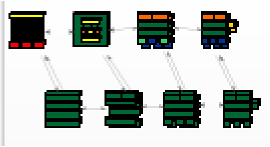
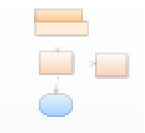
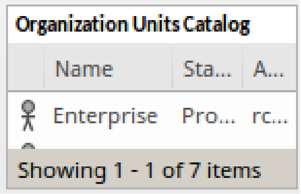
| Framework/Language | Description |
|---|---|
| The Zachman Framework  | The Zachman Framework is a widely used approach for engineering Enterprise Architecture. The Framework is a simple, logical structure that helps in organizing the information infrastructure of the Enterprise. |
| TOGAF  | The Open Group Architecture Framework (TOGAF) is one of the most widely accepted methods for developing Enterprise Architecture, providing a practical, definitive and proven step-by-step method for developing and maintaining Enterprise Architecture. |
| UPDM  | The MDG Technology for UPDM tightly integrates with Sparx Systems Enterprise Architect and provides a model-based framework for planning, designing and implementing the Unified Profile for DoDAF and MODAF (UPDM) architectures. |
| ArchiMate | ArchiMate® is a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure. |
| Extending UML | UML Profiles, MDG Technologies and SDK. |

# TOGAF

The Open Group Architecture Framework (TOGAF) is one of the most widely accepted methods for developing enterprise architecture. TOGAF is an open framework, providing a practical, definitive and proven step-by-step method for developing and maintaining enterprise architecture.

## Discussion

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

| Section | Content |
|---|---|
| Welcome | This section provides an introduction to TOGAF, and contains the formal documentation defining its use with Enterprise Architect. |
| Using TOGAF | Get started with TOGAF, learning about the model structure, templates, diagram types and more. |
| TOGAF ADM | The key to TOGAF remains a reliable, practical method - the TOGAF Architecture Development Method (ADM) - for defining business needs and developing an architecture that meets those needs, applying the elements of TOGAF and other architectural assets available to the organization. |
| The TOGAF Enterprise Contiuum<br> | The TOGAF Enterprise Continuum is a 'virtual repository' of all the architecture assets - models, Patterns, architecture descriptions and other artifacts - that exist both within the enterprise and in the IT industry at large, and that the enterprise considers itself to have available for the development of architectures for the enterprise. |
| Federal Enterprise Architecture Framework<br> | TOGAF provides diagrams and Toolbox pages specific to the Federal Enterprise Architecture Framework (FEAF). It also provides 'out-of-the-box' models of the FEAF Performance Reference model and Technical Reference model. |
| TOGAF Catalogs<br><br>Organization Units Catalog<br>Name Sta... A...<br>Enterprise Pro... rc...<br>Showing 1 - 1 of 7 items | Enterprise Architect helps you to create Model Catalog Artifacts, using the TOGAF-Catalog model Pattern, for:<br>• Actors<br>• Business Services<br>• Organization Units<br>• Principles |

| | |
|---|---|
| | • Requirements and<br>• Roles |

# Welcome to TOGAF in Enterprise Architect

Welcome to The Open Group Architecture Framework (TOGAF) integrated with Enterprise Architect.

Using this technology, users of Enterprise Architect benefit from TOGAF within a powerful modeling environment based on open standards.

## About TOGAF

The Open Group Architecture Framework is one of the most widely accepted methods for developing enterprise architecture. TOGAF is an open framework, providing a practical, definitive and proven step-by-step method for developing and maintaining enterprise architecture.

The key to TOGAF remains a reliable, practical method - the TOGAF Architecture Development Method (ADM) - for defining business needs and developing an architecture that meets those needs, applying the elements of TOGAF and other architectural assets available to the organization.

TOGAF embodies the concept of the Enterprise Continuum to reflect different levels of abstraction in an architecture development process. In this way TOGAF facilitates understanding and co-operation between actors at different levels. It provides a context for the use of multiple frameworks, models, and architecture assets in conjunction with the TOGAF ADM. By means of the Enterprise Continuum, architects are encouraged to leverage all other relevant architectural resources and assets, in addition to the TOGAF Foundation Architecture, in developing an organization-specific IT architecture.

For detailed information on TOGAF itself, visit the TOGAF website.

## Benefits of TOGAF

- Helps align business processes and IT to the business strategies and goals
- Provides support for all the phases in the ADM
- Provides support for OMG's Business Motivation Model
- Provides support for the Architecture Content Model
- Provides support for visual modeling of As-Is and To-Be architecture
- Provides support for modeling all four architecture domains specific to TOGAF (Business, Application, Data and Technology)
- Provides support for the report generation of TOGAF work products
- Provides the Open Group's TOGAF 9 deliverable templates as Linked Document templates
- Provides out-of-box FEAF reference models

## TOGAF Features

- A visual clickable Interface for ADM
- Useful starter model to help you become productive quickly
- UML profiles for FEAF Business, Performance, Service and Technical Reference Models
- Efficient relationship management for model artifacts with Enterprise Architect's Relationship Matrix and Hierarchy View
- Links to external files, audit log and report generation features of Enterprise Architect, providing additional capability for maintaining and managing your enterprise architecture

## Getting Started

For instructions on how to start using TOGAF, see *Getting Started with TOGAF* and *Using TOGAF*.

# TOGAF Copyright Notices

## TOGAF: Copyright © 2003-20018 X/Open Company Ltd, Trading as The Open Group. All Rights Reserved.

Any organization that intends use the methods, resources, and associated documentation suite known as The Open Group Architecture Framework - TOGAF Version 9 (and all earlier versions) for commercial purposes must apply to The Open Group for a commercial licence. See the *Open Group TOGAF* web site.

# TOGAF Software Product License Agreement

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

MDG Technology for TOGAF, Enterprise Architect MDG Add-In, Version 3.0.

Copyright © 2008-2019 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 the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

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

## DEFINITIONS

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

- "EULA" means this End User License Agreement
- "SPARX" means Sparx Systems Pty Ltd A.C.N 085 034 546
- "Licensee" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA
- "Registered Edition of MDG Technology for TOGAF" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: https://sparxsystems.com/products/mdg/tech/togaf/purchase.html, following a thirty (30) day free evaluation period
- "SOFTWARE PRODUCT" or "SOFTWARE" means MDG Technology for TOGAF, 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 of MDG Technology for TOGAF" 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.
- To make copies of the SOFTWARE PRODUCT for backup, archival and instructional purposes.

## EVALUATION LICENSE

The Trial Version of MDG Technology for TOGAF 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 thirty (30) days.

Upon expiration of the thirty (30) day evaluation period, the SOFTWARE PRODUCT must be removed from the computer. Unregistered use of MDG Technology for TOGAF 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 https://sparxsystems.com/products/mdg/tech/togaf/purchase.html). Upon payment of the license fee, YOU will be sent details of where to download the registered edition of MDG Technology for TOGAF and will be provided with a suitable software 'key' by email.

## ADDITIONAL RIGHTS AND LIMITATIONS

YOU hereby undertake not to sell or sub-license 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 XP, Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7 and Windows 8 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.

# Acknowledgement of Trademarks

## Trademarks of Microsoft

- Microsoft®
- Windows®

## Trademarks of the OMG

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

## Trademarks of The Open Group

- TOGAF™

# TOGAF Support

Technical support for modeling through TOGAF in Enterprise Architect is available to registered users of Enterprise Architect in exactly the same way as for Enterprise Architect itself.

# TOGAF System Requirements

TOGAF version 9.x runs under these environments:

## Operating Systems

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

## Enterprise Architect Versions

- Enterprise Architect Version 11.1 or later

# Getting Started With TOGAF

TOGAF is fully integrated with the Unified and Ultimate editions of Enterprise Architect, in which it is enabled and ready for use.

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

You can use the TOGAF profile in the Professional edition of Enterprise Architect. However, the Gap Analysis Matrix feature is not available for TOGAF in the Professional edition.

## Access TOGAF

1. Create a new Enterprise Architect project file, and click on the top-level Package.
2. Select the 'Start > View > Perspective > Open Model Wizard' option.
3. In the Model Wizard, select the Enterprise Architecture > TOGAF' Perspective and the 'Starter Model' Pattern.
4. Click on the Create Patterns button.

A new base TOGAF model is created in the Project Browser, containing the TOGAF Architecture Development Method (ADM) structures and the Enterprise Continuum asset Packages, and displaying the TOGAF-ADM (Interface) diagram.

# Using TOGAF

TOGAF provides a model-based framework for planning, designing and implementing the Architecture for an Enterprise. The starter model provided with TOGAF acts as a base upon which you can build the Enterprise Architecture. You can create the appropriate diagrams from the extended Enterprise Architect UML diagram set, using Toolbox pages that support every phase of the TOGAF Interface Diagram. You can also align models across the phases of the Architecture Development Method (ADM) using the Enterprise Architect Relationship Matrix.

## Notes

- TOGAF is integrated with the features of Enterprise Architect
- Enterprise Architect provides other Service Oriented Architecture tools such as SOMF and SoaML, and broader architecture modeling tools such as ArchiMate, SPEM and Business Rule Modeling, all of which you can use in conjunction with TOGAF to model and develop your Enterprise Architecture

# TOGAF Model Patterns

TOGAF includes a set of model Patterns that you can use to generate separate models within your TOGAF project. These are available through the Model Wizard.

## Access

Display the Model Wizard window, using any of the methods outlined here.

Once in the Model Wizard window, select the 'Model Patterns' tab, and the 'Enterprise Architecture > TOGAF' Perspective.

In the 'TOGAF Perspective' panel, expand 'TOGAF' and select from the TOGAF Patterns:

- Starter Model (includes both ADM and Enterprise Continuum)
- Architecture Development Method (ADM)
- Enterprise Continuum
- Technical Reference Model
- Catalogs

If you require additional diagrams, then also in the Model Wizard, click on the 'Diagrams' tab and (if necessary) select the 'Enterprise Architecture > TOGAF' Perspective. Then select from the diagram categories:

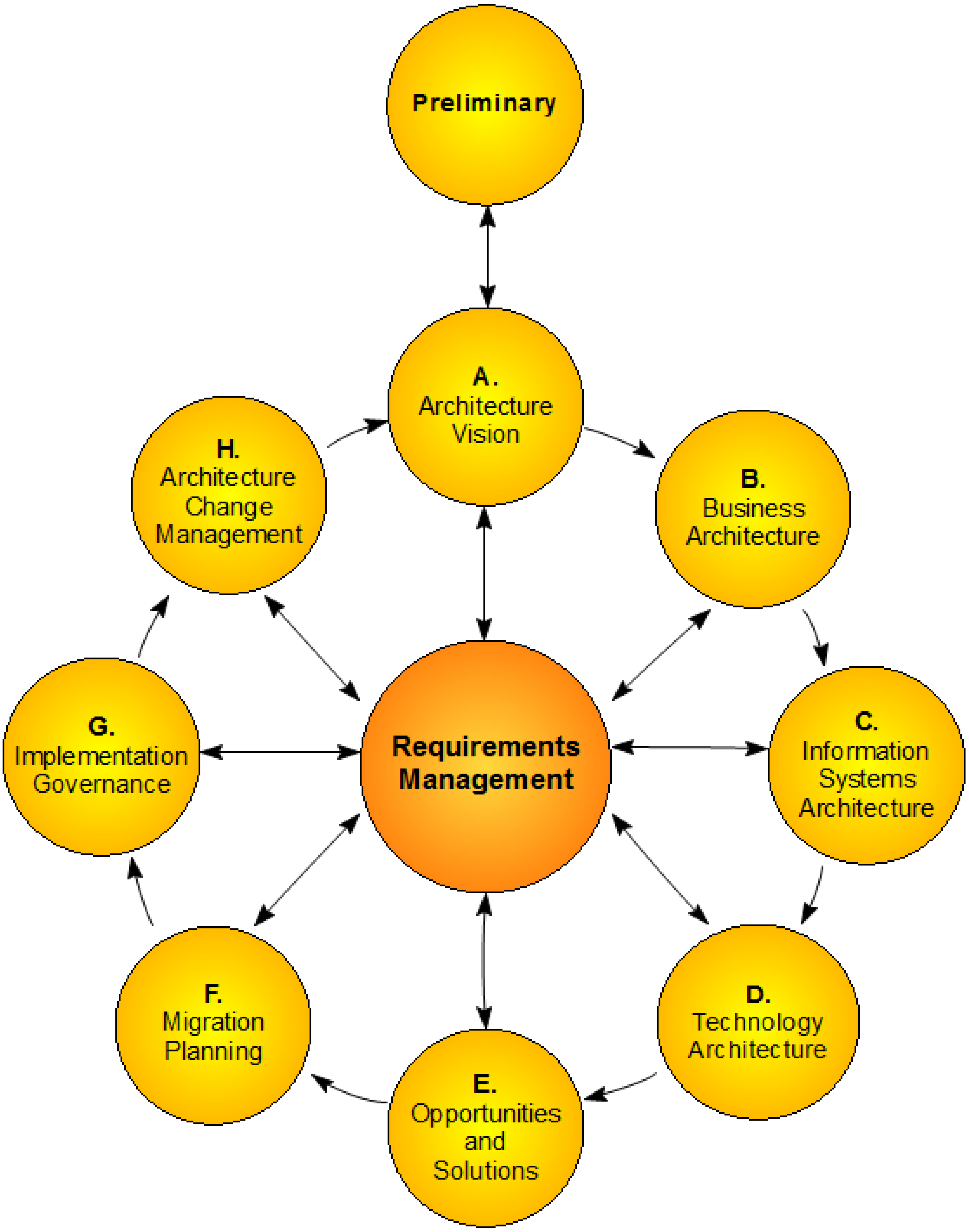
- FEAF Diagrams (Federal Enterprise Architecture Framework)
- TOGAF_BusinessArchitecture
- TOGAF_DataArchitecture
- TOGAF Diagrams

In the 'Diagram Types' panel, select the required diagram type.

| | |
|---|---|
| Ribbon | Design > Package > Insert > Insert using Model Wizard<br>Start > View > Perspective > Open Model Wizard |
| Context Menu | Right-click on Package \| Add a Model using Wizard |
| Keyboard Shortcuts | Ctrl+Shift+M |
| Other | Project Browser caption bar menu \| New Model from Pattern |

# The TOGAF Interface Diagram

In Enterprise Architect, the TOGAF Framework is presented as a predefined model. The model-level diagram of this model structure is the TOGAF Interface diagram, which serves as a user interface for the development of Enterprise Architecture based on TOGAF.

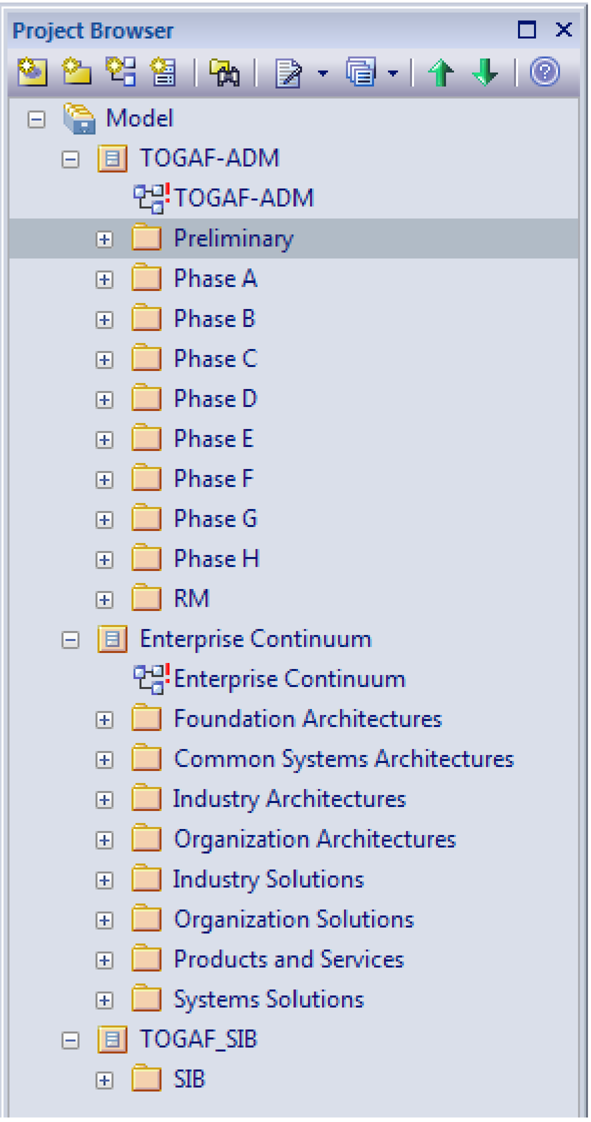

Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architecture
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

The TOGAF Framework model makes use of UML Packages, which is apparent from the model structure diagram. The Interface diagram itself is a standard UML Package diagram, using custom images.

Double-click on a cell of the Interface diagram to open the model Package and diagram corresponding to that particular ADM phase.

# The TOGAF Model Structure

Within the TOGAF Framework model, each ADM phase is modeled as the highest-level Package.



# The TOGAF Diagrams

TOGAF provides a number of diagram types to support modeling with TOGAF. These diagrams include:

TOGAF diagrams:

- TOGAF Interface
- Conceptual Framework
- Architecture Content
- Architecture Development Method
- Service Model
- Enterprise Continuum
- Standards Information Base

TOGAF_BusinessArchitecture:

- Benefits
- Business Motivation Model
- Organization Structure
- Business Logistics
- Business Process

TOGAF_DataArchitecture:

- Data Map

FEAF diagrams:

- (FEAF) Business Reference Model
- (FEAF) Service Component Reference Model
- (FEAF) Technical Reference Model
- (FEAF) Performance Reference Model

TOGAF-specific diagrams can be created in the same way as for any other diagram in Enterprise Architect. When you open a TOGAF diagram, Enterprise Architect automatically opens the appropriate Toolbox pages for that diagram.

# The TOGAF Toolbox Pages

The MDG Technology For TOGAF Toolbox pages provide elements and relationships for the full range of TOGAF diagrams supported by the Technology.

## Access

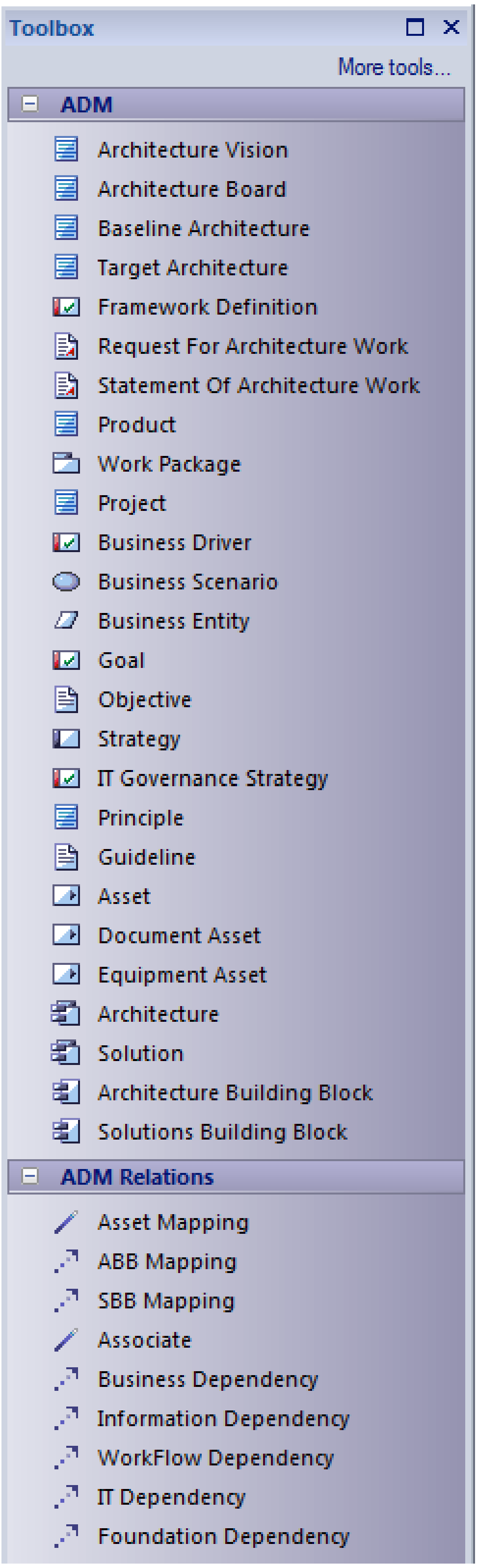
When you open a TOGAF diagram, Enterprise Architect displays the Toolbox pages that are most useful for that particular diagram type. In addition, the 'Common' and 'Artifacts' pages of UML elements and relationships display, regardless of which diagram is open.

The Diagram Toolbox pages can be docked on either side of the diagram, or free floated on top of the diagram to expose more surface for editing.

| | |
|---|---|
| Ribbon | Design > Diagram > Toolbox: 🔍 > Specify 'TOGAF' in the 'Find Toolbox Item' dialog |
| Keyboard Shortcuts | Alt+5 : 🔍 > Specify 'TOGAF' in the 'Find Toolbox Item' dialog |
| Other | You can display or hide the Diagram Toolbox by clicking on the » or « icons at the left-hand end of the Caption Bar at the top of the Diagram View. |

# Architecture Development Method Toolbox Pages

Architecture Development Method (ADM) elements are used to define and model the TOGAF specific primitives in all the phases of ADM. You use them to define the scope of the architecture.



## Architecture Development Method Toolbox

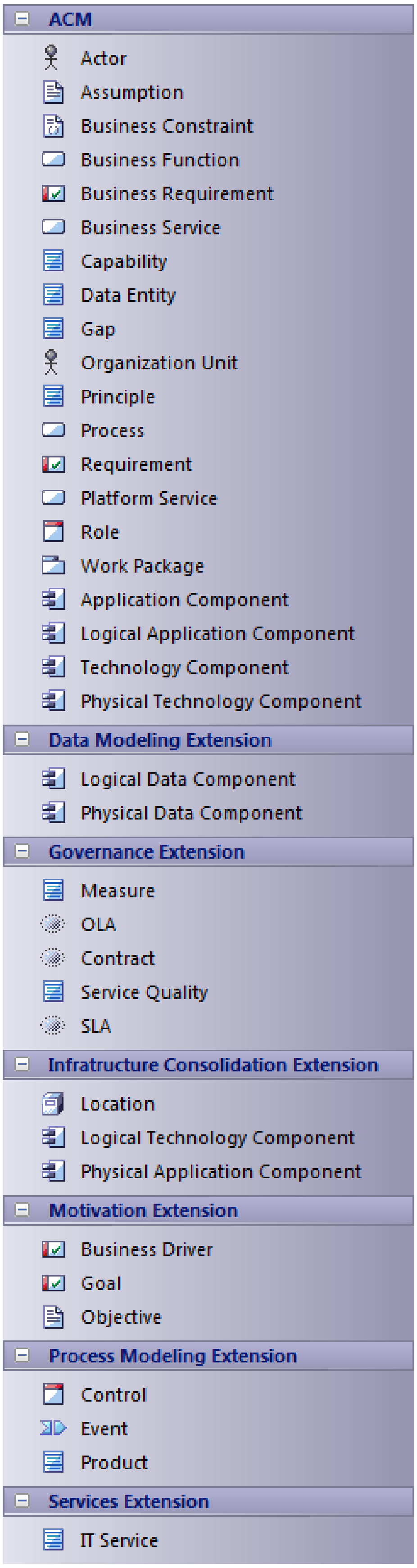
| Item | Description |
|---|---|
| Architecture Vision | Articulates a vision that enables the business goals, responds to the strategic drivers, conforms with the principles, and addresses the stakeholder concerns and objectives.<br>Tagged Values – ID, Scope, Version |
| Architecture Board | Captures the definition for a cross-organization Architecture Board. This is a key element in a successful architecture governance strategy, to oversee the implementation of the strategy.<br>This body should be representative of all the key stakeholders in the architecture, and typically comprises a group of executives responsible for the review and maintenance of the overall architecture.<br>Tagged Values – ID, Authority Limits, Responsibilities |
| Baseline Architecture | Captures the very high-level definitions of the Baseline environment from the perspective of business information systems and technology. The scope and level of detail to be defined depends on the extent to which existing architecture elements are likely to be carried over into the Target Architecture.<br>Tagged Values – ID, Type, Version |
| Target Architecture | Captures the very high-level definitions of the target environment, from the perspective of business information systems and technology.<br>Tagged Values – ID, Type, Version |
| Framework Definition | Provides a textual description of the Framework.<br>Tagged Values – ID, Version |
| Request for Architecture Work | Captures the information for the Request for Architecture Work, a major input for the ADM phases.<br>This element is designed as a Document Artifact. On creating a new element of this type, double-click on the element to open the Linked Document and select the 'TOGAF - Request for Architecture Work' template from the list of templates available for the 'Copy Template' option.<br>Tagged Values – ID, Architecturing Organization, Sponsoring Organization |
| Statement of Architecture Work | Captures the information for the Statement of Architecture Work, a major output for the ADM phases.<br>This element is designed as a Document Artifact. On creating a new element of this type, double-click on the element to open the Linked Document and select the 'TOGAF – Statement of Architecture Work' template from the list of templates available for the 'Copy Template' option.<br>Tagged Values – ID, Version |
| Product | Captures the information on a product produced by the enterprise.<br>Tagged Value – ID |
| Work Package | Defines a set of actions that achieve one or more objectives for the business. A work Package can be a part of a project, a complete project, or a program. |

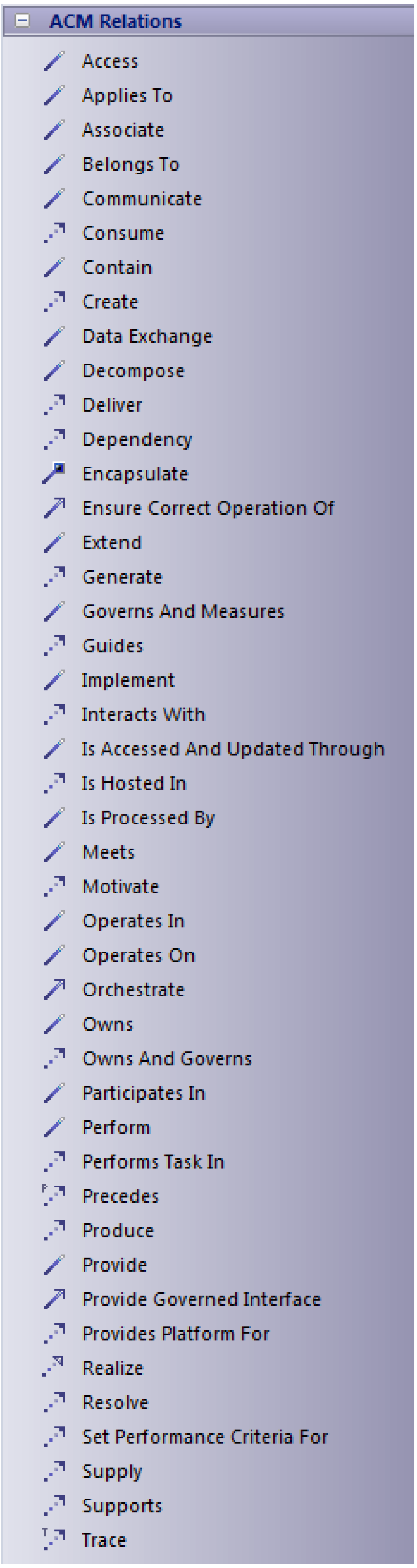
| | |
|---|---|
| | Tagged Values – CapabilityDelivered, WorkPackageCategory, ID, Source, Owner |
| Project | Captures the information to define a planned endeavor undertaken to create a product or service.<br>Tagged Values – ID, FutureDirections, Introduction, ProjectDevelopment, Process Overview, References, Target Architecture(s) Mapping |
| Business Driver | Defines the business driver in the 'Name' field.<br>Tagged Values – ID, Version |
| Business Scenario | Identifies and clarifies business needs, and thereby derives the business requirements that the architecture development has to address. Creating a business scenario involves these steps:<br>1. Identifying, documenting, and ranking the problem driving the scenario.<br>2. Identifying the business and technical environment of the scenario and documenting it in scenario models.<br>3. Identifying and documenting desired objectives.<br>4. Identifying the human actors (participants) and their place in the business model.<br>5. Identifying computer actors (computing elements) and their place in the technology model.<br>6. Identifying and documenting roles, responsibilities, and measures of success per actor; documenting the required scripts per actor, and the results of handling the situation.<br>7. Checking for 'fitness-for-purpose' and refining only if necessary.<br>A Linked Document template for Business Scenarios is provided by the Technology. To use the template, right-click on the element and select the 'Edit Linked Document' menu option. Select 'TOGAF – Business Scenario/Architecture Vision' for the 'Copy template' option.<br>Tagged Value – ID |
| Business Entity | A generic element that captures enterprise resources.<br>Tagged Values – ID, Description |
| Goal | Captures what is to be achieved by the enterprise, with specifications defined by the Tagged Values.<br>Tagged Values – Assumption, Critical Success Factor, Goal Type, ID, Key Performance Indicator, Measure, Unit Responsible, Opportunity, Strength, Threat, Weakness |
| Objective | Captures the attainable, time-targeted, and measurable target that the enterprise seeks to meet in order to achieve its goals.<br>Tagged Value – ID |
| Strategy | Captures the strategy statements for the business plan.<br>Tagged Values – Action Plan, Estimated Budget, Estimated Time Period, ID, Measure, Target Value |
| IT Governance Strategy | Defines the strategy statement for IT governance.<br>Tagged Values – ID, Version |
| | Defines and guides the organization, for the use of all assets and resources across |

| | |
|---|---|
| Principle | the enterprise. Each Principle should be linked to the relevant business objective and key architecture drivers.<br>Tagged Values – ID, Implications, Rationale, Statement, Type, Version |
| Guideline | Captures the Guidelines governing the enterprise and its functions, by providing guidance on the optimal ways to carry out design or implementation activities.<br>Tagged Value – ID |
| Asset | Captures the enterprise resources that could be estimated for value.<br>Tagged Values – ID, AssetValue, Description |
| Document Asset | A subtype of Asset that captures the important document resources of the enterprise.<br>Tagged Values – ID, AssetValue, Description |
| Equipment Asset | A subtype of Asset that captures the equipment resources of the enterprise.<br>Tagged Values – ID, AssetValue, Description |
| Architecture | Captures summary views of the Architecture Landscape (that is, the state of the enterprise) at particular points in time.<br>Tagged Values – ID, Category, Source, Owner, Subject Matter, View Point, Level Of Detail, Level Of Abstraction, Accuracy, Version, Maturity |
| Solution | Captures the summary views of a solution in place for a specific architecture.<br>Tagged Values – ID, Category, Source, Owner, Subject Matter, Time, Volatility, Version, Maturity |
| Architecture Building Block | (ABB) Relates to the Architecture Continuum, and is defined or selected as a result of the application of the ADM.<br>Tagged Values – ID, Description, Owning Organization, Rationale, ServicePortfolio |
| Solutions Building Block | (SBB) Relates to the Solutions Continuum, and can be either procured or developed.<br>Tagged Values – ID, Description, Supplier Organization |

# Architecture Content Model Toolbox Pages

The Architecture Content framework provides a structural model for architectural content that enables the major work products that an architect creates to be consistently defined, structured, and presented.





The elements in each of the Architecture Content Model Toolbox pages are described in separate topics:

- *ACM Core*

- *Data Modeling Extension*
- *Governance Extension*
- *Infrastructure Consolidation Extension*
- *Motivation Extension*
- *Process Modeling Extension*
- *Services Extension*

For information on Architecture Content Model relationships, see the topic *Architecture Content Metamodel Relationships* in the TOGAF 9 online documentation.

# ACM Core

Elements from the ACM page of the Architecture Content Model Toolbox.

## ACM Core Toolbox

| Item | Description |
|---|---|
| Actor | Identifies a person, organization or system with a role that initiates or interacts with activities. Actors can be internal or external to an organization.<br>Tagged Values – ID, Category, Source, Owner, #FTEs, ActorGoal, ActorTasks |
| Assumption | Defines a statement of probable fact that has not been fully validated at this stage, due to external constraints.<br>Tagged Values – ID, Rationale, Statement, Type |
| Business Constraint | Identifies an external factor that prevents an organization from pursuing particular approaches to meet its goals.<br>Tagged Value – ID |
| Business Function | Identifies a factor that delivers business capabilities closely aligned to an organization, but not necessarily explicitly governed by the organization.<br>Tagged Value – ID |
| Business Requirement | Defines a quantitative statement of business need that must be met by a particular architecture or work Package.<br>Tagged Value – ID |
| Business Service | Identifies a service that supports business capabilities through an explicitly defined interface and is explicitly governed by an organization.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |
| Capability | Defines a business-focused outcome that is delivered by the completion of one or more work Packages. Using a capability-based planning approach, change activities can be sequenced and grouped in order to provide continuous and incremental business value.<br>Tagged Values – ID, Category, Source, Owner, Increments, BusinessValue |
| Data Entity | Defines an encapsulation of data that is recognized by a business domain expert as an entity. Logical data entities can be tied to applications, repositories and services, and can be structured according to implementation considerations.<br>Tagged Values – ID, Category, Source, Owner, PrivacyClassification, RetentionClassification |
| Gap | Provides a statement of difference between two states. Used in the context of gap analysis, where the difference between the Baseline and Target Architecture is identified.<br>Tagged Values – ID, Category, Source, Owner |

| | |
|---|---|
| Organization Unit | Defines a self-contained unit of resources with line management responsibility, goals, objectives, and measures. Organizations can include external parties and business partner organizations.<br>Tagged Values – ID, PersonIncharge |
| Principle | Provides a qualitative statement of intent that should be met by the architecture. This has at least a supporting rationale and a measure of importance.<br>Tagged Values – ID, Type, Statement, Rationale, Implications |
| Process | Represents the flow of control between or within functions and/or services (depending on the granularity of definition). Processes represent a sequence of activities that together achieve a specified outcome, can be decomposed into sub-processes, and can show operation of a function or service (at the next level of detail). Processes can also be used to link or compose organizations, functions, services, and processes.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate, ProcessCriticality, ProcessVolumetrics, ProcessType |
| Platform Service | Defines a technical capability required to provide enabling infrastructure that supports the delivery of applications.<br>Tagged Values – ID, Category, Source, Owner, StandardClass |
| Role | Defines the usual or expected function of an Actor, or the part somebody or something plays in a particular action or event. An Actor can have a number of roles.<br>Tagged Values – ID, Category, Source, Owner, Responsibilities |
| Work Package | Identifies a set of actions to achieve one or more objectives for the business. A work Package can be a part of a project, a complete project or a program.<br>Tagged Values – ID, Category, Source, Owner, CapabilityDelivered |
| Application Component | Provides an encapsulation of application functionality aligned to implementation structure.<br>See also: 'Logical Application Component' and 'Physical Technology Component'.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |
| Logical Application Component | Provides an encapsulation of application functionality that is independent of a particular implementation.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |
| Technology Component | Provides an encapsulation of technology infrastructure that represents a class of technology product or specific technology product.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |
| Physical Technology | Defines an instance of a specific technology infrastructure product or technology |

| | |
|---|---|
| Component | infrastructure product.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate, ModuleName, ProductName, Vendor, Version |

# Data Modeling Extension

Elements from the Data Modeling Extension page of the Architecture Content Model Toolbox.

## Data Modeling Extensions Toolbox

| Item | Description |
|---|---|
| Logical Data Component | Defines a boundary zone that encapsulates related data entities to form a logical location to be held.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |
| Physical Data Component | Defines a boundary zone that encapsulates related data entities to form a physical location to be held.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |

# Governance Extension

Elements from the Governance Extension page of the Architecture Content Model Toolbox.

## Governance Extension Toolbox

| Item | Description |
|---|---|
| Measure | Identifies an indicator or factor that can be tracked, usually on an ongoing basis, to determine success or alignment with objectives and goals.<br>Tagged Values – ID, Category, Source, Owner |
| Contract | Defines an agreement between a service consumer and a service provider that establishes functional and non-functional parameters for interaction.<br>Tagged Values – ID, Source, Owner, AvailabilityCharacteristics, BehaviorCharacteristics, CapacityCharacteristics, ConsumingService, ContractControlRequirements, CredibilityCharacteristics, ExtensibilityCharacteristics, Growth, GrowthPeriod, IntegrityCharacteristics, InternationalizationCharacteristics, LocalizationCharacteristics, LocatabilityCharacteristics, ManageabilityCharacteristics, PeakProfileLongTerm, PeakProfileShortTerm, PerformanceCharacteristics, PortabilityCharacteristics, PrivacyCharacteristics, ProvidingService, QualityOfInformationRequired, RecoverabilityCharacteristics, ReliabilityCharacteristics, ResponseRequirements, ResultControlRequirements, ScalabilityCharacteristics, SecurityCharacteristics, ServiceabilityCharacteristics, ServiceQualityCharacteristics, ServiceTimes, Throughput, ThroughputPeriod |
| OLA | Defines an Operation Level Agreement.<br>Tagged Values – ID, Source, Owner, AvailabilityCharacteristics, BehaviorCharacteristics, CapacityCharacteristics, ConsumingService, ContractControlRequirements, CredibilityCharacteristics, ExtensibilityCharacteristics, Growth, GrowthPeriod, IntegrityCharacteristics, InternationalizationCharacteristics, LocalizationCharacteristics, LocatabilityCharacteristics, ManageabilityCharacteristics, PeakProfileLongTerm, PeakProfileShortTerm, PerformanceCharacteristics, PortabilityCharacteristics, PrivacyCharacteristics, ProvidingService, QualityOfInformationRequired, RecoverabilityCharacteristics, ReliabilityCharacteristics, ResponseRequirements, ResultControlRequirements, ScalabilityCharacteristics, SecurityCharacteristics, ServiceabilityCharacteristics, ServiceQualityCharacteristics, ServiceTimes, Throughput, ThroughputPeriod |
| SLA | Defines a Service Level Agreement<br>Tagged Values – ID, Source, Owner, AvailabilityCharacteristics, BehaviorCharacteristics, CapacityCharacteristics, ConsumingService, ContractControlRequirements, CredibilityCharacteristics, ExtensibilityCharacteristics, Growth, GrowthPeriod, IntegrityCharacteristics, InternationalizationCharacteristics, LocalizationCharacteristics, LocatabilityCharacteristics, ManageabilityCharacteristics, PeakProfileLongTerm, PeakProfileShortTerm, PerformanceCharacteristics, PortabilityCharacteristics, PrivacyCharacteristics, ProvidingService, QualityOfInformationRequired, RecoverabilityCharacteristics, ReliabilityCharacteristics, ResponseRequirements, ResultControlRequirements, ScalabilityCharacteristics, SecurityCharacteristics, ServiceabilityCharacteristics, ServiceQualityCharacteristics, ServiceTimes, |

| | |
|---|---|
| | Throughput, ThroughputPeriod |
| Service Quality | Defines a preset configuration of non-functional attributes that can be assigned to a service or service contract.<br>Tagged Values – ID, Category, Source, Owner |

# Infrastructure Consolidation Extension

Elements from the Infrastructure Consolidation Extension page of the Architecture Content Model Toolbox.

## Infrastructure Consolidation Extension Toolbox

| Item | Description |
|---|---|
| Location | Represents a place where business activity takes place and can be hierarchically decomposed.<br>Tagged Values – ID, Category, Source, Owner |
| Logical Technology Component | Provides an encapsulation of technology infrastructure that is independent of a particular product. A class of technology product.<br>Tagged Values – ID, Category, Source, Owner, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |
| Physical Application Component | Identifies an application, application module, application service or other deployable component of functionality.<br>Tagged Values – ID, Source, Owner, AvailabilityCharacteristics, CapacityCharacteristics, CredibilityCharacteristics, ExtensibilityCharacteristics, Growth, GrowthPeriod, IntegrityCharacteristics, InternationalizationCharacteristics, InteroperabilityCharacteristics, LocalizationCharacteristics, LocatabilityCharacteristics, ManageabilityCharacteristics, PeakProfileLongTerm, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetirementDate, PeakProfileShortTerm, PerformanceCharacteristics, PortabilityCharacteristics, PrivacyCharacteristics, RecoverabilityCharacteristics, ReliabilityCharacteristics, ScalabilityCharacteristics, SecurityCharacteristics, ServiceabilityCharacteristics, ServiceTimes, Throughput, ThroughputPeriod, LifeCycleStatus, InitialLiveDate, DateOfLastRelease, DateOfNextRelease, StandardsClass |

# Motivation Extension

Elements from the Motivation Extension page of the Architecture Content Model Toolbox.

## Motivation Extension Toolbox

| Item | Description |
|---|---|
| Business Driver | Defines an external or internal condition that motivates the organization to define its goals.<br>Tagged Values – ID, Version |
| Goal | Provides a high-level statement of intent or direction for an organization. Typically used to measure success of an organization.<br>Tagged Values – ID, Category, Source, Owner |
| Objective | Identifies a time-bounded milestone for an organization, to demonstrate progress towards a goal.<br>Tagged Values – ID |

# Process Modeling Extension

Elements from the Process Modeling Extension page of the Architecture Content Model Toolbox.

## Process Modeling Extension Toolbox

| Item | Description |
|---|---|
| Control | Defines a decision-making step with accompanying decision logic, used to determine the execution approach for a process or to ensure that a process complies with governance criteria.<br>Tagged Values – ID, Category, Source, Owner |
| Event | Defines an organizational state change that triggers processing events; can originate from inside or outside the organization and can be resolved inside or outside the organization.<br>Tagged Values – ID, Category, Source, Owner |
| Product | Defines the output generated by the business; that is, the business product of the execution of a process.<br>Tagged Values – ID, Category, Source, Owner |

# Services Extension

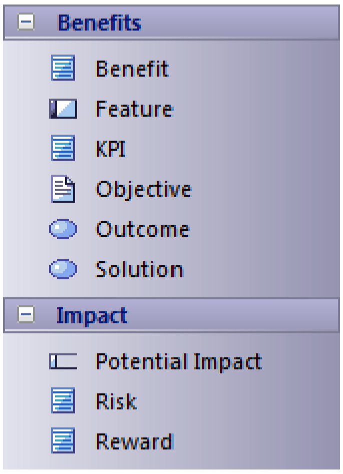
Elements from the Services Extension page of the Architecture Content Model Toolbox.

## Services Extension Toolbox

| Item | Description |
|---|---|
| IT Service | Defines the automated elements of a business service. An information system service can deliver or support part or all of one or more business services.<br>Tagged Values – ID, Category, Source, Owner, DefinitionText, ContactPoint, Availability, ChargeToUser, DependentSystems, StandardsClass, StandardCreationDate, LastStandardReviewDate, NextStandardReviewDate, RetireDate |

# Benefits Toolbox Pages

You use the Benefits Toolbox to create elements that represent and depict the opportunities identified in an architecture definition, classified according to their relative size, benefit, and complexity. The resulting Benefits diagram can be used by stakeholders to make decisions on selection, prioritization and sequencing of the identified opportunities.



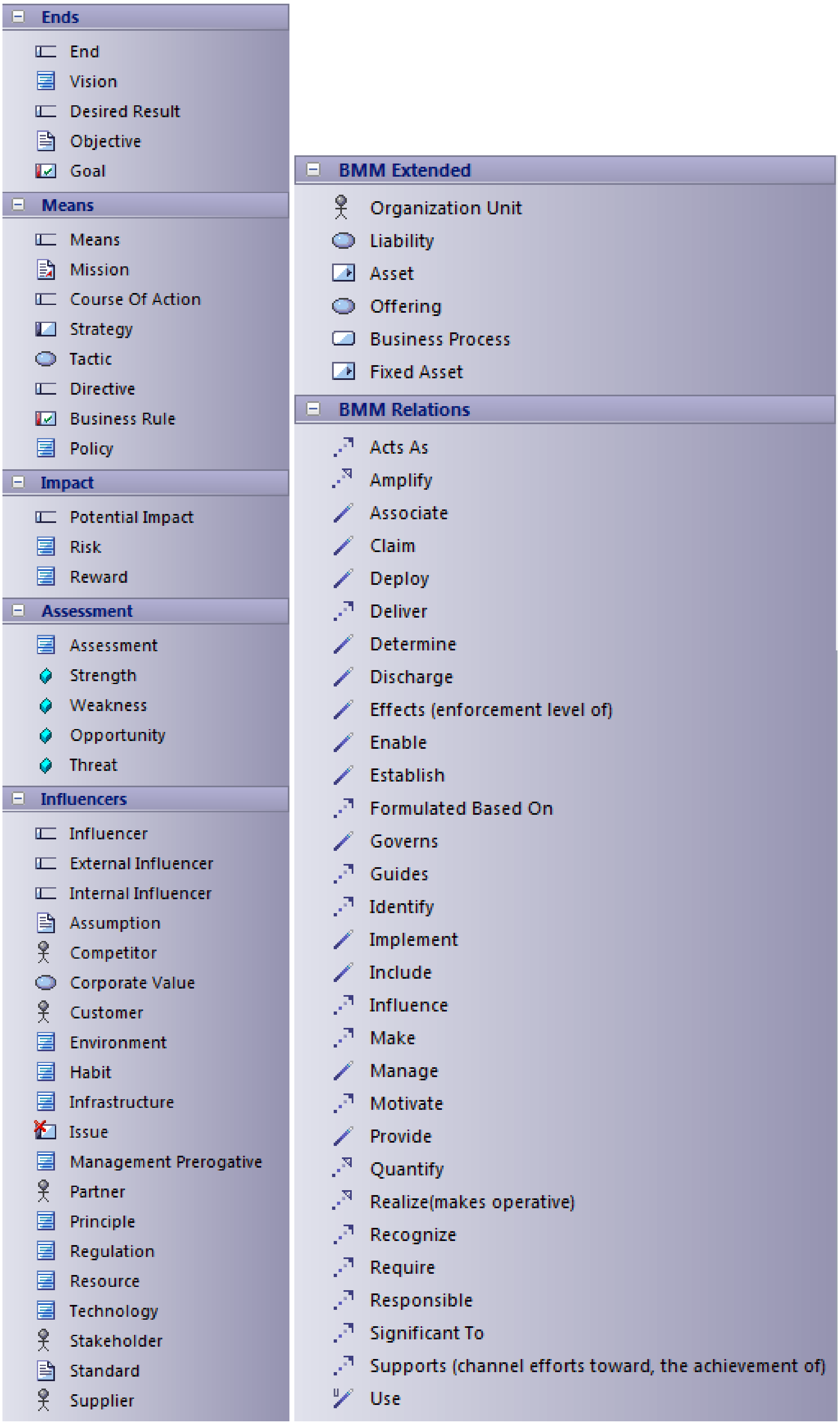
## Benefits Toolbox

| Item | Description |
|---|---|
| Benefit | An Artifact to model the benefit of an opportunity identified in the architecture definition.<br>Tagged Values – ID, Owner, Source, Category |
| Feature | Represents a characteristic of a service or solution<br>Tagged Values – ID, Owner, Source, Category |
| KPI | (Key Performance Indicator) A metric used to define and measure progress towards achieving goals or critical success factors.<br>Tagged Values – ID, Owner, Source, Category |
| Objective | A statement of an attainable, time-targeted and measurable target that the enterprise seeks to meet in order to achieve its goals. An Objective quantifies a Goal.<br>Tagged Value – ID |
| Outcome | The resulting end state of an event, decision or architecture process.<br>Tagged Values – ID, Owner, Source, Category |
| Solution | A statement of an operation or activity that supports the outcome.<br>Tagged Values – ID, Owner, Source, Category |
| Potential Impact | See the Help on the 'Impact' Page of the Business Motivation Model toolbox. |
| Risk | See the Help on the 'Impact' page of the Business Motivation Model toolbox. |
| | |

| | |
|---|---|
| Reward | See the Help on the 'Impact' Page of the Business Motivation Model toolbox. |

# Business Motivation Model Toolbox Pages

The Business Motivation Model Toolbox page is based on the OMG specification for the Business Motivation Model (BMM). These elements provide a structure for developing, communicating, and managing business plans in an organized manner.



The elements in each of the Business Motivation Model Toolbox pages are described in separate topics:

- *Ends Page*

- *Means Page*
- *Impact Page*
- *Assessment Page*
- *Influencers Page*
- *BMM Extended Page*

# Ends Page

Elements from the 'Ends' page of the Business Motivation Model Toolbox.

## Ends Toolbox

| Item | Description |
|---|---|
| End | Groups 'end' concepts (Vision and Desired Result).<br>An End is something the business seeks to accomplish. It does not include any indication of how it is to be achieved.<br>Tagged Values – ID, Category, Source, Owner |
| Vision | Describes the future state of the enterprise, without regard to how it is to be achieved.<br>A Vision is supported or made operative by Missions, and is amplified by Goals.<br>Tagged Value – ID |
| Desired Result | Groups 'desired result' concepts (Goal and Objective). A Desired Result is an End that is a state or target that the enterprise intends to maintain or sustain. A Desired Result is supported by Courses of Action. One Desired Result can include other Desired Results and can itself be included in another Desired Result.<br>Tagged Values – ID, Category, Source, Owner |
| Goal | A statement about a state or condition of the enterprise to be brought about or sustained through appropriate Means. A Goal amplifies a Vision.<br>Tagged Values – Assumption, Critical Success Factor, Goal Type, ID, Key Performance Indicator, Measure, Unit Responsible, Opportunity, Strength, Threat, Weakness |
| Objective | A statement of an attainable, time-targeted and measurable target that the enterprise seeks to meet in order to achieve its goals. An Objective quantifies a Goal.<br>Tagged Value – ID |

# Means Page

Elements from the 'Means' page of the Business Motivation Model Toolbox.

## Means Toolbox

| Item | Description |
|---|---|
| Means | Groups 'Means' concepts (Mission, Course of Action and Directive). A Means represents any capabilities that can be exploited to achieve the desired Ends.<br>Tagged Values – ID, Category, Source, Owner |
| Mission | Captures the mission statement, policies and values of the enterprise. A Mission indicates the ongoing operational activity of the enterprise, and makes a Vision operative.<br>Tagged Values – ID, Category, Source, Owner |
| Course of Action | Groups 'course of action' concepts (Strategy and Tactic). A Course of Action is an approach or plan for configuring some aspect of the enterprise involving things, processes, locations, people, timing or motivation, undertaken to achieve Desired Results.<br>A Course of Action channels efforts towards Desired Results. Courses of Action are governed by Directives. It is also possible for the Courses of Action to be formulated based on Directives. Courses of Action can be realized by Business Processes. One Course of Action can include other Courses of Action, and one Course of Action can be enabled by another Course of Action.<br>Tagged Values – ID, Category |
| Strategy | Defines the right approach to achieve a set of Goals, given the environmental constraints and risks. A Strategy usually channels efforts towards those Goals.<br>Tagged Values – Action Plan, Estimated Budget, Estimated Time Period, ID, Measure, Target Value |
| Tactic | A Course of Action that represents part of the detailing of a Strategy. A Tactic implements one or more Strategies.<br>Tagged Values – ID, Category |
| Directive | Indicates how the Course of Action should, or should not, be carried out. A Directive defines, constrains or liberates some aspect of an enterprise. It is intended to assert business structure or to control or influence the behavior of the business, and is stated in declarative form. Directives govern Courses of Action. A Directive is defined to support the achievement of a Desired Result directly.<br>Tagged Values – ID, Category |
| Business Rule | A Business Rule element captures the Business Rule statements. Business Rules provide specific, actionable governance or guidance to implement Business Policies. Business Rules guide Business Processes.<br>Tagged Values – ID, Name, Description, Effective_From, Expiry_From, Status, Version, Enforcement_Level |
| | Captures the policy definitions followed in the enterprise. A Business Policy is a |

| | |
|---|---|
| Policy | non-actionable Directive whose purpose is to govern or guide the enterprise. Business Policies provide the basis for Business Rules. Business Policies also govern Business Processes. One Business Policy can include other Business Policies.<br>Tagged Value – ID |

# Impact Page

Elements from the 'Impact' page of the Business Motivation Model Toolbox.

## Impact Toolbox

| Item | Description |
|---|---|
| Potential Impact | Groups the concepts of 'impacts' (Risk and Reward). Each Potential Impact is an evaluation that quantifies or qualifies some aspect of an Assessment in specific terms, types or dimensions.<br>An Assessment identifies some Potential Impacts. A Potential Impact can be significant to an Assessment.<br>Tagged Values – ID, Category, Source, Owner |
| Risk | A Potential Impact that indicates the possibility of loss, injury, disadvantage or destruction.<br>Tagged Value – ID |
| Reward | A Potential Impact that indicates the probability of gain.<br>Tagged Value – ID |

# Assessment Page

Elements from the 'Assessment' page of the Business Motivation Model Toolbox.

## Assessment Toolbox

| Item | Description |
|---|---|
| Assessment | A judgment on an Influencer that affects the organization's ability to employ its Means or achieve its Ends. A Directive is motivated by an Assessment. Assessments can also use other Assessments. An Assessment can support the achievement of Ends.<br>Tagged Values – ID, Source, Owner |
| Strength | This category of Assessment indicates some advantage or area of excellence within the enterprise that can impact its employment of Means or achievement of Ends. It is modeled as a parameter of the Assessment element.<br>Tagged Value – ID |
| Weakness | This category of Assessment indicates some area of inadequacy within the enterprise that can impact its employment of Means or achievement of Ends. It is modeled as a parameter of the Assessment element.<br>Tagged Value – ID |
| Opportunity | This category of Assessment indicates that some Influencer can have a favorable impact on the organization's employment of Means or achievement of Ends. It is modeled as a parameter of the Assessment element.<br>Tagged Value – ID |
| Threat | This category of Assessment indicates that some Influencer can have an unfavorable impact on the organization's employment of Means or achievement of Ends. It is modeled as a parameter of the Assessment element.<br>Tagged Value – ID |

# Influencers Page

Elements from the 'Influencers' page of the Business Motivation Model Toolbox.

## Influencers Toolbox

| Item | Description |
|---|---|
| Influencer | An Influencer element groups the elements influencing an Assessment. The Influencers are those that can impact the enterprise in its employment of Means or achievement of its Ends. This impact has influence that is judged in Assessments.<br>Tagged Values – ID, Category |
| External Influencer | An External Influencer element groups the elements having an external influence on an Assessment. External Influencers are those outside an enterprise's organizational boundaries that can impact its employment of Means or achievement of Ends.<br>Tagged Values – ID, Category |
| Internal Influencer | An Internal Influencer element groups the elements having an internal influence on an Assessment. Internal Influencers are those from within an enterprise that can impact its employment of Means or achievement of Ends.<br>Tagged Values – ID, Category |
| Assumption | An Assumption element captures the assumptions made in information manipulation; assumptions are items of information taken for granted or without proof.<br>Tagged Values – ID, Rationale, Statement, Type |
| Competitor | An External Influencer that is an individual or enterprise posing a challenge to the subject enterprise.<br>Tagged Value – ID |
| Corporate Value | An ideal, custom or institution that an enterprise promotes or agrees with (either positive or negative).<br>Tagged Value – ID |
| Customer | An External Influencer as an individual or enterprise that has investigated, ordered, received or paid for products or services from the subject enterprise.<br>Tagged Value – ID |
| Environment | An Environment element is the aggregate of surrounding conditions or Influencers affecting the existence or development of an enterprise.<br>Tagged Value – ID |
| Habit | A customary practice or use.<br>Tagged Value – ID |
| Infrastructure | An Internal Influencer forming the basic underlying framework or features of a |

| | |
|---|---|
| | system.<br>Tagged Value – ID |
| Issue | A point in question or a matter that is in dispute as between contending partners. |
| Management Prerogative | A right or privilege exercised by virtue of ownership or position in an enterprise.<br>Tagged Value – ID |
| Partner | An External Influencer as an enterprise that shares risks and profit with the subject enterprise (or is associated with the subject enterprise to share risks and profit) because this is mutually beneficial.<br>Tagged Value – ID |
| Principle | Defines and guides the organization, for use of all assets and resources across the enterprise. Each Principle should be linked to the relevant business objective and key architecture drivers.<br>Tagged Values – ID, Implications, Rationale, Statement, Type, Version |
| Regulation | An External Influencer as an order prescribed by an authority such as a government body or the management of an enterprise.<br>Tagged Value – ID |
| Resource | An internal Influencer as a resource available for carrying out the business of an enterprise, applying its influence especially by way of its quality.<br>Tagged Value – ID |
| Technology | An External Influencer as the role of technology, including its developments and limitations — there could be prerequisites for use of technology, or an enterprise activity that technology enables or restricts.<br>Tagged Value – ID |
| Stakeholder | Captures the actors interested and involved in the enterprise.<br>Tagged Value – ID |
| Standard | Defines the standards followed in the enterprise.<br>Tagged Values – ID, Statement, Type |
| Supplier | An External Influencer as an individual or enterprise that can furnish or provide products or services to the subject enterprise.<br>Tagged Value – ID |

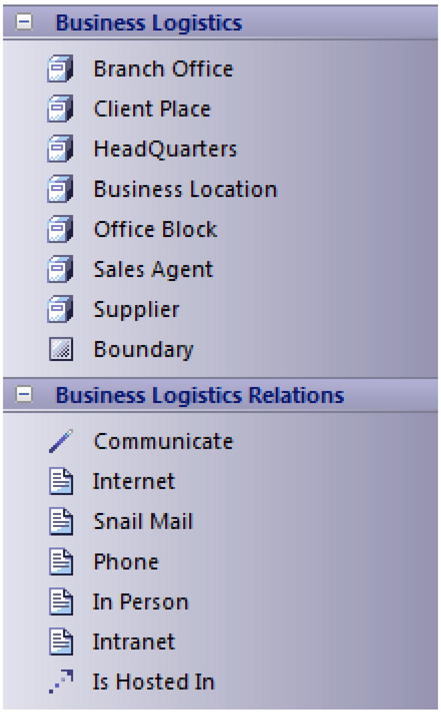
# BMM Extended Page

Elements from the 'BMM Extended' page of the Business Motivation Model Toolbox.

## BMM Extended Toolbox

| Item | Description |
|---|---|
| Organization Unit | Represents any recognized association of people in the context of the enterprise. In a hierarchical structure, it might be the corporation, a division, a department, a group or a team.<br>Tagged Values – ID, PersonInCharge |
| Liability | A Liability is a reservation of actual resources (materials, finished goods, people's time, cash) to meet commitments. A Liability can be discharged by Courses of Action, can be the responsibility of Organization Units, and can claim Resources.<br>Tagged Value – ID |
| Asset | An Asset is something of value owned by the enterprise.<br>Tagged Values – ID, Description, AssetValue |
| Offering | An Offering is a Fixed Asset that is a specification of a product or service that can be supplied by the enterprise. An Offering can be defined by Courses of Action, can be delivered by Business Processes, can require Resources and can use Fixed Assets.<br>Tagged Value – ID |
| Business Process | A function or behavior of the Enterprise or part of the Enterprise. A Business Process is the responsibility of an Organization Unit, realizes Courses of Action, is guided by Business Rules, is governed by Business Policies, can deliver Offerings and can manage Assets.<br>Tagged Values – ID, Description, ProcessType |
| Fixed Asset | A Fixed Asset is an Asset that is maintained over time and reused. A Fixed Asset can be used by Offerings and can provide Resources.<br>Tagged Values – ID, AssetValue |

# Business Logistics Toolbox Pages



Business Logistics

- Branch Office
- Client Place
- HeadQuarters
- Business Location
- Office Block
- Sales Agent
- Supplier
- Boundary

Business Logistics Relations

- Communicate
- Internet
- Snail Mail
- Phone
- In Person
- Intranet
- Is Hosted In

## Business Logistics Toolbox

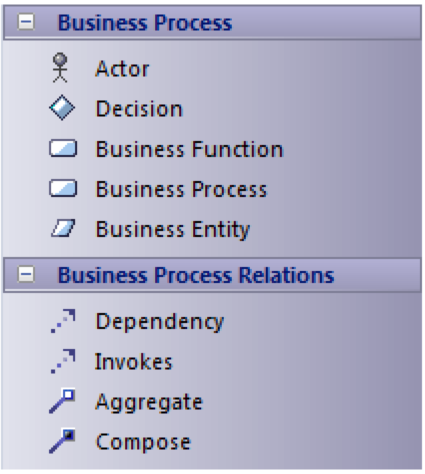
| Item | Description |
|---|---|
| Branch Office | Models a Business Location as a Branch Office. |
| Client Place | Models a Business Location as a Client Place. |
| Head Quarters | Models a Business Location as a Head Quarters. |
| Business Location | Models the location from which the business operates. |
| Office Block | Models a Business Location as an Office Block. |
| Sales Agent | Models a Business Location as a Sales Agent. |
| Supplier | Models a Business Location as a Supplier. |
| Communicate | Indicates that a business location communicates directly with another business location. |
| Internet | Indicates that the means of communication is the World Wide Web. |
| Snail Mail | Indicates that the means of communication is the postal system or courier services. |
| Phone | Indicates that the means of communication is the telephone. |

| | |
|---|---|
| In Person | Indicates that the means of communication is direct person-to-person. |
| Intranet | Indicates that the means of communication is the local intranet or WAN. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Business Process Toolbox Pages



Business Process
- Actor
- Decision
- Business Function
- Business Process
- Business Entity

Business Process Relations
- Dependency
- Invokes
- Aggregate
- Compose

## Business Process Toolbox

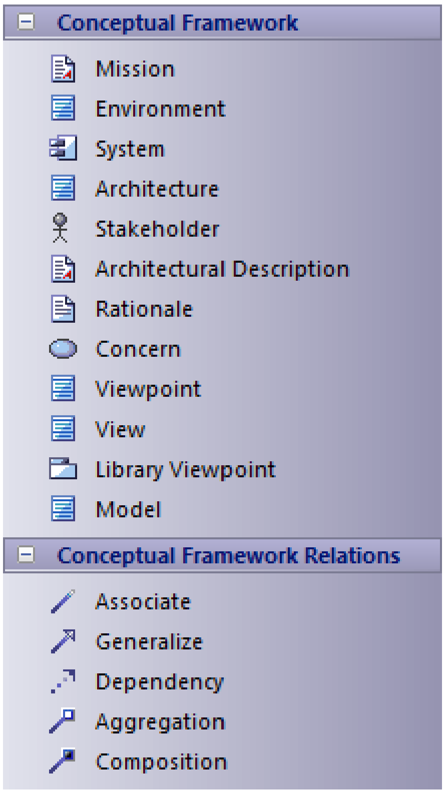
| Item | Description |
|---|---|
| Actor | Models a stakeholder or any other human resource of the Enterprise. |
| Decision | Indicates point of conditional progression where a business decision is taken. |
| Business Function | A major function performed by the Enterprise or a part of the Enterprise. |
| Business Process | A function or behavior of the Enterprise or part of the Enterprise. |
| Business Entity | A generic element to capture Enterprise resources. |
| Invokes | A relationship that defines the invocation of a business process. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Conceptual Framework Toolbox Pages

The Conceptual Framework Elements are used to model the architectural descriptions and to establish concepts for architectural thinking. The Toolbox element design is based on IEEE standard 1471 - 2000.



## Conceptual Framework Toolbox

| Item | Description |
|---|---|
| Mission | Captures the mission statement, policies and values of the enterprise.<br>Tagged Value – ID |
| Environment | Defines the developmental, operational and programmatic context of the system for the purpose of Enterprise Engineering work.<br>Tagged Value – ID |
| System | Captures details of a working component of the enterprise. System includes, for example, application, system, platform, system -of-systems, enterprise and product line.<br>Tagged Value – ID |
| Architecture | Captures the definition of the Architecture work.<br>Tagged Value – ID |
| Stakeholder | Captures the actors interested and involved in the enterprise.<br>Tagged Value – ID |
| | |

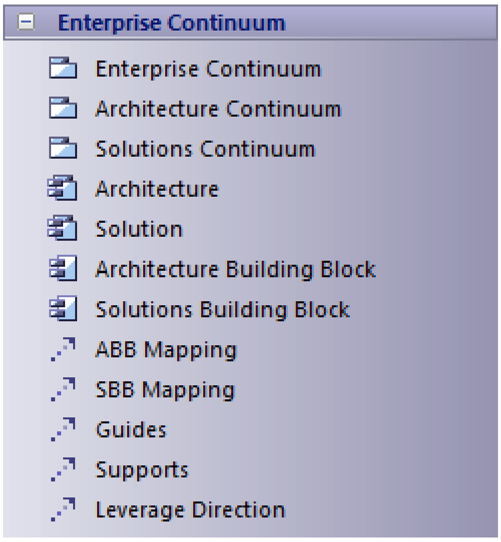
| | |
|---|---|
| Architectural Description | Captures the architectural descriptions and identifies the system's stakeholders and their concerns.<br>Tagged Value – ID |
| Rationale | Captures the statement of purpose for the Architectural Description. |
| Concern | Forms the basis for completeness. An Architectural Description addresses all stakeholders' concerns, and each Concern is addressed by an Architectural View |
| Viewpoint | A Pattern for constructing Views – Viewpoints define the rules on Views.<br>Each View corresponds to exactly one Viewpoint.<br>Tagged Value – ID |
| View | A representation of a whole system from the perspective of a set of Concerns. A View can contain one or more architectural models, so the View can use multiple notations. |
| Library Viewpoint | Captures a collection of categorized Viewpoints.<br>Tagged Value – ID |
| Model | Defines and represents a model.<br>Tagged Value – ID |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Enterprise Continuum Toolbox Page

Enterprise Continuum elements are used to model the Architecture Continuum and Solutions Continuum of an enterprise. Using these elements you can create Architecture Building Blocks or Solutions Building Blocks by mapping to the appropriate architecture models or solution models (as diagrams, elements and models).

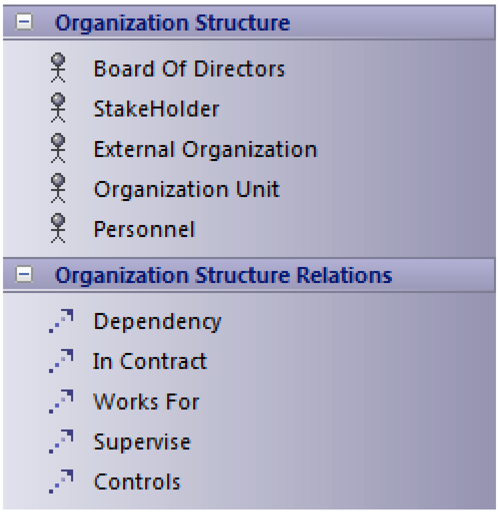


## Enterprise Continuum Toolbox

| Item | Description |
|---|---|
| Enterprise Continuum | A Package that models the Enterprise Continuum.<br>Tagged Values – ID, Architecturing Organization, Sponsoring Organization |
| Architecture Continuum | A Package that models the Architecture Continuum. |
| Solutions Continuum | A Package that models the Solutions Continuum. |
| Architecture | Captures summary views of the Architecture Landscape (such as the state of the enterprise) at particular points in time.<br>Tagged Values – ID, Category, Source, Owner, Subject Matter, View Point, Level Of Detail, Level Of Abstraction, Accuracy, Version, Maturity |
| Solution | Captures the summary views of a solution in place for a specific architecture.<br>Tagged Values – ID, Category, Source, Owner, Subject Matter, Time, Volatility, Version, Maturity |
| Architecture Building Block | Relates to the Architecture Continuum, and is defined or selected as a result of the application of the ADM.<br>Tagged Values – ID, Description, Owning Organization, Rationale, ServicePortfolio |
| Solutions Building Block | Relates to the Solutions Continuum, and can be either procured or developed.<br>Tagged Values – ID, Description, Supplier Organization |
| | |

| | |
|---|---|
| ABB Mapping | Connector to map the architectural models and artifacts to the Architecture Building Blocks. |
| SBB Mapping | Connector to map the solution models and artifacts to the Solutions Building Blocks. |
| Guides | Connector to represent guides relationships. Architecture Building Blocks guide the development of Solutions Building Blocks. |
| Supports | Connector to represent supports relationships. Solutions Building Blocks support the development of other Solutions Building Blocks. |
| Leverage Direction | Connector to represent the direction of leveraging of architecture and solution components. |

# Organization Structure Toolbox Pages



⊟ Organization Structure
- Board Of Directors
- StakeHolder
- External Organization
- Organization Unit
- Personnel

⊟ Organization Structure Relations
- Dependency
- In Contract
- Works For
- Supervise
- Controls

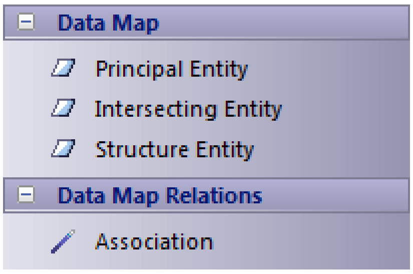
## Organization Structure Toolbox

| Item | Description |
|---|---|
| Board of Directors | Captures the details of the board of directors. |
| StakeHolder | Captures stakeholders of the enterprise. |
| External Organization | Captures any external business unit that is not under direct control of the enterprise, but has a relationship with the enterprise. |
| Organization Unit | Captures any business unit that is under direct control of the enterprise. |
| Personnel | Captures the details of personnel in an enterprise. |
| In Contract | Captures the contract-based relationships between business units. |
| Works For | Captures the details of team links; for example, Stakeholder 1 works for Organization Unit 1. |
| Supervise | Captures process supervision details. |
| Control | Captures Unit in charge or Person in charge information. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Data Map Toolbox Pages



Data Map

Principal Entity

Intersecting Entity

Structure Entity

Data Map Relations

Association

## Data Map Toolbox

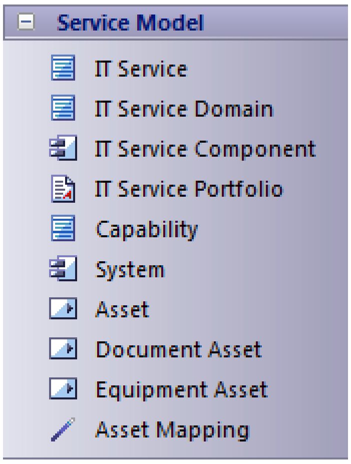
| Item | Description |
|---|---|
| Principal Entity | A business entity that forms a resource of the enterprise. |
| Intersecting Entity | Normalizes the many-to-many relationship between principal entities. |
| Structure Entity | Captures potential knowledge base entities. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Service Model Toolbox Page

Service Model elements are used to build a conceptual framework that describes the IT Service infrastructure of the enterprise.



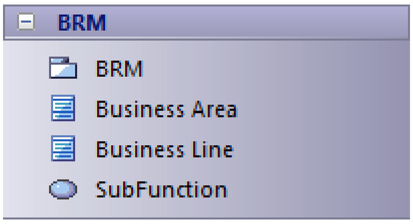
## Service Model Toolbox

| Item | Description |
|---|---|
| IT Service | Captures the IT capability offered as a consumable entity that is managed by the enterprise.<br>Tagged Values – ID, DefinitionText, Owner, Availability, Charge_to_User, ContactPoint, Dependent_Systems |
| IT Service Domain | Categorizes IT services.<br>Tagged Values – ID, Description |
| IT Service Component | Captures a set of capabilities that might be exposed through the technology interface.<br>Tagged Values – ID, Rationale |
| IT Service Portfolio | A Document Artifact that captures the information required to describe an IT service portfolio.<br>Tagged Values – ID |
| Capability | A business-focused outcome that is delivered by the completion of one or more work Packages. Using a capability-based planning approach, change activities can be sequenced and grouped in order to provide continuous and incremental business value.<br>Tagged Values – ID, Category, Increments, Business Value, Source, Owner |
| System | Captures details of a working component of the enterprise. System includes things such as application, system, platform, system-of-systems, enterprise and product line.<br>Tagged Values – ID |
| | |

| | |
|---|---|
| Asset | Captures the enterprise resources that could be estimated for value.<br>Tagged Values – ID, AssetValue, Description |
| Document Asset | Subtype of Asset that captures the important document resources of the enterprise.<br>Tagged Values – ID, AssetValue, Description |
| Equipment Asset | Subtype of Asset that captures the equipment resources of the enterprise.<br>Tagged Values – ID, AssetValue, Description |

# FEAF Business Reference Model Toolbox Page

The FEAF Business Reference Model (BRM) provides a framework facilitating a functional (rather than organizational) view of the enterprise's lines of business (LoBs), including its internal operations and its services.



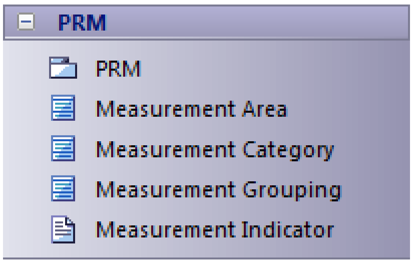
## FEAF Business Reference Model Toolbox

| Item | Description |
|---|---|
| BRM | A Package in which to capture the Business Reference Model (BRM).<br>Tagged Values – Version |
| Business Area | The high-level organizing layer of the BRM, capturing high-level categories relating to the business purpose and objectives.<br>Tagged Values – BusinessAreaID, Definition |
| Business Line | Captures the lines of business of the enterprise.<br>Tagged Values – BusinessLineID, Definition, Referencing Business Area |
| SubFunction | Represents the lowest level of granularity in the BRM, grouping functionalities related to each line of business.<br>Tagged Values – SubFunctionID, Definition, Referencing BusinessLine, Referencing Business Area |

# FEAF Performance Reference Model Toolbox Page

The FEAF Performance Reference Model (PRM) Toolbox page is designed to conform to the specification of the FEAF-PRM framework. The PRM is a framework for performance measurement providing common output measurements throughout the enterprise. It enables agencies to better manage the business at a strategic level, by providing a means for using an agency's Enterprise Architect to measure the success of IT investments and their impact on strategic outcomes.

The FEAF Performance Reference Model (PRM) facilitates resource-allocation decisions based on comparative determinations of which programs and organizations are more efficient and effective.

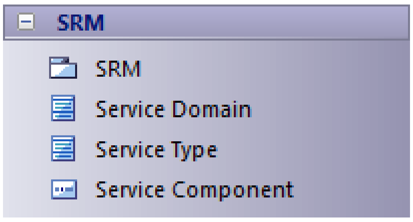


## FEAF Performance Reference Model Toolbox

| Item | Description |
|---|---|
| PRM | A Package to capture the Performance Reference Model.<br>Tagged Values – Version |
| Measurement Area | The high-level organizing layer of the PRM, capturing aspects of performance at the output levels. This layer is directly linked to the performance objectives established at the agency and program levels.<br>Tagged Values – MeasurementAreaID, Definition |
| Measurement Category | Categorizes the measurement area with respect to the attribute or characteristic to be measured.<br>Tagged Values – MeasurementCategoryID, Definition, Referencing Measurement Area |
| Measurement Grouping | Further refines Measurement Categories into specific types of Measurement Indicators.<br>Tagged Values – MeasurementGroupingID, Definition, Referencing Measurement Category |
| Measurement Indicator | Captures the specific measures.<br>Tagged Values – MeasurementIndicatorID, Definition, Referencing Measurement Grouping |

# FEAF Service Component Reference Model Toolbox Page

The FEAF Service Component Reference Model (SRM) is a business-driven, functional framework classifying Service Components according to how they support business and performance objectives. The model aids in recommending service capabilities to support the reuse of business components and services across the enterprise. The SRM should be structured across horizontal service areas that, independent of the business functions, can provide a leverage-able foundation for reuse of applications, application capabilities, components, and business services.

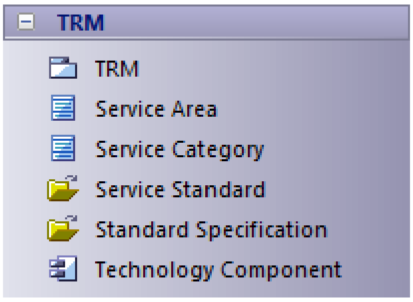


## FEAF Service Component Reference Model Toolbox

| Item | Description |
|---|---|
| SRM | A Package to capture the Service Component Reference Model.<br>Tagged Values – Version |
| Service Domain | Captures a high-level view of the services and capabilities that support enterprise and organizational processes and applications.<br>Tagged Values – ServiceDomainID, Definition |
| Service Type | Groups similar capabilities in support of the domain, providing an additional layer of categorization that defines the context of a specific capability component within a given domain.<br>Tagged Values – ServiceTypeID, Definition, Referencing Service Domain |
| Service Component | Captures a set of capabilities that might be exposed through a business or technology interface. Service Components are 'building blocks' to deliver the information management capability to the business.<br>Tagged Values – ServiceComponentID, Definition, Referencing Service Domain, Referencing Service Type |

# FEAF Technical Reference Model Toolbox Page

The FEAF Technology Reference Model (TRM) is a component-driven, technical framework categorizing the standards and technologies to support and enable the delivery of Service Components and capabilities.



## FEAF Technical Reference Model Toolbox

| Item | Description |
|---|---|
| TRM | A Package to capture the Technology Reference Model.<br>Tagged Value – Version |
| Service Area | Represents a technical tier supporting the secure construction, exchange, and delivery of a Service Component.<br>Tagged Values – ServiceAreaID, Definition |
| Service Category | Classifies a lower level of technology and standard with respect to the business or technology function it serves.<br>Tagged Values – ServiceCategoryID, Definition, Referencing Service Area |
| Service Standard | Defines a standard and technology that supports a Service Category.<br>Tagged Values – ServiceStandardID, Definition, Referencing Service Category |
| Standard Specification | Provides the specification details of the standard.<br>Tagged Value – StandardSpecificationID |

# Gap Analysis Matrix - TOGAF

The Specification document for TOGAF states:

'Gap analysis is widely used in the TOGAF Architecture Development Method (ADM) to validate an architecture that is being developed. The basic premise is to highlight a shortfall between the Baseline Architecture and the Target Architecture; that is, items that have been deliberately omitted, accidentally left out, or not yet defined.'

TOGAF provides a Gap Analysis Matrix that you can use to:

- Identify gaps between the baseline and target
- Create Gap elements (if any gaps are identified) in the repository, which can later be addressed and assigned as tasks; the Gap elements can then be used to prioritize activities
- Create and manage Gap Analysis Matrix profiles

## Notes

- This feature is not available in the Professional edition of Enterprise Architect

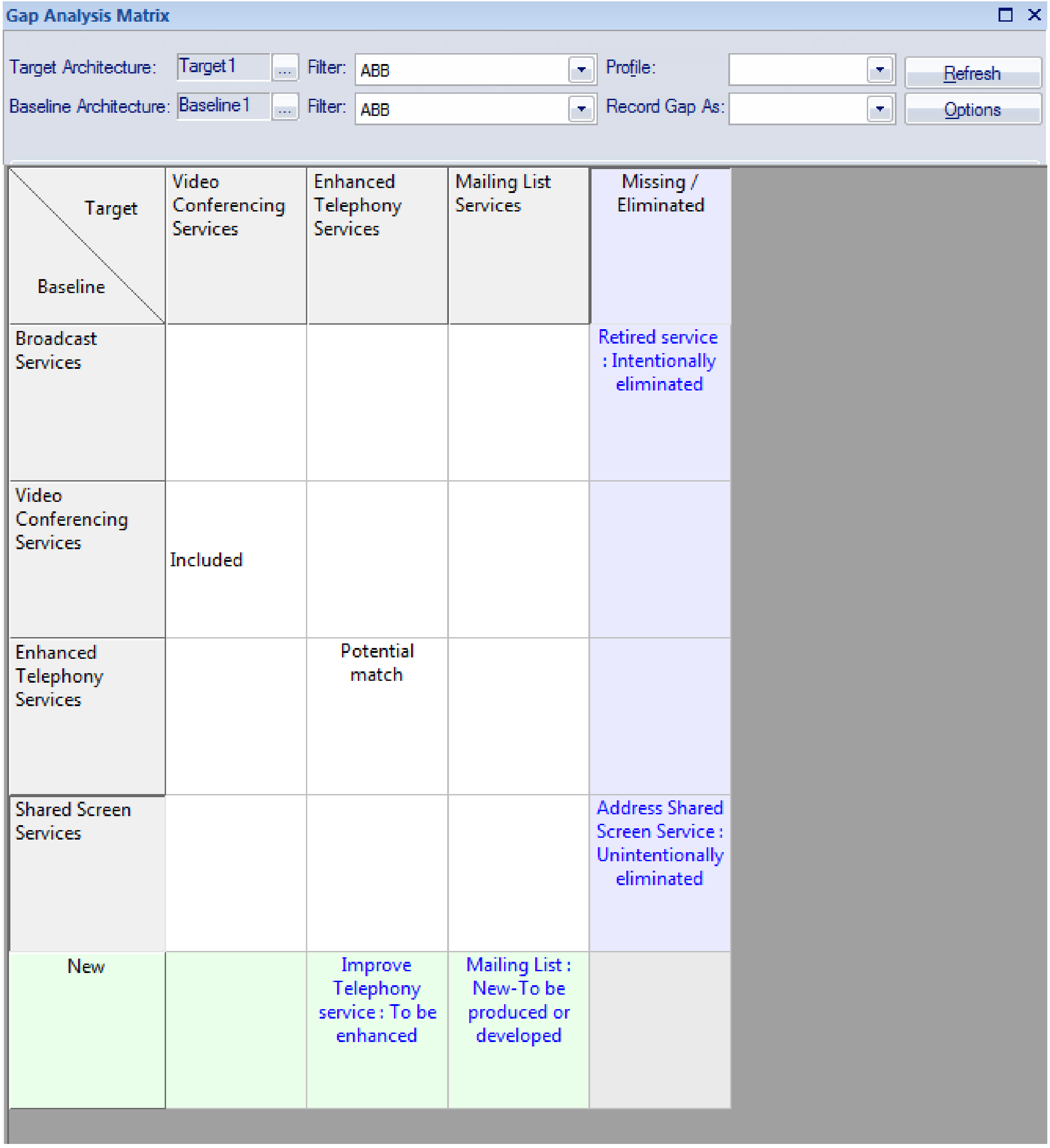
# Open the Matrix

## Access

| | |
|---|---|
| Ribbon | Design > Impact > Gap Analysis |

## Example

This Gap Analysis Matrix example is from the TOGAF Specification; it illustrates Gap Analysis for Architecture Building Blocks (ABBs) that are services from the Network Services category.



Target Architecture: Target1 Filter: ABB Profile:

Baseline Architecture: Baseline1 Filter: ABB Record Gap As:

| Target / Baseline | Video Conferencing Services | Enhanced Telephony Services | Mailing List Services | Missing / Eliminated |
|---|---|---|---|---|
| Broadcast Services | | | | Retired service : Intentionally eliminated |
| Video Conferencing Services | Included | | | |
| Enhanced Telephony Services | | Potential match | | |
| Shared Screen Services | | | | Address Shared Screen Service : Unintentionally eliminated |
| New | | Improve Telephony service : To be enhanced | Mailing List : New-To be produced or developed | |

## Using the Gap Analysis Matrix

The 'Filter' fields list all the stereotypes that can be shown in the matrix; use these fields to set a filter for each of the target and baseline Architectures.

After setting the filters, click on the ... button to the right of the 'Target Architecture' and 'Baseline Architecture' fields, and browse for and select the target Architecture Package and baseline Architecture Package.

Click on the Refresh button; the matrix lists the elements having the stereotypes you set in the 'Filter' fields. The target Architecture elements are listed horizontally as column headings, and the baseline Architecture elements are listed vertically as row titles. If you double-click on the row or column headers containing the baseline or target elements, the corresponding 'Properties' dialog displays.

To locate an object in the Project Browser, right-click on it and select the 'Find in Project Browser' option.

In the cells at the intersection of the target element columns and baseline element rows, you can create and edit Gap Analysis Notes. To edit the notes double-click on the cell, or right-click and select the 'Edit Notes' option.

Any elements that are not in the target Architecture but are available in the baseline Architecture must be addressed as Gap elements in the last column, called 'Missing / Eliminated'. Any elements that are in the target Architecture but not in the baseline Architecture must be addressed as Gap elements in the last row, called 'New'.

In the example:

- Broadcast Services and Shared Screen Services are present in the baseline Architecture but missing from the target Architecture; therefore, you must create appropriate Gap elements in the 'Missing / Eliminated' column, the last column of the matrix
- Mailing List Services is not in the baseline Architecture but it is in the target Architecture, meaning that the service is a new one in the target Architecture and it must be procured or developed; you must create a corresponding Gap element in the 'New' row, the last row of the matrix

## Notes

- Locating the baseline/target element in the Project Browser with the Traceability Window and Element Browser open helps improve gap analysis, as it shows all the elements and details such as Tagged Values that are linked to the element; for example, if an Architecture Building Block (ABB) is missing in the target Architecture, you can see what other processes and tasks depend on this ABB and what processes are impacted, which can also help you to decide whether or not an ABB must be enhanced in the target Architecture

# Create Gap Elements

## Create a Gap element to model an identified gap

1. Right-click on the cell and select the 'Create Gap Element' option. The 'Browse Project' dialog displays.
2. Select the Package in which to create the Gap element and click on the OK button. A Gap element is created in the selected Package and its 'Properties' dialog displays; enter the element name and other required properties.
3. Select the 'Tagged Values' tab of the 'Properties' dialog and set the Tagged Values listed under 'Gap Element Tagged Values'.
4. Click on the OK button. The name and category of the Gap element displays in the selected matrix cell.

## Gap Element Tagged Values

If you intend to use a Gap element that is already available in the model, right-click on the appropriate cell in the 'Missing / Eliminated' column or 'New' row and select the 'Link to Existing Gap Element' option. The 'Select Classifier' dialog displays, from which you select the existing Gap element.

Once you have created a Gap element, you can right-click on its cell and select from these context menu options:

- 'Edit Gap Element' to open the 'Properties' dialog of the Gap element and edit its properties
- 'Locate in Project Browser' to find and highlight the Gap element in the Project Browser
- 'Remove Gap Element Link' to delete the link to the element in that cell (the element still exists in its parent Package)
- 'Delete Gap Element' to delete the element from the model; this action cannot be undone

| Tagged Value | Description |
|---|---|
| ID | The unique identifier for the architecture object. |
| Owner | The owner of the architecture object. |
| Source | The location/source from which the information is collected. |
| Category | The categorization of the Gap. This can have any one of the values:<br>• Intentionally eliminated<br>• Unintentionally eliminated<br>• New – To be produced or developed<br>• To be enhanced |
| RefBaseline Architecture | The name of the baseline Architecture artifact that is linked to the Gap element. If the gap is to point to a missing element, this tag has the value of the baseline artifact that is missing. |
| RefTarget Architecture | The name of the target Architecture artifact that is linked to the Gap element. If the gap points to a new artifact that is required for the target Architecture, this tag has the value of the new target artifact. |

# Gap Analysis Matrix Profiles

On the Gap Analysis Matrix, you can create and manage profiles to save commonly-used combinations of target Architectures and stereotypes.

To work on Gap Analysis Matrix profiles, click on the Options button in the top right corner of the matrix. A submenu displays, listing options to:

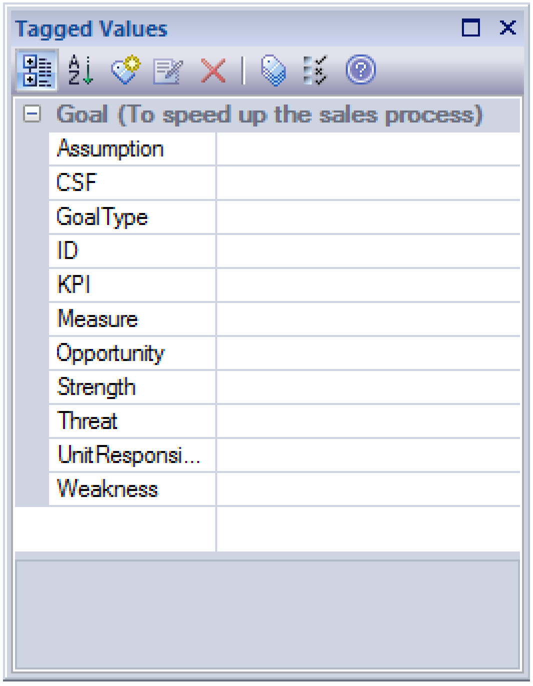
- Create a profile of the current matrix settings
- Update the currently-selected profile in the 'Profile' field
- Delete the currently-selected profile

The 'Profile' field drop-down list shows all the saved Gap Analysis Matrix profiles.

# Tagged Values in TOGAF

TOGAF makes extensive use of Tagged Values for assigning custom properties to the various elements specific to TOGAF. It is recommended that you keep the Tagged Values window docked and visible at all times when creating or viewing a TOGAF model.

To open the Tagged Values window, or bring it to the top if already opened, select the 'Start > Explore > Properties > Tagged Values' ribbon option, or press Ctrl+Shift+6.



Goal (To speed up the sales process)

| | |
|---|---|
| Assumption | |
| CSF | |
| GoalType | |
| ID | |
| KPI | |
| Measure | |
| Opportunity | |
| Strength | |
| Threat | |
| UnitResponsi... | |
| Weakness | |

## Synchronize Tagged Values

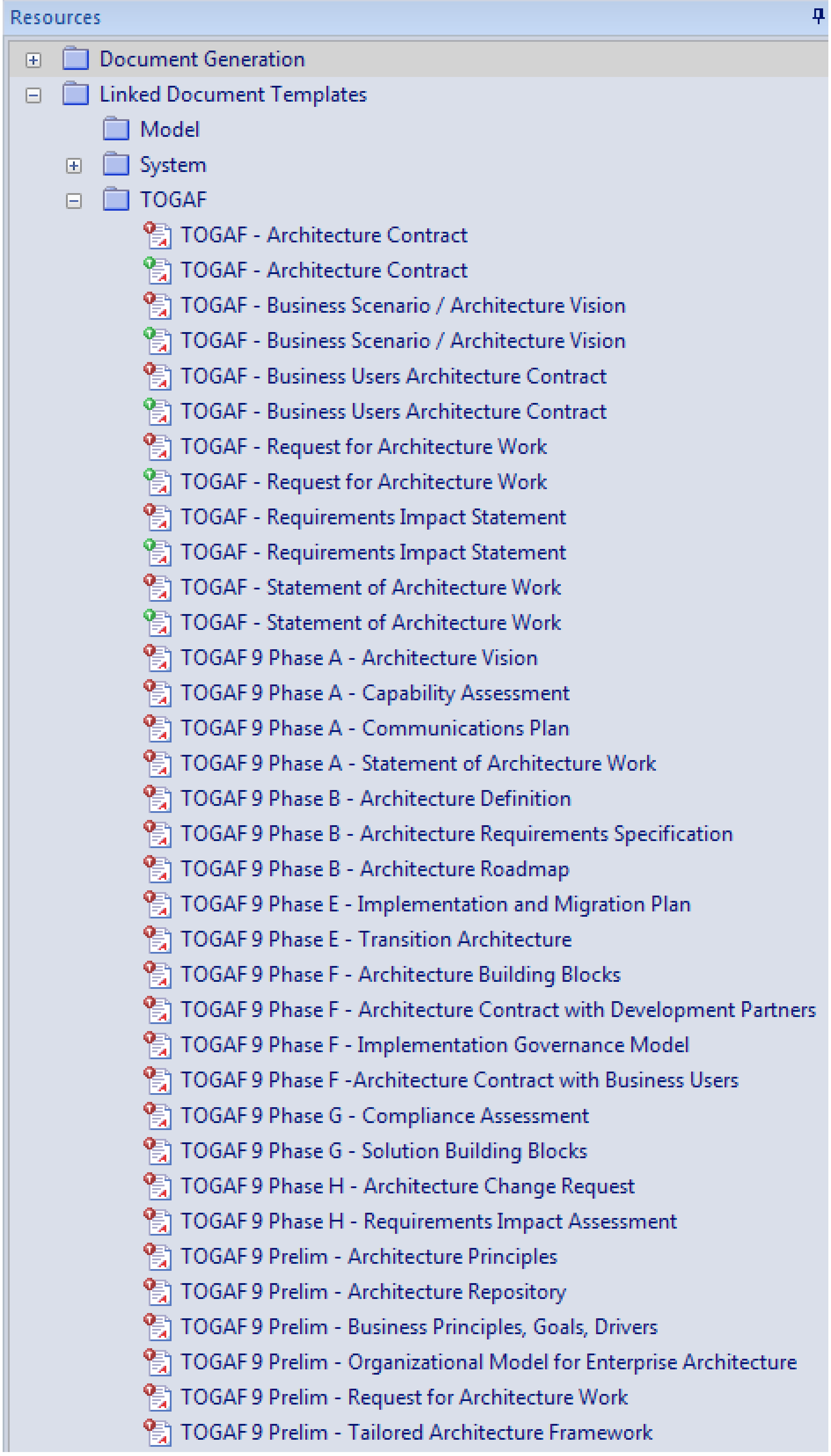
From time to time you might need to add missing Tagged Values to all elements in the model that require them, such as:

- Whenever you create a new element by any means other than directly dropping the element from the TOGAF Toolbox pages
- Before using a new version of the Technology, to update the Tagged Values of elements in existing models to the latest version of the TOGAF profile

You can do this using the 'Synchronize Stereotype' option on the icons in the TOGAF pages of the Diagram Toolbox.

# TOGAF Linked Document Templates

Enterprise Architect contains a set of Linked Document templates that are specific to TOGAF.



- Document Generation
- Linked Document Templates
  - Model
  - System
  - TOGAF
    - TOGAF - Architecture Contract
    - TOGAF - Architecture Contract
    - TOGAF - Business Scenario / Architecture Vision
    - TOGAF - Business Scenario / Architecture Vision
    - TOGAF - Business Users Architecture Contract
    - TOGAF - Business Users Architecture Contract
    - TOGAF - Request for Architecture Work
    - TOGAF - Request for Architecture Work
    - TOGAF - Requirements Impact Statement
    - TOGAF - Requirements Impact Statement
    - TOGAF - Statement of Architecture Work
    - TOGAF - Statement of Architecture Work
    - TOGAF 9 Phase A - Architecture Vision
    - TOGAF 9 Phase A - Capability Assessment
    - TOGAF 9 Phase A - Communications Plan
    - TOGAF 9 Phase A - Statement of Architecture Work
    - TOGAF 9 Phase B - Architecture Definition
    - TOGAF 9 Phase B - Architecture Requirements Specification
    - TOGAF 9 Phase B - Architecture Roadmap
    - TOGAF 9 Phase E - Implementation and Migration Plan
    - TOGAF 9 Phase E - Transition Architecture
    - TOGAF 9 Phase F - Architecture Building Blocks
    - TOGAF 9 Phase F - Architecture Contract with Development Partners
    - TOGAF 9 Phase F - Implementation Governance Model
    - TOGAF 9 Phase F -Architecture Contract with Business Users
    - TOGAF 9 Phase G - Compliance Assessment
    - TOGAF 9 Phase G - Solution Building Blocks
    - TOGAF 9 Phase H - Architecture Change Request
    - TOGAF 9 Phase H - Requirements Impact Assessment
    - TOGAF 9 Prelim - Architecture Principles
    - TOGAF 9 Prelim - Architecture Repository
    - TOGAF 9 Prelim - Business Principles, Goals, Drivers
    - TOGAF 9 Prelim - Organizational Model for Enterprise Architecture
    - TOGAF 9 Prelim - Request for Architecture Work
    - TOGAF 9 Prelim - Tailored Architecture Framework

You can also select these templates from the drop-down list in the 'New Linked Document from Template' dialog; scroll down to the 'Technology Templates' list.

The Linked Document templates are provided by The Open Group, contingent on this text being displayed in any documentation of the templates:

# The TOGAF Architecture Development Method

The key to TOGAF remains a reliable, practical method - the TOGAF Architecture Development Method (ADM) - for defining business needs and developing an architecture that meets those needs, applying the elements of TOGAF and other architectural assets available to the organization.

TOGAF embodies the concept of the Enterprise Continuum to reflect different levels of abstraction in an architecture development process. In this way TOGAF facilitates understanding and co-operation between actors at different levels. It provides a context for the use of multiple frameworks, models, and architecture assets in conjunction with the TOGAF ADM. By means of the Enterprise Continuum, architects are encouraged to leverage all other relevant architectural resources and assets, in addition to the TOGAF Foundation Architecture, in developing an organization-specific IT architecture.

## Key Points About the ADM

The ADM is iterative over the whole process, between phases and within phases; for each iteration of the ADM, a fresh decision must be taken on:

- The breadth of coverage of the enterprise to be defined
- The level of detail to be defined
- The extent of the time horizon aimed at, including the number and extent of any intermediate time horizons
- The architectural assets to be leveraged in the organization's Enterprise Continuum, including:
  - Assets created in previous iterations of the ADM cycle within the enterprise
  - Assets available elsewhere in the industry (such as other frameworks, systems models and vertical industry models)

These decisions must be made on the basis of a practical assessment of resource and competence availability, and the value that can realistically be expected to accrue to the enterprise from the chosen scope of the architecture work.

As a generic method, the ADM is intended to be used by enterprises in a wide range of different geographies and applied in different vertical sectors/industry types. As such it can be - but does not necessarily have to be - tailored to specific needs. For example, it can be used:

- In conjunction with the set of deliverables of another framework, where these are more appropriate for a specific organization; many US federal agencies have developed individual frameworks that define the deliverables specific to their particular departmental needs
- In conjunction with the well-known Zachman Framework, which is an excellent classification scheme but which lacks an openly available, well-defined methodology

# ADM Documentation

The Architecture Development Method (ADM) has ten Phases, as identified here. The approach and complete description of each Phase are provided in the TOGAF 9 documentation available on The Open Group website, to identify the objectives, inputs, steps and outputs of each phase.

## Preliminary Phase: Framework and Principles

The Preliminary Phase is about defining 'where, what, why, who, and how we do architecture' in the enterprise concerned. The main aspects are:

- Defining the enterprise
- Identifying key drivers and elements in the organizational context
- Defining the requirements for architecture work
- Defining the architecture principles that will inform any architecture work
- Defining the framework to be used
- Defining the relationships between management frameworks
- Evaluating the enterprise architecture maturity

## Phase A: Architecture Vision

Architecture Vision starts with receipt of a Request for Architecture Work from the sponsoring organization to the architecture organization. During this phase, you define the architecture scope, how to create the vision, and obtain approvals.

## Phase B: Business Architecture

Business Architecture is the first architecture activity that must be undertaken, if not catered for already in other organizational processes (such as enterprise planning, strategic business planning or business process re-engineering).

## Phase C: Information Systems Architectures

In this phase you develop the Information Systems Architectures, including the Data and Applications Architectures. Detailed steps for Phase C are given separately for each architecture domain:

- Data Architecture
- Applications Architecture

## Phase D: Technology Architecture

The steps within the Technology Architecture phase are:

- Select reference models, viewpoints, and tools
- Develop Baseline Technology Architecture Description
- Develop Target Technology Architecture Description

- Perform gap analysis
- Define roadmap components
- Resolve impacts across the Architecture Landscape
- Conduct formal stakeholder review
- Finalize the Technology Architecture
- Create Architecture Definition Document

## Phase E: Opportunities and Solutions

In the Opportunities and Solutions phase you identify the parameters of change, the major phases along the way, and the top-level projects to be undertaken in moving from the current environment to the target.

## Phase F: Migration Planning

During the Migration Planning phase you sort the various implementation projects into priority order. Activities include assessing the dependencies, costs and benefits of the various migration projects.

## Phase G: Implementation Governance

During the Implementation Governance phase you bring together all the information for successful management of the various implementation projects.

## Phase H: Architecture Change Management

In the Architecture Change Management phase you establish an architecture change management process for the new enterprise architecture baseline.

## ADM Architecture Requirements Management

The ADM is continuously driven by the Architecture Requirements Management process.

# The TOGAF Enterprise Continuum

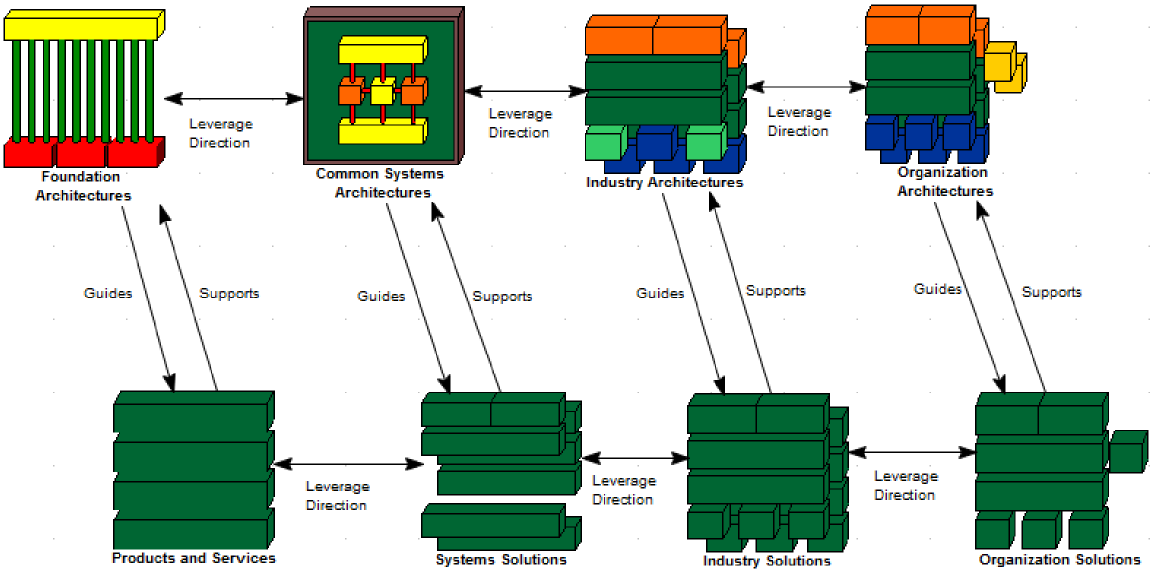
It is simplest to think of the Enterprise Continuum as a 'virtual repository' of all the architecture assets - models, Patterns, architecture descriptions and other artifacts - that exist both within the enterprise and in the IT industry at large, and that the enterprise considers itself to have available for the development of architectures for the enterprise.

Examples of 'assets within the enterprise' are the deliverables of previous architecture work that are available for re-use.

Examples of 'assets in the IT industry at large' are the wide variety of industry reference models and architecture Patterns that exist and that are continually emerging, including those that are:

- Highly generic, such as TOGAF's own Technical Reference Model (TRM)
- Specific to certain aspects of IT, such as a web services architecture, or a generic manageability architecture
- Specific to certain types of information processing, such as e-Commerce or supply chain management
- Specific to certain vertical industries; for example, the models generated by vertical consortia such as TMF (in the Telecommunications sector), ARTS (Retail) or POSC (Petrotechnical)

Enterprise Architect's support for the Enterprise Continuum is provided by the Enterprise Continuum diagram and the corresponding Diagram Toolbox page. The starter model consists of an interface to the TOGAF Enterprise Continuum.



When you double-click on an Architecture Continuum or Solution Continuum element, an Enterprise Continuum diagram displays. The Diagram Toolbox page provides the Architecture Building Block and Solutions Building Block elements and the appropriate relationship connectors.

# Support For Federal Enterprise Architecture Framework

TOGAF provides diagrams and Toolbox pages specific to the Federal Enterprise Architecture Framework (FEAF). It also provides 'out-of-the-box' models of the FEAF Performance Reference model and Technical Reference model.

To open FEAF-PRM and FEAF-TRM models:

1. Create a new Enterprise Architect project file, and click on the top-level Package.
2. Select the 'Start > View > Perspective > Open Model Wizard' option.
3. In the Model Wizard, select the 'Enterprise Architecture > TOGAF' Perspective and the required FEAF Pattern.
4. Click on the Create Patterns button.

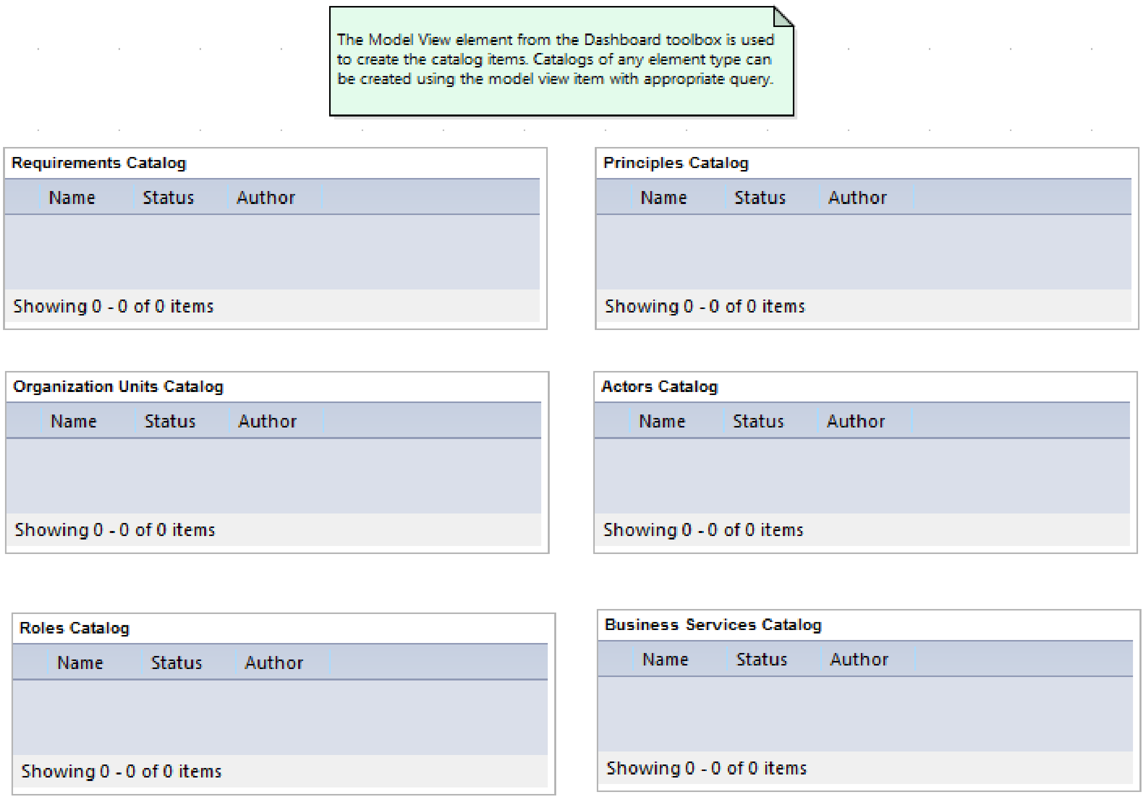
These TOGAF Toolbox pages provide specific support for FEAF:

- FEAF Business Reference Model Toolbox Page
- FEAF Performance Reference Model Toolbox Page
- FEAF Service Component Reference Model Toolbox Page
- FEAF Technical Reference Model Toolbox Page

# TOGAF Catalogs

Enterprise Architect helps you to create Model Catalog Artifacts, using the TOGAF-Catalog model Pattern. Choosing this model Pattern in the Model Wizard generates a template model in which you create TOGAF-specific catalogs for:

- Actors
- Business Services
- Organization Units
- Principles
- Requirements and
- Roles



The Model View element from the Dashboard toolbox is used to create the catalog items. Catalogs of any element type can be created using the model view item with appropriate query.

Requirements Catalog

| Name | Status | Author |
|---|---|---|

Showing 0 - 0 of 0 items

Principles Catalog

| Name | Status | Author |
|---|---|---|

Showing 0 - 0 of 0 items

Organization Units Catalog

| Name | Status | Author |
|---|---|---|

Showing 0 - 0 of 0 items

Actors Catalog

| Name | Status | Author |
|---|---|---|

Showing 0 - 0 of 0 items

Roles Catalog

| Name | Status | Author |
|---|---|---|

Showing 0 - 0 of 0 items

Business Services Catalog

| Name | Status | Author |
|---|---|---|

Showing 0 - 0 of 0 items

Each Model View will list all objects of the corresponding type in the entire model.

Alternatively, you can create TOGAF Catalogs in a diagram using Model View elements from the 'Dashboard' pages of the Diagram Toolbox.

# UPDM

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

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

## Discussion

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

| Section | Content |
|---|---|
| Welcome | This section provides an introduction to UPDM, and contains the formal documentation defining its use with Enterprise Architect. |
| Using UPDM | Get started with UPDM, learning about the model structure, templates, diagram types and more. |
| Model Validation | Learn how to develop and configure model validation for UPDM. |

# Welcome to the MDG Technology for UPDM

Welcome to the MDG Technology for UPDM 2.0, in Sparx Systems Enterprise Architect.

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

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

The Technology is already integrated with the Enterprise Architect Ultimate and Unified editions; it can be purchased separately to be used with the Enterprise Architect Professional or Corporate editions.

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

## Getting Started

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

# MDG Technology for UPDM Copyright Notice

## Copyright © 2010 - 2019 Sparx Systems Pty. Ltd. All rights reserved.

Sparx Systems Pty. Ltd.

99 Albert St,

Creswick, Victoria 3363,

AUSTRALIA

Phone: +61 (3) 5345 1140

Fax: +61 (3) 5345 1104

Support Email: support@sparxsystems.com

Sales Email: sales@sparxsystems.com

Website: sparxsystems.com

# MDG Technology for UPDM Software Product License Agreement

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

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

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 the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

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

## DEFINITIONS

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

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

## GRANT OF LICENSE

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

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

additional fee.

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

## EVALUATION LICENSE

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

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

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

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

## ADDITIONAL RIGHTS AND LIMITATIONS

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

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

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

## ASSIGNMENT

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

## TERMINATION

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

## WARRANTIES AND LIABILITY

WARRANTIES

SPARX warrants that:

- The SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and

- Any SUPPORT SERVICES provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX SUPPORT ENGINEERS will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

EXCLUSIONS

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

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

LIMITATION

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

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

## TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and Licenses. Windows, Windows XP, Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7 and Windows 8 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.

# Acknowledgement of Trademarks - UPDM

## Trademarks of Microsoft

- Microsoft®
- Windows®

## Trademarks of the OMG

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

# MDG Technology for UPDM Support

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

# MDG Technology for UPDM System Requirements

The MDG Technology for UPDM version 2.0 runs under these environments:

## Operating Systems

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

## Enterprise Architect Versions

- Enterprise Architect Version 9.0 or later

# Getting Started with MDG Technology for UPDM

When you install the Unified or Ultimate editions of Enterprise Architect, the MDG Technology for UPDM is fully enabled and ready to use.

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

## Access the MDG Technology

1. Create a new Enterprise Architect project file, and click on the top-level Package.
2. Select the 'Start > View > Perspective > Open Model Wizard' option.
3. In the Model Wizard, select the 'Enterprise Architecture > UPDM' Perspective and the 'UPDM Frameworks' Pattern Group; select either the 'DoDAF Framework' Pattern or the 'MODAF Framework' Pattern.
4. Click on the Create Patterns button.

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

# Using UPDM

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

You can use the MDG Technology for UPDM to perform UPDM modeling within Enterprise Architect. The Technology provides:

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

# Model Wizard in the MDG Technology for UPDM

You can create UPDM models within your project using templates selected from the Enterprise Architect Model Wizard

## Access

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

## Notes

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

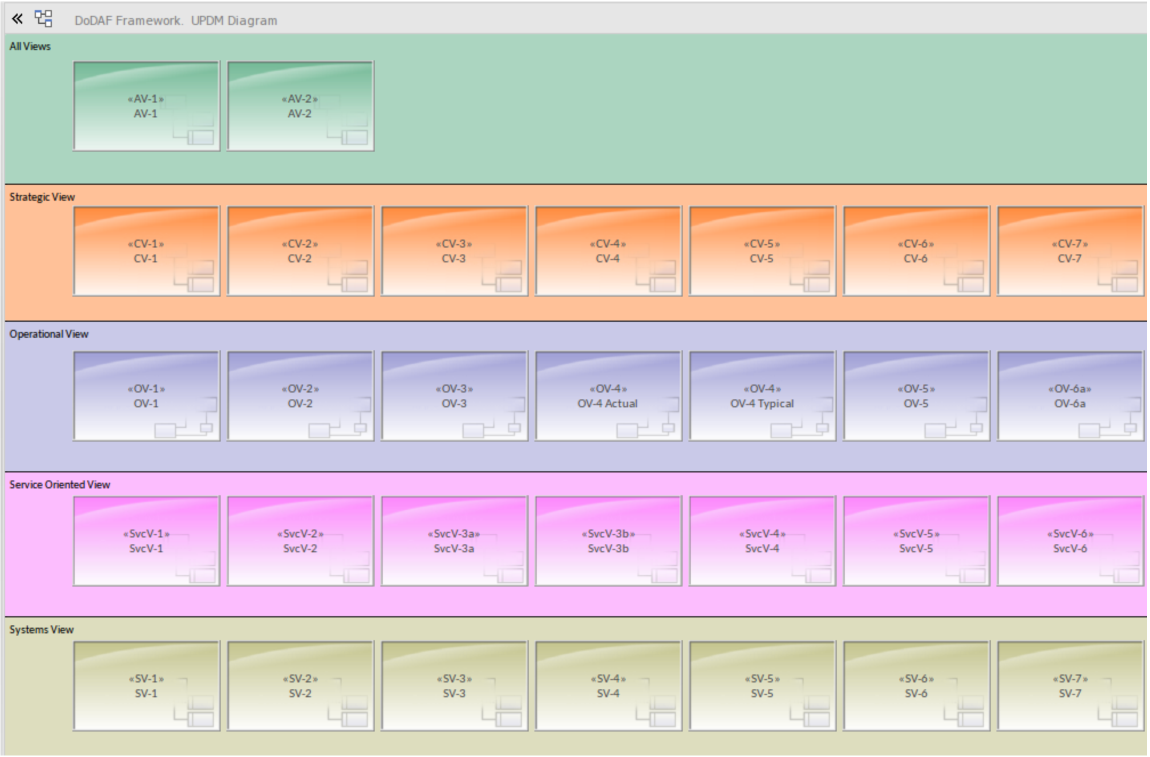
# UPDM Framework Diagram

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

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

## Create a UPDM Framework Diagram

1. In the Model Wizard, click on the '... Perspective' button and select 'System Engineering > UPDM'.
2. Expand the 'UPDM Frameworks' group and click on the required Pattern, either 'DODAF Framework' or 'MODAF Framework'.
3. Click on the Create Patterns button to generate the corresponding UPDM model structures in your project.



## Editing Swimlanes

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

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

## Changing Appearances

Each Package, document and hyperlink on the Framework diagram has an alternative image. To load these images into your model, select 'Specialize > Technologies > UPDM > Import UPDM Images'.

If you want to apply your own bitmap images to the UPDM elements, you must first import the images into the model using the 'Configure > Reference Data > Images' option. 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 this Shape Script:

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

# UPDM Extensions Menu

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

## Access

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

## Options

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

# UPDM 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 Diagram Toolbox pages for the diagram type.

You can use the UPDM diagrams that are generated by the Model Wizard, or create a new UPDM diagram.

## Access

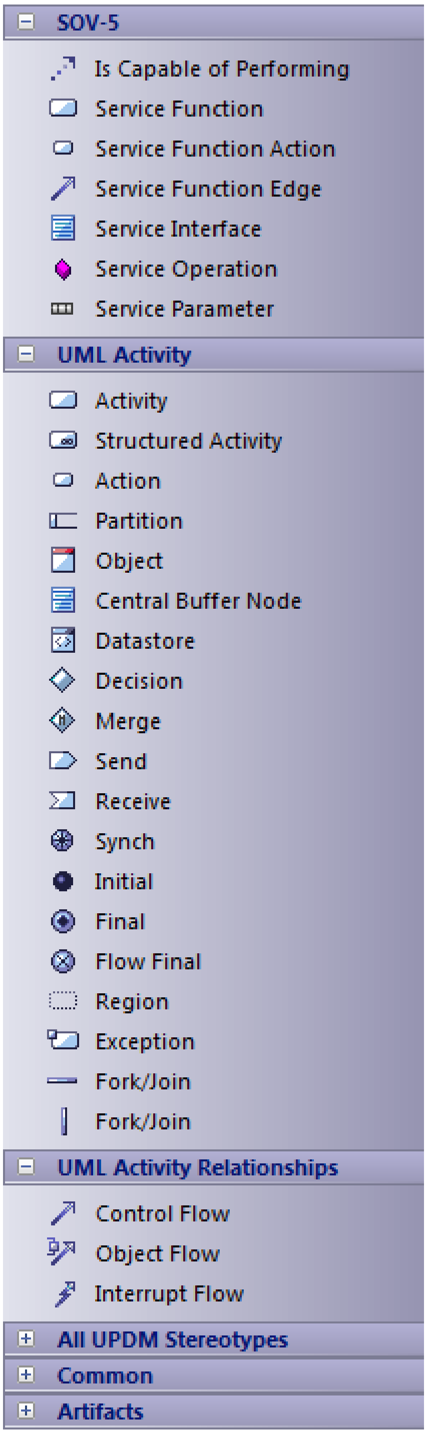
| | |
|---|---|
| Ribbon | Design > Diagram > Insert |
| Context Menu | Project Browser \| Right-click on Package \| Add Diagram |

## Notes

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

# UPDM Toolboxes

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



In addition, the 'Common' elements and relationships page and the 'Artifacts' page of the Diagram Toolbox are 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 Diagram Toolbox page is provided that includes every stereotype in the UPDM profile, listed in alphabetical order. If you cannot remember which context-sensitive Toolbox page a stereotype appears in, just go to the 'All UPDM Stereotypes' Toolbox page instead. To make this page available at all times, either:

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

# UPDM Stereotypes

## ActualMeasurementSet

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

Extensions:

- Object

Constraints:

- Classifier must be a MeasurementSet

Use:

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

## ActualOrganization

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

Extensions:

- Object

Generalizations:

- ActualOrganizationalResource

Constraints:

- Classifier must be an Organization

Use:

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

## ActualOrganizationRelationship

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

Extensions:

- InformationFlow

Constraints:

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

## ActualPerson

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

Extensions:

- Object

Constraints:

- Classifier must be a Person

Use:

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

## ActualPost

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

Extensions:

- Object

Generalizations:

- ActualOrganizationalResource

Constraints:

- Classifier must be a Post

Use:

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

## ActualProject

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

Extensions:

- Object

Constraints:

- Classifier must be a Project

Use:

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

## ActualProjectMilestone

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

See also: IncrementMilestone, OutOfServiceMilestone, NoLongerUsedMilestone and DeployedMilestone.

Extensions:

- Object

Specializations:

- IncrementMilestone
- OutOfServiceMilestone
- NoLongerUsedMilestone
- DeployedMilestone

Constraints:

- Classifier must be a ProjectMilestone

Use:

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

## Alias

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

Extensions:

- Note

Constraints:

- AnnotatedElement must be a UPDMElement

Use:

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

## Arbitrary Relationship

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

Extensions:

- Dependency

Constraints:

- Client and Supplier must both be stereotyped ConceptRole

Use:

- Drag a Quicklink from a ConceptRole

## ArchitecturalDescription

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

Extensions:

- Package

Use:

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

## ArchitecturalReference

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

Extensions:

- Dependency

Constraints:

- Client and Supplier must both be stereotyped ArchitecturalDescription

Use:

- Drag a Quicklink from an ArchitecturalDescription.

## ArchitectureMetadata

Information on architectural description; used in AV-1.

Extensions:

- Note

Generalizations:

- Metadata

Constraints:

- AnnotatedElement must be an ArchitecturalDescription

Use:

- Drag a quicklink from an ArchitecturalDescription

## Capability

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

Extensions:

- Class

Generalizations:

- SubjectOfForecast

Use:

- Can have a set of associated environment conditions stereotyped Environment
- Capabilities can be composed of Capabilities (Composite aggregation)
- Capabilities can be dependent on Capabilities (Dependency)

- Capabilities can sub-class Capabilities (Generalization)
- Can be supplier or client of a Forecast (both must be same stereotype) (from SubjectOfForecast)

## CapabilityConfiguration

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

Extensions:

- Class

Generalizations:

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

Specializations:

- SystemsNode

Use: Can:

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

## Climate

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

Extensions:

- Class

Generalizations:

- EnvironmentalType

Use:

- Can be the type of an EnvironmentProperty

## Commands

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

Extensions:

- InformationFlow

Generalizations:

- ResourceInteraction

Constraints:

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

Use:

- Conveys a DataElement

## CompatibleWith

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

Extensions:

- Dependency

Constraints:

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

Use:

- Drag a Quicklink from a Node

## Competence

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

Extensions:

- Class

Generalizations:

- SubjectOfForecast

Use: Can be:

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

## ConceptRole

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

Extensions:

- Part

Constraints:

- Type is a ConceptItem

Use:

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

## ConfigurationExchange

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

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Constraints:

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

Use:

- Conveys a CapabilityConfiguration

## Controls

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

Extensions:

- InformationFlow

Generalizations:

- ResourceInteraction

Constraints:

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

Use:

- Conveys a DataElement

## DataElement

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

Extensions:

- Class

Generalizations:

- SubjectOfResourceConstraint
- ResourceInteractionItem
- SystemsElement
- SubjectOfResourceStateMachine

Use:

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

## DataExchange

A DoDAF alias for ResourceInteraction.

Extensions:

- InformationFlow

Generalizations:

- ResourceInteraction
- SystemsElement

Use:

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

## DefinesArchitecture

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

Extensions:

- Realization

Constraints:

- Client is an ArchitecturalDescription
- Supplier is an EnterprisePhase

Use:

- Drag a Quicklink from an ArchitecturalDescription

## Definition

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

Extensions:

- Note

Constraints:

- Annotated Element is a UPDMElement

Use:

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

## DeployedMilestone

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

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone (from ActualProjectMilestone)

Use: Can:

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

## EnduringTask

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

Extensions:

- Class

Use:

- Target of association from EnterprisePhase

## Energy

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

Extensions:

- Class

Generalizations:

- ResourceInteractionItem
- OperationalExchangeItem

Use:

- Conveyed on a EnergyExchange information flow

## EnergyExchange

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

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange
- OperationalElement

Constraints:

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

Use:

- Conveys a Class stereotyped Energy

## EnterpriseGoal

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

Extensions:

- Class

Use:

- Has an association to one EnterprisePhase

## EnterprisePhase

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

Extensions:

- Class

Specializations:

- WholeLifeEnterprise

Use:

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

- Can be Supplier of a DefinesArchitecture Realization

## EnterpriseVision

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

Extensions:

- Class

Use:

- Has an association to one EnterprisePhase

## EntityAttribute

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

Extensions:

- Attribute

Use:

- Is owned by an EntityItem

## EntityItem

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

Extensions:

- Class

Constraints:

- Owned attributes must be stereotyped EntityAttribute

Use: Can:

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

## EntityRelationship

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

Extensions:

- Association

Constraints:

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

## Environment

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

Extensions:

- Class

Constraints:

- Owned attributes must be EnvironmentProperty

## EnvironmentProperty

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

Extensions:

- Attribute

Constraints:

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

Use:

- Owned by an Environment element

## Equipment

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

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

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

Use:

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

## ExhibitsCapability

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

Extensions:

- Dependency

Constraints:

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

## Expose

Assertion that a service interface exposes a capability.

Extensions:

- Dependency

Constraints:

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

## ExternalIndividual

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

Extensions:

- Object

Use:

- Can be the supplier of a SameAs dependency

## ExternalNode

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

Extensions:

- Class

Generalizations:

- Node
- Performer

Use: Can:

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

## ExternalType

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

Extensions:

- Class

Use:

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

## FieldedCapability

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

Extensions:

- Object

Constraints:

- Its classifier must be a CapabilityConfiguration

## FillsPost

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

Extensions:

- Dependency

Constraints:

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

## Forecast

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

Extensions:

- Dependency

Specializations:

- TechnologyForecast

Constraints:

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

## Function

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

Extensions:

- Activity

Generalizations:

- PerformedActivity
- SystemsElement
- SubjectOfResourceConstraint

Constraints:

- Owned parameters are FunctionParameter

Use: Can:

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

## FunctionAction

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

Extensions:

- Action (Call Behavior)

Specializations:

- SystemFunctionAction

Constraints:

- Activity is stereotyped Function

Use:

- Ctrl+L to set the function

## FunctionEdge

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

Extensions:

- ControlFlow

Generalizations:

- SystemsElement

Specializations:

- SystemFunctionEdge

Constraints:

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

Use:

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

## FunctionParameter

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

Extensions:

- ActivityParameter

Constraints:

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

Use:

- Owned by a Function

## HighLevelOperationalConcept

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

Extensions:

- Class

Constraints:

- Owned attributes are ConceptRole

Use:

- Can have a set of described Mission

## HostedSoftware

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

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

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

Use: Can:

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

## HumanResource

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

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

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

Use: Can:

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

## ImplementsOperational

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

Extensions:

- Dependency

Constraints:

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

## IncrementMilestone

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

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone (from ActualProjectMilestone)

Use:

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

## InformationElement

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

Extensions:

- Class

Generalizations:

- OperationalExchangeItem
- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine
- OperationalElement

Use: Can:

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

## InformationExchange

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

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Constraints:

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

## InternalDataModel

DoDAF alias for PhysicalDataModel; used in SV-11.

Extensions:

- Package

Generalizations:

- PhysicalDataModel
- DataModel

Constraints:

- Owns EntityItem elements (from DataModel)

## KnownResource

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

Extensions:

- Part

Generalizations:

- NodeChild

Constraints:

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

## LightCondition

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

Extensions:

- Class

Generalizations:

- EnvironmentalType

Use:

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

## Location

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

Extensions:

- Class

Generalizations:

- ReferredLocation
- ConceptItem
- EnvironmentalType

Use: Can be:

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

## LogicalArchitecture

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

Extensions:

- Class

Generalizations:

- NodeParent

Use:

- Can own ProblemDomain properties

## LogicalDataModel

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

Extensions:

- Package

Generalizations:

- DataModel

Constraints:

- Owns EntityItem elements (from DataModel)

## MapsToCapability

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

Extensions:

- Dependency

Constraints:

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

## MaterielExchange

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

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Constraints:

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

Use:

- Can convey a ResourceArtifact or Software

## Measurement

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

Extensions:

- Attribute

Specializations:

- PerformanceParameter

Use:

- Owned by a MeasurementSet Class

## MeasurementSet

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

Extensions:

- Class

Constraints:

- Owned attributes must be Measurement

Use:

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

## Metadata

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

Extensions:

- Note

Specializations:

- ArchitectureMetadata

## MilestoneSequence

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

Extensions:

- Dependency

Constraints:

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

## Mission

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

Extensions:

- UseCase

Generalizations:

- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine

Use:

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

## MovementOfPeople

MODAF alias for OrganizationalExchange.

Extensions:

- InformationFlow

Generalizations:

- OrganizationalExchange

Use:

- Conveys an OrganizationalResource (Organization or Post)

## Needline

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

Extensions:

- Association
- Connector

Generalizations:

- OperationalElement

Constraints:

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

Use:

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

## NoLongerUsedMilestone

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

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone (from ActualProjectMilestone)

Use:

- Has set of associations to 'noLongerUsedBy' ActualOrganizationalResource (ActualOrganization or ActualPost) objects
- Can have a set of associated Resource (from ActualProjectMilestone)
- Can be client/supplier of a MilestoneSequence (from ActualProjectMilestone)
- Has a set of associations with 'configuration' CapabilityConfiguration Classes

## Node

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

Extensions:

- Class

Generalizations:

- Performer
- ConceptItem
- NodeParent

- SubjectOfOperationalConstraint
- SubjectOfOperationalStateMachine
- OperationalElement

Specializations:

- OperationalNode

Constraints:

- Owned ports must be NodePort, RequestPoint or ServicePoint

Use: Can:

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

## NodePort

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

Extensions:

- Port

Constraints:

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

Use:

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

## NodeRole

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

Extensions:

- Part

Generalizations:

- NodeChild

Specializations:

- ProblemDomain

Constraints:

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

## OperationalActivity

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

Extensions:

- Activity

Generalizations:

- PerformedActivity
- SubjectOfOperationalConstraint
- OperationalElement
- SubjectOfOperationalStateMachine

Specializations:

- StandardOperationalActivity

Constraints:

- Owned parameters must be OperationalParameter

Use: Can:

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

## OperationalActivityAction

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

Extensions:

- CallBehaviorAction

Constraints:

- Activity/Behavior must be an OperationalActivity

Use:

- Can be the Source or Target of an OperationalActivityEdge

## OperationalActivityEdge

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

Extensions:

- ControlFlow

Generalizations:

- OperationalElement

Constraints:

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

Use: Can:

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

## OperationalConstraint

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

Extensions:

- Constraint

Specializations:

- OperationalRule

Constraints:

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

## OperationalMessage

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

Extensions:

- Message

Generalizations:

- OperationalElement

Use:

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

## OperationalNode

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

Extensions:

- Class

Generalizations:

- Node

Constraints:

- Owned ports must be NodePort, RequestPoint or ServicePoint

Use: Can:

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

## OperationalParameter

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

Extensions:

- ActivityParameter

Constraints:

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

Use:

- Can be owned by an OperationalActivity

## OperationalRule

A DoDAF alias for OperationalConstraint.

Extensions:

- Constraint

Generalizations:

- OperationalConstraint

Constraints:

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

## OperationalStateMachine

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

Extensions:

- StateMachine

Constraints:

- Owner is SubjectOfOperationalStateMachine (Mission, InformationElement or Node)

## Organization

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

Extensions:

- Class

Generalizations:

- OrganizationalResource
- Resource, Performer
- SubjectOfForecast
- SubjectOfResourceConstraint

Use: Can:

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

## OrganizationalExchange

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

Extensions:

- InformationFlow

Generalizations:

- OperationalExchange

Specializations:

- MovementOfPeople

Constraints:

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

## OutOfServiceMilestone

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

Extensions:

- Object

Generalizations:

- ActualProjectMilestone

Constraints:

- Classifier must be a ProjectMilestone

Use:

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

## OwnsProcess

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

Extensions:

- Dependency

Constraints:

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

## Part

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

Extensions:

- Part

Generalizations:

- ResourceRole

Specializations:

- SubSystemPart

Constraints:

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

Use: Can have:

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

## PerformanceParameter

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

Extensions:

- Attribute

Generalizations:

- Measurement

Use:

- Owned by a MeasurementSet class

## Performs

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

Extensions:

- Dependency

Constraints:

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

## Person

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

Extensions:

- Class

Use:

- Can be Classifier of an ActualPerson

## PhysicalDataModel

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

Extensions:

- Package

Generalizations:

- DataModel

Specializations:

- InternalDataModel

Constraints:

- Owns EntityItem elements (from DataModel)

## PhysicalLocation

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

Extensions:

- Class

Generalizations:

- ReferredLocation
- ConceptItem
- EnvironmentalType

Use: Can be:

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

## Platform

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

Extensions:

- Part

Generalizations:

- ResourceComponent
- ResourceRole

Constraints:

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

Use:

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

## Post

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

Extensions:

- Class

Generalizations:

- OrganizationalResource
- Resource
- Performer
- SubjectOfForecast
- SubjectOfResourceConstraint

Use: Can:

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

## PostRole

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

Extensions:

- Part

Generalizations:

- OrganizationRole
- ResourceRole

Constraints:

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

Use: Can have a:

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

## ProblemDomain

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

Extensions:

- Part

Generalizations:

- NodeRole
- NodeChild

Constraints:

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

## Project

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

Extensions:

- Class

Use: Can:

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

## ProjectMilestone

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

Extensions:

- Class

Constraints:

- Owned attributes must be ProjectTheme

Use: Can:

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

## ProjectSequence

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

Extensions:

- Dependency

Constraints:

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

## ProjectTheme

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

Extensions:

- Attribute

Constraints:

- Type must be a ProjectThemeStatus

Use:

- Owned by ProjectMilestone

## ProjectThemeStatus

Specifies a status for a ProjectTheme.

Extensions:

- Class

Use:

- The type of a ProjectTheme

## Protocol

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

Extensions:

- Class

Generalizations:

- Standard
- SubjectOfForecast

Use: Can:

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

## ProtocolLayer

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

Extensions:

- Attribute

Constraints:

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

## ProvidesCompetence

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

Extensions:

- Dependency

Constraints:

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

## RealizesCapability

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

Extensions:

- Realization

Constraints:

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

## RequestPoint

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

Extensions:

- Port

Constraints:

- Type must be a ServiceInterface

Use:

- Can be owned by a Node or a Resource

## RequiresCompetence

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

Extensions:

- Dependency

Constraints:

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

## ResourceArtifact

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

Extensions:

- Class

Generalizations:

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

Specializations:

- System

Use: Can:

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

## ResourceComponent

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

Extensions:

- Part

Generalizations:

- ResourceRole

Specializations:

- Platform

Constraints:

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

Use: Can have a:

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

## ResourceConnector

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

Extensions:

- Connector

Generalizations:

- ProtocolImplementation

Constraints:

- End roles must be ResourcePort

Use: Can:

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

## ResourceConstraint

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

Extensions:

- Constraint

Constraints:

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

## ResourceInteraction

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

Extensions:

- InformationFlow

Generalizations:

- SystemsElement
- ProtocolImplementation

Specializations:

- Controls
- Commands
- DataExchange

Constraints:

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

Use:

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

## ResourceInterface

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

Extensions:

- Association
- Connector

Generalizations:

- SystemsElement

Specializations:

- SystemConnector

Constraints:

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

Use:

- Can realize a ResourceInteraction

## ResourceMessage

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

Extensions:

- Message

Generalizations:

- SystemsElement

Use:

- Can have a set of ResourceInteraction that it carries

## ResourcePort

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

Extensions:

- Port

Generalizations:

- ProtocolImplementation

Constraints:

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

Use:

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

## ResourceStateMachine

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

Extensions:

- StateMachine

Generalizations:

- SystemsElement

Constraints:

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

## SameAs

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

Extensions:

- Dependency

Constraints:

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

## ServiceAttribute

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

Extensions:

- Attribute

Use:

- Owned by a ServiceInterface

## ServiceFunction

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

Extensions:

- Activity

Use: Can:

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

## ServiceFunctionAction

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

Extensions:

- CallBehaviorAction

Constraints:

- Behavior must be a ServiceFunction

## ServiceInteraction

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

Extensions:

- Interaction

## ServiceInterface

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

Extensions:

- Class

Constraints:

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

Use: Can:

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

## ServiceMessage

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

Extensions:

- Message

Use:

- Can carry a set of ResourceInteractions

## ServiceOperation

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

Extensions:

- Operation

Constraints:

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

Use: Can:

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

## ServiceOperationAction

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

Extensions:

- CallOperationAction

Constraints:

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

Use:

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

## ServicePoint

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

Extensions:

- Port

Constraints:

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

Use:

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

## ServicePolicy

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

Extensions:

- Constraint

Use:

- Rule can be owned by a ServiceInterface

## ServiceStateMachine

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

Extensions:

- StateMachine

## Software

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

Extensions:

- Class

Generalizations:

- ManufacturedResourceType

- Resource
- SubjectOfForecast
- ResourceInteractionItem
- Performer
- SubjectOfResourceConstraint

Use: Can:

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

## Standard

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

Extensions:

- Class

Generalizations:

- SubjectOfForecast

Specializations:

- Protocol

Use:

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

## StandardConfiguration

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

Extensions:

- Note

Constraints:

- The annotated element must be a CapabilityConfiguration

## StandardOperationalActivity

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

Extensions:

- Activity

Generalizations:

- OperationalActivity
- PerformedActivity
- SubjectOfOperationalConstraint
- OperationalElement
- SubjectOfOperationalStateMachine

Constraints:

- Owned parameters must be OperationalParameter (from OperationalActivity)

Use: Can:

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

## StereotypeExtension

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

Extensions:

- Note

Constraints:

- Annotated element must be a UPDMElement

Use:

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

## StructuralPart

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

Extensions:

- Part

Constraints:

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

## SubOrganization

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

Extensions:

- Part

Generalizations:

- OrganizationRole
- ResourceRole

Constraints:

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

Use: Can:

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

## SubSystemPart

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

Extensions:

- Part

Generalizations:

- Part
- ResourceRole

Constraints:

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

Use: Can:

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

## SupportsOperationalActivity

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

Extensions:

- Dependency

Constraints:

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

## System

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

Extensions:

- Class

Generalizations:

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

Use: Can:

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

## SystemConnector

A link between two systems.

Extensions:

- Association
- Connector

Generalizations:

- ResourceInterface
- SystemsElement

Specializations:

- SystemConnector

Constraints:

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

Use:

- Can realize a ResourceInteraction (from ResourceInterface)

## SystemFunction

A DoDAF alias for Function.

Extensions:

- Activity

Generalizations:

- Function
- PerformedActivity
- SystemsElement
- SubjectOfResourceConstraint

Constraints:

- Owned parameters are FunctionParameter (from Function)

Use: Can:

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

## SystemFunctionAction

A DoDAF alias for FunctionAction.

Extensions:

- CallBehaviorAction

Generalizations:

- FunctionAction

Constraints:

- Activity is stereotyped Function (from FunctionAction)

Use:

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

## SystemFunctionEdge

An alias for FunctionEdge.

Extensions:

- A DoDAF ControlFlow

Generalizations:

- FunctionEdge
- SystemsElement

Constraints:

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

Use:

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

## SystemsNode

A DoDAF alias for CapabilityConfiguration.

Extensions:

- Class

Generalizations:

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

Use: Can:

- Have a set of associated deployed milestones, stereotyped DeployedMilestone (from CapabilityConfiguration)
- Have an optional associated no longer used milestone, stereotyped NoLongerUsedMilestone (from CapabilityConfiguration)
- Have a set of associated increment milestones, stereotyped IncrementMilestone (from CapabilityConfiguration)
- Have an optional associated out of service milestone, stereotyped OutOfServiceMilestone (from CapabilityConfiguration)
- Be annotated by a StandardConfiguration note (from CapabilityConfiguration)
- Be the type of a ConceptRole (from ConceptItem)

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

## TechnologyForecast

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

Extensions:

- Forecast
- Dependency

Constraints:

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

## TemporalPart

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

Extensions:

- Part

Constraints:

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

## UsedConfiguration

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

Extensions:

- Part

Generalizations:

- ResourceRole

Constraints:

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

Use: Can:

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

## VisionStatement

A high-level textual description of an EnterpriseVision.

Extensions:

- Note

## WholeLifeEnterprise

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

Extensions:

- Class

Generalizations:

- EnterprisePhase

Use: Can:

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

# Abstract Stereotypes

## Stereotype Specializations

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

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

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

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

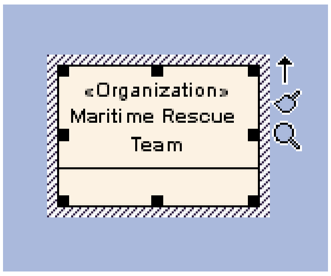
| | |
|---|---|
| | • InformationElement<br>• Node<br>• Mission |
| SubjectOfOperationalStateMachine | The element being described by the StateMachine.<br>Specializations:<br>• OperationalActivity<br>• InformationElement<br>• Node<br>• Mission |
| SubjectOfResourceConstraint | Anything that can be constrained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

| | |
|---|---|
| | elements in a common way.<br>Specializations:<br>• All UPDM stereotypes |

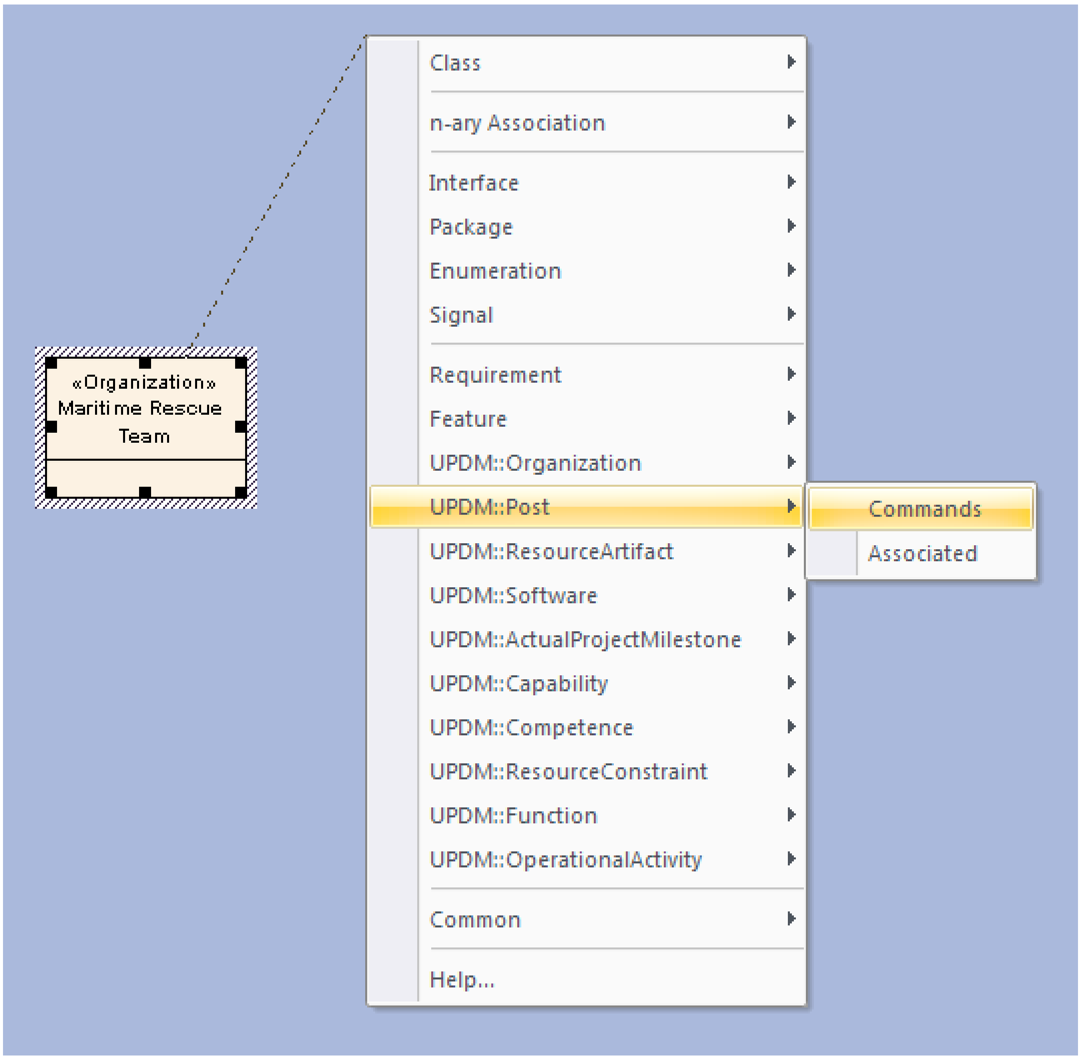
# Quicklinks

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

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



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



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

# Tagged Values for UPDM

UPDM is an extension of UML, which is extended by applying stereotypes to elements, which stereotypes can in turn apply Tagged Values. These 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 open and docked at all times.

## Synchronize Tagged Values

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

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

## The URL/URI Tagged Value

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

# Model Validation in MDG Technology for UPDM

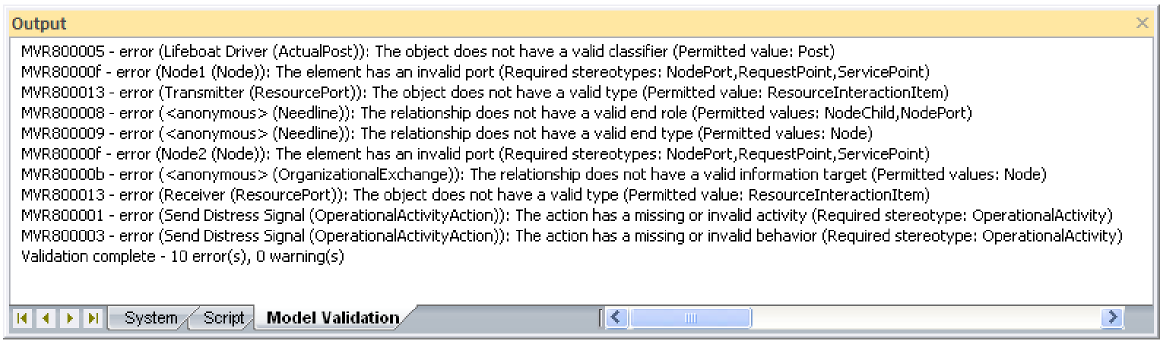
The MDG Technology for UPDM provides model validation of UPDM models, validating and reporting errors against more than 160 different rules.

## Configure Model Validation

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

## Perform Model Validation

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



```
MVR800005 - error (Lifeboat Driver (ActualPost)): The object does not have a valid classifier (Permitted value: Post)
MVR80000f - error (Node1 (Node)): The element has an invalid port (Required stereotypes: NodePort,RequestPoint,ServicePoint)
MVR800013 - error (Transmitter (ResourcePort)): The object does not have a valid type (Permitted value: ResourceInteractionItem)
MVR800008 - error (<anonymous> (Needline)): The relationship does not have a valid end role (Permitted values: NodeChild,NodePort)
MVR800009 - error (<anonymous> (Needline)): The relationship does not have a valid end type (Permitted values: Node)
MVR80000f - error (Node2 (Node)): The element has an invalid port (Required stereotypes: NodePort,RequestPoint,ServicePoint)
MVR80000b - error (<anonymous> (OrganizationalExchange)): The relationship does not have a valid information target (Permitted values: Node)
MVR800013 - error (Receiver (ResourcePort)): The object does not have a valid type (Permitted value: ResourceInteractionItem)
MVR800001 - error (Send Distress Signal (OperationalActivityAction)): The action has a missing or invalid activity (Required stereotype: OperationalActivity)
MVR800003 - error (Send Distress Signal (OperationalActivityAction)): The action has a missing or invalid behavior (Required stereotype: OperationalActivity)
Validation complete - 10 error(s), 0 warning(s)
```

# Model Validation Rules

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

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

## MVRxx0001 - activity

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

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

Solution: Locate the Action in the 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.

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

## MVRxx0002 - annotatedElement

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

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

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

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

## MVRxx0003 - behavior

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

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

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

| Action Stereotypes | Behavior Stereotypes |
|---|---|
| OperationalActivityAction | OperationalActivity |
| ServiceFunctionAction | ServiceFunction |

## MVRxx0004 - class

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

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

Solution: Locate the property in the 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.

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

| | |
|---|---|
| TemporalPart | EnterprisePhase |
| UsedConfiguration | CapabilityConfiguration |

## MVRxx0005 - classifier

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

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

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

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

## MVRxx0006 - client

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

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

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

| Relationship Stereotypes | Client Element Stereotypes |
|---|---|
| ArbitraryRelationship | HighLevelOperationalConcept |
| ArchitecturalReference | ArchitecturalDescription |
| CompatibleWith | Node |
| DefinesArchitecture | ArchitecturalDescription |
| ExhibitsCapability | Node |

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

## MVRxx0007 - constrainedElement

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

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

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

| Constraint Stereotypes | Constrained Element Stereotypes |
|---|---|
| OperationalConstraint | SubjectOfOperationalConstraint |
| ResourceConstraint | SubjectOfResourceConstraint |

## MVRxx0008 - endRoles

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

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

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

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

## MVRxx0009 - endType

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

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

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

| Connector Stereotypes | End Type Stereotypes |
|---|---|
| EntityRelationship | EntityItem |
| Needline | Node |
| ResourceInterface | Resource |

## MVRxx000a - informationSource

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

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

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

| InformationFlow Stereotypes | Source Element Stereotypes |
|---|---|
| ActualOrganizationRelationship | ActualOrganizationalResource |
| Commands | OrganizationalResource |

| | |
|---|---|
| Controls | OrganizationalResource |
| OperationalExchange | Node |
| ResourceInteraction | Resource |

## MVRxx000b - informationTarget

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

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

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

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

## MVRxx000c - ownedAttribute

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

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

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

| Class Stereotypes | Attribute Stereotypes |
|---|---|
| EntityItem | EntityAttribute |
| Environment | EnvironmentProperty |
| HighLevelOperationalConcept | ConceptRole |
| MeasurementSet | Measurement |
| ProjectMilestoneType | ProjectTheme |

| | |
|---|---|
| ServiceInterface | ServiceAttribute |

## MVRxx000d - ownedOperation

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

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

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

| Class Stereotype | Operation Stereotype |
|---|---|
| ServiceInterface | ServiceOperation |

## MVRxx000e - ownedParameter

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

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

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

| Activity Stereotypes | ActivityParameter Stereotypes |
|---|---|
| Function | FunctionParameter |
| OperationalActivity | OperationalParameter |

## MVRxx000f - ownedPort

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

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

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

| Class Stereotypes | Port Stereotypes |
|---|---|
| Node | NodePort |
| Node | RequestPoint |
| Node | ServicePoint |
| Resource | RequestPoint |

| | |
|---|---|
| Resource | ResourcePort |
| Resource | ServicePoint |

## MVRxx0010 - source

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

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

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

| ActivityEdge Stereotypes | Source Element Stereotypes |
|---|---|
| FunctionEdge | ServiceOperationAction |
| OperationalActivityEdge | OperationalActivityAction |

## MVRxx0011 - supplier

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

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

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

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

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

## MVRxx0012 - target

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

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

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

| ActivityEdge Stereotypes | Target Element Stereotypes |
|---|---|
| FunctionEdge | ServiceOperationAction |
| OperationalActivityEdge | OperationalActivityAction |

## MVRxx0013 - type

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

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

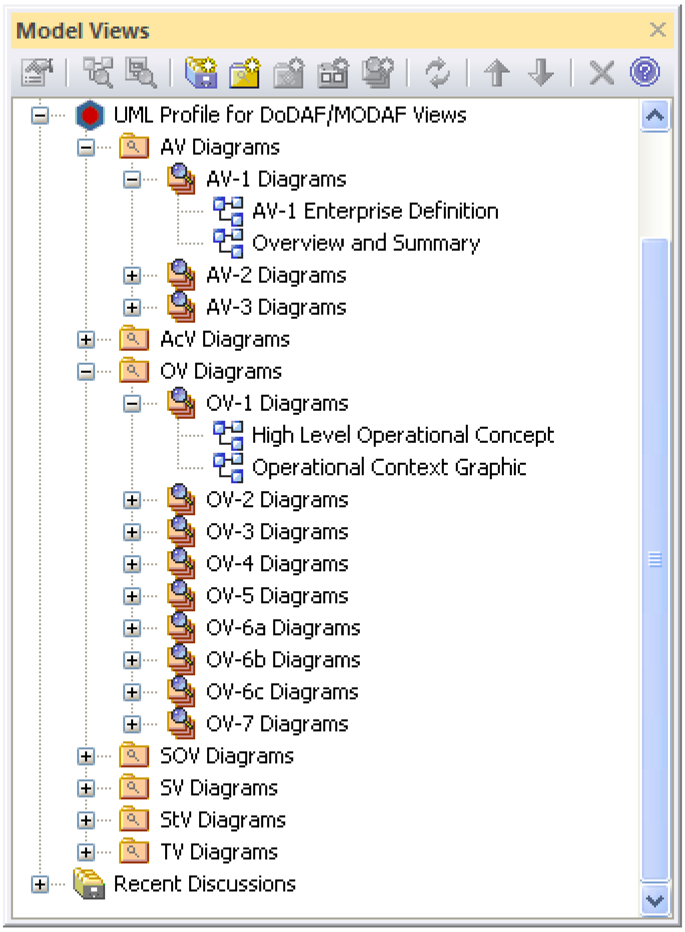
Solution: For Parts, right-click on the Part and select 'Advanced | Set Property Type', or press Ctrl+L, and select a type element with the named stereotype (or one of its specializations). For attributes, open the Features window for the attribute and select a type element with the named stereotype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

| Property Stereotypes | Type Element Stereotypes |
|---|---|
| ConceptRole | ConceptItem |

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

# Model Views in MDG Technology for UPDM

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.



- UML Profile for DoDAF/MODAF Views
  - AV Diagrams
    - AV-1 Diagrams
      - AV-1 Enterprise Definition
      - Overview and Summary
    - AV-2 Diagrams
    - AV-3 Diagrams
  - AcV Diagrams
  - OV Diagrams
    - OV-1 Diagrams
      - High Level Operational Concept
      - Operational Context Graphic
    - OV-2 Diagrams
    - OV-3 Diagrams
    - OV-4 Diagrams
    - OV-5 Diagrams
    - OV-6a Diagrams
    - OV-6b Diagrams
    - OV-6c Diagrams
    - OV-7 Diagrams
  - SOV Diagrams
  - SV Diagrams
  - StV Diagrams
  - TV Diagrams
- Recent Discussions

To open the Model Views window, select 'Start > Explore > Browse > Model Views'. Double-click on the required diagram to open it.

# Glossary

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

## Import Glossary

You import the Glossary definitions into each model individually. To do this, select the 'Specialize > Technologies > UPDM > Import UPDM Glossary' ribbon option.

## View the Glossary

To view the Glossary, select one of:

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

# Using Enterprise Architect Elements

## Creating an instance from a Class

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

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

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

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

## Set the run state of an object

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

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

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

## Properties

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

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

1. Double-click on the element to display the 'Properties' dialog.
2. Select the 'Roles' tab.

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

Stereotypes that extend Property are:

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

# UPDM Example Model

The UPDM Example Model shows the common civilian maritime search and rescue operation for a yacht in distress. It is based on the Search and Rescue (SAR) example given in the UPDM specification document, which illustrates how UPDM can support DoDAF and MODAF, providing a model defining a sample of DoDAF and MODAF views addressing a typical problem space.

## Access

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

## Framework Diagram

When you open the example model, its default diagram displays. This is a UPDM Framework Diagram. which 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 can access a View by double-clicking on the corresponding element.

## Notes

- Most of the Views are Package elements; double-clicking on one opens the Package's child diagram
- The AV-1 Overview and Summary Information element is a Document Artifact; double-clicking on it opens the document in the Enterprise Architect document editor
- The StV-5 and SOV-3 elements are hyperlinks to Relationship Matrix profiles; double-clicking them opens the Relationship Matrix
- The OV-3 element is a hyperlink to a custom SQL query; double-clicking it executes the query and presents the results in a tabular form

# The Zachman Framework

The Zachman Framework is a widely used approach for engineering Enterprise Architecture. The Framework is a simple, logical structure that helps in organizing the information infrastructure of the Enterprise and provides many benefits in helping align technology with business needs.

## Discussion

The topics described here provide an introduction to, and procedural explanation of, using the MDG Technology for the Zachman Framework in Enterprise Architect.

| Section | Content |
|---|---|
| Welcome | This section provides an introduction to the Zachman Framework, and contains the formal documentation defining its use with Enterprise Architect. |
| Using the Zachman Framework | Get started with the Zachman Framework, learning about the model structure, templates, diagram types and more. |
| Model Validation | Learn how to develop and configure model validation for the Zachman Framework. |

# Welcome

Welcome to the Zachman Framework in Enterprise Architect.

Using this technology with Enterprise Architect, you can employ the Zachman Framework with the associated benefits of a powerful, open-standard modeling system. The Zachman Framework is already integrated with the Ultimate and Unified editions; it can be purchased separately to be used with the Enterprise Architect Professional or Corporate Editions.

## About the Zachman Framework

The Zachman Framework is a widely used approach for engineering Enterprise Architecture. The Framework is a simple, logical structure that helps in organizing the information infrastructure of the Enterprise.

While conceptually simple, the Zachman Framework provides many benefits in helping align technology with business needs. It has become a popular approach in defining Enterprise Architecture because it:

- Is platform neutral
- Is a powerful planning device
- Is both comprehensive and readily understood by non-technical people
- Assists in problem solving
- Helps in documenting enterprise-wide information system architecture

Under the Zachman Framework, an Enterprise is modeled by answering six questions: What? How? Where? Who? When? and Why? with respect to six role perspectives: the Planner, Owner, Designer, Builder, Subcontractor and Functioning Enterprise.

For further information, visit the Zachman Framework website.

## Getting Started

For instructions on how to use the Zachman Framework, see the topics:

- *Getting Started with the Zachman Framework* and
- *Using the Zachman Framework*

# Zachman Framework Copyright Notice

## Copyright © 2007-2019 Sparx Systems Pty. Ltd. All rights reserved.

Sparx Systems Pty. Ltd.

99 Albert St,

Creswick, Victoria 3363,

AUSTRALIA

Phone: +61 (3) 5345 1140

Fax: +61 (3) 5345 1104

Support Email: support@sparxsystems.com

Sales Email: sales@sparxsystems.com

Website: sparxsystems.com

# MDG Technology for Zachman Framework Software Product License Agreement

This Software Product License Agreement relates to the separately-purchased MDG Technology for Zachman Framework for use with the Professional and Corporate editions of Sparx Systems Enterprise Architect. The MDG Technology integrated with the Unified and Ultimate editions of Enterprise Architect is subject to the Sparx Systems Enterprise Architect Modelling Tool.

MDG Technology for Zachman Framework - Enterprise Architect MDG Add-In, Version 1.1

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 the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

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

## DEFINITIONS

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

- "EULA" means this End User 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.
- "Registered Edition of MDG Technology for Zachman Framework" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: https://sparxsystems.com/products/mdg/tech/zachman/purchase.html
- "SOFTWARE PRODUCT" or "SOFTWARE" means MDG Technology for Zachman Framework, 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 Enterprise Architect, 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.

## 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.
- To make copies of the SOFTWARE PRODUCT for backup, archival and instructional purposes.

## EVALUATION LICENSE

The Trial Version of MDG Technology for Zachman Framework 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 thirty (30) days.

Upon expiration of the thirty (30) days, the SOFTWARE PRODUCT must be removed from the computer. Unregistered use of MDG Technology for Zachman Framework 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 https://sparxsystems.com/products/mdg/tech/zachman/purchase.html). Upon payment of the license fee, YOU will be sent details on where to download the registered edition of MDG Technology for Zachman Framework 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 XP, Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7 and Windows 8 are trademarks of Microsoft.

The Zachman Framework for Enterprise Architecture TM is a trademark of John A. Zachman and Zachman International.

## GOVERNING LAW

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

# Acknowledgement of Trademarks

Sparx Systems acknowledge these trademarks, which are used throughout the MDG for Zachman Framework documentation.

## Trademarks of Microsoft

- Microsoft Word
- Microsoft Office
- Windows®

## Trademarks of the Object Management Group

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

## Trademark of John A. Zachman and Zachman International

- The Zachman Framework For Enterprise Architecture ™

# Support for the Zachman Framework

Technical support for the Zachman Framework is available to registered users of Enterprise Architect through the same channels as for Enterprise Architect itself.

# Zachman Framework System Requirements

Zachman Framework version 1.1.4 runs under the environments identified here.

## Microsoft® Operating Systems Supported

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

## Enterprise Architect Versions Supported

- Enterprise Architect Version 7.1 or later

## Notes

- 32 bit and 64 bit operating systems supported

# Getting Started with the Zachman Framework

When you install the Unified or Ultimate editions of Enterprise Architect, the Zachman Framework is fully enabled and ready to use.

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

## Access the MDG Technology For Zachman Framework

1. Create a new Enterprise Architect project file, and click on the top-level Package.
2. Select the 'Start > View > Perspective > Open Model Wizard' option.
3. In the Model Wizard, select the 'Enterprise Architecture > Zachman' Perspective and the 'Zachman Framework' Pattern.
4. Click on the Create Patterns button.

A new base Zachman model is created in the Project Browser, containing the Zachman Framework diagram and the Planner, Owner, Designer, Builder, Subcontractor and Functioning Enterprise Packages.

# Using the Zachman Framework

The Zachman Framework provides a model-based framework for planning, designing and implementing the Architecture for an Enterprise. The starter model provided with the Technology acts as a base upon which you can build the Enterprise Architecture. You can create the appropriate diagrams from the extended Enterprise Architect UML diagram set, using Toolbox pages that support every cell of the Zachman classification framework.

The Technology also provides model validation and reporting capabilities for strategic project plans.

Within Enterprise Architect you can choose between Diagram View and Element List View. Element List View can be used in cells where you prefer to define only the model artifacts.

You can also align cells across the framework (horizontally and vertically) through the Enterprise Architect Relationship Matrix.

You can view a demonstration video of the MDG Technology For Zachman Framework in use, on the Sparx Systems website.

The Zachman Framework Help topics provide a detailed exploration of the Zachman Framework tools and features, such as.

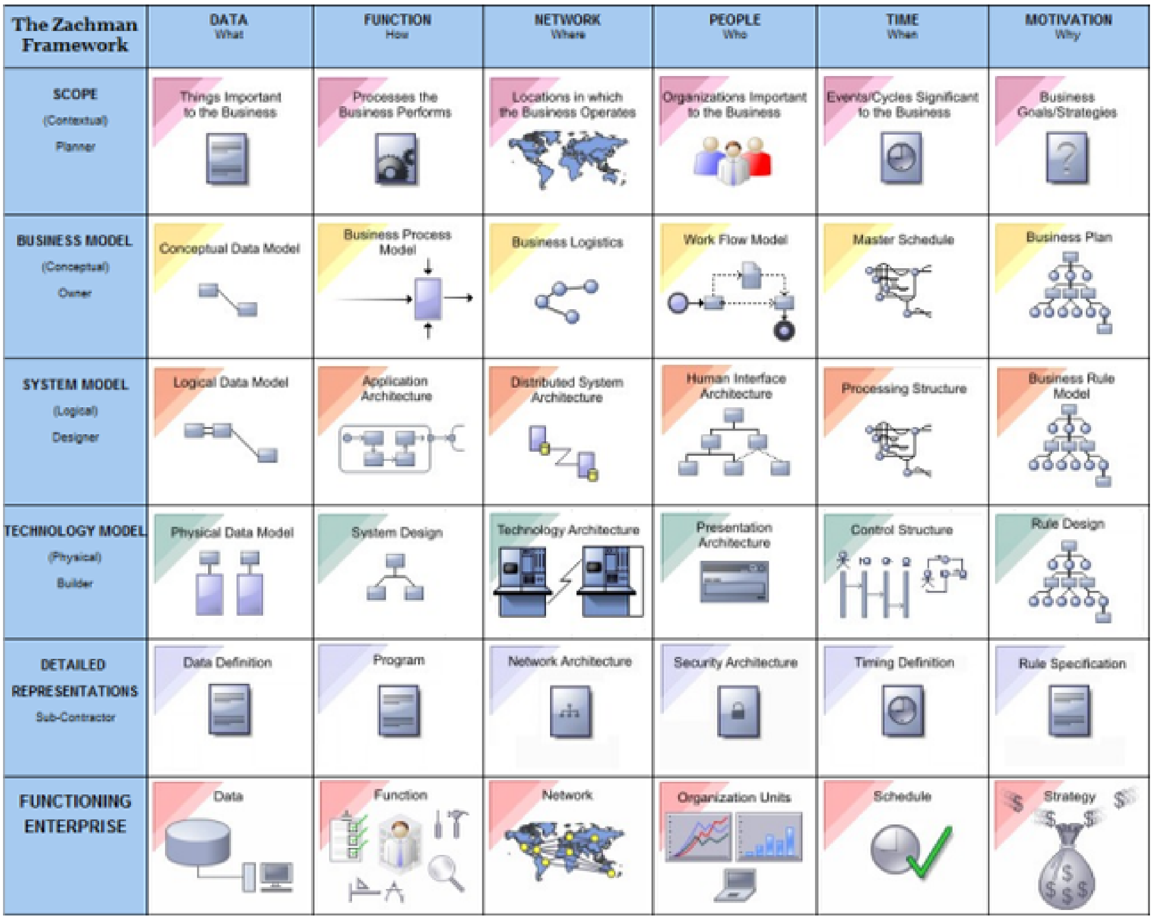
- The example Enterprise Architect model for the Zachman Framework
- UML profiles (Toolbox pages) for use within specific Zachman Framework cells
- A diagram interface for the Zachman Framework
- New diagram types specific to the Zachman Framework
- A flexible model starter-structure
- Report generation capabilities for strategic project plans

The MDG Technology For Zachman Framework is integrated with the features of Enterprise Architect.

# The Zachman Framework Interface Diagram

The Zachman Framework is a predefined model in Enterprise Architect. The model-level diagram of the model structure is the Zachman Framework Interface diagram, which serves as a template for the development of Enterprise Architecture based on the Zachman classification framework.

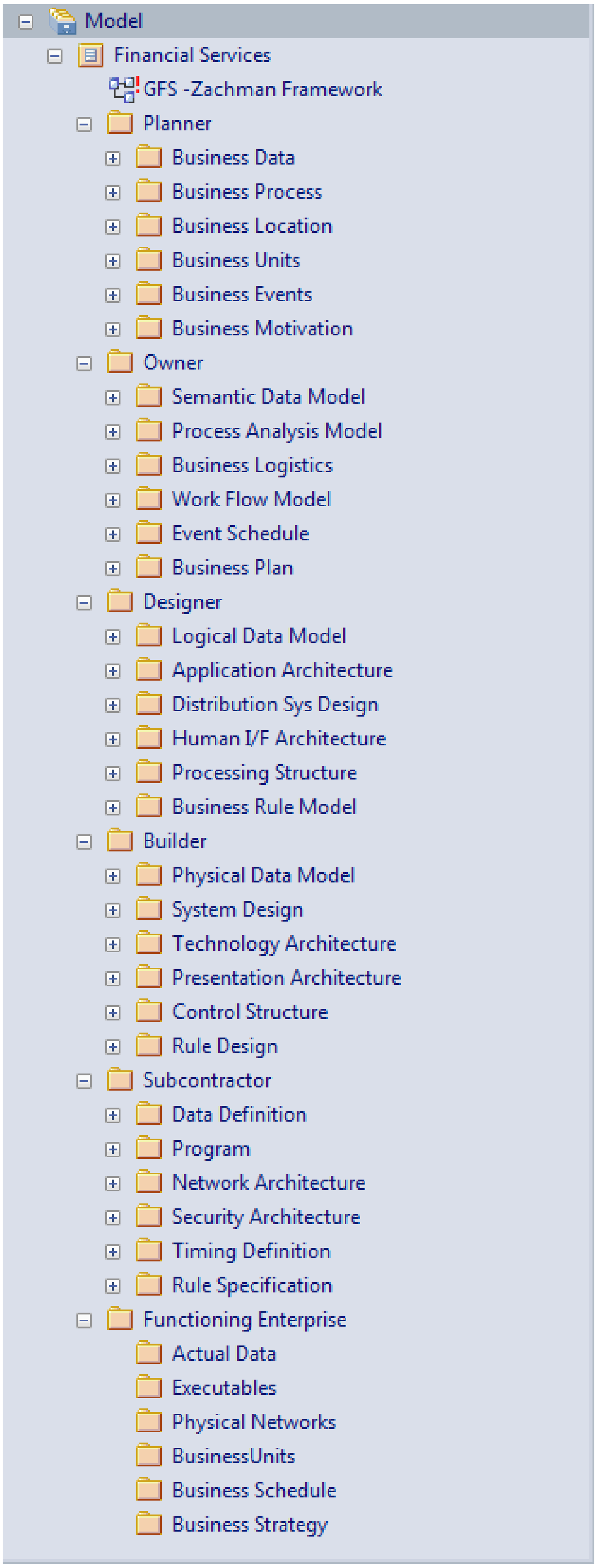
Each cell links to the relevant Zachman Framework diagram in the child Packages in the base model.



| The Zachman Framework | DATA<br>What | FUNCTION<br>How | NETWORK<br>Where | PEOPLE<br>Who | TIME<br>When | MOTIVATION<br>Why |
|---|---|---|---|---|---|---|
| SCOPE<br>(Contextual)<br>Planner | Things Important to the Business | Processes the Business Performs | Locations in which the Business Operates | Organizations Important to the Business | Events/Cycles Significant to the Business | Business Goals/Strategies |
| BUSINESS MODEL<br>(Conceptual)<br>Owner | Conceptual Data Model | Business Process Model | Business Logistics | Work Flow Model | Master Schedule | Business Plan |
| SYSTEM MODEL<br>(Logical)<br>Designer | Logical Data Model | Application Architecture | Distributed System Architecture | Human Interface Architecture | Processing Structure | Business Rule Model |
| TECHNOLOGY MODEL<br>(Physical)<br>Builder | Physical Data Model | System Design | Technology Architecture | Presentation Architecture | Control Structure | Rule Design |
| DETAILED REPRESENTATIONS<br>Sub-Contractor | Data Definition | Program | Network Architecture | Security Architecture | Timing Definition | Rule Specification |
| FUNCTIONING ENTERPRISE | Data | Function | Network | Organization Units | Schedule | Strategy |

# Zachman Framework Model Structure

The Zachman Framework provides a Framework model template, in which each Zachman Perspective (or row) is modeled as the highest-level Package inside the model. Cells belonging to the Perspectives are modeled as child Packages of the appropriate row Package.



- Model
  - Financial Services
    - GFS -Zachman Framework
    - Planner
      - Business Data
      - Business Process
      - Business Location
      - Business Units
      - Business Events
      - Business Motivation
    - Owner
      - Semantic Data Model
      - Process Analysis Model
      - Business Logistics
      - Work Flow Model
      - Event Schedule
      - Business Plan
    - Designer
      - Logical Data Model
      - Application Architecture
      - Distribution Sys Design
      - Human I/F Architecture
      - Processing Structure
      - Business Rule Model
    - Builder
      - Physical Data Model
      - System Design
      - Technology Architecture
      - Presentation Architecture
      - Control Structure
      - Rule Design
    - Subcontractor
      - Data Definition
      - Program
      - Network Architecture
      - Security Architecture
      - Timing Definition
      - Rule Specification
    - Functioning Enterprise
      - Actual Data
      - Executables
      - Physical Networks
      - BusinessUnits
      - Business Schedule
      - Business Strategy

# The Zachman Framework Model Template

The Zachman Framework Model Template provides the model skeleton from which you can develop your Enterprise definition.

## Add a new Zachman Framework model to the project

1. Right-click on the root node and select 'Add a Model using Wizard'. The 'Model Wizard' view displays.
2. On the 'Model Patterns' tab, click on the <name> Perspective button and select 'Enterprise Architecture > Zachman' from the list.
3. Select the 'Zachman Framework' pattern.
4. Click on the Create Patterns button.

# Zachman Framework Diagrams

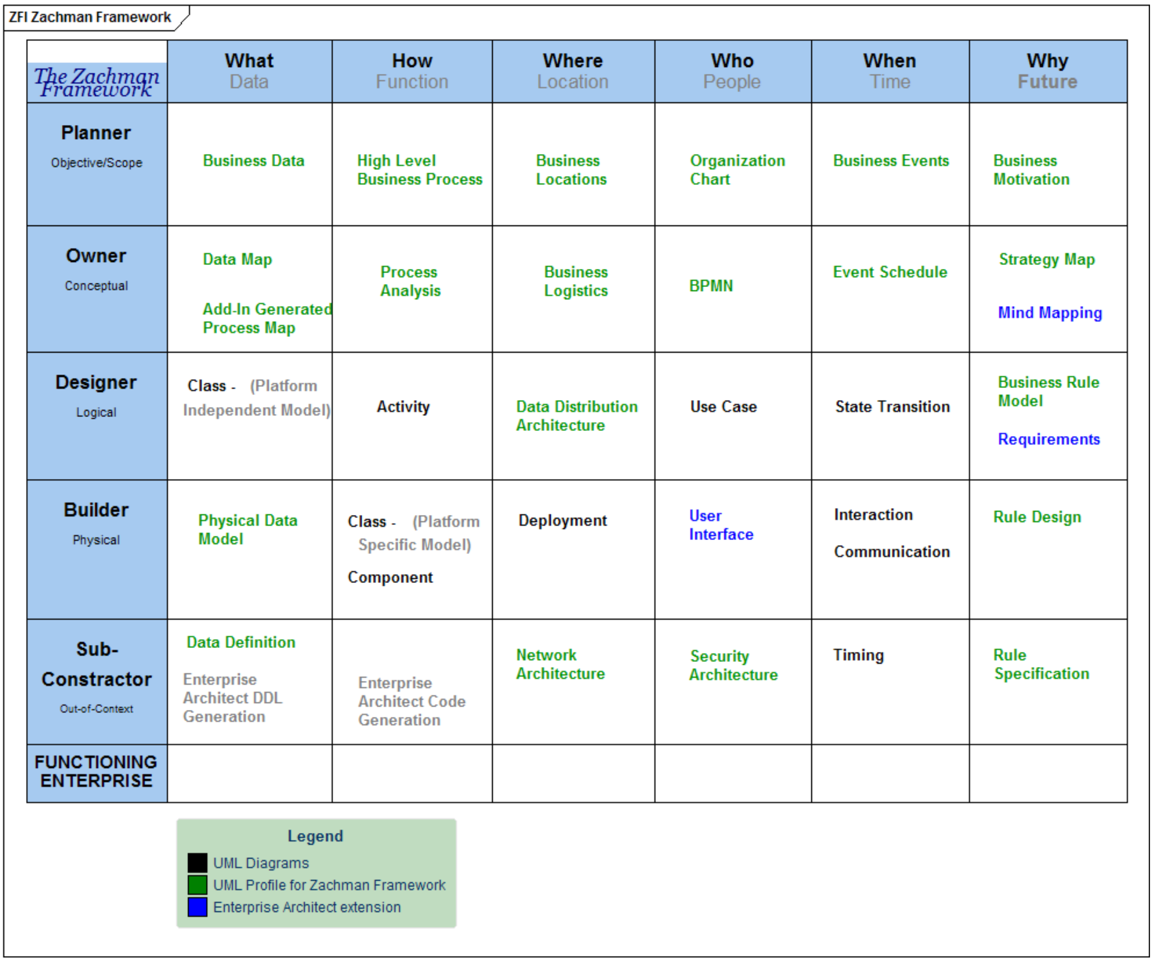
The Zachman Framework introduces new diagram types that support modeling of the Zachman Classification Framework. A Zachman Framework diagram is created in the same way as any other diagram in Enterprise Architect.

The Technology provides access to these categories of diagram through the 'New Diagram' dialog:

- Planner
- Owner
- Designer
- Builder
- Subcontractor
- Zachman Framework Interface

# Zachman Framework Diagram Types

The Zachman Framework further extends the Enterprise Architect diagram set to support the Framework, with diagram types appropriate to each cell of the Zachman Framework.



ZFI Zachman Framework

| The Zachman Framework | What<br>Data | How<br>Function | Where<br>Location | Who<br>People | When<br>Time | Why<br>Future |
|---|---|---|---|---|---|---|
| Planner<br>Objective/Scope | Business Data | High Level Business Process | Business Locations | Organization Chart | Business Events | Business Motivation |
| Owner<br>Conceptual | Data Map<br>Add-In Generated Process Map | Process Analysis | Business Logistics | BPMN | Event Schedule | Strategy Map<br>Mind Mapping |
| Designer<br>Logical | Class - (Platform Independent Model) | Activity | Data Distribution Architecture | Use Case | State Transition | Business Rule Model<br>Requirements |
| Builder<br>Physical | Physical Data Model | Class - (Platform Specific Model)<br>Component | Deployment | User Interface | Interaction<br>Communication | Rule Design |
| Sub-Constractor<br>Out-of-Context | Data Definition<br>Enterprise Architect DDL Generation | Enterprise Architect Code Generation | Network Architecture | Security Architecture | Timing | Rule Specification |
| FUNCTIONING ENTERPRISE | | | | | | |

Legend

- ⬛ UML Diagrams
- 🟩 UML Profile for Zachman Framework
- 🟦 Enterprise Architect extension

# The Zachman Framework Toolbox

The Zachman Framework pages of the Diagram Toolbox provide elements and relationships for all the Zachman Framework diagrams that the MDG Technology supports. The Zachman Framework Toolbox pages can be accessed by clicking on 🔍 and specifying 'Zachman' in the 'Find Toolbox Item' dialog. The Diagram Toolbox can be docked on either side of the diagram, or free floated on top of the diagram to expose more surface for editing.

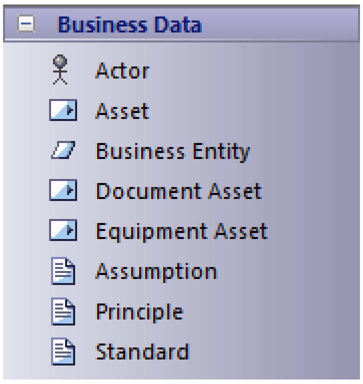
## Diagrams for Toolboxes

This table shows, for each Zachman Framework cell, the diagram that could be used.

| Zachman Cell | Diagram |
|---|---|
| Planner - Data | Business Data |
| Planner - Function | Business Process |
| Planner - Location | Business Locations |
| Planner - People | Organization Chart |
| Planner - Timing | Business Events |
| Planner - Motivation | Business Motivation |
| Owner - Data | Data Map and<br>Process Map (Generated by Add-In) |
| Owner - Function | Process Analysis |
| Owner - Location | Business Logistics |
| Owner - People | BPMN |
| Owner - Timing | Event Schedule |
| Owner - Motivation | Enterprise Architect Mind Mapping diagram and<br>Strategy Map |
| Designer - Data | Class |
| Designer – Function | Activity |
| Designer - Location | Data Distribution Architecture |
| Designer - People | Use Case |
| Designer - Timing | State Transition |
| | |

| | |
|---|---|
| Designer - Motivation | Business Rule Model |
| Builder - Data | Physical Data Model |
| Builder - Function | Class and<br>Component |
| Builder - Location | Deployment |
| Builder - People | User Interface |
| Builder - Timing | Communication and<br>Interaction |
| Builder - Motivation | Rule Design |
| Subcontractor - Data | Data Definition; default toolbox for the diagram is Custom. |
| Subcontractor – Function | No diagram defined – Code generation is done in this cell. |
| Subcontractor - Location | Network Architecture |
| Subcontractor - People | Security Architecture |
| Subcontractor - Timing | Timing |
| Subcontractor - Motivation | Rule Specification |

# Business Data Page



Business Data

Actor

Asset

Business Entity

Document Asset

Equipment Asset

Assumption

Principle

Standard

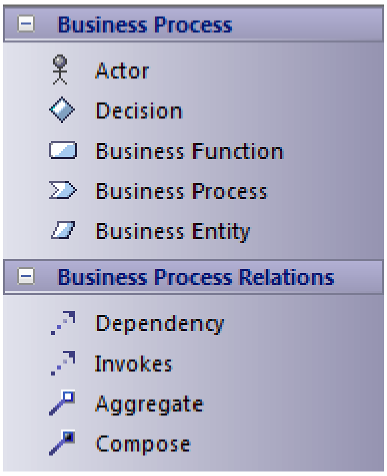
## Business Data Toolbox

| Item | Description |
|---|---|
| Actor | Models a stakeholder or any other human resource of the enterprise. |
| Asset | Represents the enterprise resources that could be estimated for value. |
| Business Entity | Represents generic enterprise resources. |
| Document Asset | A subtype of Asset that captures the important documents of the enterprise. |
| Equipment Asset | A subtype of Asset that captures the equipment resources of the enterprise. |
| Assumption | Captures the assumptions made in information manipulation.<br>Applies the Tagged Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Principle | Defines the Principles framed and followed in the enterprise.<br>Applies the Tagged Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Standard | Defines the standards followed in the Enterprise.<br>Applies the Tagged Value Type = Enterprise / Business / System / Application / Technology / Data. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Business Process Pages



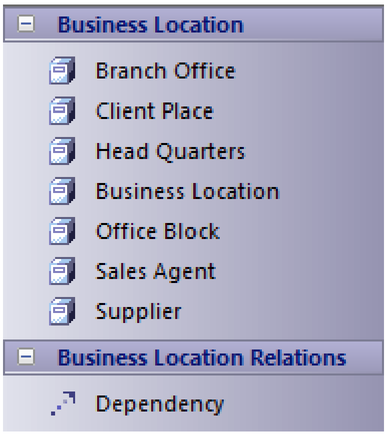
## Business Process Toolbox

| Item | Description |
|---|---|
| Actor | Models a stakeholder or any other human resource of the Enterprise. |
| Decision | Indicates the point of conditional progression where a business decision is taken. |
| Business Function | Represents a major function performed by the enterprise or a part of the enterprise. |
| Business Process | Represents a function or behavior of the enterprise or part of the enterprise. |
| Business Entity | Represents generic enterprise resources. |
| Invokes | A relationship that defines the invocation of a business process. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Business Location Page



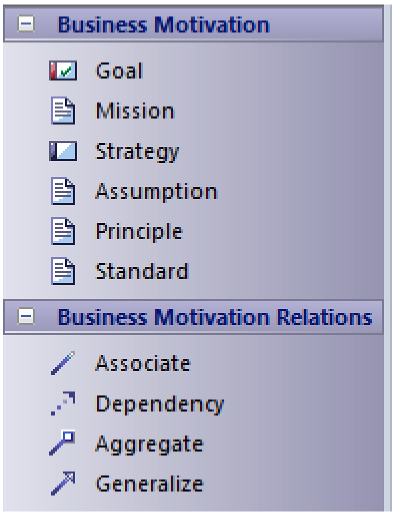
## Business Location Toolbox

| Item | Description |
|---|---|
| Branch Office | Models a Business Location as a Branch Office. |
| Client Place | Models a Business Location as a Client Place. |
| Head Quarters | Models a Business Location as a Head Quarters. |
| Business Location | Models the location from which the business operates. |
| Office Block | Models a Business Location as an Office Block. |
| Sales Agent | Models a Business Location as a Sales Agent. |
| Supplier | Models a Business Location as a Supplier. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Business Motivation Pages



Business Motivation

Goal

Mission

Strategy

Assumption

Principle

Standard

Business Motivation Relations

Associate

Dependency

Aggregate

Generalize

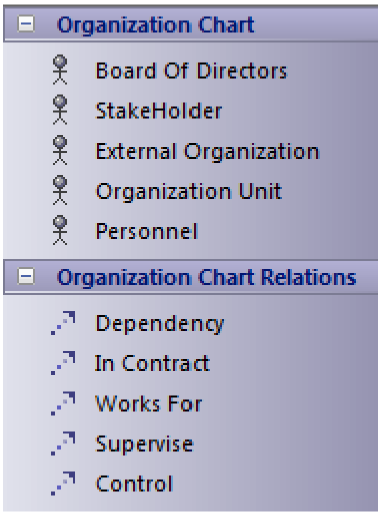
## Business Motivation Toolbox

| Item | Description |
|---|---|
| Goal | Models what is to be achieved by the enterprise, with specifications defined by the Tagged Values. |
| Mission | Models the mission statement, policies and values of the enterprise. |
| Strategy | Models the strategy statements for the business plan. |
| Assumption | Models the assumptions made in information manipulation.<br>Tagged Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Principle | Defines the Principles framed and followed in the enterprise.<br>Tagged Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Standard | Defines the standards followed in the enterprise.<br>Tagged Value Type = Enterprise / Business / System / Application / Technology / Data. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Organization Chart Pages



Organization Chart

Board Of Directors

StakeHolder

External Organization

Organization Unit

Personnel

Organization Chart Relations

Dependency

In Contract

Works For

Supervise

Control

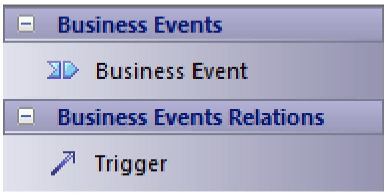
## Organization Chart Toolbox

| Item | Description |
|---|---|
| Board of Directors | Captures the details of the board of directors. |
| StakeHolder | Defines a stakeholder of the enterprise. |
| External Organization | Defines any external business unit that is not under direct control of the enterprise, but has a relationship with the enterprise. |
| Organization Unit | Defines any business unit that is under direct control of the enterprise. |
| Personnel | Captures the details of personnel in an enterprise. |
| In Contract | A connector that represents the contract-based relationships between business units. |
| Works For | A connector that captures the details of team links; for example, Stakeholder 1 works for Organization Unit 1. |
| Supervise | A connector that captures process supervision details. |
| Control | A connector that captures Unit in charge or Person in charge information. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Business Events Pages


Business Events

Business Event

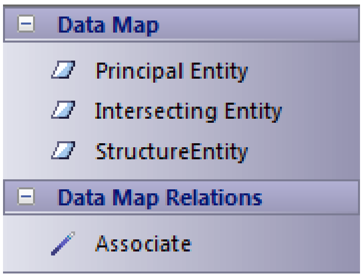
Business Events Relations

Trigger

## Business Event Toolbox

| Item | Description |
|---|---|
| Business Event | Captures major business events of the enterprise. |
| Trigger | Indicates that a Business Event triggers another event or a business process. |

# Data Map Pages


Data Map
Principal Entity
Intersecting Entity
StructureEntity
Data Map Relations
Associate

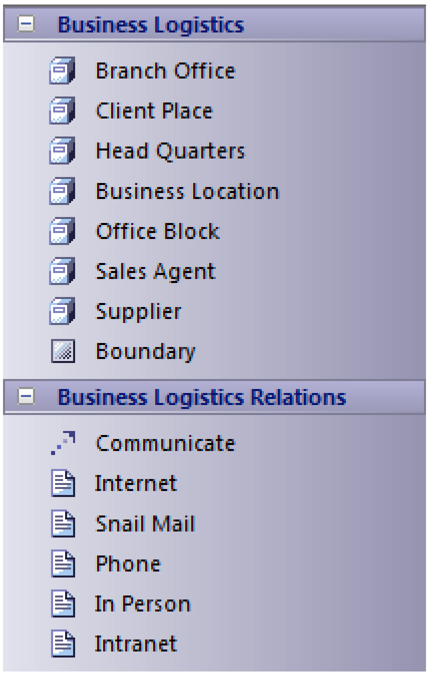
## Data Map Toolbox

| Item | Description |
|---|---|
| Principal Entity | Represents a business entity that forms a resource of the enterprise. |
| Intersecting Entity | Normalizes the many-to-many relationship between principal entities. |
| Structure Entity | Captures potential knowledge-based entities. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Business Logistics Pages



## Business Logistics Items and Relations

| Item | Description |
|---|---|
| Branch Office | Models a Business Location as a Branch Office. |
| Client Place | Models a Business Location as a Client location |
| Head Quarters | Models a Business Location as Head Quarters. |
| Business Location | Models the location from which the business operates. |
| Office Block | Models a Business Location as an Office Block. |
| Sales Agent | Models a Business Location as a Sales Agent. |
| Supplier | Models a Business Location as a Supplier. |
| Communicate | Indicates that a business location communicates directly with another business location. |
| Internet | Indicates that the means of communication is the World Wide Web. |
| Snail Mail | Indicates that the means of communication is the postal system or courier services. |
| Phone | Indicates that the means of communication is the telephone. |
| | |

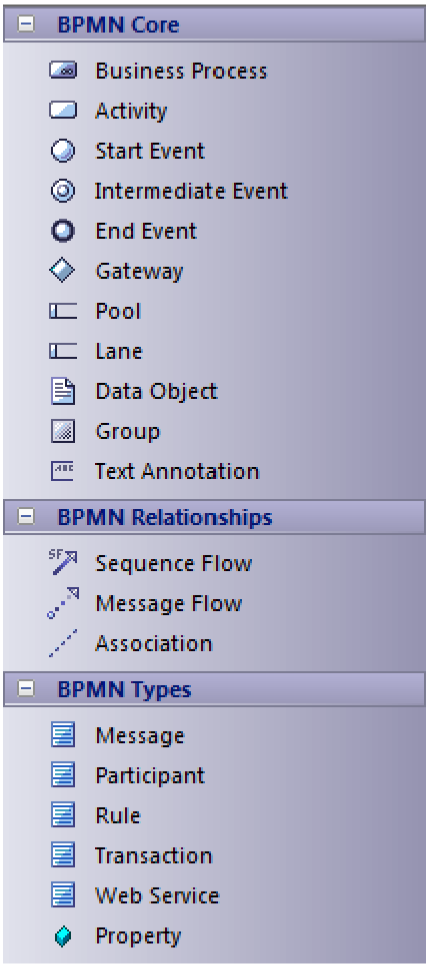
| | |
|---|---|
| In Person | Indicates that the means of communication is direct person-to-person. |
| Intranet | Indicates that the means of communication is the local intranet or WAN. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# BPMN Pages

The BPMN Toolbox pages provide the graphical (Core) and non-graphical (Types) Business Process Model and Notation (BPMN) elements for use on Business Process diagrams through the Zachman Framework Technology. Specifications of these elements and relationships are defined by Tagged Values.



BPMN Core
- Business Process
- Activity
- Start Event
- Intermediate Event
- End Event
- Gateway
- Pool
- Lane
- Data Object
- Group
- Text Annotation

BPMN Relationships
- Sequence Flow
- Message Flow
- Association

BPMN Types
- Message
- Participant
- Rule
- Transaction
- Web Service
- Property

## BPMN Toolbox

| Item | Description |
|---|---|
| Business Process | Defines a business process; an extension of a composite Activity. |
| Activity | Defines an activity within a business process. |
| Start Event | Defines the initiating event in a process. |
| Intermediate Event | Defines an intermediate event in a process. |
| End Event | Defines the terminating event in a process. |
| Gateway | Defines a decision point in a business process. If a condition is true, then processing continues one way; if not, then another. |

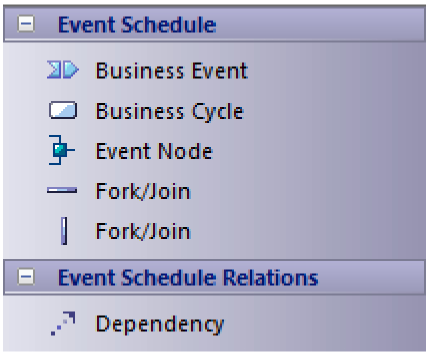
| | |
|---|---|
| Pool | Logically organizes an Activity; an extension of a Partition element. |
| Lane | Subdivides a Pool; an extension of a Partition element. |
| Data Object | Defines a physical piece of information used or produced by a system; an extension of an Artifact element. |
| Group | Groups a number of other elements; an extension of a Boundary element. |
| Text Annotation | A comment. |
| Sequence Flow | Defines the flow of an activity; an extension of a Control Flow relationship. |
| Message Flow | Defines the flow of communications in a process; an extension of a Control Flow relationship. |
| Association | Associates information and artifacts with flow objects. |
| Message | Defines a message; an extension of a Class element. |
| Participant | Defines a participant in an activity; an extension of a Class element. |
| Rule | Defines business rule statements; an extension of a Class element. |
| Transaction | Defines a transaction in an activity; an extension of a Class element. |
| Web Service | Defines a web service; an extension of a Class element. |
| Property | Assigns a property to an element; an extension of an attribute. |

## Notes

- Enterprise Architect is delivered with the BPMN Technologies (for BPMN 1.0. 1.1 and 2.0) automatically installed, providing BPMN profiles and Toolboxes separate from this Zachman version; to make even further use of BPMN facilities, download the BPMN Add-In from:

  https://sparxsystems.com/products/mdg_bpmn.html

# Event Schedule Pages



Event Schedule

Business Event

Business Cycle

Event Node

Fork/Join

Fork/Join

Event Schedule Relations

Dependency

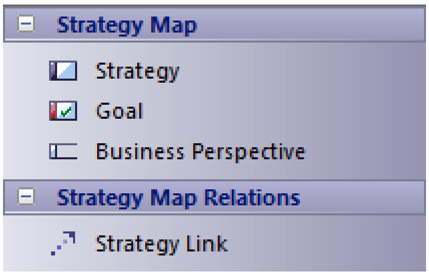
## Event Schedule Toolbox

| Item | Description |
|---|---|
| Business Event | Captures major business events of the enterprise. |
| Business Cycle | Captures major business cycles of the enterprise. |
| Event Node | Captures the event points in a business cycle. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Strategy Map Pages



Strategy Map

Strategy

Goal

Business Perspective

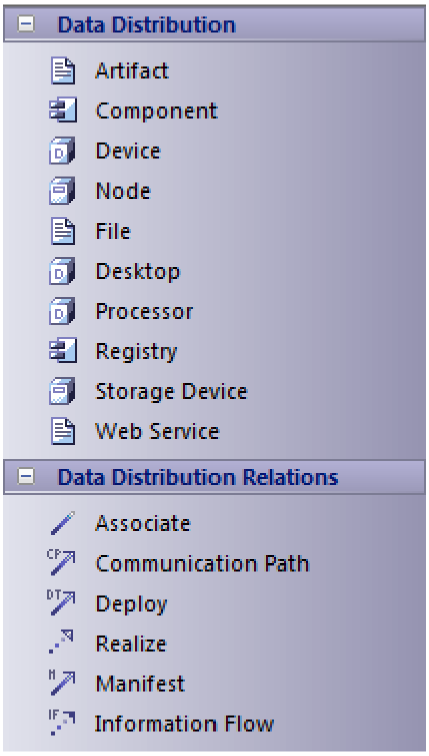
Strategy Map Relations

Strategy Link

## Strategy Map Toolbox

| Item | Description |
|---|---|
| Strategy | Captures the strategy statements for the business plan. |
| Goal | Captures what is to be achieved by the enterprise, with specifications defined by the Tagged Values. |
| Business Perspective | Relates the strategies to a specific category. |
| Strategy Link | Indicates that a strategy is linked to another strategy or goal. |

# Data Distribution Architecture Pages



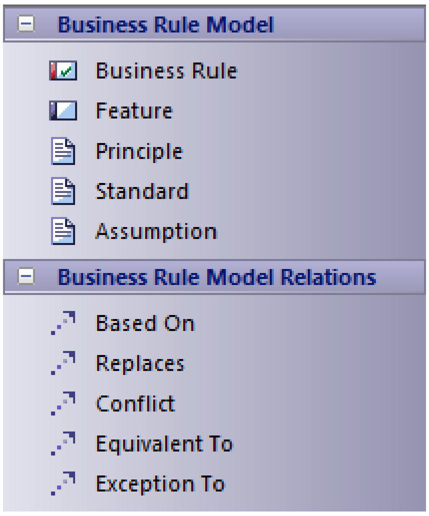
## Data Distribution Architecture Toolbox

| Item | Description |
|---|---|
| File | Represents a file. |
| Desktop | Represents a desktop. |
| Processor | Represents a processor. |
| Registry | Represents a registry. |
| Storage Device | Represents a storage device. |
| Web Service | Represents a web service. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Business Rule Model Pages



Business Rule Model
- Business Rule
- Feature
- Principle
- Standard
- Assumption

Business Rule Model Relations
- Based On
- Replaces
- Conflict
- Equivalent To
- Exception To

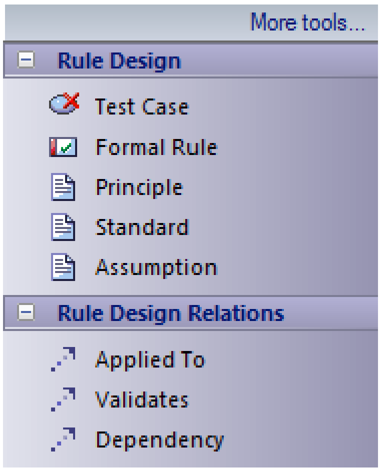
## Business Rule Model Toolbox

| Item | Description |
|---|---|
| Business Rule | Captures the Business Rule statements. |
| Principle | Defines the Principles framed and followed in the Enterprise.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Standard | Defines the standards followed in the Enterprise.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Assumption | Captures the assumptions made in information manipulation.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Based On | Indicates that a rule is based on another model element, which forms the rationale for the rule. |
| Replaces | Indicates that a new rule replaces another rule. |
| Conflict | Indicates that a rule conflicts with another defined rule. |
| Equivalent To | Indicates that a rule is equivalent to another rule. |
| Exception To | Indicates exceptions for a rule. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Rule Design Pages



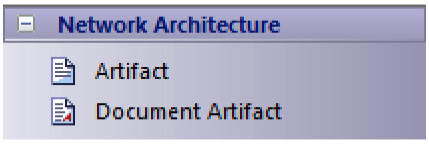
## Rule Design Toolbox

| Item | Description |
|---|---|
| Formal Rule | Represents a business rule transformed to a technology-specific logical rule or constraint statement. |
| Principle | Defines the Principles framed and followed in the Enterprise.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Standard | Used to define the Standards followed in the Enterprise.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Assumption | Used to capture the assumptions made in information manipulation.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Applied To | Indicates that a Formal Rule is applied to other model artifacts such as Scenarios or Activities. |
| Validates | Indicates that a model artifact validates a Formal Rule. |

## Notes

- Elements and connectors common to Enterprise Architect UML and Extended diagrams are documented in the Object Toolbox section

# Network Architecture Pages



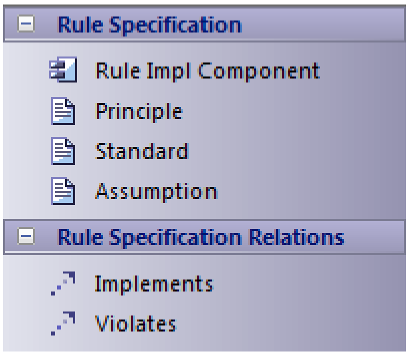
## Network Architecture Toolbox

| Item | Description |
|---|---|
| Artifact | Generic graphical element used to capture information. |
| Document Artifact | Generic graphical element used to capture detailed information such as network configuration details. |

## Notes

- For a full description of Artifact elements, see the Artifact topic

# Rule Specification Pages



Rule Specification

Rule Impl Component

Principle

Standard

Assumption

Rule Specification Relations

Implements

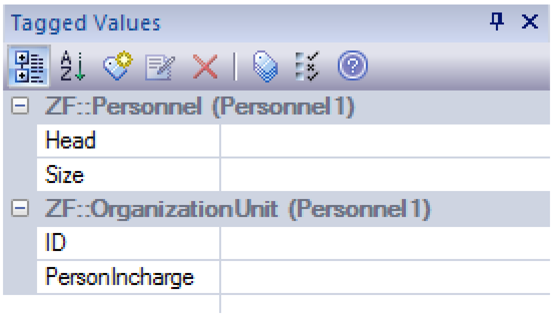
Violates

## Rule Specification Toolbox

| Item | Description |
|---|---|
| Rule Impl Component | Captures the component implementing a rule. |
| Principle | Defines the Principles framed and followed in the enterprise.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Standard | Defines the Standards followed in the enterprise.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Assumption | Captures the assumptions made in information manipulation.<br>Tag Value Type = Enterprise / Business / System / Application / Technology / Data. |
| Implements | Indicates that a Rule Impl Component implements a rule. |
| Violates | Indicates that the rule is violated by the connecting model element. |

# Tagged Values for Zachman Framework

The Zachman Framework makes extensive use of Tagged Values to assign custom properties to the various Zachman Framework elements. It is recommended that you keep the Tagged Values window docked and visible at all times when creating or viewing a Zachman Framework model.



ZF::Personnel (Personnel1)

| | |
|---|---|
| Head | |
| Size | |

ZF::OrganizationUnit (Personnel1)

| | |
|---|---|
| ID | |
| PersonIncharge | |

## Access

| | |
|---|---|
| Ribbon | Start > Explore > Properties > Tagged Values<br>Show > Portals > Window > Properties > Tagged Values |
| Keyboard Shortcuts | Ctrl+Shift+6 |

## Synchronize Tagged Values

From time to time you might need to add missing Tagged Values to all elements in the model that require them, such as:

- Whenever you create a new element by any means other than directly dropping the element from the Zachman Framework Toolbox pages
- Before using a new version of the Technology, to update the Tagged Values of elements in existing models to the latest version of the Zachman Framework profile

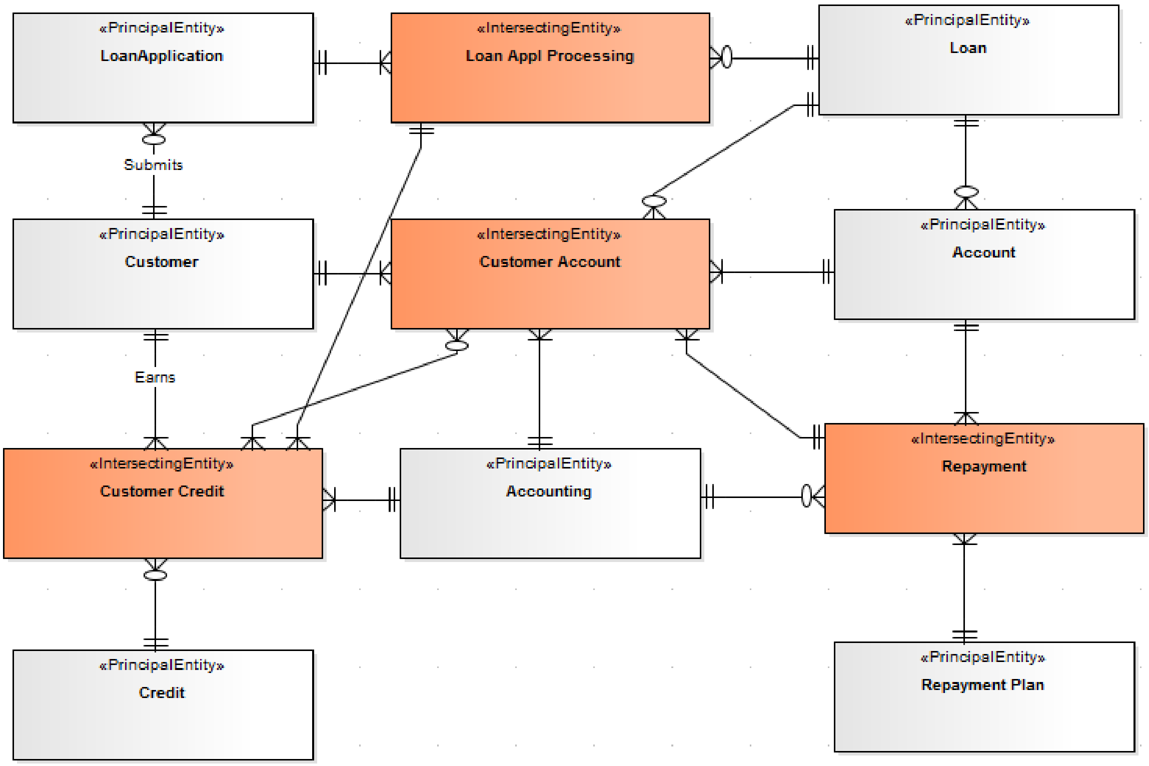
You can do this using the 'Synchronize Stereotype' option on the icons in the Zachman Framework pages of the Diagram Toolbox.

# Data Map Analysis

A valid Data Map diagram is basically an Entity Relationship diagram constructed using Principal Entity, Structure Entity and Intersecting Entity elements. The relationships between them are defined by the business rules.

- Principal Entities are identified from the Business Entities in scope
- Intersecting Entities are used to break a many-to-many association between Principal Entities, which form potential business processes
- Structure Entities represent the existence of a potential knowledge base

This is an example of a valid Data Map diagram:



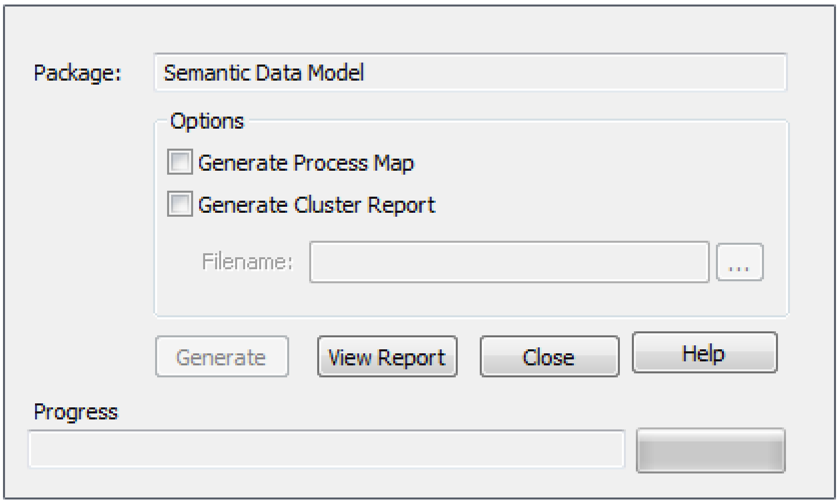
Cluster Reports and Process Maps are deliverables of a valid Data Map diagram analysis.

## Perform a Data Map diagram analysis

With the Data Map diagram to be analyzed open and active, either:

- Select the 'Specialize > Add-Ins > Zachman Framework > Do Data-Map Analysis' ribbon option, or
- Right-click on the Data Map diagram in the Project Browser, and select the 'Specialize | Zachman Framework | Do Data-Map Analysis' context menu option

The 'Data Map Analysis' dialog displays.



Package: Semantic Data Model

Options

□ Generate Process Map

□ Generate Cluster Report

Filename:

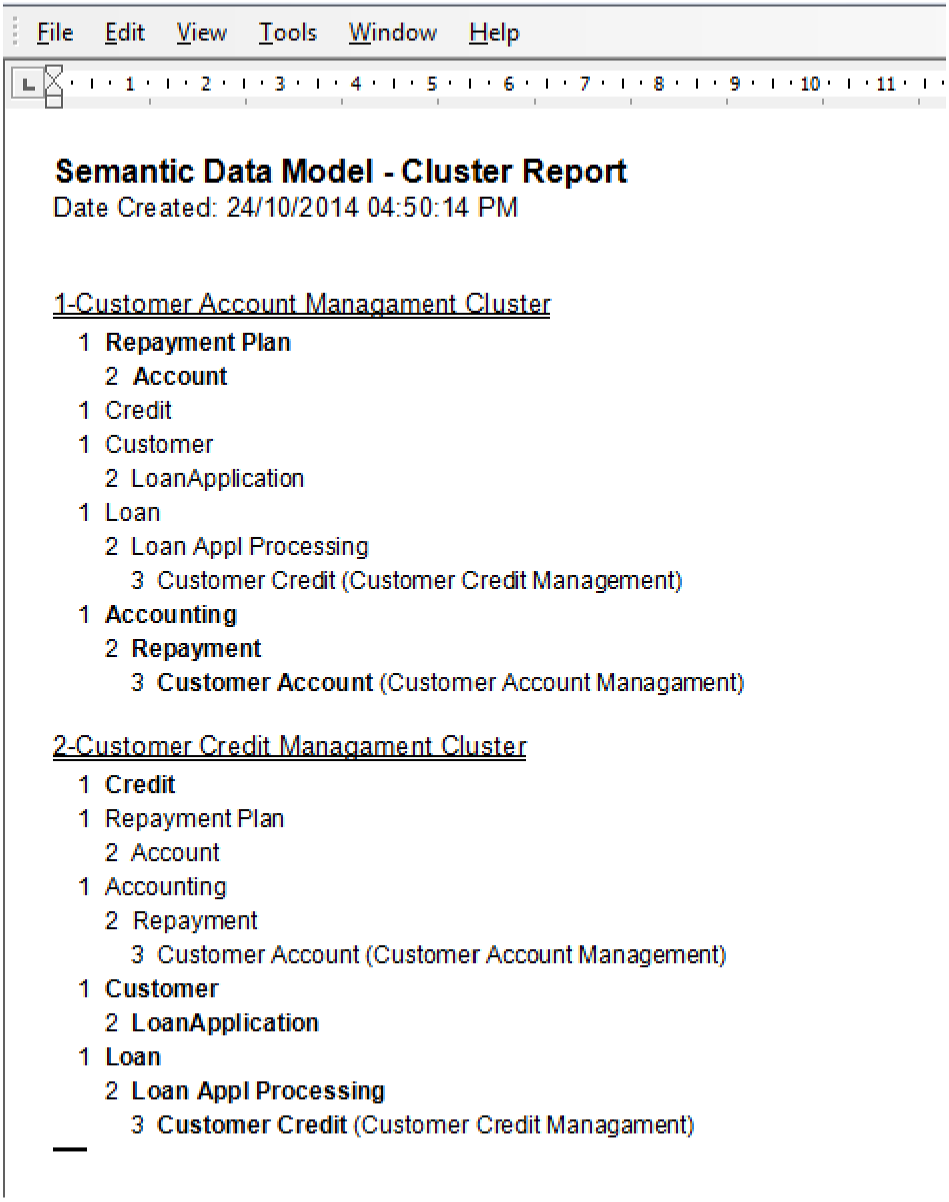
Click on the checkbox against each deliverable required. If you have selected 'Generate Cluster Report', also enter the file pathname under which to save the report.

Click on the Generate button.

# Cluster Report

A cluster is a logically related group of processes arranged in a sequence, this being the plan of the order of the execution of processes.

This Cluster Report was generated for the sample Data Map diagram, in .rtf format.



**Semantic Data Model - Cluster Report**
Date Created: 24/10/2014 04:50:14 PM

1-Customer Account Managament Cluster

- 1 **Repayment Plan**
  - 2 **Account**
- 1 Credit
- 1 Customer
  - 2 LoanApplication
- 1 Loan
  - 2 Loan Appl Processing
    - 3 Customer Credit (Customer Credit Management)
- 1 **Accounting**
  - 2 **Repayment**
    - 3 **Customer Account** (Customer Account Managament)

2-Customer Credit Managament Cluster

- 1 **Credit**
- 1 Repayment Plan
  - 2 Account
- 1 Accounting
  - 2 Repayment
    - 3 Customer Account (Customer Account Management)
- 1 **Customer**
  - 2 **LoanApplication**
- 1 **Loan**
  - 2 **Loan Appl Processing**
    - 3 **Customer Credit** (Customer Credit Managament)

The report shows how each cluster is a logical group of processes or tasks forming a major business process.

The number preceding each entity name is the phase number for the entity. Phase 1 against an entity means that the entity forms a potential resource/element that must be procured/framed before proceeding with the business process.

Entities with phase numbers greater than 1 are potential processes, with their sequence of execution set after procuring/framing the phase 1 entities in the cluster.

After successful completion of Data Map analysis, the phase property of each entity in the Data Map diagram is set accordingly.

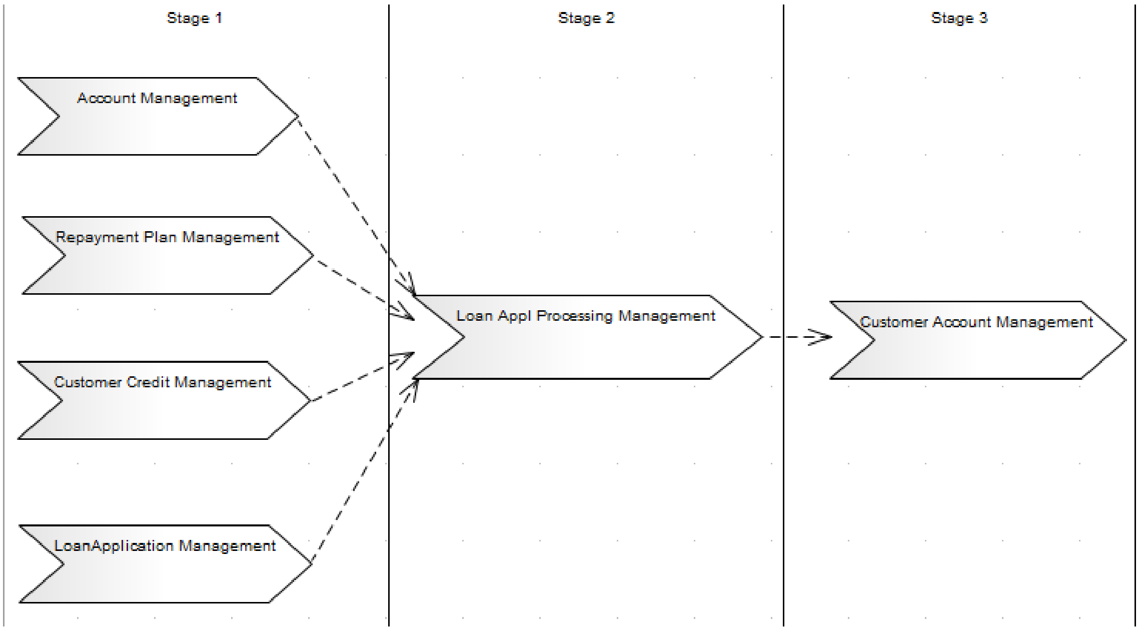
## Acknowledgement

The algorithm for Cluster Report generation is derived from the book *Enterprise Architecture for Integration: Rapid Delivery Methods and Technologies* (Clive Finkelstein; April 2006).

# Process Map

A Process Map is the visual model of the Cluster Report; however, the Phase 1 entities in the Cluster Report are not shown. The Process Map groups the identified Business Processes into the stages of the project, arranged as a guide for the project.

This is the Process Map generated for the sample Data Map diagram.


Stage 1
Stage 2
Stage 3
Account Management
Repayment Plan Management
Customer Credit Management
LoanApplication Management
Loan Appl Processing Management
Customer Account Management

# Business Scorecard Report Template

To aid your strategic management methods, the Zachman Framework provides a report template for creating Business Scorecards.

## Generate a Business Scorecard

| Step | Action |
|---|---|
| 1 | In the Project Browser, click on the Package containing your Business Perspectives and Strategies (an Owner \| Business Plan \| Strategic Plan Package).<br>The Business Perspectives must own the respective strategies. |
| 2 | Either:<br>• Press F8, or<br>• Select the 'Publish > Report > Documentation > Generate Rich Text Documentation' menu option<br>The 'Generate Documentation' dialog displays. |
| 3 | In the 'Use Template' field, click on the drop-down arrow and select 'Balanced Score Card'. |
| 4 | Click on the Generate button. |

# Model Validation

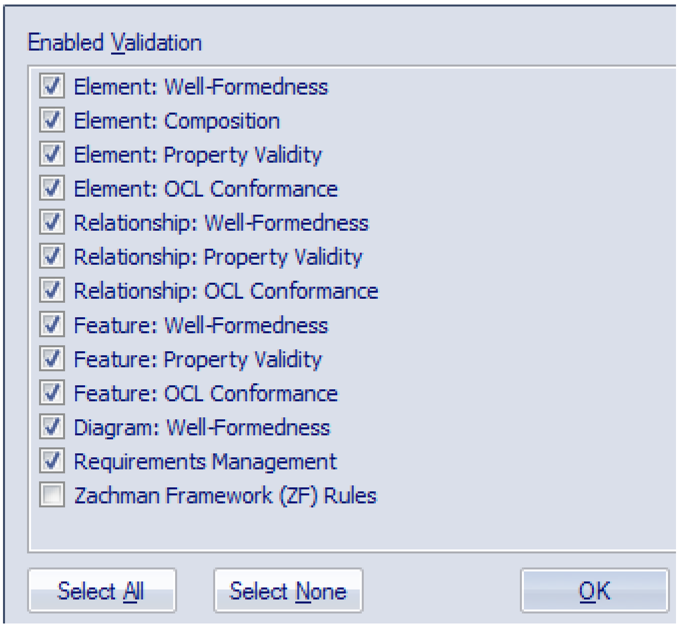
The Zachman Framework registers with Enterprise Architect to receive model validation requests from users.

## Configure Model Validation

To configure Enterprise Architect to perform Zachman Framework model validation, select:

- 'Design > Package > Edit > Validation > Configure Validation Rules'

The 'Model Validation Configuration' dialog displays.



Enabled Validation

☑ Element: Well-Formedness<br>
☑ Element: Composition<br>
☑ Element: Property Validity<br>
☑ Element: OCL Conformance<br>
☑ Relationship: Well-Formedness<br>
☑ Relationship: Property Validity<br>
☑ Relationship: OCL Conformance<br>
☑ Feature: Well-Formedness<br>
☑ Feature: Property Validity<br>
☑ Feature: OCL Conformance<br>
☑ Diagram: Well-Formedness<br>
☑ Requirements Management<br>
□ Zachman Framework (ZF) Rules

To perform validation on Zachman Framework models only, click on the Select None button and then click on the checkbox for 'Zachman Framework (ZF) Rules'. Click on the OK button.

## Validate Zachman Framework Model

You can validate, against the Zachman Framework rules:

- An element and any connectors attached to it
- A diagram and all its elements, or
- A Package and all its diagrams and elements

To do this, click on the element, diagram or Package and then select:

- 'Design > Package > Edit > Validation > Validate Current Package'

The 'Model Validation status' dialog displays, showing the progress of the validation.

# Validation Messages for Elements

These error messages can be output by the validation of a Zachman Framework element.

## Messages

| Element | Diagram and Message |
|---|---|
| Event Node | Event Schedule<br>Message: Event Nodes must be used only with Business Cycles<br>Meaning: An Event Node has been used with elements other than Business Cycle. |
| Event Node | Event Schedule<br>Message: Message triggered Event Node must have a message defined<br>Meaning: An Event Node with the 'Trigger' Tagged Value set to 'Message' does not have the 'MessageDetail' Tagged Value set. |
| Event Node | Event Schedule<br>Message: Rule triggered Event Node must have Rule defined<br>Meaning: An Event Node with the 'Trigger' Tagged Value set to 'Rule' does not have the 'Rule' Tagged Value set. |
| Event Node | Event Schedule<br>Message: Error triggered Event Node must have the Error defined<br>Meaning: An Event Node with the 'Trigger' Tagged Value set to 'ErrorDetail' does not have the 'Error' Tagged Value set. |
| Event Node | Event Schedule<br>Message: Multiple triggered Event Node must have a defined list of Triggers<br>Meaning: An Event Node with the 'Trigger' Tagged Value set to 'Multiple' does not have the 'Trigger' Tagged Value set. |
| Business Cycle | Event Schedule<br>Message: Business Cycles must have Event Nodes defined<br>Meaning: A Business Cycle element does not have any Event Nodes defined. |
| Goal | Business Motivation/ Strategy Map<br>Message: Goal not realized<br>Meaning: A Goal has no relationship defined with other model artifacts. |
| Strategy | Business Motivation/ Strategy Map<br>Message: Strategy not realized<br>Meaning: A Strategy has no relationship defined with other model artifacts. |

# Validation Messages for Connectors

These error messages can be output by the validation of a Zachman Framework connector.

## Messages

| Connector | Diagram and Message |
|---|---|
| Association | Data Map<br>Message: DataMap Association must have a valid source element<br>Meaning: An Association has a source element other than Principal Entity, Structure Entity or Intersecting Entity. |
| Association | Data Map<br>Message: DataMap Association must have a valid target element<br>Meaning: An Association has a target element other than Principal Entity, Structure Entity or Intersecting Entity. |
| Association | Data Map<br>Message: Possibility of an Intersecting entity < name> which might represent a Potential Business Process exists – This is a warning message.<br>Meaning: An Association has a many-to-many relationship, informing that the relationship could be normalized. |
| Strategy Link | Strategy Map<br>Message: StrategyMap Association must have a valid source element<br>Meaning: A Strategy Link has a source element other than Strategy and Goal. |
| Strategy Link | Strategy Map<br>Message: StrategyMap Association must have a valid target element<br>Meaning: A Strategy Link has a target element other than Strategy and Goal. |

# Validation Messages for Diagrams

These error message can be output by the validation of a Zachman Framework diagram.

## Messages

| Diagram | Message |
|---|---|
| Data Map | Entities must have relations in DataMap<br>Meaning: In the Data Map diagram there are entities with no relationships defined. |

# ArchiMate

The MDG Technology for ArchiMate®3 is one of the business modeling tools integrated with Enterprise Architect.

ArchiMate® is an open-standard enterprise architecture language from The Open Group, based on the IEEE 1471 standard. It offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure, that can be used by Enterprise Architects to describe, analyze and clearly visualize the relationships between business domains.

## Access

On the Diagram Toolbox, click on 🔍 to display the 'Find Toolbox Item' dialog and specify 'ArchiMate 3 <diagram type>'.

| | |
|---|---|
| Ribbon | Design > Diagram > Toolbox |
| Keyboard Shortcuts | Alt+5 |

## ArchiMate Integration

| Aspect | Detail |
|---|---|
| ArchiMate Toolbox Pages | For each of the Business, Application and Technology diagram types, the Toolbox pages show separate categories of elements - Information, Behavior and Structure - which provides clear differentiation between elements in different layers; for example, Business Service, Application Service and Infrastructure Service.<br>Similarly, there are separate connector pages - Structural, Dynamic, and Other - although the connectors have parallels with standard UML connectors.<br>The ArchiMate integration provides two further diagram types:<br>• Motivation<br>- For modeling stakeholders, drivers for change, business goals, principles and requirements<br>- Contains additional elements - Stakeholder, Driver, Assessment, Goal, Requirement, Constraint and Principle<br>- Contains an additional relationship - Influence<br>• Implementation and Migration<br>- Supports project portfolio management, gap analysis, and transition and migration planning<br>- Contains additional elements - Work Package, Deliverable, Plateau and Gap<br>You can modify the appearance of elements by toggling the 'Advanced \| Use Rectangle Notation' context menu option on the element in a diagram.<br>• For Artifact, Process, Function, Interaction, Service, Event, Actor, Role, Collaboration, Interface, Component, Node and Device elements:<br>- Select 'Use Rectangle Notation' to show a rectangle with a decoration in the top corner<br>- Deselect 'Use Rectangle Notation' to show the iconic representation of the element<br>Because there is a large range of ArchiMate elements, using the Quick Linker |

| | |
|---|---|
| | arrow is very helpful in guiding you in selecting appropriate source and target elements and relationship types to model your enterprise architecture. |
| ArchiMate in Enterprise Architect | Developing ArchiMate diagrams is fast and simple using the ArchiMate MDG Technology. The ArchiMate facilities are provided in the form of:<br>• Several ArchiMate diagram types, accessed through the 'New Diagram' dialog<br>• A comprehensive set of ArchiMate pages in the Toolbox, for each diagram type<br>• ArchiMate element and relationship entries in the 'Toolbox Shortcut' menu and Quick Linker<br>Enterprise Architect currently supports integration with releases of ArchiMate up to and including ArchiMate 3. |
| Generate Model Exchange File | You can export any model created using the built-in MDG Technology for ArchiMate 2 or Archimate 3 in Model Exchange File Format. |
| Import Model Exchange File | If you have a valid ArchiMate 2 or Archimate 3 Model Exchange File, you can import it into an Enterprise Architect project as a UML model. |

## Notes

- ArchiMate® is a Registered Trademark of The Open Group
- Enterprise Architect also supports ArchiMate and ArchiMate 2; if necessary, you can migrate your ArchiMate model to ArchiMate 2, or ArchiMate 2 to ArchiMate 3

# Migrate Model to ArchiMate 2 or Archimate 3

If you have created a model under one version of ArchiMate, you can migrate all or part of it to a later version; for example, migrate the model from ArchiMate 2 to ArchiMate 3. You perform the migration using the Automation Project Interface function Migrate(), which updates the Tagged Values and, if required, stereotypes to the later version of ArchiMate for all elements, attributes, connectors and diagrams under the selected Package or element.

When you migrate your model, you must have both releases of the Technology enabled; when the migration is complete, disable the older release of the Technology.

## Migrate ArchiMate 2 model to ArchiMate 3

The MDG Technology for ArchiMate 3 has a migration script built into it, to upgrade a model built for ArchiMate 2 to ArchiMate 3. To execute this script:

1. Select the 'Code > Tools > Scripting' ribbon option.
2. Expand the *ArchiMate® 3* folder.
3. Right-click on the 'Migrate ArchiMate 2 to ArchiMate 3' script.
4. Click on the 'Run Script' option from the context menu.
5. Monitor the execution of the script in the System Output window.

When the script has finished executing, disable the ArchiMate 2 technology.

## Notes (Migrate ArchiMate 2 to ArchiMate 3)

- If you want to migrate from an ArchiMate model to an ArchiMate 3 model, you must do so in two stages: from ArchiMate to ArchiMate 2, and from ArchiMate 2 to ArchiMate 3

## Migrate ArchiMate model to ArchiMate 2

You must create a script to call the Migrate() function to migrate a Package or element to ArchiMate 2: this is a VB script example of such a script:

```
Sub MigrateElement (sGUID, lngPackageID)
   Dim proj as EA.Project
   set proj = Repository.GetProjectInterface
   proj.Migrate sGUID, "ArchiMate", "ArchiMate 2"
   'refresh the model
   If lngPackageID<>0 Then
       Repository.RefreshModelView (lngPackageID)
   End If
End Sub
Sub MigrateSelectedItem
   Dim selType
   Dim selElement as EA.Element
   Dim selPackage as EA.Package
   selType = GetTreeSelectedItemType
   If selType = 4 Then 'means Element
       set selElement = GetTreeSelectedObject
       MigrateElement selElement.ElementGUID, selElement.PackageID
       MsgBox "Element Migration Completed",0,"ArchiMate 2 Migration"
   ElseIf selType = 5 Then 'means Package
       set selPackage = GetTreeSelectedObject
       MigrateElement selPackage.PackageGUID, selPackage.PackageID
       MsgBox "Package Migration Completed",0,"ArchiMate 2 Migration"
   Else
       MsgBox "Select a Package or Element in the Project Browser to initiate migration",0,"ArchiMate 2 Migration"
   End If
End Sub
Sub Main
   MigrateSelectedItem
End Sub
Main
```

## Notes (Migrate ArchiMate to ArchiMate 2)

- All diagrams are converted to Business Layer diagrams
- Collaboration elements are converted to Business Collaboration or Application Collaboration depending on the value of the 'collaborationtype' Tagged Value
- Object elements are converted to Business Object, Contract or Data Object depending on the value of the 'objecttype' Tagged Value
- Interface elements are converted to Business Interface, Application Interface or Infrastructure Interface depending on the value of the 'interfacetype' Tagged Value
- Function elements are converted to Business Function or Application Function depending on the value of the 'functiontype' Tagged Value
- Interaction elements are converted to Business Interaction or Application Interaction depending on the value of the 'interactiontype' Tagged Value
- Service elements are converted to Business Service, Application Service or Infrastructure Service depending on the value of the 'servicetype' Tagged Value
- The 'iconstyle' Tagged Value is removed and the 'Use Rectangle Notation' menu option is set on or off as appropriate
- Process elements become Business Process elements
- Event elements become Business Event elements
- Actor elements become Business Actor elements
- Role elements become Business Role elements
- Component elements become Application Component elements
- Software elements become System Software elements
- Specialisation connectors become Specialization connectors
- Realisation connectors become Realization connectors
- Network Connector connectors become Network
- Node elements now extend UML Class elements

- Software elements now extend UML Class elements
- Junction elements now extend UML Decision elements

# Generate Model Exchange File

After you have created an ArchiMate® 2 or ArchiMate® 3 model using the built-in MDG Technology for ArchiMate 2 or Archimate 3 respectively, you can generate a Model Exchange File from the model Package.

## Access

| | |
|---|---|
| Ribbon | Specialize > Technologies > ArchiMate > Generate Model Exchange File |
| Context Menu | Right-click on Package \| Specialize \| ArchiMate \| Generate Model Exchange File |

## Generate Model Exchange File

| Option | Action |
|---|---|
| Package | Displays the name of the currently-selected Package. If this is not the required Package, click on the 📂 button and select the correct Package. |
| Filename | Type in the file path and name of the file to be generated, or click on the ... button and browse for the location. |
| Version | Click on the drop-down arrow and select the version of the Model Exchange File. |
| Language | (Optional) Select the language identifier for the content of the elements in the generated file. |
| Include | Select the appropriate checkboxes:<br>• 'Element Relationships' - to include connectors between the elements being generated<br>• 'Tagged Values' - to include the Tagged Values for each element and relationship being generated<br>• 'Element Organization' - to include the structural organization of the elements inside the selected Package<br>• 'Diagrams' - include the ArchiMate diagrams inside the selected Package |
| Generate | Click on this button to generate the Model Exchange File.<br>The progress of the file generation is reported in the 'ArchiMate' tab of the System Output window. A message box also displays to indicate when the generation is complete; click on the OK button to clear the message. |
| View File | Click on this button to display the contents of the generated file. |
| Close | Click on this button to close this 'Generate Model Exchange File Format' dialog. |
| Help | Click on this button to display this Help topic. |

## Notes

- This facility is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect
- The 'Language' field displays the 2-letter codes from the Language Subtag Registry: http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry
- The Model Exchange File generated by Enterprise Architect adheres to The Open Group Specified Schema
- Generated elements and diagrams will be displayed in the System Output window – double-click on an item in the System Output window to highlight it in the Project Browser
- Elements, connectors and diagrams that are not from the built-in MDG Technology for ArchiMate 2 or Archimate 3 will be ignored during generation
- The Network and Communication Path connectors from the 'ArchiMate 2 Technology' toolbox in Enterprise Architect will not currently be generated as they do not have any mapping in The Open Group Specified Schema
- The Technology Object element, Path connector, Communication Network connector and Distribution Network connector from the 'ArchiMate 3 Technology' toolbox in Enterprise Architect will not currently be generated as they do not have any mapping in The Open Group Specified Schema

# Import ArchiMate Model Exchange File

If you have a valid ArchiMate® 2 or ArchiMate® 3 Model Exchange File, you can import it into your Enterprise Architect project as a UML model.

## Access

| | |
|---|---|
| Ribbon | Specialize > Technologies > ArchiMate > Import Model Exchange File |
| Context Menu | Right-click on Package \| Specialize \| ArchiMate \| Import Model Exchange File |

## Import Model Exchange File

| Option | Action |
|---|---|
| Package | Displays the name of the currently-selected Package.<br>If this is not the required Package, click on the 📂 button to select the Package Browser and select the correct Package. |
| Filename | Type in the file path of the file to be imported, or click on the ... button and browse for the location. |
| View File | Click on this button to open and view the file to be imported. |
| Import | Click on this button to import the selected Model Exchange file into the specified Package.<br>The progress of the file import is reported in the 'ArchiMate' tab of the System Output window. A message box also displays to indicate when the import is complete; click on the OK button to clear the message. |
| Close | Click on this button to close the dialog. |
| Help | Click on this button to display this Help topic. |

## Notes

- This facility is available in the Professional, Corporate, Unified and Ultimate editions of Enterprise Architect
- The Model Exchange File should adhere to The Open Group Specified Schema
- Elements, connectors and diagrams that are not specified in *The Open Group Schema* will be ignored during import
- Imported elements and diagrams will be displayed in the System Output window – double-click on an item in this window to highlight it in the Project Browser