ENTERPRISE ARCHITECT

用户指南系列

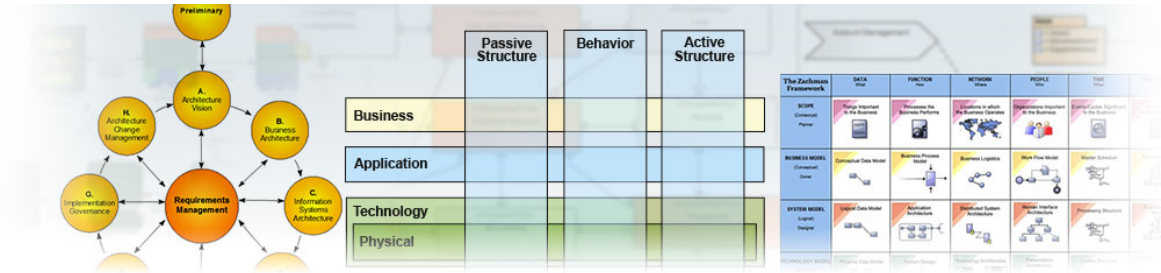
建模框架

Author: Sparx Systems
Date: 13/11/2024
Version: 17.0

创建于 ENTERPRISE ARCHITECT

目录

建模框架



为企业建模是一项艰巨的任务，但使用久经考验的行业框架可以降低一些风险，并增加您从创建企业架构模型所使用的努力和资源中获得的收益。 Enterprise Architect支持最流行的框架，包括 TOGAF、UPDM、UAF、Zachman 框架和联邦企业架构框架。该工具提供了许多预先构建的模式，在使用时将减少所需的建模工作并确保您开发符合行业标准的最佳实践模型，从而使您可以将精力集中在企业的架构描述上。

The Zachman Framework	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why
SCOPE (Contextual) 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 (Conceptual) Owner	Conceptual Data Model	Business Process Model	Business Logistics	Work Flow Model	Master Schedule	Business Plan
SYSTEM MODEL (Logical) Designer	Logical Data Model	Application Architecture	Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
TECHNOLOGY MODEL (Physical) Builder	Physical Data Model	System Design	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
DETAILED REPRESENTATIONS Sub-Contractor	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification
FUNCTIONING ENTERPRISE	Data	Function	Network	Organization Units	Schedule	Strategy

*Zachman Framework Interfaces*图表显示了*Enterprise Architect*模型支持的单元。

Enterprise Architect允许您单独或组合使用每个框架。例如，您可以将 TOGAF 用于您的整体企业架构治理和描述，但使用 Zachman 框架作为您的内容框架来描述和编目参考架构景观和参考库、标准和治理日志中的工件。在本主题中，您将学习如何安装和使用每个框架。

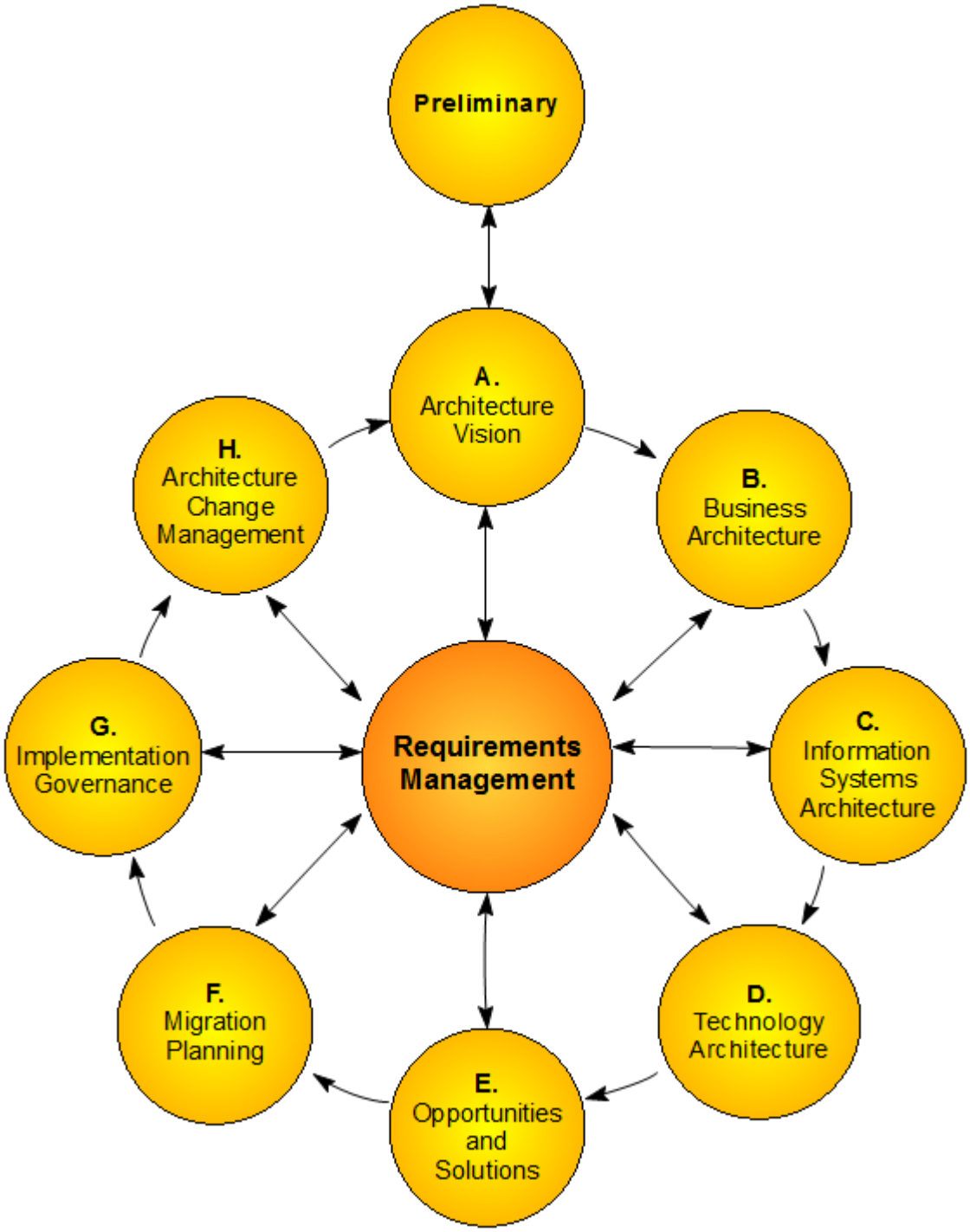
建模框架

框架/语言	描述
	ArchiMate 是由 The Open Group 定义的框架，它提供了一种用于业务架构分

ArchiMate 框架	析和设计的可视化建模语言。它的使用范围主要面向企业建模。
TOGAF 	Open Group架构框架(TOGAF) 是开发企业架构最广泛接受的方法之一，它为开发和维护企业架构提供了一种实用、明确和经过验证的分步方法。
UPDM 	UAF/UPDM 与Sparx Systems Enterprise Architect紧密集成，并为规划、设计和实施 DoDAF 和 MODAF (UPDM) 架构的统一配置文件提供基于模型的框架。
扎克曼框架 	Zachman 框架是一种广泛使用的工程企业架构方法。框架是一个简单的逻辑结构，有助于组织描述企业的内容模型。
ArcGIS 地理数据库	ArcGIS 框架使用基于UML的建模标准支持 Esri ArcGIS 平台的地理数据库结构的建模和设计。 ArcGIS 模型可以通过 ArcGIS XML工作空间文档进行导出和导入。
谷歌云平台(GCP)	Enterprise Architect的谷歌云平台建模提供了用于创建用于指定新的云基础设施和平台以及记录现有 GCP 结构的富有表现力的 GCP 图的结构。
Amazon Web Services (AWS)	Enterprise Architect的Amazon Web Services建模构造允许您创建详细的图表，指定基于 AWS 基础设施的云基础设施和平台。这包括定义 IaaS（基础设施即服务）和 PaaS（平台即服务）环境。
微软天青	Microsoft Azure 提供服务来定义 IaaS（基础设施即服务）、PaaS（平台即服务）和 SaaS（软件即服务）云环境。 Enterprise Architect提供建模构造，允许您创建富有表现力的 Azure 图表。
MDG 技术	使用MDG 技术，您可以扩展核心UML结构以创建基于公开定义的模型语言或您自己定义的语言的建模框架。作为技术开发人员，可以使用Enterprise Architect开发您自己的定制建模语言和解决方案。

The Open Group Architecture Framework (TOGAF)

The Open Group Architecture Framework (TOGAF)是开发企业架构最广泛接受的方法之一。TOGAF 是一个开放的框架，为开发和维护企业架构提供了一种实用、明确且经过验证的分步方法。您可以使用Enterprise Architect中的 TOGAF功能来模型任何规模的企业，并且可以创建或导入任意数量的工件包括目录、矩阵和图表，这些工件都可以方便地存储在将用作架构存储库的存储库中。所有工件都按照 TOGAF 元模型进行存储；参考库、标准和治理日志都可以在该工具中建模。



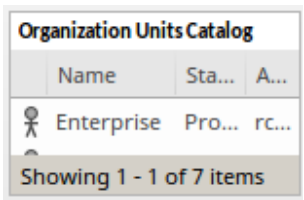
The Open Group 的 *TOGAF*架构开发方法

企业架构是一门重要的学科，因为组织需要了解其业务的基本方面，以便跟上不断发展的世界中全球市场和技术的变化。 Enterprise Architect内置了对所有重要企业架构框架和企业建模语言的支持，使您可以从业务目标和

驱动因素到基于云的基础设施服务来模型企业。在本主题中，您将学习如何使用 TOGAF 来模型企业，包括使用 ADM 和元模型属性。

讨论

此处描述的主题提供了在Enterprise Architect中使用 TOGAF 的介绍和程序说明。

部分	内容
欢迎 	本节介绍 TOGAF，并包含定义其与Enterprise Architect一起使用的正式文档。
使用 TOGAF 	开始使用 TOGAF，了解模型结构、模板、图表类型等。
TOGAF ADM 	TOGAF 的关键仍然是一种可靠、实用的方法 - TOGAF架构开发方法 (ADM) - 用于定义业务需求和开发满足这些需求的架构，应用 TOGAF 的元素和组织可用的其他架构资产。
TOGAF企业连续体 	TOGAF企业Continuum 是企业和整个 IT 行业中存在的所有架构资产（模型、模式、架构描述和其他工件）的 虚拟存储库”，企业认为自己可以使用这些资产为企业开发架构。
联邦企业架构框架 	TOGAF 提供特定于联邦企业架构框架 (FEAF) 的图表和工具箱页面。它还提供了 FEAF 性能参考模型和技术参考模型。
TOGAF 目录 	Enterprise Architect模型帮助您使用工件-Catalog模式创建模型Catalog模型，用于： • 演员 • 业务服务 • 组织单位 • 原则 • 需求和 • 角色

简单的介绍

欢迎使用与Enterprise Architect集成的 The Open Group架构框架(TOGAF)。

使用这项技术， Enterprise Architect的用户可以在基于开放标准的多功能建模环境中受益于 TOGAF。

关于 TOGAF

Open Group架构框架是开发企业架构最广泛接受的方法之一。 TOGAF 是一个开放框架，为开发和维护企业架构提供了一种实用、明确且经过验证的分步方法。

TOGAF 的关键仍然是一种可靠、实用的方法 - TOGAF架构开发方法 (ADM) - 用于定义业务需求和开发满足这些需求的架构，应用 TOGAF 的元素和组织可用的其他架构资产。

TOGAF 体现了企业连续体的概念，以反映架构开发过程中的不同抽象级别。通过这种方式，TOGAF 促进了不同层次参与者之间的理解和合作。它为与上下文ADM 结合使用多个框架、模型和架构资产提供了时间。通过企业Continuum，除了 TOGAF 基础架构之外，鼓励架构师利用所有其他相关架构资源和资产来开发特定于组织的IT架构。

有关 TOGAF 本身的详细信息，请访问 TOGAF 网站。

TOGAF 的好处

- 帮助使业务流程和 IT 与业务战略和目标保持一致
- 为 ADM 中的所有阶段提供支持
- 为OMG的业务动机模型提供支持
- 为架构内容模型提供支持
- 为 As-Is 和 To-Be架构的可视化建模提供支持
- 为 TOGAF 特有的所有四个架构域（业务、应用程序、数据和技术）建模提供支持
- 为 TOGAF 工作产品的报告生成提供支持
- 提供 Open Group 的 TOGAF 可交付模板作为链接文档模板
- 提供开箱即用的 FEAF 参考模型

TOGAF特征

- 用于架构开发方法 (ADM) A可视可点击接口
- 有用的入门模型可帮助您快速提高工作效率
- FEAF业务、性能、服务和技术参考模型的UML配置文件
- 使用Enterprise Architect的关系矩阵和层次结构视图对模型工件进行有效的关系管理
- Enterprise Architect的外部文件、审计log和报告生成特征的链接，提供额外的功能来维护和管理您的企业架构
- 该技术A TOGAF 特定词汇表

开始

有关如何在Enterprise Architect中开始使用 TOGAF 的说明，请参阅使用 *TOGAF*帮助主题。

TOGAF系统需求

TOGAF 9.x 版在以下环境下运行：

操作系统

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

Enterprise Architect版本

- Enterprise Architect版本11. 1或更高版本

TOGAF 支持

Enterprise Architect 中的 TOGAF 建模技术支持向Enterprise Architect的注册用户提供，其方式与Enterprise Architect Enterprise Architect完全相同。

许可版权和商标

TOGAF 由 The Open Group 拥有和管理，任何想要将材料用于商业用途的组织都必须向 The Open Group 申请商业许可。请参阅*The Open Group TOGAF*网站。

TOGAF 版权声明

TOGAF 软件产品许可协议

本软件产品许可协议涉及单独购买的用于 TOGAF 的MDG 技术，用于Sparx Systems Enterprise Architect的企业版和专业版。与 Enterprise Architect 的终极版和统一版集成的 TOGAF 的MDG 技术包含在Enterprise Architect的 Sparx Systems Enterprise Architect Modelling Tool。

MDG 技术for TOGAF, Enterprise Architect MDG插件

,版本3.0。

重要 - 请仔细阅读：本最终用户许可协议（“EULA”）是您作为被许可人与 SPARX 之间就上述软件产品达成的法律协议。通过安装、复制或以其他方式使用软件产品，您同意受本 EULA 条款的约束。如果您不同意本 EULA 的条款，请立即删除未使用的软件产品。

软件产品及其文档的版权归Sparx Systems Pty Ltd, A所有。 B 。 N 38 085 034 546. 根据本 EULA 的条款，您被授予在 EULA 有效期内使用软件产品的非排他性权利。您不会根据本 EULA 获得软件产品任何部分的版权或其他知识产权的属性。

您使用本软件即表示您接受本 EULA 和保修。

定义

在本最终用户许可协议中，除非出现相反意图：

- “EULA”是指本最终用户许可协议
- A SPARX”是指Sparx Systems Pty Ltd A.C.电话号码N 034 546
- 被许可人”是指您或您代表其接受 EULA 的组织（如果有）
- “TOGAF 的MDG 技术注册版”是指可从以下网站购买的软件产品版本： https ://sparxsystems.com/products/mdg/tech/togaf/purchase.html ，在三十 (30) 天免费评估期
- 软件产品”或 软件”是指 TOGAF 的MDG 技术，包括计算机软件和相关媒体和印刷材料，可能包括在线或电子文档
- 支持服务”是指 SPARX 提供的基于电子邮件的支持，包括有关软件产品使用的建议、错误调查、修复、模型维修（如果适用）以及一般产品支持
- “SPARX 支持工程师”是指提供在线支持服务的 SPARX 员工
- “TOGAF 的MDG 技术Trial Edition ”是指在三十 (30) 天内免费提供用于评估目的的软件产品版本

授予许可

根据本 EULA 的条款，您被授予以下权利：

- 在单台计算机上安装和使用软件产品的一个副本，或替代同一操作系统的任何先前版本；作为安装本软件产品的计算机的主要用户，您可以制作第二份副本，供您在家或便携式计算机上独家使用
- 在存储设备（例如网络服务器）上存储或安装软件产品的副本，仅用于通过内部网络安装或运行软件产品
- 为备份、存档和指导目的制作软件产品的副本

评估许可证

TOGAF 的MDG 技术Trial Edition不是免费软件。根据本协议的条款，特此授权您在三十 (30) 天内免费使用本软件进行评估。

三十 (30) 天评估期到期后，必须从计算机中删除软件产品。在 30 天评估期后未注册使用 TOGAF 的MDG 技术 Trial Edition违反了澳大利亚、 U . S 。和国际版权法。

SPARX 可根据要求自行决定延长评估期。

如果您选择在 30 天评估期后使用此软件，则必须购买许可证（如https中所述）。支付许可费后，您将收到有关在何处下载 TOGAF MDG 技术注册版的详细信息，并将通过电子邮件向您提供合适的软件 密钥”。

附加权利和限制

您在此承诺，除非本 EULA 明确授权，否则不会出售或再许可软件产品。

没有保修。软件产品按 原样”提供，不提供任何形式的保证，SPARX 明确否认与软件产品有关的所有明示、暗示或法定保证和/或条件，包括但不限于暗示保证和/或适销性、令人满意的质量、适用于特定目的、准确性、安静享受和不侵犯第三方权利的条件。

局限性

在任何情况下，SPARX 均不对因本许可或您使用、复制、修改、分发软件产品或其任何部分而引起或与之相关的任何偶然、特殊、间接或后果性损害承担责任，无论是否根据合同理论, 保证, 严格责任或其他, 即使版权持有人已被告知此类损害的可能性, 尽管任何补救措施的基本目的失败。

商标

本 EULA、软件产品或随附文档中使用的所有产品和公司名称可能是其相应所有者的商标。它们在本 EULA 中的使用旨在遵守相应的指南和许可。

适用法律

本协议应根据维多利亚州的状态联邦法律解释。

商标确认

微软的商标

- 微软®
- 视窗®

OMG 的商标

- 天啊™
- 物件管理组™
- UML™
- 统一建模Language™

The Open Group 的商标

- TOGAF™

使用 TOGAF

TOGAF 为企业架构的规划、设计和实施提供了一个基于模型的框架。 TOGAF 提供的启动器模型作为您构建企业架构的基础。您可以从扩展的Enterprise Architect UML图集创建适当的图，使用支持相接口图表的每一个方面的工具箱页面。您还可以使用Enterprise Architect关系矩阵在架构开发方法 (ADM) 的各个阶段调整模型。

注记

- TOGAF 与Enterprise Architect的特征集成
- Enterprise Architect与其他面向服务的架构工具（例如 SOMF 和 SoaML）以及更广泛的架构建模工具（例如 ArchiMate、SPEM 和企业规则建模建模工具）集成，您可以将所有这些工具与模型结合使用来建模和开发您的企业架构

开始使用 TOGAF

TOGAF 与Enterprise Architect的统一和终极版本完全集成，在其中启用并准备使用。

如果您有Enterprise Architect的企业版，您可以单独购买并安装TOGAF的MDG 技术；一旦您为 TOGAF 输入了 MDG 技术的注册密钥，它就会自动在Enterprise Architect中可用并集成到统一版和终极版中。

您可以在Enterprise Architect的专业版中使用 TOGAF 配置文件。但是，专业版中的 TOGAF 没有差距分析矩阵特征。

访问TOGAF

1. 创建一个新的Enterprise Architect项目文件，然后单击顶层包。
2. 选择功能区选项 设计>包>模型构建器”。
3. 在“模型Builder”对话框中，选择 企业架构> TOGAF”蓝图和 Starter模型”模式。
4. 单击 创建模型”按钮。

浏览器窗口中将创建A新的基本 TOGAF模型，其中包含 TOGAF架构开发方法 (ADM) 结构和企业Continuum 资产包，并显示 TOGAF-ADM (接口) 图。

TOGAF模型模式

TOGAF 包含一组模型模式，您可以使用它们在 TOGAF 项目中生成单独的模型。这些模型模式可通过模型生成器获取。

访问

使用这里概述的任何方法显示模型构建器。

进入模型构建器后，选择 企业架构> TOGAF"蓝图。

从 TOGAF模式中选择：

- 入门模型（包括 ADM 和企业）
- 架构开发方法（ADM）
- 连续企业
- 技术参考模型
- 目录

如果您需要其他图表，则在模型构建器中，单击 图表构建器"选项卡并（如有必要）选择 企业架构> TOGAF"蓝图。然 从图表类别中选择：

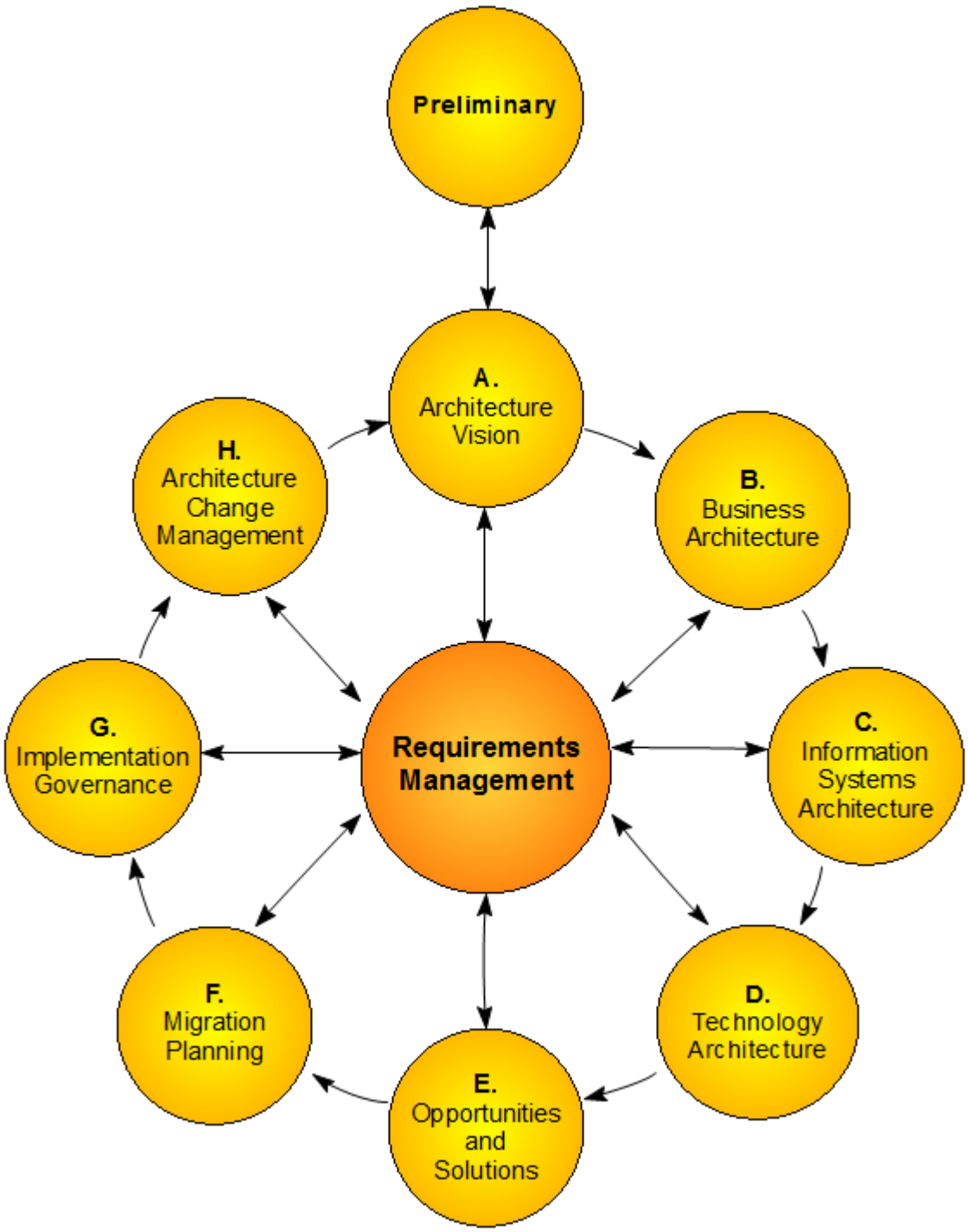
- FEAF图表（联邦企业架构框架）
- TOGAF_业务架构
- TOGAF_数据架构
- TOGAF图表

在 图表类型"面板中，选择所需的图表类型。

功能区	开始> 个人>图表生成器 设计>包>模型生成器
上下文菜单	右键点击包|模型（模式库）
键盘快捷键	Ctrl+Shift+M
其它	浏览器窗口标题栏：  |模型生成器（模式库）

TOGAF接口图表

在Enterprise Architect中，TOGAF 框架以预定义模型的形式呈现。该模型结构的模型级图为TOGAF接口图，作为基于TOGAF开发企业架构的用户界面。

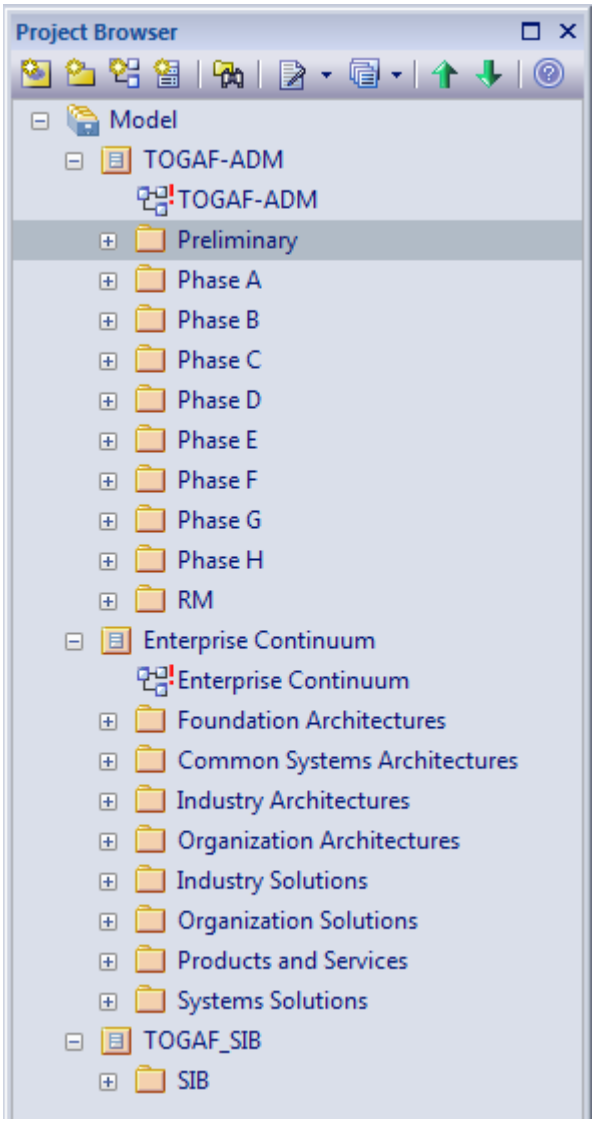


TOGAF 框架模型使用UML包，从模型结构图中可以看出。接口图本身是一个标准的UML包图，使用自定义图像。

双击接口图的一个单元打开模型包和对应于该特定 ADM相的图。

TOGAF模型结构

在 TOGAF 框架模型中，每个 ADM相都被建模为最高级别的包。



图表

TOGAF 提供了多种图表类型来支持使用 TOGAF 进行建模。这些图表包括：

TOGAF 图：

- TOGAF接口
- 概念框架
- 架构内容
- 架构开发方式
- 服务模型
- 企业连续体
- 标准信息库

TOGAF_BusinessArchitecture：

- 好处
- 业务动机模型
- 组织的结构
- 业务物流
- 业务流程

TOGAF_DataArchitecture：

- 资料图

FEAF 图：

- (FEAF)业务参考模型
- (FEAF) 服务部件参考模型
- (FEAF) 技术参考模型
- (FEAF) 性能参考模型

可以以与Enterprise Architect中的任何其他图表相同的方式创建 TOGAF 特定图表。当您打开 TOGAF 图时，Enterprise Architect会自动为该图打开相应的工具箱页面。

TOGAF工具箱页面

MDG 技术对于 TOGAF工具箱页面为技术支持的所有 TOGAF 图提供元素和关系。

访问

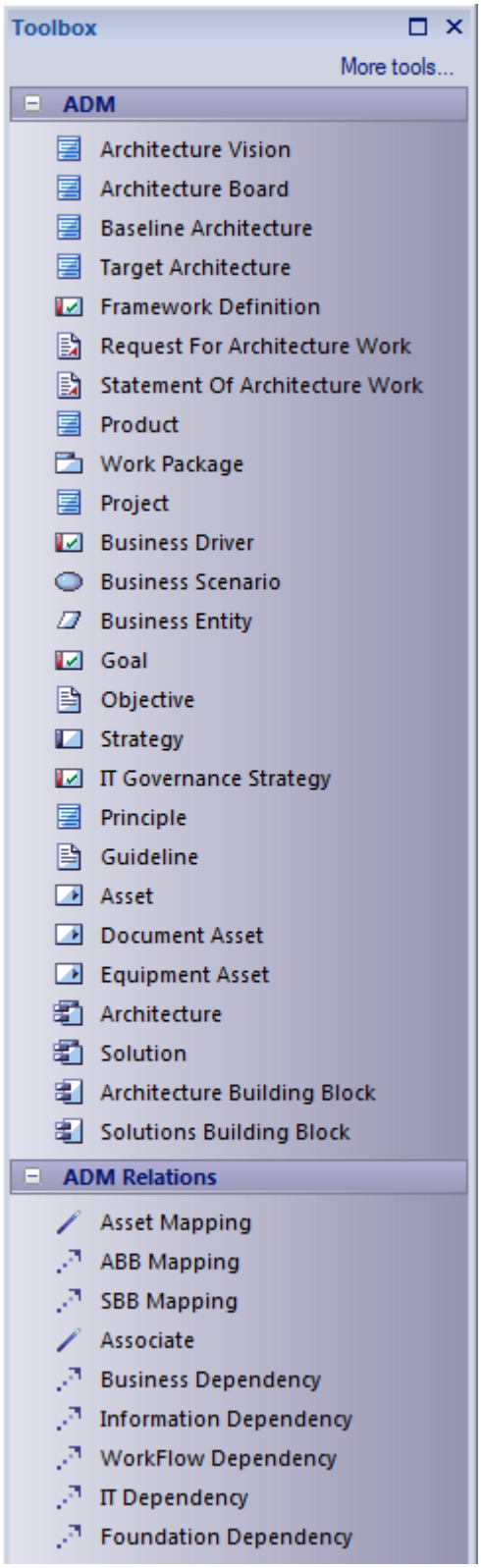
当您打开 TOGAF 图表时， Enterprise Architect会显示对该特定图表类型最有用的工具箱页面。此外，无论打开哪个图表， UML元素和关系的 "公共元素"和 "公共关系"页面都会显示。

图表工具箱页面可以停靠在图表的任一侧，或自由浮动在图表顶部以暴露更多表面以供编辑。

功能区	设计>  工具箱图表 "查找工具箱项"对话框中指定 "TOGAF"
键盘快捷键	Ctrl+Shift+3 :  > 在 "查找工具箱项目"对话框中指定 "TOGAF"
其它	您可以通过单击  图表工具箱显示或隐藏图形图表视图。

架构开发方法工具箱Pages

架构开发方法 (ADM) 元素用于定义和模型ADM 所有阶段中的 TOGAF 特定原语。您使用它们来定义架构的范围。



架构开发方法工具箱

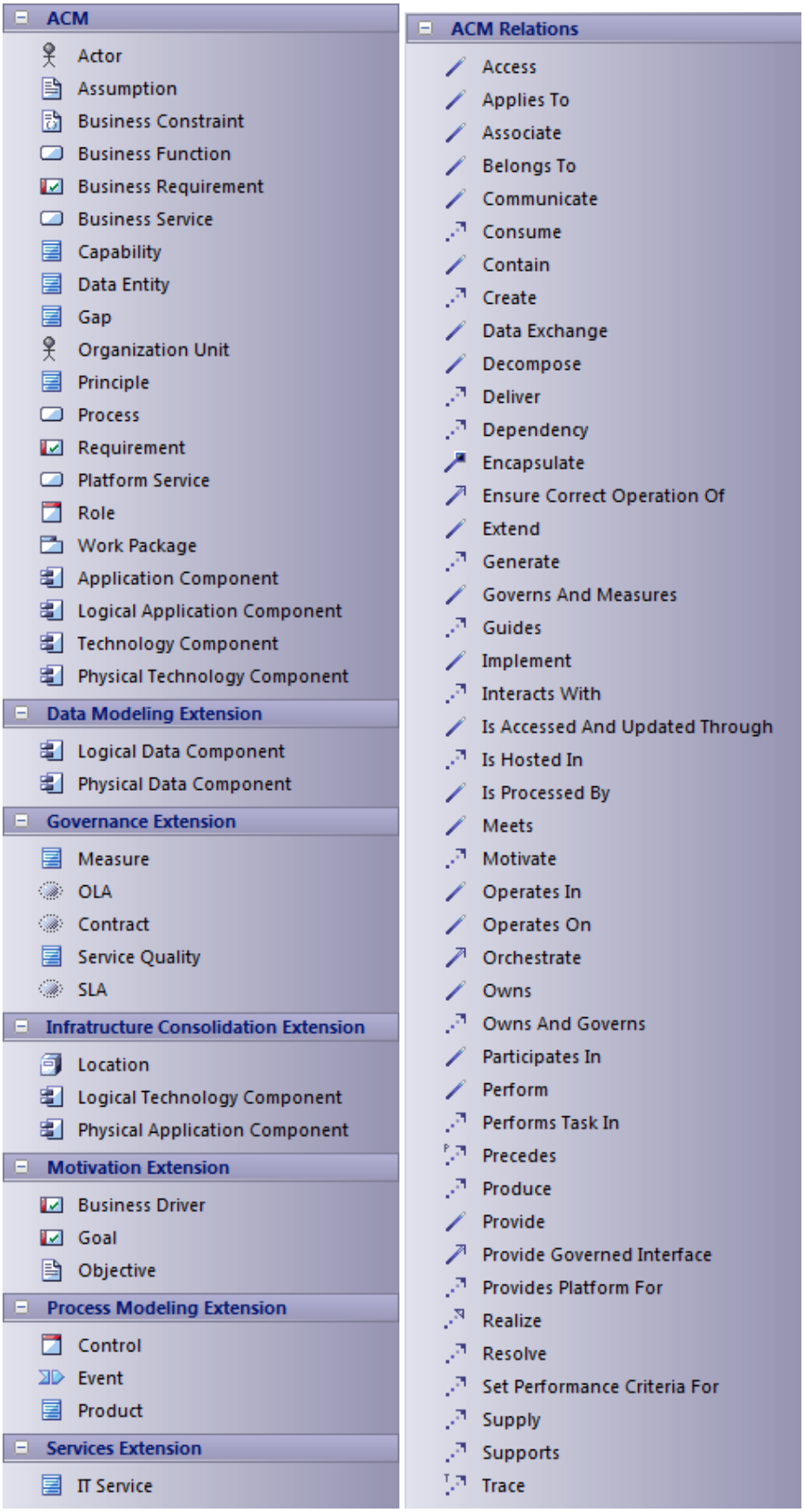
物品	描述
架构愿景	阐明实现业务目标的愿景，响应战略驱动因素，遵守原则，并解决利益利益相关者的担忧和目标。 标记值、范围、版本
架构板	捕获跨组织架构板的定义。这是成功架构治理战略的关键元素，监督战略的实施。 该机构应代表架构中的所有关键利益相关者，通常由一组负责审阅和维护整体架构的高管组成。 标记值、权限范围、责任
基线架构	从业务信息系统和技术的角度，捕捉高度基线环境的定义。要定义的细节的范围和级别取决于现有架构元素可能被转移到目标架构中的程度。 标记值、类型、版本
目标架构	从业务信息系统和技术的角度捕获目标环境的高级定义。 标记值、类型、版本
框架定义	提供框架的文本描述。 值– ID、版本标记值
申请架构工作	捕获申请架构工作的信息，这是 ADM 阶段的主要输入。 本元素被设计为文档工件。在创建此类型的新元素时，双击该元素以打开链接文档并从 复制模板”选项可用的模板列表中选择 TOGAF - 请求架构工作 ”模板。 标记值– ID、建筑组织、赞助组织
架构工作声明	捕获架构工作声明的信息，这是 ADM 阶段的主要输出。 本元素被设计为文档工件。在创建这种类型的新元素时，双击该元素以打开链接文档并从 复制模板”选项可用的模板列表中选择 TOGAF –架构工作声 ”模板。 值– ID、版本标记值
产品	获取企业生产的产品信息。 标记值– ID
工作包	定义一组为企业实现一个或多个目标的行动。工作包A是项目、已完成项目或计划的一部分。 标记值– CapabilityDelivered、 WorkPackageCategory、 ID、源、拥有着
项目	捕获信息以定义为创建产品或服务而进行的计划努力。 标记值– ID,进程, Introduction, ProjectDevelopment Process,Overview, References,目标架构映射
业务司机	在 名称”字段中定义业务驱动程序。 值– ID、版本标记值

业务场景	识别和阐明业务需求，从而得出架构开发必须解决的业务需求。创建业务场景涉及以下步骤： 1. 识别、记录和排列驱动场景的问题。 2. 识别场景的业务和技术环境并将其记录在场景模型中。 3. 识别和记录期望的目标。 4. 识别人类参与者（参与者）及其在业务模型中的位置。 5. 识别计算机参与者（计算元素）及其在技术模型中的位置。 6. 识别并记录每个参与者的角色、责任和成功衡量标准；记录每个参与者所需的脚本，以及处理情况的结果。 7. 检查 适合目的"并仅在必要时进行改进。 业务场景A链接文档模板由技术提供。要使用模板，请右键单击元素并选择 编辑链接文档"菜单选项。为 复制模板"选项选择 TOGAF业务场景/架构愿景"。 标记值– ID
实体业务	捕获企业资源A通用元素。 值– ID,描述标记值
目标	捕捉企业要实现的目标，以标记值定义的规范。 标记值成功因素、目标类型、ID、关键绩效指标、度量、责任单位、机会、优势、威胁、劣势
客观的	捕获企业为实现其目标而寻求满足的可实现的、有时间目标的和可衡量的目标。 标记值– ID
战略	捕获业务计划的战略声明。 行动计划、标记值预算、预计时间周期、ID、度量、目标
IT 治理策略	定义 IT 治理的战略声明。 值– ID、版本标记值
原则	定义和指导组织使用整个企业的所有资产和资源。每项原则都应与相关的业务目标和关键架构驱动因素相关联。 标记值、基本原理、语句、类型、版本
指南	通过提供有关开展设计或实施活动的最佳方式的指导，获取管理企业及其职能的指南。 标记值– ID
资产	捕获可以评估价值的企业资源。 标记值– ID, AssetValue,描述
文件资产	捕获企业重要文档资源A资产子类型。 标记值– ID, AssetValue,描述
设备资产	捕获企业设备资源A资产子类型。 标记值– ID, AssetValue,描述

架构	在特定时间点捕获架构景观（即企业的状态）的摘要视图。 标记值源、拥有着、主题、视图点、详细程度、抽象程度、准确性、版本、成熟度
解决方案	捕获针对特定架构的解决方案的摘要视图。 标记值源、拥有着、Subject Matter、时间、Volatility、版本、Maturity
架构构建块	(ABB) 与架构连续统一体相关，并根据 ADM 的应用进行定义或选择。 标记值– ID、描述、所属组织、Rationale、ServicePortfolio
解决方案组件	(SBB) 与解决方案连续体相关，可以采购或开发。 标记值– ID、描述、供应商组织

架构内容模型工具箱页面

架构内容框架为架构内容提供了一个结构模型，使架构师创建的主要工作产品能够被一致地定义、结构化和呈现。



每个架构内容模型工具箱页面中的元素在单独的主题中进行了描述：

- *ACM* 核心

- 数据建模扩展
- 治理扩展
- 基础设施整合扩展
- 动机扩展
- 进程建模扩展
- 服务扩展

有关架构内容模型关系的信息，请参阅TOGAF 在线文档中的主题架构内容元模型关系。

ACM 核心

元素来自架构内容模型工具箱的 ACM 页面。

ACM核心工具箱

物品	描述
参与者	识别具有启动活动或与活动交互的角色的个人、组织或系统。参与者可以是组织内部或外部的。 标记值源、拥有着、#FTEs、ActorGoal、ActorTasks
假设	定义由于外部约束而在此阶段尚未完全验证的可能事实的陈述。 标记值、标注理由、声明、类型
业务约束	识别阻止组织采用特定方法来实现其目标的外部因素。 标记值– ID
函数业务	确定提供与组织紧密一致的业务能力的因素，但不一定由组织明确管理。 标记值– ID
业务需求	定义特定架构或工作包必须满足的业务需求的定量陈述。 标记值– ID
业务服务	标识通过明确定义的接口支持业务功能并由组织明确管理的服务。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate
能力	定义通过完成一个或多个工作包交付的以业务为中心的结果。使用基于能力的规划方法，可以对变更活动进行排序和分组，以提供持续和增量的业务价值。 标记值源、拥有着、Increments、BusinessValue
数据实体	定义被业务领域专家识别为实体的数据封装。逻辑数据实体可以绑定到应用程序、存储库和服务，并且可以根据实施考虑进行结构化。 标记值、Category、标记源、拥有着、PrivacyClassification、RetentionClassification
间隙	提供两种状态之间差异的陈述。用于上下文分析和目标架构之间的基线，在哪里确定差异。 标记值、类别、源、拥有着
组织单位	定义具有直线管理职责、目标、目的和措施的自包含资源单元。组织可以包括外部各方和业务合作伙伴组织。 标记值– ID, PersonIncharge
原则	提供架构应满足的定性意向声明。这至少有一个支持的理由和重要性的衡量标准。

	值– ID、类型、标记值声明、基本原理、含义
进程	代表功能和/或服务之间或内部的控制流（取决于定义的粒度）。流程代表序列活动，这些活动共同实现指定的结果，可以分解为子流程，并且可以显示函数或服务的操作（在下一个细节级别）。流程也可用于链接或组合组织、功能、服务和流程。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate、ProcessCritality、ProcessVolumetrics、ProcessType
平台服务	定义提供支持应用程序交付的支持基础设施所需的技术能力。 标记值、Category、源、拥有着、StandardClass
角色	定义函数通常或预期的功能，或某人或参与者在特定动作或事件中扮演的角色。一个参与者可以有多个角色。 标记值、Category、标记源、拥有着、Responsibilities
工作包	确定为实现一个或多个业务目标而采取的一组行动。工作包A是项目、完成项目或计划的一部分。 标记值、Category、源、拥有着、CapabilityDelivered
应用程序部件	提供与实现结构一致的应用程序功能封装。 也见： 逻辑应用部件”和 物理技术部件”。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate
逻辑应用程序部件	提供独立于特定实现的应用程序功能封装。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate
技术部件	提供代表一类技术产品或特定技术产品的技术基础设施的封装。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate
物理技术部件	定义特定技术基础架构产品或技术基础架构产品的实例。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate、ModuleName、ProductName、Vendor、版本

数据建模扩展

元素来自架构内容模型工具箱的数据建模模型扩展页面。

数据建模Extensions工具箱

物品	描述
逻辑数据部件	定义一个边界区域，将相关数据实体封装起来，形成一个要保存的逻辑位置。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate
物理数据部件	定义一个边界区域，将相关数据实体封装起来，形成一个物理位置来保存。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate

治理扩展

架构内容模型工具箱的 Governance扩展页面中的元素。

治理扩展工具箱

物品	描述
措施	确定可以跟踪的指标或因素，通常是在持续的基础上，以确定成功或与目标和目标的一致性。 标记值、类别、源、拥有着
合同	定义服务消费者和服务提供者之间的协议，为交互建立功能性和非功能性参数。 标记值– ID,源,拥有着, 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
法律事务厅	定义操作级别协议。 标记值– ID,源,拥有着, 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
服务水平协议	定义服务水平协议 标记值– ID,源,拥有着, 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
服务质量	定义可分配给服务或服务合同的非功能属性的预设配置。 标记值、类别、源、拥有着

基础设施整合扩展

来自架构内容模型工具箱的 Infrastructure Consolidation扩展页面的元素。

Infrastructure Consolidation扩展工具箱

物品	描述
地点	代表业务活动发生的地方，可以分层分解。 标记值、类别、源、拥有着
逻辑技术部件	提供独立于特定产品的技术基础架构封装。 A类科技产品。 标记值源、拥有着、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate
物理应用部件	标识应用程序、应用程序模块、应用程序服务或其他可部署的功能组件。 标记值– ID,源,拥有着, 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

动机扩展

元素来自架构内容模型工具箱的 Motivation扩展页面。

动机扩展工具箱

物品	描述
业务司机	定义激励组织定义其目标的外部或内部条件。 值– ID、版本标记值
目标	为组织提供高层次的意图或方向声明。通常用于衡量组织的成功。 标记值、类别、源、拥有着
客观的	确定组织的有时限里程碑，以展示实现目标的进展。 标记值– ID

进程建模扩展

元素来自架构内容模型工具箱的进程建模扩展页面。

进程建模扩展工具箱

物品	描述
控件	定义一个带有决策逻辑的决策步骤，用于确定流程的执行方法或确保流程符合治理标准。 标记值、类别、源、拥有着
事件	定义触发处理事件的组织状态变化；可以来自组织内部或外部，可以在组织内部或外部解决。 标记值、类别、源、拥有着
产品	定义业务产生的输出；即流程执行的业务产品。 标记值、类别、源、拥有着

服务扩展

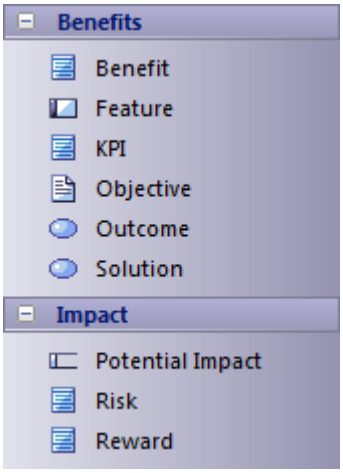
元素来自架构内容模型工具箱的服务扩展页面。

服务扩展工具箱

物品	描述
资讯科技服务	定义业务服务的自动化元素。信息系统服务可以交付或支持一项或多项业务服务的部分或全部。 标记值源、拥有着、DefinitionText、ContactPoint、Availability、ChargeToUser、DependentSystems、StandardsClass、StandardCreationDate、LastStandardReviewDate、NextStandardReviewDate、RetireDate

福利工具箱页面

您可以使用收益工具箱创建元素来表示和描述架构定义中确定的机会，并根据它们的相对大小、收益和复杂性进行分类。利益相关者可以使用生成的收益图来对已识别机会的选择、优先级和排序做出决策。

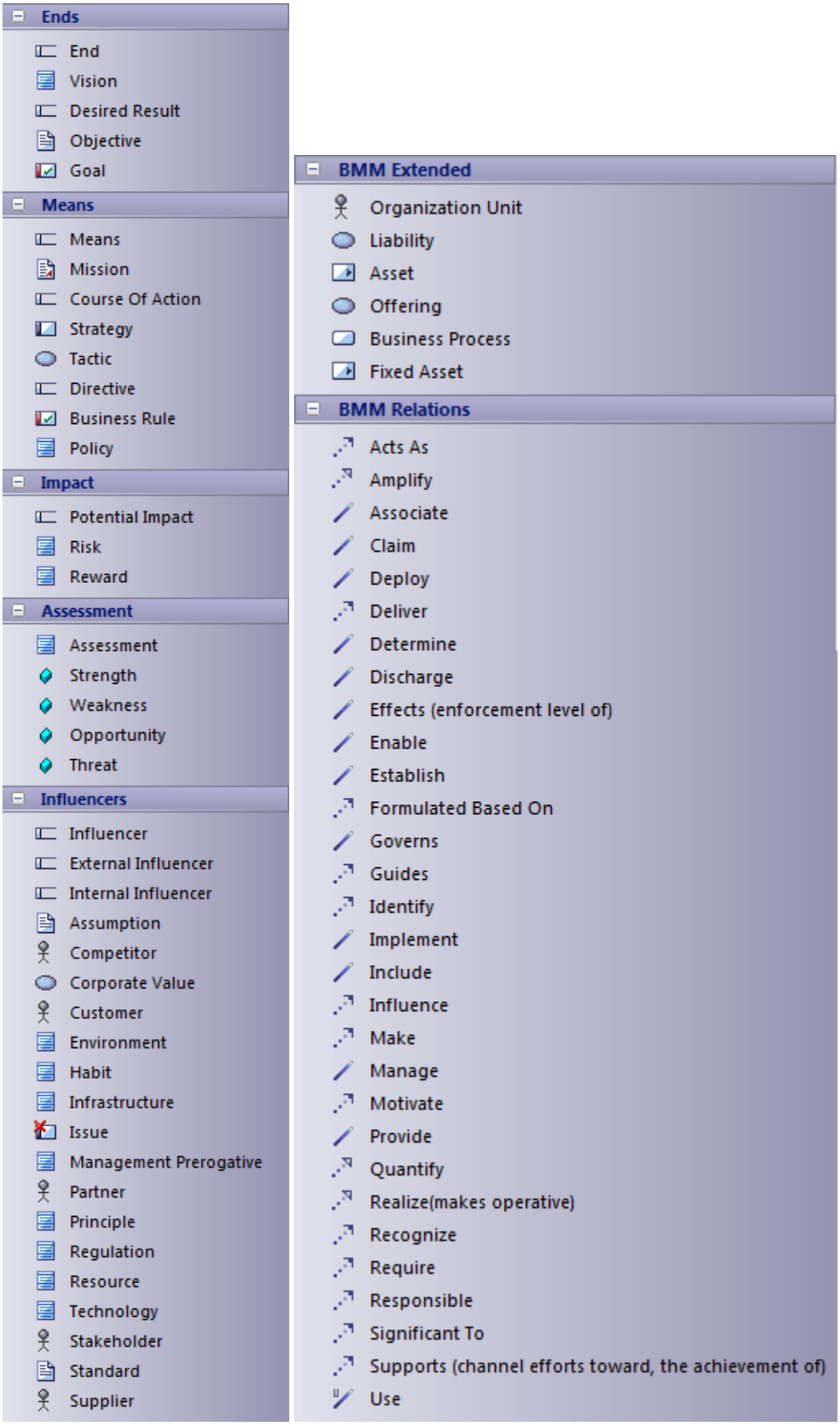


福利工具箱

物品	描述
益处	在架构定义中识别模型的工件。 值– ID,拥有着标记值,源,类别
特征	代表服务或解决方案的特征 值– ID,拥有着标记值,源,类别
关键绩效指标	（关键绩效指标）A定义和衡量实现目标或关键成功因素的进度的指标。 值– ID,拥有着标记值,源,类别
客观的	企业为实现其目标而寻求达到的可实现的、有时间目标的和可衡量的目标A声明。一个目标量化了一个目标。 标记值– ID
结果	事件、决策或架构过程的最终状态。 值– ID,拥有着标记值,源,类别
解决方案	支持结果的操作或活动A陈述。 值– ID,拥有着标记值,源,类别
潜在影响	有关业务动机模型，请参见 影响”帮助页面上的工具箱。
风险	有关业务动机模型，请参见 影响”帮助页面上的工具箱。
报酬	有关业务动机模型，请参见 影响”帮助页面上的工具箱。

业务动机模型工具箱Pages

业务动机模型工具箱页面基于 OMG 规范的业务动机模型（BMM）。这些元素为以有组织的方式制定、沟通和管理业务计划提供了一个结构。



每个业务动机模型工具箱页面中的元素在单独的主题中进行了描述：

- 结束页面

- 意味着页面
- 影响页面
- 评估页面
- 影响者页面
- *BMM*扩展页面

结束页面

元素来自业务动机模型工具箱的 Ends"页面。

结束工具箱

物品	描述
结尾	将 结束"概念（愿景和期望结果）分组。 目的是企业寻求完成的事情。它不包括如何实现它的任何指示。 标记值、类别、源、拥有着
愿景	描述企业的未来状态，而不考虑如何实现。 愿景A使命支持或实施，并由目标放大。 标记值– ID
期望的结果	将 期望结果"概念（目标和目标）分组。期望结果是企业打算维持或维持的状态或目标A结束。行动课程支持期望A结果。一个期望结果可以包括其他期望结果，并且本身可以包含在另一个期望结果中。 标记值、类别、源、拥有着
目标	关于通过适当方式实现或维持的企业状态或状况A声明。 A目标放大了一个愿景。 标记值成功因素、目标类型、ID、关键绩效指标、度量、责任单位、机会、优势、威胁、劣势
客观的	企业为实现其目标而寻求达到的可实现的、有时间目标的和可衡量的目标A声明。一个目标量化了一个目标。 标记值– ID

意味着页面

元素来自动机业务模型工具箱的 手段”页面。

手段工具箱

物品	描述
方法	分组 手段”概念（使命、行动课程和指令）。手段代表任何可以用来实现预期目的A能力。 标记值、类别、源、拥有着
使命	捕捉企业的使命宣言、政策和价值观。使命表明企业正在进行A经营活动，使愿景具有可操作性。 标记值、类别、源、拥有着
行动	分组 行动过程”概念（战略和战术）。行动过程是A方法或计划，用于配置企业的某些方面，涉及事物、流程、位置、人员、时间或动机，为实现预期结果而采取。 A Course of行动将努力引导到期望的结果。行动课程由指令管理。也可以根据指令制定行动课程。行动过程可以通过业务流程来实现。一个行动课程可以包含其他行动，一个行动可以由另一个行动课程行动。 标记值– ID, 分类
战略	在给定环境限制和风险的情况下，定义实现一组目标的正确方法。 A通常引导努力实现这些目标。 行动计划、标记值预算、预计时间周期、ID、度量、目标
战术	行动过程，代表战略细节A一部分。战术实施A或多个战略。 标记值– ID, 分类
指示	表示应该或不应该如何执行行动过程。指令定义、约束或解放企业A某些方面。它旨在断言业务结构或控制或影响业务行为，并以声明形式进行说明。指令管理行动课程。指令被定义为直接支持期望结果A实现。 标记值– ID, 分类
企业规则	企业规则元素捕获企业规则声明。业务规则提供具体A 、可操作的治理或指导来实施。业务业务规则指导业务流程。 标记值– ID、名称、标记、Effective_From、描述、状态、版本、Enforcement_Level
政策	捕获企业中遵循的策略定义。 A业务政策是一项不可采取行动的指令，其目的是管理或指导企业。业务政策为业务规则提供基础。业务政策也管辖业务流程。一个业务策略可以包括其他业务策略。 标记值– ID

影响页面

元素来自业务动机模型工具箱的 影响”页面。

冲击工具箱

物品	描述
潜在影响	对 影响”（风险和奖励）的概念进行分组。每个潜在影响都是以特定术语、类型或维度对评估的某些方面进行量化或限定的评估。 评估确定了一些潜在影响。潜在影响A评估可能很重要。 标记值、类别、源、拥有着
风险	表示损失、伤害、不利或破坏可能性A潜在影响。 标记值– ID
报酬	表示获得概率A潜在影响。 标记值– ID

评估页面

元素来自业务动机模型工具箱的 评估”页面。

评估工具箱

物品	描述
评估	对影响者A判断会影响组织使用其手段或实现其目的的能力。指令A评估激发。评估也可以使用其他评估。评估可以支持目标的实现。 标记值、源、拥有着标记
力量	此类评估表明企业内的某些优势或卓越领域可能会影响其使用手段或实现目标。它被建模为评估元素的参数。 标记值– ID
弱点	此类评估表明企业内某些领域的不足可能会影响其使用手段或实现目标。它被建模为评估元素的参数。 标记值– ID
机会	此类评估表明，某些影响者可以对组织使用手段或实现目标产生有利影响。它被建模为评估元素的参数。 标记值– ID
威胁	此类评估表明，某些影响者可能会对组织对手段的使用或目的的实现产生不利影响。它被建模为评估元素的参数。 标记值– ID

影响者页面

元素来自业务动机模型工具箱的 影响者”页面。

影响工具箱

物品	描述
影响者	影响者元素对影响评估的元素进行分组。影响者是那些可以影响企业使用手段或实现其目的的人。这种影响具有在评估中判断的影响。 标记值– ID, 分类
外部影响者	外部影响者元素对对评估有外部影响的元素进行分组。外部影响者是企业组织边界之外的那些可以影响其使用手段或实现目的的人。 标记值– ID, 分类
内部影响者	内部影响者元素对对评估有内部影响的元素进行分组。内部影响者是来自企业内部的那些可以影响其使用手段或实现目的的人。 标记值– ID, 分类
假设	一个假设元素捕获了在信息操作中所做的假设；假设是被视为理所当然或没有证据的信息项目。 标记值、标注理由、声明、类型
竞争者	外部影响者，即对主题企业构成挑战的个人或企业。 标记值– ID
企业价值	企业提倡或赞同（正面或负面）的理想、习俗或制度。 标记值– ID
顾客	外部影响者是指从目标企业调查、订购、接收或支付产品或服务的个人或企业。 标记值– ID
环境	环境元素是影响企业生存或发展的周边条件或影响因素的集合。 标记值– ID
习惯	习惯做法或用途A 标记值– ID
基础设施	形成系统基本框架或特征的内部影响者。 标记值– ID
问题	有争议A问题或竞争伙伴之间有争议的事项。
管理特权	凭借在企业中的所有权或地位而行使A权利或特权。 标记值– ID

伙伴	外部影响者作为与标的企业分担风险和利润（或与标的企业有关联以分担风险和利润）的企业，因为这是互惠互利的。 标记值– ID
原则	定义和指导组织使用整个企业的所有资产和资源。每项原则都应与相关的业务目标和关键架构驱动因素相关联。 标记值、基本原理、语句、类型、版本
规定	外部影响者是政府机构或企业管理层等权威机构规定的命令。 标记值– ID
资源	内部影响者作为可用于开展企业业务的资源，尤其是通过其质量来应用其影响力。 标记值– ID
技术	外部影响者作为技术的角色，包括其发展和局限性——可能存在使用技术的先决条件，或者技术支持或限制的企业活动。 标记值– ID
利益相关者	捕捉对企业感兴趣并参与其中的参与者。 标记值– ID
标准	定义企业遵循的标准。 标记值、Statement、标记类型
供应商	外部影响者，即可以向目标企业提供或提供产品或服务的个人或企业。 标记值– ID

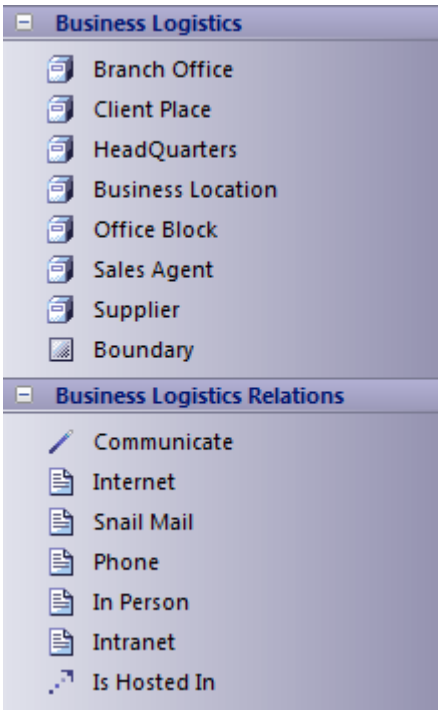
BMM扩展页面

元素来自业务动机模型工具箱的 'BMM扩展"页面。

BMM扩展工具箱

物品	描述
组织单位	代表在企业时间范围内任何公认的人的上下文。在层次结构中，它可能是公司、部门、部门、小组或团队。 标记值– ID, PersonInCharge
责任	负债是为履行承诺A保留的实际资源（材料、成品、人员时间、现金）。 A Liability 可以通过 Courses of行动来履行，可以是组织单位的责任，并且可以要求资源。 标记值– ID
资产	资产是企业拥有的有价值的东西。 标记值– ID、描述、资产值
提供	产品是一种固定资产，是企业可以提供的产品或服务的规范。产品可以由行动课程定义，可以通过业务流程交付，可以需要资源并且可以使用固定资产。 标记值– ID
业务流程	企业或企业的一部分A函数或行为。业务流程是组织单位A责任，实现行动课程，受业务规则指导，受业务政策约束，可以交付产品并可以管理资产。 标记值– ID、描述、标记类型
固定资产	固定资产是随着时间的推移维护和重复使用A资产。固定资产可以被产品A并且可以提供资源。 标记值– ID, AssetValue 标记

业务物流工具箱Pages



业务物流工具箱

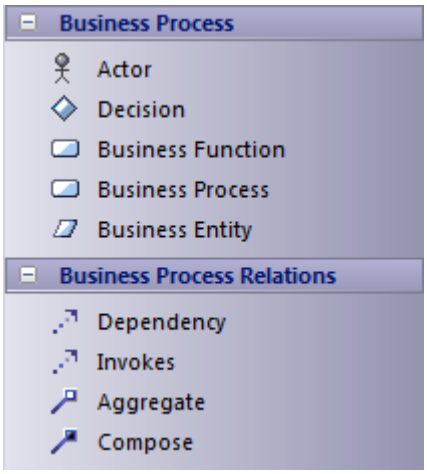
物品	描述
分支机构	将业务位置建模为分支机构。
客户场所	将业务位置建模为客户场所。
总部	将业务位置建模为总部。
业务地点	对业务运营的位置进行建模。
办公块	将业务位置建模为 Office块。
销售代理人	将业务地点建模为销售代理人。
供应商	将业务地点建模为供应商。
交流	表示一个营业地点直接与另一个营业地点通信。
互联网	表示通信方式是网络。
蜗牛邮件	表示通讯方式是邮政系统或快递服务。
电话	表示通讯方式是电话。

Person	表示沟通方式是直接的人对人。
内联网	表示通信方式是本地 Intranet 或 WAN。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

业务流程工具箱Pages



业务流程工具箱

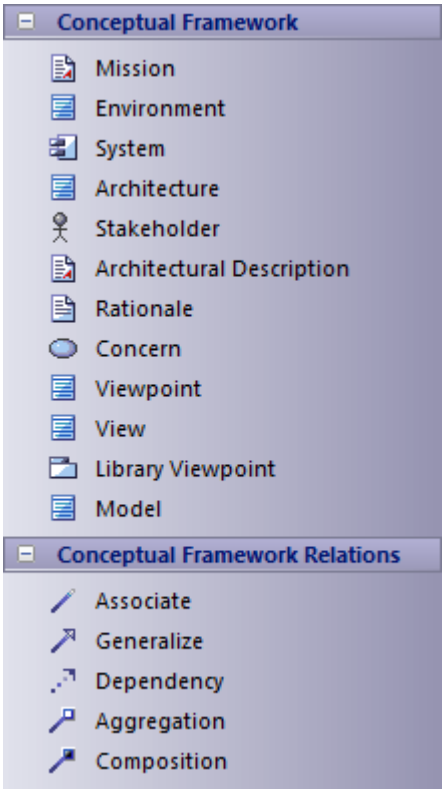
物品	描述
参与者	为企业的利益相关者或任何其他人力资源建模。
决策	表示做出业务决策的条件进展点。
函数业务	由企业或企业的一部分执行A主要函数。
业务流程	企业或企业的一部分A函数或行为。
实体业务	捕获企业资源A通用元素。
调用	A关系定义业务流程的调用。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

概念框架工具箱Pages

概念框架元素用于模型建筑描述并建立建筑思维的概念。工具箱元素设计基于 IEEE 标准 1471 - 2000。



概念框架工具箱

物品	描述
使命	捕捉企业的使命宣言、政策和价值观。 标记值– ID
环境	为企业工程工作的目的定义系统的开发、操作和程序上下文。 标记值– ID
系统	捕获企业工作组件的详细信息。系统包括，例如，应用程序、系统、平台、系统系统、企业和产品线。 标记值– ID
架构	捕获架构工作的定义。 标记值– ID
利益相关者	捕捉对企业感兴趣并参与其中的参与者。 标记值– ID
建筑描述	捕获架构描述并识别系统的利益相关者及其关注点。 标记值– ID

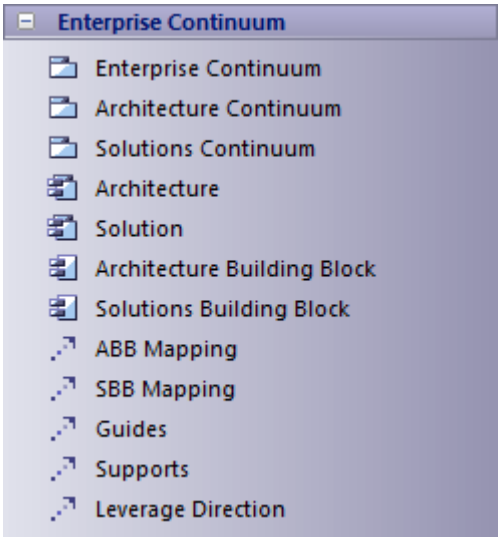
基本原理	捕获建筑描述的目的陈述。
关注	构成完整性的基础。建筑描述解决了所有利益相关者的关注点，每个关注点都由建筑视图
观点	视图A模式——视点定义了视图的视图。 每个视图对应一个视图。 标记值– ID
视图	从A组关注点的角度对整个系统的表示。 A视图可以包含一个或多个建筑模型，因此该视图可以使用多个符号。
图书馆观点	捕获分类视点的集合。 标记值– ID
模型	定义和表示模型。 标记值– ID

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

企业Continuum工具箱页面

企业Continuum 元素用于模型企业的架构Continuum 和Solution Continuum。使用这些元素，您可以通过映射到适当的架构模型或解决方案模型（如图表、元素和模型）来创建架构建造块或解决方案建造块。



企业Continuum工具箱

物品	描述
企业连续体	A模仿企业连续体的包。 标记值– ID、建筑组织、赞助组织
架构连续体	模拟架构Continuum A包。
解决方案连续体	A模拟解决方案连续体的包。
架构	在特定时间点捕获架构景观的摘要视图（例如企业的状态）。 标记值源、拥有着、主题、视图点、详细程度、抽象程度、准确性、版本、成熟度
解决方案	捕获针对特定架构的解决方案的摘要视图。 标记值源、拥有着、Subject Matter、时间、Volatility、版本、Maturity
架构构建块	与架构连续统一体相关，并根据 ADM 的应用进行定义或选择。 标记值– ID、描述、所属组织、Rationale、ServicePortfolio
解决方案组件	与解决方案连续体相关，可以采购或开发。 标记值– ID、描述、供应商组织
ABB 映射	连接器将架构模型和工件映射到架构建造块。
SBB 映射	连接器将解决方案模型和工件映射到解决方案建造块。

指南	连接器代表指导关系。架构构建建造指导建造的发展。
支持	连接器代表支持关系。建造支持开发其他的建造。
杠杆方向	连接器表示架构和解决方案组件的利用方向。

组织结构工具箱页面

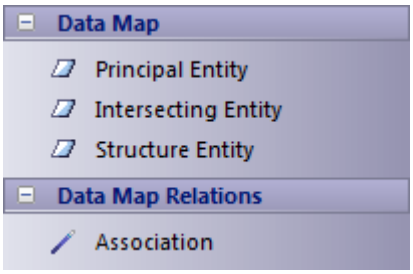
组织结构工具箱

物品	描述
董事会	捕获董事会的详细信息。
持股人	捕获企业的利益相关者。
外部组织	捕获不受企业直接控制但与企业有关系的任何外部业务单位。
组织单位	捕获受企业直接控制的任何业务单位。
人员	捕获企业中人员的详细信息。
合同中	捕获业务单位之间基于合同的关系。
效劳于	捕获团队链接的详细信息；例如，利益相关者1为组织单位1工作。
监督	捕获过程监督细节。
控件	获取负责单位或负责Person信息。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

资料图工具箱Pages



数据图工具箱

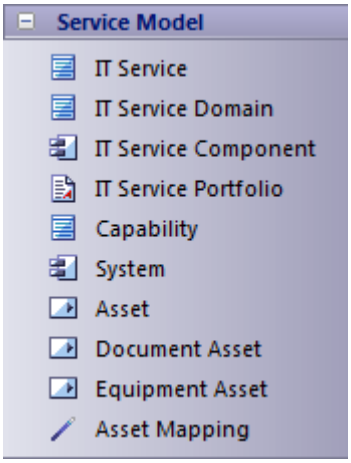
物品	描述
校长实体	构成企业资源A业务实体。
相交实体	规范主体实体之间的多对多关系。
结构实体	捕获潜在的知识库实体。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

服务模型工具箱页面

服务模型元素用于构建描述企业 IT 服务基础设施的概念框架。



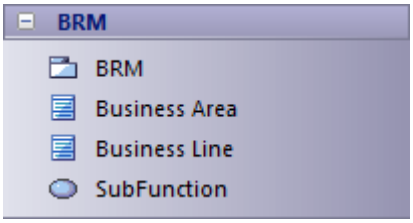
工具箱模型

物品	描述
资讯科技服务	捕获作为由企业管理的消耗性实体提供的 IT 功能。 标记值– ID、DefinitionText、拥有着、Availability、Charge_to_User、ContactPoint、Dependent_Systems
IT服务领域	对 IT 服务进行分类。 值– ID,描述标记值
IT 服务部件	捕获一组可能通过技术接口公开的功能。 标记值原理
IT 服务组合	文档工件A捕获描述 IT 服务组合所需的信息 标记值– ID
能力	通过完成A或多个工作包交付的以业务为中心的结果。使用基于能力的规划方法，可以对变更活动进行排序和分组，以提供持续和增量的业务价值。 标记值、分类、增量、业务值、源、拥有着
系统	捕获企业工作组件的详细信息。系统包括应用程序、系统、平台、系统系统、企业和产品线等内容。 标记值– ID
资产	捕获可以评估价值的企业资源。 标记值– ID, AssetValue,描述
文件资产	捕获企业重要文档资源的资产子类型。 标记值– ID, AssetValue,描述

设备资产	捕获企业设备资源的资产子类型。 标记值– ID, AssetValue,描述

FEAF业务参考模型工具箱页面

FEAF业务参考模型(BRM) 提供了一个框架，便于对企业的业务线 (LoB) 进行功能（而不是组织）视图，包括其内部运营和服务。



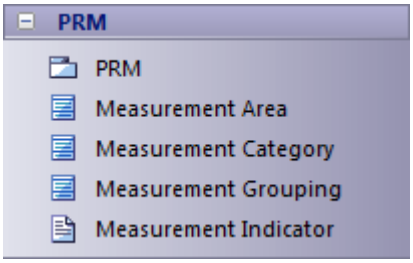
FEAF业务参考模型工具箱

物品	描述
BRM	用于捕获业务参考模型（BRM） A包。 值–版本标记值
业务区	BRM 的高级组织层，捕获与业务目的和目标相关的高级类别。 标记值– BusinessAreaID, 定义
业务线	捕获企业的业务线。 标记值– BusinessLineID、定义、参考业务区域
子函数	代表BRM 中最低级别的粒度，对与每个业务线相关的功能进行分组。 标注标记值– SubFunctionID、定义、参考业务线、参考业务区

FEAF Performance参考模型工具箱页面

FEAF 性能参考模型（PRM）工具箱页面旨在符合 FEAF-PRM 框架的规范。 PRM 是一个绩效测量框架，提供整个企业的通用输出测量。它通过提供一种使用机构的Enterprise Architect来衡量 IT 投资的成功及其对战略成果的影响的方法，使机构能够在战略层面更好地管理业务。

FEAF 绩效参考模型(PRM) 有助于基于比较确定哪些计划和组织更高效和有效的资源分配决策。

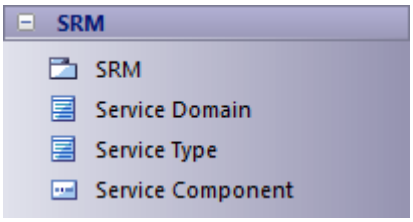


FEAF Performance参考模型工具箱

物品	描述
PRM	用于捕获性能参考模型A包。 值–版本标记值
测量区域	PRM 的高级组织层，捕获输出级别的性能方面。这一层与在机构和项目层面建立的绩效目标直接相关。 标记值– MeasurementAreaID, 定义
测量类别	根据要测量的属性或特征对测量区域进行分类。 标记值– MeasurementCategoryID, Definition, Reference Measurement Area
测量分组	进一步细化测量类别为特定类型的测量指标。 标记值– MeasurementGroupingID、定义、参考测量类别
测量指标	捕获具体措施。 标记值、定义、参考测量分组

FEAF Service部件参考模型工具箱页面

FEAF 服务部件参考模型(SRM) 是一个业务驱动的功能框架，根据服务组件支持业务和性能目标的方式对它们进行分类。该模型有助于推荐服务能力，以支持跨企业重用业务组件和服务。 SRM 应该跨水平服务区域构建，独立于业务功能，可以为应用程序、应用程序功能、组件和业务服务的重用提供可利用的基础。

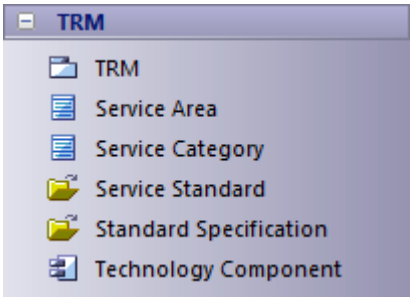


FEAF 服务部件参考模型工具箱

物品	描述
SRM	用于捕获服务部件参考模型A包。 值–版本标记值
服务领域	捕获支持企业和组织流程和应用程序的服务和功能的高级视图。 标记值– ServiceDomainID，定义
服务类型	对支持领域的类似能力进行分组，提供额外的分类层，定义给定上下文内特定能力组件的时间。 标记值——ServiceTypeID、定义、引用服务域
服务部件	捕获一组可能通过业务或技术接口公开的功能。服务组件是向企业提供信息管理能力的 构建块”。 标记值、定义、引用服务域、引用服务类型

FEAF技术参考模型工具箱页面

FEAF技术参考模型(TRM) 是一个组件驱动的技术框架，对标准和技术进行分类，以支持和实现服务组件和功能的交付。



FEAF 技术参考模型工具箱

物品	描述
TRM	捕获技术参考模型A包。 值–版本标记值
服务区	代表支持服务部件的安全构建、交换和部件的技术层。 标记值– ServiceAreaID, 定义
服务类别	就函数服务的业务或技术功能对较低级别的技术和标准进行分类。 标记值– ServiceCategoryID、定义、参考服务区域
服务标准	定义支持服务类别的标准和技术。 标记值ID、定义、参考服务类别
标准规格	提供标准的规范细节。 标记值

差距分析矩阵

TOGAF 的规范文档指出：

'间隙分析广泛用于 TOGAF架构开发方法 (ADM) 以验证正在开发的架构。基本前提突出显示基线架构和目标架构之间的差距；也就是说，故意省略、意外遗漏或尚未定义的项目。

TOGAF 提供了一个差距分析矩阵，您可以使用它：

- 确定基线和目标之间的差距
- 在存储库中创建间隙元素（如果识别出任何差距），以后可以将其作为任务处理和分配；然后可以使用间隙元素来确定活动的优先级
- 创建和管理差距分析矩阵

注记

- Enterprise Architect专业版中没有此特征

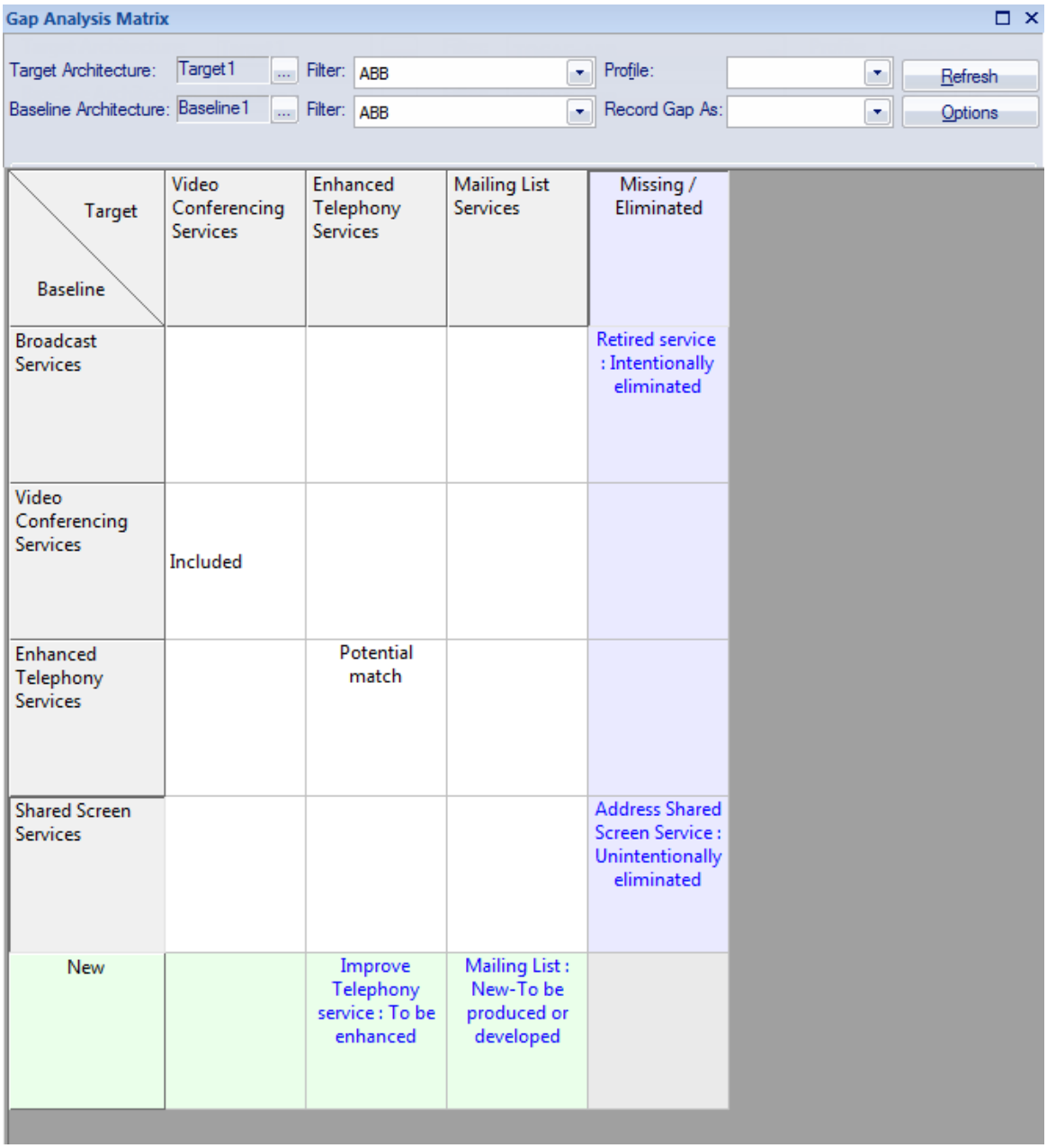
打开矩阵

访问

功能区	设计>包>间隙分析

示例

这个差距分析矩阵示例来自 TOGAF 规范；它说明了来自网络服务类别的服务的间隙分析架构建造块 (ABB)。



使用差距分析矩阵

 对滤器”字段列出了可以在矩阵中显示的所有构造型；使用这些字段为每个目标和基线设置过滤架构。

设置好过滤器后，在“  ”按钮和 基线架构目标架构”字段中浏览，在目标架构基线架构包右侧包选择。

点击刷新按钮；该矩阵列出了具有您在 对滤器”字段中设置的刻板印象的元素。横向列出的元素目标架构为列标题，基线架构列出的对象为行标题。如果双击包含基线或目标元素的相应行或列标题，则显示相应的 属性”对话框

要从浏览器窗口中的矩阵定位object ，请右键单击它并选择 在项目中查找浏览器”选项。

在单元列和目标元素行的交叉点上，您可以间隙分析元素注记基线。要编辑注记，请双击单元，或右键单击并选择 编辑间隙注记”选项。

任何不在支持目标架构中的元素，必须在最后一栏中称为 间隙丢失/消除消除”的元素中作为基线架构处理。任何在目标架构基线架构但不在最后一行的元素，必须作为间隙在最后一行的空间中处理，称为 新”。

在示例中：

- 广播服务和屏幕服务存在于基线架构中，但缺少目标架构；因此，您必须在矩阵的最后一列 丢失/消除”列中创建适当的间隙元素
- 邮件列表服务不在目标基线架构中，而是在目标架构中，这意味着该服务必须在目标架构中进行推广或开发；您必须在矩阵的最后一行 新”行中创建一个相应的间隙元素

注记

- 找到基线器，然后在 项目”选项卡的 项目”选项卡中找到浏览器和目标元素，并打开可显示可追溯性检验员的 详细信息”选项卡，以帮助间隙分析，因为它显示了所有内容与元素相关的元素和细节，例如标记值；例如，如果目标架构中缺少架构构建块（ABB），您可以查看其他哪些流程和任务依赖于此 ABB，以及哪些流程受到影响，这也可以帮助您决定是否必须加入 ABB增强了目标架构

间隙元素

创建一个间隙到模型的间隙识别的元素

1. 右键单击单元并选择 创建间隙元素”选项。将显示 浏览项目”对话框。
2. 选择要在其中创建间隙元素的包，然后单击确定按钮。在间隙包中创建A元素并显示其 属性”对话框；输入元素名称和其他必需的属性。
3. 选择 属性”对话框的 属性”选项卡，设置 间隙元素标记值标记值”下的标记值。
4. 点击确定按钮。所选单元矩阵中显示间隙元素的名称和类别。

间隙元素标记值

如果您打算使用模型中间隙的元素，请右键单击 丢失/消除”列或 删除/元素”行中的相应单元，然后选择 链接到现有间隙”选项。将显示 选择分类器”对话框，从中选择现有的间隙元素。

创建元素间隙后，您可以右键单击其单元并从以下上下文菜单选项中进行选择：

- 编辑间隙元素”打开间隙元素的 属性”对话框并编辑其属性
- 'Locate in Project' 查找并突出浏览器浏览器窗口中的间隙元素
- 'Remove元素间隙单元元素元素包中）
- '删除间隙元素' 删除模型中的元素；此操作无法撤消

标记值	描述
ID	架构object的唯一标识符。
拥有着	架构object的所有者。
源	收集信息的位置/源。
类别	间隙的分类。这可以具有以下任一值： • 故意淘汰 • 无意消除 • 新的——待生产或开发 • 有待加强
RefBaseline架构	基线架构元素的工件的间隙。如果间隙是缺少的元素，那么该标签对于该标签的工件是基线的。
RefTarget架构	与工件相关联的间隙目标架构的元素。如果目标间隙的新工件是该目标架构具有新的目标工件。

差距分析矩阵**Profiles**

在差距分析矩阵上，您可以创建和管理配置文件以保存目标架构和刻板印象的常用组合。

要处理差距分析矩阵配置文件，请单击矩阵右上角的选项按钮。将显示A子菜单，列出以下选项：

- 创建当前矩阵设置的配置文件
- 更新 配置文件”字段中当前选择的配置文件
- 删除当前选择的个人资料

配置文件”字段下拉列表显示所有保存的差距分析矩阵文件。

TOGAF标记值

TOGAF 广泛使用标记值将自定义属性分配给特定于特定于各种元素的 TOGAF。创建或查看 TOGAF模型时，建议您始终保持属性窗口停靠且可见，并展开 TOGAF 部分。

同步标记值

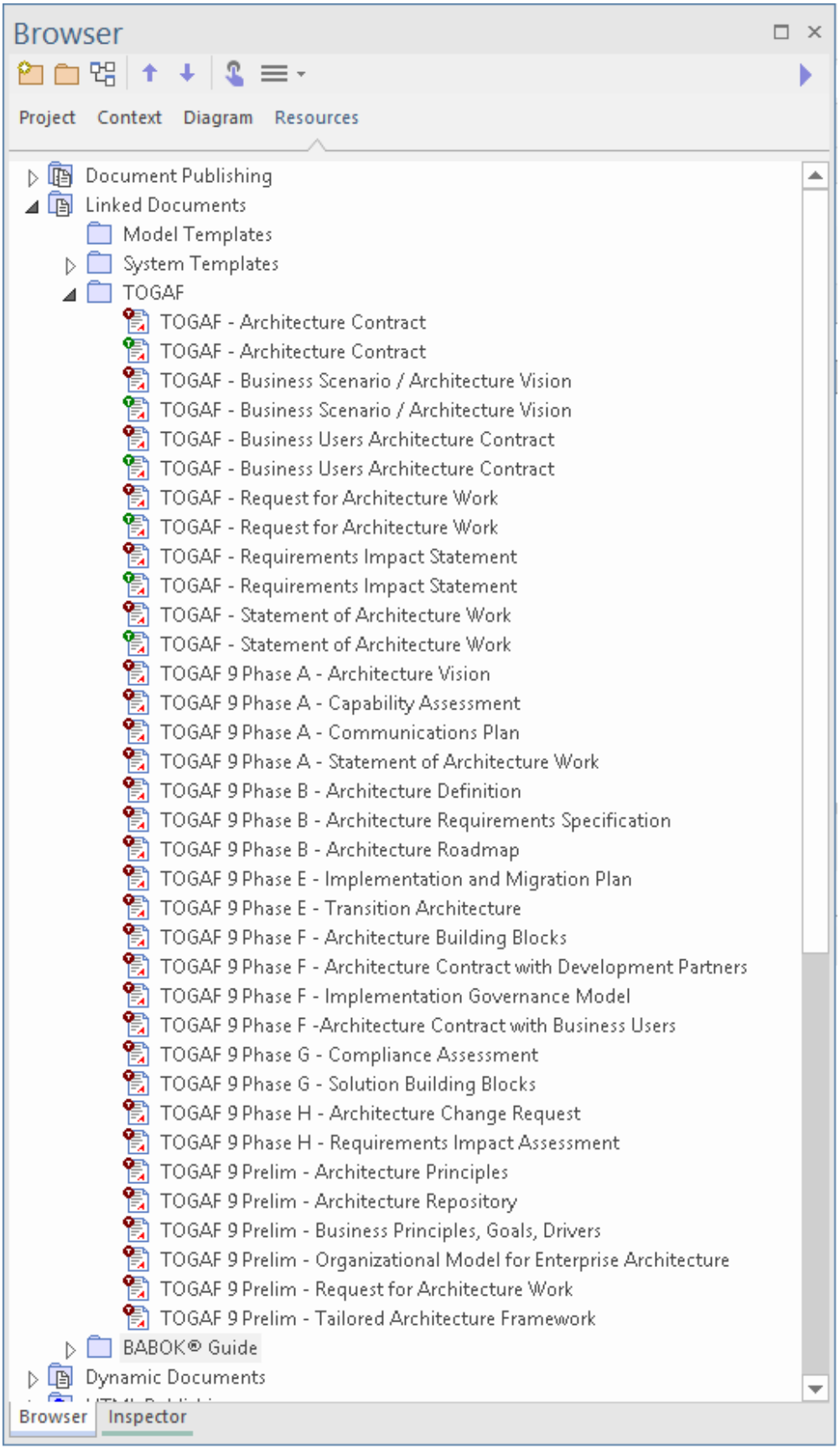
有时您可能需要在模型中需要它们的所有元素中添加缺失的标记值，例如：

- 每当您通过任何方式创建新元素时，除了直接从 TOGAF工具箱页面中删除元素
- 在使用新版本技术之前，将现有模型中元素的标注标记值更新为最新版本的TOGAF配置文件

您可以使用工具箱图表页面中图标上的 同步构造型”选项来执行此工具箱。

TOGAF 链接文档模板

Enterprise Architect包含一组特定于 TOGAF 的链接文档模板。



您还可以从 从模板新建链接文档”对话框的下拉列表中选择这些模板；向下滚动到 技术模板”列表。

链接文档模板由 The Open Group 提供，视此文本是否显示在模板的任何文档中而定：

“Open Group TOGAF 9 模板和示例。

架构开发方法 (ADM)

TOGAF 的关键仍然是一种可靠、实用的方法——TOGAF架构开发方法 (ADM)——用 定义业务需求和开发满足这些需求的架构，应用 TOGAF 的元素和组织可用的其他架构资产。

TOGAF 体现了企业连续体的概念，以反映架构开发过程中的不同抽象级别。通过这种方式，TOGAF 促进了不同层次参与者之间的理解和合作。它为与上下文ADM 结合使用多个框架、模型和架构资产提供了时间。通过企业连续统一体，鼓励架构师利用除 TOGAF 基础架构之外的所有其他相关架构资源和资产来开发特定于组织的IT架构。

关于 ADM 的要点

ADM 在整个过程中、阶段之间和阶段内迭代；对于 ADM 的每次迭代，必须对以下方面做出新的决定：

- 企业覆盖范围待定
- 待定义的细节级别
- 目标时间范围的范围，包括任何中间时间范围的数量和范围
- 要在组织的企业连续体中利用的架构资产，包括：
  - 在企业内 ADM 周期的先前迭代中创建的资产
  - 行业其他地方可用的资产（例如其他框架、系统模型和垂直行业模型）

这些决定必须基于对资源和能力可用性的实际评估，以及从所选择的架构工作范围中可以实际预期为企业带来的价值。

作为一种通用方法，ADM 旨在被广泛不同地区的企业使用，并应用于不同的垂直部门/行业类型。因此，它可以（但不一定必须）根据特定需求进行定制。例如，它可以用于：

- 与另一个框架的一组可交付成果相结合，这些成果更适合特定组织；许多美国联邦机构已经开发了单独的框架来定义特定于其特定部门需求的可交付成果
- 与著名的 Zachman 框架相结合，这是一个出色的分类方案，但缺乏公开可用的、明确定义的方法

ADM 阶段

架构开发方法 (ADM) 有十个阶段，如此处所述。 The Open Group 网站上提供的相文档中提供了每个阶段的方法和完成描述，以确定每个相的目标、输入、步骤和输出。

初步相：框架与原则

初步相是关于在相关企业中定义 我们在哪里、做什么、为什么、谁和如何做架构”。主要方面有：

- 定义企业
- 识别组织时间中的关键驱动因素和上下文
- 定义架构工作的要求
- 定义将告知任何架构工作的架构原则
- 定义要使用的框架
- 定义管理框架之间的关系
- 评估企业架构成熟度

相A ：架构愿景

架构愿景始于从赞助组织收到架构工作请求到架构组织。在此相，您将定义架构范围、如何创建愿景并获得批准。

相B ：业务架构

业务架构是必须进行的第一个架构活动，如果在其他组织流程（例如企业规划、战略业务规划或业务流程再造）中尚未满足的话。

相C：信息系统架构

在此相，您将开发信息系统架构，包括数据和应用程序架构。针对每个架构域分别给出了相C的详细步骤：

- 数据架构
- 应用程序架构

相D ：技术架构

技术架构相的步骤是：

- 选择参考模型、视点和工具
- 基线技术架构描述
- 开发目标技术架构描述
- 执行差距分析
- 定义路线图组件

- 解决整个架构的影响
- 进行正式的利益相关者审阅
- 敲定技术架构
- 创建架构定义文档

相E ：机遇与解决方案

在机遇和解决相中，您确定了变化的参数、沿途的主要阶段以及从当前环境转移到目标的顶级项目。

相F：移民规划

在迁移计划相，您将各种实施项目按优先顺序排序。活动包括评估各种迁移项目的依赖性、成本和收益。

相G：实施治理

在实施治理相，您将所有信息汇集在一起，以成功管理各种实施项目。

相H：架构更改管理

在架构更改管理相，您可以为新的企业架构基线建立一个架构变更管理流程。

ADM架构需求管理

ADM 由架构需求管理流程持续驱动。

TOGAF企业连续体

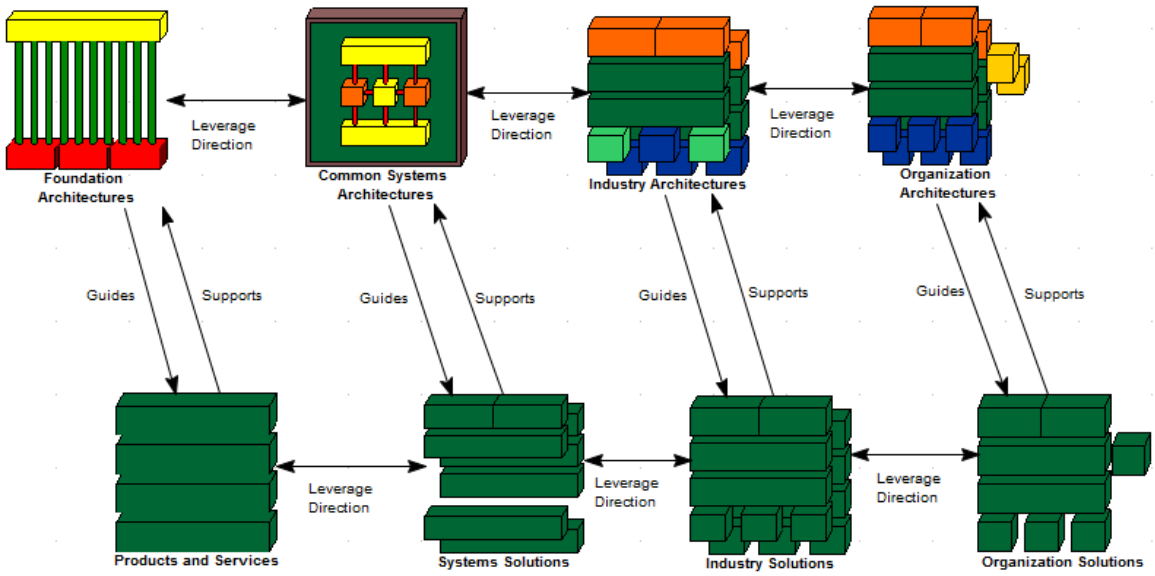
最简单的做法是将企业架构模式所有存在于企业和整个 IT架构中并且企业认为本身可用于为企业开发架构。

企业内的资产”的示例是可重复使用的先前架构工作的可交付成果。

整个 IT 行业的资产”的示例是存在并不断出现的各种行业参考模型和架构模式，包括：

- 高度通用，例如 TOGAF 自己的技术参考模型（TRM）
- 特定于 IT 的某些方面，例如 Web 服务架构或通用可管理性架构
- 特定于某些类型的信息处理，例如电子商务或供应链管理
- 特定于某些垂直行业；例如，由 TMF（电信部门）、ARTS（零售）或 POSC（石油技术）等垂直联盟生成的模型

Enterprise Architect对企业Continuum 的支持由企业图表图和相应的图表工具箱页面提供。启动器模型由TOGAF企业连续体的接口组成。



当您双击架构连续体或解决方案连续体元素时，将显示企业连续体图。图表工具箱提供了架构构建块和解决方案组件以及适当的关系连接器。

支持联邦企业架构Framework

TOGAF 提供特定于联邦企业架构框架 (FEAF) 的图表和工具箱页面。它还提供 FEAF 性能参考模型和技术参考模型的 开箱即用”模型。

要打开 FEAF-PRM 和 FEAF-TRM 模型：

1. 创建一个新的Enterprise Architect项目文件，然后单击顶层包。
2. 选择功能区选项 设计>包>模型构建器”。
3. 在模型构建器对话框中，选择 企业架构> TOGAF”蓝图和所需的 FEAF模式。
4. 单击创建模型按钮。

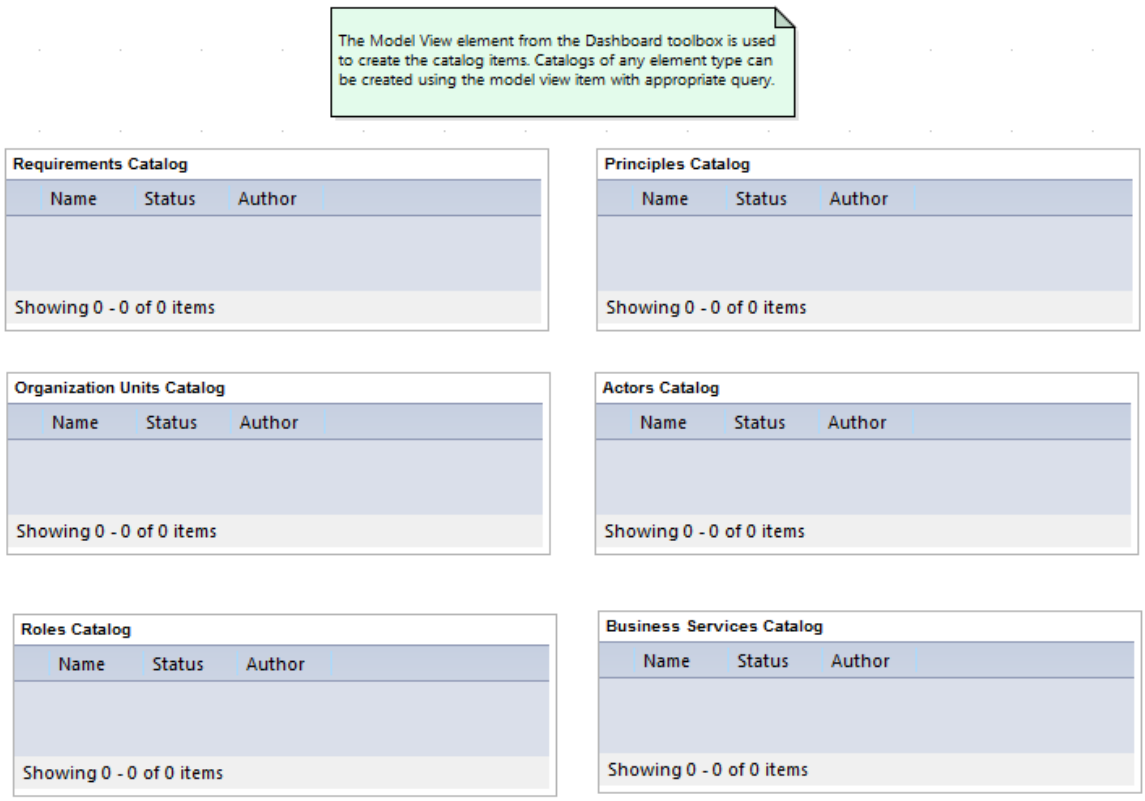
这些 TOGAF工具箱页面为 FEAF 提供特定支持：

- 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 目录

Enterprise Architect使用 TOGAF-Catalog模型模式帮助您创建模型目录工件。在模型构建器中选择此模型模式会生成一个模板模型，您可以在其中创建 TOGAF 特定的目录：

- 演员
- 业务Services
- 组织单位
- 原则
- 需求和
- 角色



每个模型视图都会列出整个模型中相应类型的所有对象。

或者，您可以使用图表工具箱 仪表板”页面中的模型视图元素在图表中创建 TOGAF 目录。

更多信息

Sparx Systems Enterprise Architect无缝集成了The Open Group Architecture Framework (TOGAF) ，为用户提供了基于开放标准的强大建模环境。TOGAF 以其实用且经过验证的企业架构开发方法而闻名，它提供 TOGAF架构开发方法 (ADM) 作为定义业务需求和定制架构的可靠方法。

借助 TOGAF 的企业连续体概念，架构师可以利用各种框架、模型和资产以及 TOGAF ADM，促进协作并创建特定于组织的 IT 架构。这种集成使用户能够在Enterprise Architect功能丰富的环境中充分利用 TOGAF 的潜力，确保有效的企业架构开发和维护。为了进一步了解 TOGAF，鼓励用户探索以下链接。

Unified Profile for DoDAF/MODAF (UPDM)

UPDM（DoDAF-MODAF 的统一配置文件）提供了一个UML配置文件，该配置文件扩展了Enterprise Architect的能力，为建模系统和企业架构提供标准方法以支持 DoDAF 和 MODAF。

DoDAF是Department of Defense架构框架(USA)的缩写； MODAF 是 Ministry of Defence架构框架（UK）的缩写。

讨论

此处描述的主题提供了在Enterprise Architect中使用 UPDM 的介绍和程序说明。

部分	内容
欢迎 	本节介绍了 UPDM，并解释了可用于框架的支持及其使用的系统要求。
许可版权和商标 	这些主题包含定义MDG 技术用于 UPDM 和Enterprise Architect的正式文档。
使用 UPDM 	使用 UPDM，了解模型结构、模板、图表类型等。
验证模型 	了解如何为 UPDM 开发和配置模型验证。

简单的介绍

欢迎使用Sparx Systems Enterprise Architect中的 UPDM 2.0 配置文件。

此UML配置文件扩展了Enterprise Architect的功能，以支持为 DoDAF 和 MODAF (UPDM)架构模型创建统一配置文件。 DoDAF是Department of Defense架构框架(USA)的缩写； MODAF 是 Ministry of Defence架构框架（UK）的缩写。

UPDM 配置文件为支持 DoDAF 和 MODAF 的建模系统和企业架构提供了标准方法。它提高了架构建模工具之间架构数据的互操作性，增强了架构数据的重用性，并改善了 DoDAF 和 MODAF 利益相关者之间的沟通。

UPDM 已与Enterprise Architect终极版和统一版集成； MDG 技术可以单独购买，与Enterprise Architect专业或企业Editions 一起使用。

该技术基于 DoDAF-MODAF (UPDM) 版本1 .0 的统一配置文件。 UPDM 1 .0 基于 DoDAF 版本1 .5 和 MODAF 版本1 .2。可以从物件管理组 (OMG) 网站获取配置文件的完整详细信息，包括最新规范。

开始

有关如何使用 UPDM 的说明，请参阅主题开始使用 *UPDM*和使用 *UPDM* 。

UPDM 支持

Enterprise Architect的注册用户可以使用与Enterprise Architect本身完全相同的方式来获得对 UPDM 配置文件的技术支持。

UPDM系统需求

UPDM 配置文件 2.0 版在以下环境下运行：

操作系统

- 窗口10
- 窗口8
- 窗口7
- 窗口2008服务器
- 窗口2003服务器
- 窗口XP Service Pack 2

Enterprise Architect版本

- Enterprise Architect 9.0 或更高版本

许可版权和商标

对于 UPDM 的Sparx Systems MDG 技术，本主题提供：

- 版权声明
- 软件产品许可协议（或最终用户许可协议）和
- 用户接口和文档中引用的其他产品的商标的确认

MDG 技术for UPDM 版权声明

此法律信息涉及Enterprise Architect和任何需要版权所有权声明的第三方代码工具的版权所有权。

版权所有 © 2010 - 2022 Sparx Systems Pty. Ltd. 保留所有权利。

Sparx Systems Pty. Ltd.

99 Albert St,

Creswick, Victoria 3363,

澳大利亚

电话：+61 (3) 5345 1140

传真：+61 (3) 5345 1104

支持电子邮件： support@sparxsystems.com

销售电子邮件： sales@sparxsystems.com

网站： sparxsystems.com

UPDM 软件产品许可协议的MDG 技术

本软件产品许可协议涉及为 UPDM 单独购买的MDG 技术，用于Sparx Systems Enterprise Architect的企业版和专业版。如果 UPDM 的MDG 技术与Enterprise Architect的终极版和统一版集成，则它包含在 Enterprise Architect 的 Sparx Systems Enterprise Architect Modelling Tool。

MDG 技术for UPDM, Enterprise Architect MDG插件

,版本2.0。

重要 - 请仔细阅读：本最终用户许可协议（“EULA”）是您作为被许可人与 SPARX 之间就上述软件产品达成的法律协议。通过安装、复制或以其他方式使用软件产品，您同意受本 EULA 条款的约束。如果您不同意本 EULA 的条款，请立即删除未使用的软件产品。

您使用本软件即表示您接受本 EULA 和保修。

定义

在本最终用户许可协议中，除非出现相反意图：

- “EULA”是指本最终用户许可协议
- A SPARX”是指Sparx Systems Pty Ltd A.C.电话号码N 034 546
- 被许可人”是指您或您代表其接受 EULA 的组织（如果有）
- “UPDM 的MDG 技术注册版”是指软件产品的版本，可从以下网站购买： https ://sparxsystems.com/updm/purchase.html ，经过 30 天的免费评估期
- 软件产品”或 软件”是指 UPDM 的MDG 技术，包括计算机软件和相关媒体和印刷材料，可能包括在线或电子文档
- 支持服务”是指 SPARX 提供的基于电子邮件的支持，包括有关软件产品使用的建议、错误调查、修复、模型维修（如果适用）以及一般产品支持
- “SPARX 支持工程师”是指提供在线支持服务的 SPARX 员工
- 试用版”是指软件产品的版本，可在三十 (30) 天内免费用于评估目的

授予许可

根据本 EULA 的条款，您被授予以下权利：

- 在单台计算机上安装和使用软件产品的一个副本，或代替它的同一操作系统的任何先前版本；作为安装本软件产品的计算机的主要用户，您可以制作第二份副本，供您在家或便携式计算机上独家使用
- 在存储设备（例如网络服务器）上存储或安装软件产品的副本，仅用于通过内部网络安装或运行软件产品；如果您希望增加有权同时访问软件产品的用户数量，您必须通知 SPARX 并同意支付额外费用
- 制作软件产品的副本仅用于备份和存档目的

评估许可证

试用版不是免费软件。根据本协议的条款，您在此获准在三十 (30) 天内免费使用本软件产品进行评估。

三十 (30) 天到期后，必须从计算机中删除软件产品。在 30 天评估期后未注册使用软件产品违反了澳大利亚、 U 。 S 。和国际版权法。

SPARX 可根据要求自行决定延长评估期。

如果您选择在 30 天评估期后使用软件产品，则必须购买许可证（如https中所述）。支付许可费后，您将收到有关在何处下载软件产品注册版本的详细信息，并将通过电子邮件向您提供合适的软件 密钥”。

附加权利和限制

您在此承诺不出售、出租、租赁、翻译、改编、改变、修改、反编译、反汇编、逆向工程、创建衍生作品、修改、再许可、出借或分发软件产品，除非本协议明确授权最终用户许可协议。

您进一步承诺不复制或分发许可证密钥代码，除非得到 SPARX 的明确书面许可。

如果购买的软件产品是学术版，您承认该许可仅限于在教育上下文使用，无论是用于自学还是在注册的教学机构中使用。未经 SPARX 明确书面许可，不得将学术版用于生产商业软件产品或在商业环境中使用。

任务

如果您向受让人提供本 EULA 的副本和所有其他文件（包括所有权证明），您只能将您在本 EULA 下的所有权利和义务转让给另一方。然后您的许可证将被终止。

终止

在不损害任何其他权利的情况下，如果您未能遵守条款和条件，SPARX 可以终止本 EULA。终止后，您或您的代表应销毁软件产品及其所有组件的所有副本，或按照 SPARX 指示的方式退回或处置此类材料。

保证和责任

保证

SPARX 保证：

- 自收到之日起九十 (90) 天内，软件产品将基本按照随附的书面材料执行，并且
- SPARX 提供的任何支持服务应与 SPARX 提供给您的适用书面材料中的描述基本一致，并且 SPARX 支持工程师将做出商业上合理的努力来解决与软件产品相关的任何问题。

除外条款

在法律允许的最大范围内，SPARX 对其自身和包含在软件产品中的任何软件供应商不承担任何直接或间接针对您或招致或遭受的所有索赔、费用、损失、损害和成本的责任（包括但不限于由于以下原因造成的成本、利润和数据损失）：

- 您对软件产品的使用或误用；
- 您无法使用或获得对软件产品的访问权；
- SPARX 或其员工、承包商或代理，或软件产品中包含的任何软件供应商在履行 SPARX 在本 EULA 下的义务方面的疏忽；或者
- 任何一方出于任何原因终止本 EULA。

局限性

软件产品和任何文档均按 原样”提供，以及以任何方式与本 EULA 的主题或本 EULA 相关的所有明示、暗示、法定或其他形式的保证，包括但不限于以下方面的保证：质量;健康;适销性；正确性；准确性;可靠性;符合您或

任何其他要求的任何描述或样品；不间断使用；遵守任何相关法律；不包括错误或无病毒。如果任何法律在本 EULA 中暗示任何条款，并且该法律避免或禁止合同中排除或修改此类条款的条款，则该条款应被视为包含在本 EULA 中。但是，如果法律允许，SPARX 对任何违反该条款的责任应仅限于 SPARX 在退回软件产品和收据副本时选择以下任何一项或多项：

- 如果违规与软件产品有关：
- 软件产品的更换或等效软件产品的供应；
- 此类软件产品的维修，或更换软件产品或购买同等软件产品的费用的支付；或者
- 支付修理软件产品的费用。
- 如果违规涉及与软件产品相关的服务：
- 再次提供服务；或者
- 支付再次提供服务的费用。

商标

本 EULA、软件产品或随附文档中使用的所有产品和公司名称可能是其相应所有者的商标。它们在本 EULA 中的使用旨在遵守相应的指南和许可证。

适用法律

本协议应根据维多利亚州的状态联邦法律解释。

商标确认 - UPDM

微软的商标

- 微软®
- 视窗®

OMG 的商标

- 天啊™
- 物件管理组™
- UML™
- 统一建模Language™

使用 UPDM

UPDM是国防部架构框架（DoDAF）和国防部架构框架（MODAF）的统一配置文件。 UPDM 是一个物件管理组 (OMG) 倡议；该规范可从 OMG 网站获取。

您可以在Enterprise Architect中执行 UPDM 建模，使用以下功能：

- UPDM配置文件，定义用于UPDM建模的刻板UML元素
- 每个 UPDM 视图的自定义图表类型
- 针对每种 UPDM 图表类型自定义图表工具箱包页面，可轻松访问该类型图表上使用的元素
- 模型生成器中的选项可用于为每个 UPDM 视图导入模板包，并提供视图的简要描述以及对建模者的期望
- 刻板元素的快速链接可指导您在元素之间创建正确的关系
- 验证模型规则，可用于检查模型的正确性
- 用于显示元素之间关系关系矩阵配置文件
- 模型视图可帮助您快速导航模型更轻松地找到特定图表
- 词汇A导入，其中的项目描述每个 UPDM 刻板印象，以便于参考
- 标记值，可用于输入特定于 UPDM 元素的元数据
- 示例模型展示了一个典型的 UPDM 问题及其解决方案，使用Enterprise Architect实现

开始使用 UPDM

当您安装Enterprise Architect的统一或终极版时，UPDM 配置文件已完全启用并可以使用。

如果您有Enterprise Architect的企业版或专业版，您可以单独购买和安装 UPDM 的MDG 技术；一旦您为 UPDM 输入了MDG 技术的注册密钥，它就会自动在Enterprise Architect中可用并集成到统一版和终极版中。

访问MDG 技术

1. 创建一个新的Enterprise Architect项目文件，然后单击顶层包。
2. 选择功能区选项 设计>包>模型生成器"。 将显示模型构建器。
3. 在模型构建器对话框中，选择 系统工程> UPDM"蓝图和 "UPDM Frameworks"模 组；选择 "DoDAF Framework"模 或"MODAF Framework"模 。
4. 单击创建模型按钮。

浏览器窗口中将创建A新的基本 DoDAF 或 MODAF模型。

UPDM中的模型构建器

您可以使用从Enterprise Architect模型生成器中选择的模板在项目中创建 UPDM 模型。

访问

功能区	开始> 个人>模型构建器 设计>包>模型生成器
上下文菜单	浏览器窗口 |右键包|模型（模式库）>模型模式
键盘快捷键	Ctrl+Shift+M

注记

- 在模型生成器对话框中，单击 <perspective name> 按钮并选择 系统Engineering > UPDM”
- 展开 ŏUPDM 框架”组或 ŏDoDAF”或“MODAF”组之一，然后单击该组中所需的模式
- 点击创建模型按钮，生成项目中相应的UPDM模型结构

UPDM 扩展菜单

您可以使用 UPDM技术菜单对 UPDM模型执行各种任务。

访问

功能区	特定>技术> UPDM 2.0
上下文菜单	右键单击包、图表或元素|特定| UPDM 2.0

选项

选项	行动
同步标记值	为模型中需要它们的所有元素添加缺失的标记值。
导入词汇表	导入信息导入Enterprise Architect Glossary。
导入图片	将替代图像（在导入框架图中使用）导入当前模型。 您可以使用这些图像来装饰您自己的模型（选择图表object ，右键单击并选择外观|选择替代图像”），或者您可以设计自己的模型。
帮助	显示此帮助主题。
关于	显示您正在使用的 UPDM 的MDG 技术版本。版本号的格式为1 .0.001，其中1 .0 是受支持的 UPDM 规范的版本，而 001 是增量内部版本号。

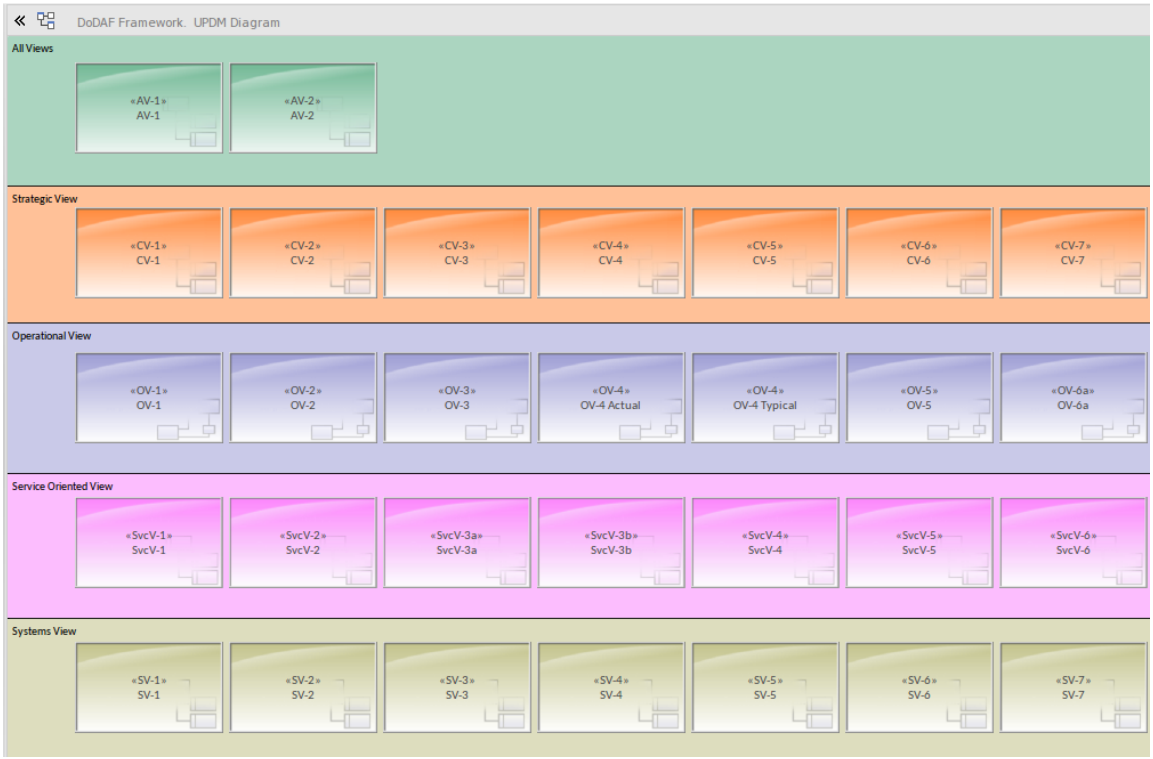
UPDM 框架图表

在开发和分发模型时，拥有一个包含模型中所有重要信息的超链接的首页图很有用。这就是两个 UPDM 框架图（一个用于 DoDAF，一个用于 MODAF）的目的，它们是为每组视图使用颜色编码的泳道创建的。您可以拖放到这些图表上：

- 包，充当他们拥有的文档的超链接
- 文档工件
- 任何指向其子图的复合元素
- 指向自定义的超链接#
  查询、关系矩阵配置文件或外部文件

创建一个 UPDM 框架图表

1. 打开模型生成器对话框（Ctrl+Shift+M），然后单击<视角名称>按钮并选择 "系统Engineering > UPDM"。
2. 展开 "UPDM 框架"组并单击所需的模式，"DODAF 框架"或"MODAF 框架"。
3. 点击创建模型按钮，在您的项目中生成相应的UPDM模型结构。



编辑泳道

您可以添加、删除和修改框架图上的泳道。 选择 "设计>图表>管理>泳道"。
要更改泳道的宽度，请使用鼠标拖动其边界。

改变外观

框架图上的每个包、文档和超链接都有一个替代图像。要将这些图像加载到您的模型中，请选择 设置 >参考> 图像”选项。

如果要将自己的位图图像应用到 UPDM 元素，则必须首先将图像导入模型，同时使用 设置 >参考> 图像”选项。然后您可以选择元素并按 Ctrl+Shift+W 将替代图像添加到元素，或者您可以应用自己的构造型将形状脚本应用到元素。例如，您可以使用此形状脚本定义一个构造型：

```
主要形状
{
v_align="中心";
h_align="中心";
defSize(90,70);
image("myBitMap.bmp",0,0,100,100);
printWrapped("#name#");
}
```

图表类型

UPDM 在Enterprise Architect中引入了许多自定义图表类型。这些图表类型大部分是扩展的UML图表。打开 UPDM 图表时， Enterprise Architect会自动打开适合该图表类型的 UPDM图表工具箱页面。

您可以使用模型生成器生成的 UPDM 图，或者创建一个新的 UPDM 图。

访问

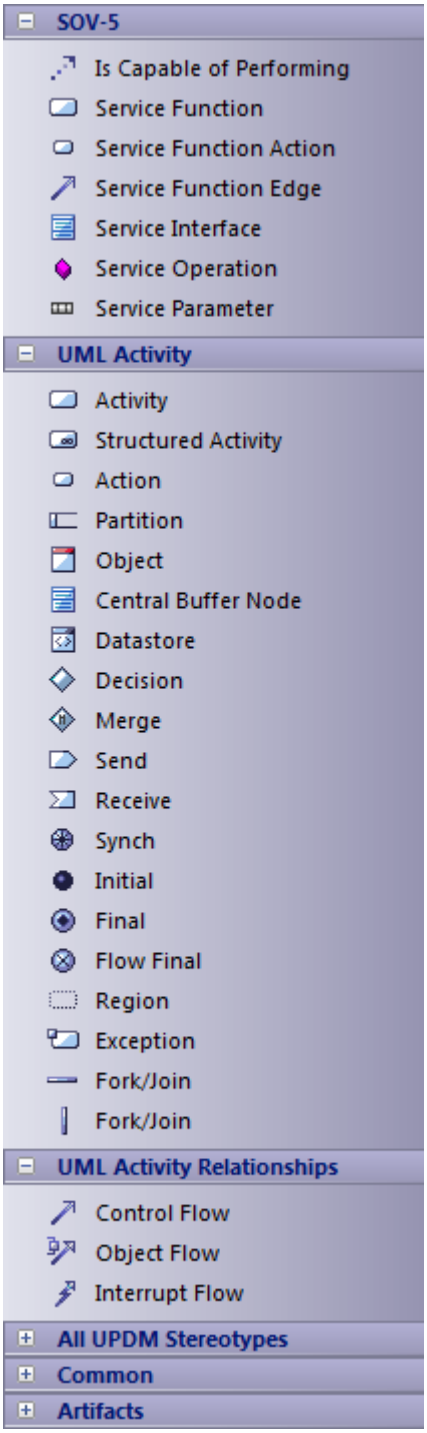
功能区	设计>图表> 添加图表
上下文菜单	浏览器窗口 |右键单击包|添加图表

注记

- 在 图表生成器”对话框中，在 选择自”面板中选择 “UPDM”，并在 图表类型”面板中选择适当的图表类型
- 单击 创建图表”按钮打开图表视图其中显示空图表

UPDM工具箱Pages

当您打开一个图表时， Enterprise Architect会打开对该特定图表类型最有用的图表工具箱页面。对于 UPDM 图表， Enterprise Architect会打开工具箱页面，其中包含适合图表所属的特定视图的元素和关系，以及图表类型的页面。例如，如果您打开 SOV-5活动图， Enterprise Architect会打开 "SOV-5元素"页面、 "UML活动"页面和 "UML活动关系"页面。



此外，无论打开哪个图表，图形工具箱的 "公共元素"和 "公共关系"页面以及各种全局任务页面图表可用。

如果隐藏了默认的工具箱页面，又想找回来，只需切换到首页，回到当前图表，当前图表类型的所有默认工具箱页面都会重新打开。

所有 UPDM构造型

为方便起见，提供了一个工具箱页面，其中包括图表配置文件中的每个原型，按字母顺序列出。如果您不记得构造型出现在哪个上下文相关的工具箱页面中，只需转到 所有 UPDM构造型”工具箱页面即可。要使此页面始终可用，请执行以下任一操作：

- 选择 特定>技术>管理技术”功能区选项，在表中选择 UPDM技术”，然后单击 设置活动”按钮，或
- 从默认工具工具栏上的列表框中选择 UPDM 2.0”

UPDM构造型

实际测量集

A组或一组测量值；用于 AV-3、OV-3、SV-6 和 SV-7。

扩展：

- 物件

约束：

- 分类器必须是一个MeasurementSet

使用：

- 按 Ctrl 并从浏览器窗口拖动一个 MeasurementSet元素以创建实例，或从工具箱中拖放一个图表并按 Ctrl+L 设置分类器；设置运行状态并为每个分类器的属性输入实际值

实际组织

一个实际的特定组织作为一个组织类的实例；用于 AcV-1、OV-4、StV-5、TV-1 和 TV-2。

扩展：

- 物件

概括：

- 实际组织资源

约束：

- 分类器必须是一个组织

使用：

- 按下 Ctrl 并从浏览器窗口拖动一个组织来创建一个实例，或者从工具箱图表按下 Ctrl+L 来设置分类器
- 可以有一套'ratifiedStandards'（标准）
- 可以是 'responsibleFor' 一组 ActualProject
- 可以是 ActualOrganizationRelationship 的客户和/或供应商
- 可以是 OperationalActivity 的 OwnsProcess 依赖项的客户端

实际组织关系

两个实际组织资源（组织或职位）之间A关系；用于OV-4。

扩展：

- 信息流

约束：

- 供应商必须是 ActualOrganizationalResource（ActualOrganization 或 ActualPost）
- 客户必须是 ActualOrganizationalResource（ActualOrganization 或 ActualPost）
- 实现一个 ResourceInteraction

实际人

履行 ActualPost A指定个人；用于OV-4。

扩展：

- 物件

约束：

- 分类器必须是一个Person

使用：

- 按 Ctrl 并从浏览器窗口拖动一个Person以创建一个实例，或从工具箱工具箱中拖放一个图表并按 Ctrl+L 设置分类器
- 可以是 ActualPost 的 FillsPost 依赖项的客户端

实际邮政

一个实际的、具体的帖子，作为 Post类的一个实例；用于 AcV-1、OV-4 和 StV-5。

扩展：

- 物件

概括：

- 实际组织资源

约束：

- 分类器必须是一个帖子

使用：

- 按下 Ctrl 并从浏览器窗口中拖动一个 Post 来创建一个实例，或者从工具箱中拖放一个图表并按下 Ctrl+L 来设置分类器
- 可以负责一组ActualProject
- 可以是 ActualOrganizationRelationship 的客户和/或供应商
- 可以是 OperationalActivity 的 OwnsProcess 依赖项的客户端
- 可以是来自 ActualPerson 的 FillsPost 依赖项的提供者

实际项目

A时间限制地尝试创建一组特定的产品或服务；用于 AcV-1、AcV-2、StV-3、StV-5 和 SV-8。

扩展：

- 物件

约束：

- 分类器必须是一个项目

使用：

- 按 Ctrl 并从浏览器窗口中拖动一个项目来创建一个实例，或者从工具箱工具箱中拖放一个图表并按 Ctrl+L 来设置分类器
- 可以与另一个 ActualProject 进行聚合
- 可以有一组 'ownedMilestones'（类型 ActualProjectMilestone，包括 IncrementMilestone、OutOfServiceMilestone、NoLongerUsedMilestone 和 DeployedMilestone）

实际项目里程碑

项目中衡量进度的事件；用于 AcV-2、StV-3、StV-5 和 SV-8。

也见：IncrementMilestone、OutOfServiceMilestone、NoLongerUsedMilestone 和 DeployedMilestone。

扩展：

- 物件

专长：

- 增量里程碑
- 停止服务里程碑
- 不再使用里程碑
- 部署里程碑

约束：

- 分类器必须是一个项目里程碑

使用：

- 可以有一组关联的资源
- 可以是 MilestoneSequence 的客户/供应商

别名

用于定义元素的替代名称A注释；用于 AV-2。

扩展：

- 注记

约束：

- AnnotatedElement 必须是 UPDMElement

使用：

- 只需将 Quicklink NoteLink 从别名拖到带注释的元素

任意关系

代表在高级操作概念图中使用的连接的视觉指示。这些连接纯粹是视觉上的，不能与任何架构语义相关联；用于OV-1。

扩展：

- 依赖

约束：

- Client 和 Supplier 都必须是刻板的 ConceptRole

使用：

- 从 ConceptRole 拖动快速链接

架构描述

技术层面A系统规范，也提供了业务上下文；用于 AV-1。

扩展：

- 包

使用：

- 可以对 EnterprisePhase 进行 DefinesArchitecture 实现
- 可以对另一个 ArchitecturalDescription 有 ArchitecturalReference 依赖
- 可以用 ArchitectureMetadata注记注解

架构参考

断言一个架构描述引用了另一个；用于 AV-1。

扩展：

- 依赖

约束：

- 客户和供应商都必须是定型的 ArchitecturalDescription

使用：

- 从 ArchitecturalDescription 中拖动快速链接。

架构元数据

架构描述信息；用于 AV-1。

扩展：

- 注记

概括：

- 元数据

约束：

- AnnotatedElement 必须是 ArchitecturalDescription

使用：

- 从 ArchitecturalDescription 中拖动快速链接

能力

对企业能力A高层次规范；用于 AV-1、OV-2、SOV-3、StV-1、StV-2、StV-3、StV-4、StV-5、StV-6、SV-1 和 SV-9。

扩展：

- 类

概括：

- 预测主题

使用：

- 可以有一组关联的环境条件定型环境

- Capabilities可以由Capabilities（复合聚合）组成
- 能力可以依赖于能力（Dependency）
- 能力子类能力（概括）可以
- 可以是预测的供应商或客户（两者必须是相同的原型）（来自 SubjectOfForecast）

能力配置

A组物理和人力资源（及其交互）配置为提供一种能力；用于 OV-1、OV-2、OV-3、StV-3、StV-5、SV-1、SV-3、SV-9、SV-10a、SV-12、TV-1、TV-2 和AcV-2。

扩展：

- 类

概括：

- 资源
- 概念项目
- 演员
- 资源交互项
- 资源约束的主题
- 预测主题
- 系统元素
- 资源状态机的主题
- 资源交互项

专长：

- 系统节点

使用: 可以:

- 有一组相关的已部署里程碑，原型 DeployedMilestone
- 有一个可选的关联不再使用的里程碑，原型 NoLongerUsedMilestone
- 有一组相关的增量里程碑，定型的 IncrementMilestone
- 有一个可选的关联停止服务里程碑，定型的 OutOfServiceMilestone
- 由 StandardConfiguration注记注解
- 成为 ConceptRole 的类型（来自 ConceptItem）
- 有一组相关的里程碑，定型的 ActualProjectMilestone（来自 Resource）
- 成为实现能力的客户（来自资源）
- 成为提供能力的客户 对能力的依赖（来自资源）
- 有一个附加的 ResourceConstraint（来自 Resource，SubjectOfResourceConstraint）
- 成为预测依赖项的供应商或客户（两者必须具有相同的刻板印象）（来自 SubjectOfForecast）
- 拥有一个 ServicePoint（来自 Resource）
- 拥有一个 RequestPoint（来自 Resource）
- 拥有一个 ResourcePort（来自 Resource）
- 成为 ResourceInteraction 的源和目标（来自 Resource）
- 拥有一个 ServiceOperation（来自 Resource）
- 成为 KnownResource 的类型（来自 Resource）
- 成为 ResourceRole 的类型（来自 Resource）

- 对 PerformedActivity（函数或 OperationalActivity）有执行依赖（来自 Performer）

气候

表演者在其中表演A天气条件或天气条件的组合；用于 StV-2。
扩展：
- 类

概括：
- 环境类型

使用：
- 可以是 EnvironmentProperty 的类型

命令

断言一个组织资源命令另一个组织资源；用于 OV-4、SV-1 和 SV-10c。
扩展：
- 信息流

概括：
- 资源交互

约束：
- 源必须是一个组织资源
- 目标必须是一个组织资源

使用：
- 传递数据元素

兼容

将节点与位置相关联，以断言操作节点必须位于该位置；用于OV-2。
扩展：
- 依赖

约束：
- 客户端是一个节点
- 供应商是ReferredLocation（位置或物理位置）

使用：
- 从节点

权限

由知识、技能和态度定义的A组特定能力；用于 OV-4、SV-1 和 SV-9。
扩展：

- 类

概括：

- 预测主题

使用: 可以是:

- 预测依赖的供应商或客户（两者必须具有相同的原型）（来自 SubjectOfForecast）
- ProvidesCompetence 依赖项的供应商
- RequiresCompetence 依赖项的供应商

概念角色

A关系断言 ConceptItem 构成高级操作概念的一部分；用于OV-1。

扩展：

- 部件

约束：

- 类型是一个概念项

使用：

- 由 HighLevelOperationalConcept 拥有
- 可以是任意关系依赖项的供应商和客户

配置交换

节点之间交换的能力配置；用于 OV-2、OV-3 和 OV-6c。

扩展：

- 信息流

概括：

- 操作交换

约束：

- 源是一个节点（来自 OperationalExchange）
- 目标是一个节点（来自 OperationalExchange）

使用：

- 传达 CapabilityConfiguration

控件

ResourceInteraction A类型，其中一个资源控制另一个资源；用于 SV-1 和 SV-10c。

扩展：

- 信息流

概括：

- 资源交互

约束：

- 源是一个组织资源（组织或职位）
- 目标是 ManufacturedResourceType（ResourceArtifact 或 Software）

使用：

- 传递数据元素

数据元素

由系统管理或在系统之间交换的数据A正式表示；用于 OV-4、SV-1、SV-2、SV-4、SV-6、SV-10a、SV-10b 和 SV-11。

扩展：

- 类

概括：

- 资源约束的主题
- 资源交互项
- 系统元素
- 资源状态机的主题

使用：

- 可以有一个附加的 ResourceConstraint（来自 SubjectOfResourceConstraint）
- 可以有一组关联的已定义实体项
- 可以在控制或命令信息流上传达

数据交换

ResourceInteraction A DoDAF 别名。

扩展：

- 信息流

概括：

- 资源交互
- 系统元素

使用：

- 传达 ResourceInteractionItem（Energy、Post、Organization、CapabilityConfiguration、Software、ResourceArtifact 或 DataElement）

定义架构

在 ArchitecturalDescription 和 EnterprisePhase 之间建立关系；用于 AV-1。

扩展：

- 实现

约束：

- 客户是一个架构描述
- 供应商是 EnterprisePhase

使用：

- 从 ArchitecturalDescription 中拖动快速链接

定义

架构中元素A定义；用于 AV-2。

扩展：

- 注记

约束：

- 带注释的元素是一个 UPDMElement

使用：

- 从工具箱中删除并将 NoteLink 拖到任何 UPDM元素

部署里程碑

断言 ActualOrganizationResource 从特定时间点开始使用或计划开始使用 CapabilityConfiguration；在 StV-5 中使用。

扩展：

- 物件

概括：

- 实际项目里程碑

约束：

- 分类器必须是一个 ProjectMilestone（来自 ActualProjectMilestone）

使用: 可以:

- 拥有一组关联的（usedBy）ActualOrganizationalResource（ActualOrganization 或 ActualPost）
- 拥有一组关联资源（来自 ActualProjectMilestone）
- 成为 MilestoneSequence 的客户/供应商（来自 ActualProjectMilestone）

持久任务

企业认为对实现其目标至关重要的A行为，即企业所做工作的战略规范；在 StV-1 中使用。

扩展：

- 类

使用：

- EnterprisePhase协会的目标

活力

节点之间要交换的能量；用于 OV-2、OV-3、OV-5、SV-1、SV-4 和 SV-6。

扩展：

- 类

概括：

- 资源交互项
- 操作交换项目

使用：

- 在 EnergyExchange 信息流上传达

能源交换

指定节点之间需要交换能量A关系；用于 OV-2、OV-3 和 OV-6c。

扩展：

- 信息流

概括：

- 操作交换
- 操作元素

约束：

- 源是一个节点（来自 OperationalExchange）
- 目标是一个节点（来自 OperationalExchange）

使用：

- 传达一种类印象的能量

企业目标

架构所代表的企业A特定要求目标；在 StV-1 中使用。

扩展：

- 类

使用：

- 与一个 EnterprisePhase 有关联

企业阶段

架构所代表的企业A特定、必需的目标；用于 AV-1、StV-1、StV-2、StV-5 和 SV-9。

扩展：

- 类

专长：

- 终身企业

使用：

- 可以有一组关联（statementTasks）到 EnduringTask类
- 可以有一组关联（展览）到 Capability类
- 可以有一组关联（栖息）到环境类

- 可以有一组与 EnterpriseGoal类的关联（目标）
- 可以与 EnterpriseVision类有一组关联（愿景）
- 可以是 StructuralPart 或 TemporalPart 的类型
- 使命用例
- 可以是 DefinesArchitecture 实现的供应商

企业愿景

企业在一定时期内的总体目标；在 StV-1 中使用。
扩展：
- 类

使用：
- 与一个 EnterprisePhase 有关联

实体属性

属性A定义属性；用于 OV-7 和 SV-11。
扩展：
- 属性

使用：
- 由 EntityItem 拥有

实体项

感兴趣项目A定义（类型）；用于 OV-7 和 SV-11。
扩展：
- 类

约束：
- 拥有的属性必须是原型 EntityAttribute

使用: 可以:
- 归某数据模型所有
- 成为 EntityRelationship 的最终类型
- 有一组关联的（definedBy）数据元素
- 有一组关联的（代表）信息元素
- 在命令或控制信息流上传达

实体关系

断言两个 EntityItem 之间存在关系；用于 OV-7 和 SV-11。
扩展：

- 关联

约束：

- 任一端的任何object的类型都必须是原型 EntityItem

环境

企业存在或运作的条件A定义；用于 AV-1 和 StV-2。

扩展：

- 类

约束：

- 拥有的属性必须是 EnvironmentProperty

环境属性

断言环境具有一个或多个属性，例如气候、位置或光照条件；用于 StV-2。

扩展：

- 属性

约束：

- 类型必须是环境类型（LightCondition、Location、PhysicalLocation 或 Climate）

使用：

- 归环境元素所有

设备

用于在系统或环境中完成任务或函数A物理资源；用于 SV-1。

扩展：

- 部件

概括：

- 资源角色

约束：

- 类必须是一个组织资源（组织或帖子）
- 类型必须是 ResourceArtifact

使用：

- 可以对 Competence 有 RequiresCompetence 依赖项（来自 ResourceRole）
- 可以有一组关联（usedFunctions）到函数（来自 ResourceRole）

展品能力

断言一个节点需要有一个能力；用于OV-2。

扩展：

- 依赖

约束：

- 客户端必须是节点
- 供应商必须是能力

暴露

断言服务接口公开了一项功能。

扩展：

- 依赖

约束：

- 客户端必须是 ServiceInterface
- 供应商必须是能力

外部个人

由外部本体定义的个体；用于 AV-2。

扩展：

- 物件

使用：

- 可以是 SameAs 依赖项的提供者

外部节点

架构外部的操作节点；用于OV-2。

扩展：

- 类

概括：

- 节点
- 演员

使用: 可以:

- 拥有一个RequestPoint端口（来自节点）
- 拥有一个ServicePoint端口（来自节点）
- 成为 Capability 的 ExhibitsCapability 依赖项的客户端（来自节点）
- 对 PerformedActivity（函数或 OperationalActivity）有 Performs 依赖（来自 Performer）
- 对 ReferredLocation（PhysicalLocation 或 Location）有 CompatibleWith 依赖（来自节点）

外部类型

由外部本体定义A类型；用于 AV-2。

扩展：

- 类

使用：

- 可以是 SameAs 依赖项的供应商
- 任何概括元素都可以有一个外部类型

现场能力

已部署并完全实现A能力实例；用于 SV-2。

扩展：

- 物件

约束：

- 它的分类器必须是一个 CapabilityConfiguration

填报

断言 ActualPerson 填写 ActualPost；用于OV-4。

扩展：

- 依赖

约束：

- 客户必须是 ActualPerson
- 供应商必须是 ActualPost

预报

系统在项目里程碑处的实际或预测状态；用于 SV-9。

扩展：

- 依赖

专长：

- 技术预测

约束：

- 客户和供应商都是预测对象（标准、能力、能力、能力配置、组织、职位、资源工件或软件）
- 客户和供应商必须是 SubjectOfForecast 的同一专业化

函数

在执行它的资源的时间中指定的上下文；用于 OV-4、SV-1、SV-4、SV-5 和 SV-10a。

扩展：

- 活动

概括：

- 执行活动
- 系统元素
- 资源约束的主题

约束：

- 拥有的参数是 FunctionParameter

使用: 可以:

- 成为 Performs 依赖项的供应商（来自 PerformedActivity）
- 拥有ServiceOperationAction、FunctionAction和FunctionEdge
- 成为 OperationalActivity 的 ImplementsOperational 依赖项的客户端（来自 SystemsElement）
- 有一个附加的 ResourceConstraint（来自 SubjectOfResourceConstraint）

功能动作

调用需要执行的函数A调用行为动作；用于 SV-4。

扩展：

- 行动（调用行为）

专长：

- SystemFunction动作

约束：

- 活动是刻板的函数

使用：

- Ctrl+L 设置函数

功能边缘

通过函数对控制/对象的流进行建模；用于 SV-4。

扩展：

- 控制流

概括：

- 系统元素

专长：

- 系统功能边缘

约束：

- 源必须是 ServiceOperationAction
- 目标必须是一个 ServiceOperationAction

使用：

- 可以实现一个ResourceInteraction（右击|高级|实现信息流）

功能参数

代表函数的输入和输出；用于 SV-4。

扩展：

- 活动参数

约束：

- 类型必须是 ResourceInteractionItem（Energy、DataElement、CapabilityConfiguration、Organization、Post、ResourceArtifact 或 Software）

使用：

- 由一个函数拥有

高级操作概念

A通用的操作模型；用于OV-1。

扩展：

- 类

约束：

- 拥有的属性是 ConceptRole

使用：

- 可以有一套描述使命

托管软件

断言软件托管在 ResourceArtifact 上；用于 SV-1。

扩展：

- 部件

概括：

- 资源角色

约束：

- 拥有类必须是 ResourceArtifact
- 类型必须是软件

使用: 可以:

- 对能力有 RequiresCompetence 依赖项（来自 ResourceRole）
- 与 使用”函数有一组关联（来自 ResourceRole）

人力资源

职位或组织在 CapabilityConfiguration 中的角色；用于 SV-1。

扩展：

- 部件

概括：

- 资源角色

约束：

- 拥有类必须是 CapabilityConfiguration
- 类型必须是组织资源（组织或职位）

使用: 可以:

- 对能力有 RequiresCompetence 依赖项（来自 ResourceRole）
- 与 使用"函数有一组关联（来自 ResourceRole）

实现操作

系统元素与其实现的操作元素之间的关系；用于 SV-5。

扩展：

- 依赖

约束：

- 客户端必须是 SystemsElement (函数)
- 供应商必须是 OperationalElement (OperationalActivity)

增量里程碑

指示项目预计交付或已交付能力的时间点的实际项目里程碑；用于 AcV-2、StV-3 和 SV-8。

扩展：

- 物件

概括：

- 实际项目里程碑

约束：

- 分类器必须是一个 ProjectMilestone（来自 ActualProjectMilestone）

使用：

- 可以是 MilestoneSequence 依赖项的供应商或客户（来自 ActualProjectMilestone）
- 可以有一组关联的资源（来自 ActualProjectMilestone）
- 与 CapabilityConfiguration 有一组关联

信息元素

节点之间交换的信息；用于 OV-2、OV-3、OV-5、OV-6a、OV-6b 和 OV-7。

扩展：

- 类

概括：

- 操作交换项目
- 操作约束的主题
- SubjectOfOperationalStateMachine
- 操作元素

使用: 可以:

- 与（由）EntityItem 类有一组关联
- 在 InformationExchange 上传递 - 右键单击 >高级> 传递的信息项
- 有一个附加的 OperationalConstraint（来自 SubjectOfOperationalConstraint）
- 拥有 OperationalStateMachine（来自 SubjectOfOperationalStateMachine）

信息交换

指定节点之间交换信息的需要A关系；用于 OV-2、OV-3 和 OV-6c。

扩展：

- 信息流

概括：

- 操作交换

约束：

- 传达一个信息元素
- 源是一个节点（来自 OperationalExchange）
- 目标是一个节点（来自 OperationalExchange）

内部数据模型

PhysicalDataModel 的 DoDAF 别名；用于 SV-11。

扩展：

- 包

概括：

- 物理数据模型
- 数据模型

约束：

- 拥有 EntityItem 元素（来自数据模型）

已知资源

断言已知资源在架构中发挥作用；用于OV-2。

扩展：

- 部件

概括：

- 节点子

约束：

- 类型必须是资源（Post、Organization、CapabilityConfiguration、SystemsNode、Software 或 ResourceArtifact）
- 类必须是NodeParent（节点或LogicalArchitecture）（来自NodeChild）

光照条件

环境照明条件A规范；用于 StV-2。

扩展：

- 类

概括：

- 环境类型

使用：

- 可以是 EnvironmentProperty 的类型（来自 EnvironmentalType）

地点

可能发生操作的环境/场景A一般规范。示例包括 沙漠"、 北极"、 海上"；用于 OV-1 和 OV-2。

扩展：

- 类

概括：

- 推荐位置
- 概念项目
- 环境类型

使用: 可以是:

- 来自节点的 CompatibleWith 依赖的供应商（来自 ReferredLocation）
- 类型的类型（来自 ConceptItem）
- EnvironmentProperty 的类型（来自 EnvironmentalType）

逻辑架构

A复合结构模型，其部分是 NodeRoles、ProblemDomains 或 KnownResources；用于OV-2。

扩展：

- 类

概括：

- 父节点

使用：

- 可以拥有属性

逻辑数据模型

A正式数据结构的业务信息需求规范；用于OV-7。

扩展：

- 包

概括：

- 数据模型

约束：

- 拥有 EntityItem 元素（来自数据模型）

MapsToCapability

断言 StandardOperationalActivity 在某种程度上是能力的一部分；在 StV-6 中使用。

扩展：

- 依赖

约束：

- 客户端必须是 StandardOperationalActivity
- 供应商必须是能力

物资交换

节点之间交换的物资；用于 OV-2、OV-3 和 OV-6c。

扩展：

- 信息流

概括：

- 操作交换

约束：

- 源是一个节点（来自 OperationalExchange）
- 目标是一个节点（来自 OperationalExchange）

使用：

- 可以传达 ResourceArtifact 或 Software

测量

A类措施；用于 AV-3、OV-2 和 SV-7。

扩展：

- 属性

专长：

- 性能参数

使用：

- 由 MeasurementSet类拥有

测量集

A组或一组测量值；用于 AV-3、OV-3 和 SV-7。

扩展：

- 类

约束：

- 拥有的属性必须是测量

使用：

- 与 (measuredElement) UPDMElement 有一组关联
- 是 ActualMeasurementSet object的分类器

元数据

可以应用于架构中任何元素的注释；用于 AV-1。

扩展：

- 注记

专长：

- 架构元数据

里程碑序列

两个里程碑之间A关系；用于 AcV-2 和 SV-8。

扩展：

- 依赖

约束：

- 客户必须是 ActualProjectMilestone
- 供应商必须是 ActualProjectMilestone

使命

A人、组织或自治系统的任务目标；用于 AV-1、OV-1、OV-6a 和 OV-6b。

扩展：

- 用例

概括：

- 操作约束的主题
- SubjectOfOperationalStateMachine

使用：

- 由 EnterprisePhase 完成
- 可以附加 OperationalConstraint（来自 SubjectOfOperationalConstraint）
- 可以拥有 OperationalStateMachine（来自 SubjectOfOperationalStateMachine）

人民运动

OrganizationalExchange 的 MODAF 别名。

扩展：

- 信息流

概括：

- 组织交流

使用：

- 传达组织资源（组织或职位）

针线

记录节点之间交换信息的要求；用于 OV-2 和 OV-3。

扩展：

- 关联
- 连接器

概括：

- 操作元素

约束：

- End Types 必须是节点
- 最终角色必须是 NodePort
- 结束角色必须是 NodeChild（NodeRole、ProblemDomain、KnownResource）

使用：

- 实现 OperationalExchange - 在与 OperationalExchange 相同的两个元素之间创建一条需求线，然后右键单击需求线并选择 高级> 实现的信息流"

不再使用里程碑

断言 ActualOrganizationResource 从特定时间点停止使用或计划停止使用 CapabilityConfiguration；在 StV-5 中使用。

扩展：

- 物件

概括：

- 实际项目里程碑

约束：

- 分类器必须是一个 ProjectMilestone（来自 ActualProjectMilestone）

使用：

- 具有与 "noLongerUsedBy"ActualOrganizationalResource（ActualOrganization 或 ActualPost）对象的关联集
- 可以有一组关联的资源（来自 ActualProjectMilestone）
- 可以是 MilestoneSequence 的客户/供应商（来自 ActualProjectMilestone）
- 具有一组与 配置"CapabilityConfiguration 类的关联

节点

执行操作活动的逻辑实体；用于 OV-1、OV-2、OV-3、OV-5、OV-6a、OV-6b 和 OV-6c。

扩展：

- 类

概括：

- 演员
- 概念项目
- 父节点
- 操作约束的主题
- SubjectOfOperationalStateMachine
- 操作元素

专长：

- 操作节点

约束：

- 拥有的端口必须是 NodePort、RequestPoint 或 ServicePoint

使用: 可以:

- 对 PerformedActivity（函数或 OperationalActivity）有 Performs 依赖（来自 Performer）
- 成为 CompatibleWith 依赖于 ReferredLocation（Location 或 PhysicalLocation）的客户端
- 成为 ConceptRole 的类型（来自 ConceptItem）
- 拥有一个 RequestPoint 端口
- 拥有一个 ServicePoint 端口
- 成为 ExhibitsCapability 对 Capability 的依赖项的客户端
- 自己的 NodeChild（NodeRole、KnownResource、ProblemDomain）（来自 NodeParent）
- OperationalExchange（ConfigurationExchange、EnergyExchange、InformationExchange、MaterielExchange 或 OrganizationalExchange）信息流的源和目标
- 成为 Needline 协会的最终类型
- 有一个附加的 OperationalConstraint（来自 SubjectOfOperationalConstraint）
- 拥有 OperationalStateMachine（来自 SubjectOfOperationalStateMachine）
- 成为 NodeRole 的类型
- 自己的服务运营

节点端口

节点的A属性，它指定节点与其环境之间或节点与其内部部分之间的不同交互点。

扩展：

- 端口

约束：

- 类型必须是 OperationalExchangeItem（Post、Organization、ResourceArtifact 或系统）

使用：

- 归一个节点所有
- 可以是针线的末端

节点角色

用于连接一个父节点到它的子节点；用于 OV-2、OV-3 和 OV-6c。
扩展：
- 部件

概括：
- 节点子

专长：
- 问题域

约束：
- 类必须是节点
- 类型必须是一个节点

运营活动

A逻辑过程，独立于过程的执行方式进行指定；用于 OV-2、OV-3、OV-4、OV-5、OV-6a、OV-6b 和 SV-5。
扩展：
- 活动

概括：
- 执行活动
- 操作约束的主题
- 操作元素
- SubjectOfOperationalStateMachine

专长：
- 标准作业活动

约束：
- 拥有的参数必须是 OperationalParameter

使用: 可以:
- 成为 Performs 依赖项的供应商（来自 PerformedActivity）
- 成为 OwnersProcess 依赖项的供应商
- 成为 OperationalActivityAction 的活动/行为
- 成为 OperationalActivityEdge 的所有者
- 有一个附加的 OperationalConstraint（来自 SubjectOfOperationalConstraint）
- 成为 SupportsOperationalActivity 依赖项的供应商
- 拥有 OperationalStateMachine（来自 SubjectOfOperationalStateMachine）

OperationalActivityAction

调用要执行的活动A调用行为动作；用于OV-5。
扩展：
- 呼叫行为动作

约束：

- 活动/行为必须是 OperationalActivity

使用：

- 可以是 OperationalActivityEdge 的源或目标

OperationalActivityEdge

通过 OperationalActivity 对控制/对象的流进行建模；用于OV-5。

扩展：

- 控制流

概括：

- 操作元素

约束：

- 必须由 OperationalActivity 拥有
- 源必须是 OperationalActivityAction
- 目标必须是 OperationalActivityAction

使用: 可以:

- 拥有一组可实现的 OperationalExchange（ConfigurationExchange、EnergyExchange、InformationExchange、MaterielExchange 或 OrganizationalExchange）信息流
- 携带一组 OperationalExchangeItem（Post、Organization、ResourceArtifact 或系统）

操作约束

管理操作行为或属性A规则；用于OV-6a。

扩展：

- 约束

专长：

- 操作规则

约束：

- 约束元素必须是SubjectOfOperationalConstraint（OperationalActivity、节点、InformationElement或使命）

操作信息

用于运营事件跟踪的信息，该跟踪跟踪包含 OperationalExchange 的任何子类型；用于OV-6c。

扩展：

- 信息

概括：

- 操作元素

使用：

- 可以有一组它实现的 OperationalExchange（ConfigurationExchange、EnergyExchange、InformationExchange、MaterialExchange 或 OrganizationalExchange）信息流

操作节点

生产、使用或处理信息的操作架构的一个元素。

扩展：

- 类

概括：

- 节点

约束：

- 拥有的端口必须是 NodePort、RequestPoint 或 ServicePoint

使用: 可以:

- 对 PerformedActivity（函数，OperationalActivity）有 Performs 依赖（来自 Performer）
- 成为 CompatibleWith 依赖于 ReferredLocation（Location 或 PhysicalLocation）的客户端
- 成为 ConceptRole 的类型（来自 ConceptItem）
- 拥有一个 RequestPoint 端口
- 拥有一个 ServicePoint 端口
- 成为 ExhibitsCapability 对 Capability 的依赖项的客户端
- 自己的 NodeChild（NodeRole、KnownResource、ProblemDomain）（来自 NodeParent）
- OperationalExchange（ConfigurationExchange、EnergyExchange、InformationExchange、MaterielExchange 或 OrganizationalExchange）信息流的源和目标
- 成为 Needline 协会的最终类型
- 有一个附加的 OperationalConstraint（来自 SubjectOfOperationalConstraint）
- 拥有 OperationalStateMachine（来自 SubjectOfOperationalStateMachine）
- 成为 NodeRole 的类型
- 自己的服务运营

操作参数

代表业务活动的投入和产出；用于OV-5。

扩展：

- 活动参数

约束：

- 类型必须是 OperationalExchangeItem（Post、Organization、ResourceArtifact 或系统）

使用：

- 可以由 OperationalActivity 拥有

操作规则

OperationalConstraint A DoDAF 别名。

扩展：

- 约束

概括：

- 操作约束

约束：

- 约束元素必须是SubjectOfOperationalConstraint（OperationalActivity、节点、InformationElement或使命）（来自OperationalConstraint）

操作状态机

描述操作行为或属性A状态机；用于OV-6b。

扩展：

- 状态机

约束：

- 拥有着是SubjectOfOperationalStateMachine（使命、InformationElement或节点）

组织

为特定目的而联系在一起A一群人；用于 OV-4、SV-1、SV-3、SV-9、SV-10a 和 SV-12。

扩展：

- 类

概括：

- 组织资源
- 资源，表演者
- 预测主题
- 资源约束的主题

使用: 可以:

- 成为一个实际组织的分类器
- 命令信息流的源或目标（来自组织资源）
- 成为类的拥有者
- 是子组织的类或类型
- 成为设备的类（来自组织资源）
- 由 OrganizationalExchange 传达（来自 OrganizationalResource）
- 成为 HumanResource 的类型（来自 OrganizationalResource）
- 成为控制信息流的源（来自 OrganizationalResource）
- 有一组相关的里程碑，定型的 ActualProjectMilestone（来自 Resource）
- 将 RealizesCapability 实现为 Capability 的客户端（来自 Resource）
- 成为 Competence 的 ProvidesCompetence 依赖项的客户端（来自 Resource）
- 有一个附加的 ResourceConstraint（来自 Resource，SubjectOfResourceConstraint）
- 成为 Forecast 依赖项的供应商或客户（两者必须具有相同的构造型）（来自 SubjectOfForecast）
- 拥有一个 ServicePoint（来自 Resource）
- 拥有一个 RequestPoint（来自 Resource）
- 拥有一个 ResourcePort（来自 Resource）

- ResourceInteraction 的源和目标（来自 Resource）
- 拥有一个 ServiceOperation（来自 Resource）
- 成为 KnownResource 的类型（来自 Resource）
- 成为 ResourceRole 的类型（来自 Resource）
- 对 PerformedActivity（函数或 OperationalActivity）有 Performs 依赖（来自 Performer）

组织交流

指定跨组织人员流动A关系；用于 OV-2、OV-3 和 OV-6c。

扩展：

- 信息流

概括：

- 操作交换

专长：

- 人民运动

约束：

- 传达的元素必须是组织资源（组织或职位）
- 源是一个节点（来自 OperationalExchange）
- 目标是一个节点（来自 OperationalExchange）

停止服务里程碑

表明项目可交付成果A项目里程碑将停止服务；用于 AcV-2、StV-3 和 SV-8。

扩展：

- 物件

概括：

- 实际项目里程碑

约束：

- 分类器必须是一个项目里程碑

使用：

- 与 CapabilityConfiguration 有一组关联（“配置”）
- 可以有一组关联的资源（来自 ActualProjectMilestone）
- 可以是 MilestoneSequence 的客户/供应商（来自 ActualProjectMilestone）

拥有进程

A关系断言 ActualOrganizationalResource 对 OperationalActivity 负责；用于OV-4。

扩展：

- 依赖

约束：

- 客户必须是 ActualOrganizationalResource（ActualPost 或 ActualOrganization）
- 供应商必须是 OperationalActivity

部件

将使用用作另一个 ResourceArtifact 的一部分；用于 SV-1。

扩展：

- 部件

概括：

- 资源角色

专长：

- 子系统部分

约束：

- 类必须是 ResourceArtifact
- 类型必须是 ResourceArtifact

使用: 可以有:

- RequiresCompetence 对 Competence A依赖（来自 ResourceRole）
- A组与 使用"函数的关联（来自 ResourceRole）

性能参数

A类质量度量，用于解决执行者满足能力需求的程度。

扩展：

- 属性

概括：

- 测量

使用：

- 由 MeasurementSet 类拥有

施行

将执行者链接到它可以执行的行为；用于 OV-2、OV-3、OV-4、OV-5、SV-1 和 SV-4。

扩展：

- 依赖

约束：

- Client 必须是 Performer（节点、ExternalNode、OperationalNode、Post、Organization、CapabilityConfiguration、SystemsNode、Software 或 ResourceArtifact）
- 供应商必须是 PerformedActivity (OperationalActivity 或函数)

Person

A人类；用于OV-4。
扩展：
- 类

使用：
- 可以是一个实际人的分类器

物理数据模型

数据结构的可实现规范；用于 SV-11。
扩展：
- 包

概括：
- 数据模型

专长：
- 内部数据模型

约束：
- 拥有 EntityItem 元素（来自数据模型）

物理位置

可以指定的Anywhere ；用于 OV-1 和 OV-2。
扩展：
- 类

概括：
- 推荐位置
- 概念项目
- 环境类型

使用: 可以是:
- 来自节点的 CompatibleWith 依赖的供应商（来自 ReferredLocation）
- 类型的类型（来自 ConceptItem）
- EnvironmentProperty 的类型（来自 EnvironmentalType）

平台

使用工件作为特定资源配置中的平台；用于 SV-1。
扩展：
- 部件

概括：
- 资源组件
- 资源角色

约束：

- 类必须是 CapabilityConfiguration
- 类型必须是 ResourceArtifact

使用：

- 可以对 Competence 有 RequiresCompetence 依赖项（来自 ResourceRole）
- 可以与 使用"函数有一组关联（来自 ResourceRole）

邮政

A联络点或负责人；用于 OV-4、SV-1、SV-3、SV-9、SV-10a 和 SV-12。

扩展：

- 类

概括：

- 组织资源
- 资源
- 演员
- 预测主题
- 资源约束的主题

使用: 可以:

- 成为实际帖子的分类器
- 成为 PostRole 的类型
- 命令信息流的源或目标（来自组织资源）
- 成为设备的类（来自组织资源）
- 由 OrganizationalExchange 传达（来自 OrganizationalResource）
- 成为 HumanResource 的类型（来自 OrganizationalResource）
- 成为控制信息流的源（来自 OrganizationalResource）
- 有一组相关的里程碑，定型的 ActualProjectMilestone（来自 Resource）
- 成为实现能力的客户（来自资源）
- 成为 Competence 的 ProvidesCompetence 依赖项的客户端（来自 Resource）
- 有一个附加的 ResourceConstraint（来自 Resource，SubjectOfResourceConstraint）
- 成为 Forecast 依赖项的供应商或客户（两者必须具有相同的构造型）（来自 SubjectOfForecast）
- 拥有一个 ServicePoint（来自 Resource）
- 拥有一个 RequestPoint（来自 Resource）
- 拥有一个 ResourcePort（来自 Resource）
- ResourceInteraction 的源和目标（来自 Resource）
- 拥有一个 ServiceOperation（来自 Resource）
- 成为 KnownResource 的类型（来自 Resource）
- 成为 ResourceRole 的类型（来自 Resource）
- 对 PerformedActivity（函数，OperationalActivity）有 Performs 依赖（来自 Performer）

岗位角色

断言组织中存在职位；用于 OV-4 和 SV-1。
扩展：
- 部件

概括：
- 组织角色
- 资源角色

约束：
- 类必须是一个组织
- 类型必须是帖子

使用: 可以有:
- RequiresCompetence 对能力的依赖（来自 ResourceRole）
- 'used'函数的关联集（来自 ResourceRole）

问题域

包含功能资源可以实现的节点的边界；用于OV-2。
扩展：
- 部件

概括：
- 节点角色
- 节点子

约束：
- 类必须是LogicalArchitecture
- 类型必须是节点（来自 NodeRole）

项目

用于定义项目的一个类别；用于 AcV-1。
扩展：
- 类

使用: 可以:
- 成为实际项目的分类器
- 与 ProjectMilestone类有关联

项目里程碑

A项目里程碑；用于 AcV-2。
扩展：
- 类

约束：

- 拥有的属性必须是 ProjectTheme

使用: 可以:

- 成为实际项目里程碑的分类器
- 有一个项目类的关联

项目序列

断言一个 ActualProject 是继另一个之后的；用于 AcV-2。

扩展：

- 依赖

约束：

- 客户必须是实际项目
- 供应商必须是实际项目

项目主题

衡量各种项目进度的一个方面；用于 AcV-2。

扩展：

- 属性

约束：

- 类型必须是 ProjectThemeStatus

使用：

- 由 ProjectMilestone 拥有

项目主题状态

为 ProjectTheme 指定状态。

扩展：

- 类

使用：

- ProjectTheme 的类型

协议

A标准；用于 SV-2、TV-1 和 TV-2。

扩展：

- 类

概括：

- 标准
- 预测主题

使用: 可以:

- 与 ('ratifiedBy') ActualOrganization 对象有一组关联（来自标准）
- 有协议层
- 成为 ProtocolLayers 的类型
- 成为 Forecast 依赖项的客户和供应商

协议层

断言一个协议使用另一个协议；用于 TV-1 和 TV-2。

扩展：

- 属性

约束：

- 拥有类必须是协议
- 类型必须是协议

提供能力

断言资源提供了能力；用于OV-4。

扩展：

- 依赖

约束：

- 客户端必须是资源（Post、Organization、CapabilityConfiguration、SystemsNode、Software 或 ResourceArtifact）
- 供应商必须是能力

实现能力

断言资源提供了能力；用于 SOV-3、StV-3、StV-5 和 SV-1。

扩展：

- 实现

约束：

- Client 必须是 Resource 或 ServiceInterface
- 供应商必须是能力

请求点

服务通信的机制；用于 OV-2 和 SV-1。

扩展：

- 端口

约束：

- 类型必须是 ServiceInterface

使用：

- 可以由节点或资源拥有

需要能力

断言角色需要能力；用于 SV-1。

扩展：

- 依赖

约束：

- 客户端必须是 ResourceRole
- 供应商必须是能力

资源工件

A人造object ；用于 OV-2、OV-3、OV-5、SV-1、SV-3、SV-9、SV-10a 和 SV-12。

扩展：

- 类

概括：

- 操作交换项目
- 制造资源类型
- 资源
- 预测主题
- 资源交互项
- 演员
- 资源约束的主题

专长：

- 系统

使用: 可以:

- 由物资交易所运送
- 成为 OperationalParameter 的类型（来自 OperationalExchangeItem）
- 自己的托管软件
- 成为一个元件的类和部件
- 成为 ResourceComponent 的类型
- 成为设备的类型
- 成为控制流的目标（来自 ManfacturedResourceType）
- 有一组相关的里程碑，定型的 ActualProjectMilestone（来自 Resource）
- 成为实现能力的客户（来自资源）
- 成为 Competence 的 ProvidesCompetence 依赖项的客户端（来自 Resource）
- 有一个附加的 ResourceConstraint（来自 Resource，SubjectOfResourceConstraint）
- 成为 Forecast 依赖项的供应商或客户（两者必须具有相同的构造型）（来自 SubjectOfForecast）

- 拥有一个 ServicePoint（来自 Resource）
- 拥有一个 RequestPoint（来自 Resource）
- 拥有一个 ResourcePort（来自 Resource）
- ResourceInteraction 的源和目标（来自 Resource）
- 拥有一个 ServiceOperation（来自 Resource）
- 成为 KnownResource 的类型（来自 Resource）
- 成为 ResourceRole 的类型（来自 Resource）
- 对 PerformedActivity（函数或 OperationalActivity）有 Performs 依赖（来自 Performer）

资源组件

CapabilityConfiguration 用来完成A能力的定义明确的资源；用于 SV-1。
扩展：
- 部件

概括：
- 资源角色

专长：
- 平台

约束：
- 类型必须是 ResourceArtifact
- 拥有类必须是 CapabilityConfiguration

使用: 可以有:
- RequiresCompetence 对能力的依赖（来自 ResourceRole）
- 'used'函数的关联集（来自 ResourceRole）

资源连接器

两个资源之间A物理连接，实现了源资源可以通过该协议将项目传输到目标资源的协议；用于 SV-2。
扩展：
- 连接器

概括：
- 协议实施

约束：
- 结束角色必须是 ResourcePort

使用: 可以:
- 有一套它实现的ResourceInterface
- 实现资源交互

资源约束

指定管理系统结构或功能方面的规则集；用于 SV-10a。

扩展：

- 约束

约束：

- 约束元素必须是SubjectOfResourceConstraint（DataElement、函数、SystemFunction、CapabilityConfiguration、SystemsNode、Software、ResourceArtifact、系统、Post或Organization）

资源交互

代表资源之间交换的数据；用于 OV-4、SOV-4c、SV-1、SV-2、SV-3、SV-4、SV-6 和 SV-10c。

扩展：

- 信息流

概括：

- 系统元素
- 协议实施

专长：

- 控件
- 命令
- 数据交换

约束：

- 实现连接器是一个 ResourceInterface
- 实现活动边是一个FunctionEdge
- 传达的元素必须是 ResourceInteractionItem（DataElement、Energy、Post、Organization、CapabilityConfiguration、系统、Software、ResourceArtifact 或 System）
- 源必须是资源（Post、Organization、CapabilityConfiguration、SystemsNode、Software、ResourceArtifact 或系统）
- 目标必须是资源（Post、Organization、CapabilityConfiguration、SystemsNode、Software、ResourceArtifact 或系统）

使用：

- 可以实现OperationalExchange（OrganizationalExchange、InformationExchange、EnergyExchange或MaterielExchange）
- 可以实现一个ActualOrganizationRelationship
- 与（ 实现”）协议有关联（来自 ProtocolImplementation）

资源接口

实现协议的两个资源之间A合同协议；用于 OV-4、SV-1、SV-2、SV-3 和 SV-6。

扩展：

- 关联
- 连接器

概括：

- 系统元素

专长：

- 系统连接器

约束：

- 最终角色必须是 ResourceRole
- 结束类型必须是资源

使用：

- 可以实现一个ResourceInteraction

资源消息

用于 Resource 事件跟踪的信息，实现 ResourceInteraction；用于 SV-10c。

扩展：

- 信息

概括：

- 系统元素

使用：

- 可以有一组它携带的 ResourceInteraction

资源端口

资源的交互点，通过它可以与外部环境交互；用于 SV-2。

扩展：

- 端口

概括：

- 协议实施

约束：

- 类型必须是 ResourceInteractionItem（Energy、Post、Organization、CapabilityConfiguration、Software、ResourceArtifact 或 DataElement）

使用：

- 可由资源拥有
- 与它 "实现"的协议类有关联（来自 ProtocolImplementation）
- 可以是 ResourceConnector 的最终角色

资源状态机

工件应用程序将UML状态机扩展到资源用于 SV-10b。

扩展：

- 状态机

概括：

- 系统元素

约束：

- 拥有着必须是 SubjectOfResourceStateMachine (Post, Organization, CapabilityConfiguration,系统, Software, ResourceArtifact, system or DataElement)

如同

断言两个元素指的是同一个现实世界的事物；用于 AV-2。

扩展：

- 依赖

约束：

- 客户端必须是 UPDMElement
- 供应商必须是 ExternalIndividual 或 ExternalType

服务属性

允许捕获性能、可靠性和成本值的服务接口A属性；用于 SOV-1。

扩展：

- 属性

使用：

- 由 ServiceInterface 拥有

服务功能

描述 ServiceOperations 的抽象行为，与实际实现无关；用于 SOV-5。

扩展：

- 活动

使用: 可以:

- 成为 ServiceFunctionAction 的行为
- 成为 ServiceOperationAction 的活动
- 自己的 ServicePoint 端口

服务功能动作

调用要执行的 ServiceFunction A调用行为动作；用于 SOV-5。

扩展：

- 呼叫行为动作

约束：

- 行为必须是 ServiceFunction

服务交互

交互服务接口；用于 SOV-4c。

扩展：

- 交互

服务接口

两个资源之间A合同协议，实现源服务与目标资源交互的协议；用于 SOV-1、SOV-2、SOV-3、SOV-4a、SOV-4b、SOV-4c 和 SOV-5。

扩展：

- 类

约束：

- 拥有的属性必须是 ServiceAttribute
- 拥有的操作必须是 ServiceOperation

使用: 可以:

- 成为 OperationalActivity 的 SupportsOperationalActivity 依赖项的客户端
- 成为 Capability 的 RealizesCapability 实现的客户
- 自己的服务政策
- 与 ServiceStateMachine 有一个关联
- 与 ServiceInteraction 有一个关联
- 是 RequestPoint 或 ServicePoint 端口的类型
- 依赖另一个 ServiceInterface
- 成为向能力公开依赖项的客户

服务讯息

用于服务交互规范的信息，实现资源交互；用于 SOV-4c。

扩展：

- 信息

使用：

- 可以携带一组ResourceInteractions

服务运营

为调用所提供服务的行为提供访问点；用于 SOV-2 和 SOV-5。

扩展：

- 手术

约束：

- 拥有着必须是一个资源（Post、Organization、CapabilityConfiguration、SystemsNode、Software、ResourceArtifact 或系统）
- 拥有着必须是一个节点

使用: 可以:

- 与（具体行为）函数有关联
- 由 ServiceInterface 拥有
- 成为一个 ServiceOperationAction 的操作
- 与（抽象行为）ServiceFunction 有关联

服务操作动作

表示调用 ServiceOperation 的 Resource 或 ServiceFunction A调用操作；用于 SOV-5。

扩展：

- 呼叫操作动作

约束：

- 活动必须是一个ServiceFunction
- 活动必须是一个函数
- 操作必须是 ServiceOperation

使用：

- 可以是 FunctionEdge 控制流的源和目标

服务点

服务通信的机制；用于 OV-2、SV-1 和 SV-12。

扩展：

- 端口

约束：

- 类型必须是 ServiceInterface
- 拥有的行为是一个 ServiceFunction

使用：

- 可由节点或资源（Post、Organization、CapabilityConfiguration、SystemsNode、Software、ResourceArtifact 或系统）拥有

服务政策

约束服务A消费者和提供者；用于 SOV-4a。

扩展：

- 约束

使用：

- 规则可以由 ServiceInterface 拥有

服务状态机

工件扩展UML状态机用于 SOV-4b。
扩展：
- 状态机

软件

系统运行所需的软件；用于 OV-2、OV-3、SV-1、SV-3、SV-9、SV-10a 和 SV-12。
扩展：
- 类

概括：
- 制造资源类型
- 资源
- 预测主题
- 资源交互项
- 演员
- 资源约束的主题

使用: 可以:
- 在 MaterielExchange 信息流上传送
- 是 HostedSoftware 的类型
- 成为控制流的目标（来自 ManfacturedResourceType）
- 有一组相关的里程碑，定型的 ActualProjectMilestone（来自 Resource）
- 成为实现能力的客户（来自资源）
- 成为 Competence 的 ProvidesCompetence 依赖项的客户端（来自 Resource）
- 有一个附加的 ResourceConstraint（来自 Resource，SubjectOfResourceConstraint）
- 成为 Forecast 依赖项的供应商或客户（两者必须具有相同的构造型）（来自 SubjectOfForecast）
- 拥有一个 ServicePoint（来自 Resource）
- 拥有一个 RequestPoint（来自 Resource）
- 拥有一个 ResourcePort（来自 Resource）
- ResourceInteraction 的源和目标（来自 Resource）
- 拥有一个 ServiceOperation（来自 Resource）
- 成为 KnownResource 的一种（来自 Resource）
- 成为 ResourceRole 的一种（来自 Resource）
- 对 PerformedActivity（函数或 OperationalActivity）有 Performs 依赖（来自 Performer）

标准

用于指导和/或约束任何 UPDM元素A已批准规则集；用于 SV-9、TV-1 和 TV-2。
扩展：
- 类

概括：
- 预测主题

专长：

- 协议

使用：

- 任何 UPDMElement 都可以具有与标准的 “conformsTo”关联
- 可以与 ActualOrganization 建立关联 (ratifiedBy)
- 可以是预测的供应商或客户（两者必须是相同的原型）（来自 SubjectOfForecast）

标准配置

附加到 CapabilityConfiguration A注释，表明注释的 CapabilityConfiguration 是一种标准模式，可在架构中重用；用于 TV1 和 TV-2。

扩展：

- 注记

约束：

- 被注解的元素必须是 CapabilityConfiguration

标准作业活动

一个操作活动，它是一个标准的过程并且是教义的；用于 OV-5 和 StV-6。

扩展：

- 活动

概括：

- 运营活动
- 执行活动
- 操作约束的主题
- 操作元素
- SubjectOfOperationalStateMachine

约束：

- 拥有的参数必须是 OperationalParameter（来自 OperationalActivity）

使用: 可以:

- 成为 Capability类的 MapsToCapability 依赖项的客户端
- 成为 Performs 依赖项的供应商（来自 PerformedActivity）
- 成为 OwnsProcess 依赖项的供应商（来自 OperationalActivity）
- 成为 OperationalActivityAction 的活动/行为（来自 OperationalActivity）
- 成为 OperationalActivityEdge 的所有者（来自 OperationalActivity）
- 有一个附加的 OperationalConstraint（来自 SubjectOfOperationalConstraint）
- 成为 SupportsOperationalActivity 依赖项的供应商（来自 OperationalActivity）
- 拥有 OperationalStateMachine（来自 SubjectOfOperationalStateMachine）

刻板印象扩展

定义架构中使用的附加构造型，该构造型未在此元模型中定义；用于 AV-2。
扩展：
- 注记

约束：
- 带注释的元素必须是 UPDMElement

使用：
- 可以有一组与 ExternalType 的关联（ontologyReference）

结构件

描述 EnterprisePhase 的结构部分；用于 AV-1。
扩展：
- 部件

约束：
- 类型必须是 EnterprisePhase
- 类必须是 EnterprisePhase

子组织

断言一种类型的组织通常是另一种类型的父级；用于 OV-4 和 SV-1。
扩展：
- 部件

概括：
- 组织角色
- 资源角色

约束：
- 类型必须是一个组织
- 类必须是一个组织

使用: 可以:
- 对能力有 RequiresCompetence 依赖项（来自 ResourceRole）
- 与 使用”函数有一组关联（来自 ResourceRole）

子系统部分

表示一个子系统是另一个系统的一部分；用于 SV-1。
扩展：
- 部件

概括：
- 部件
- 资源角色

约束：

- 类必须是 ResourceArtifact（来自部件）
- 类型必须是 ResourceArtifact（来自部件）

使用: 可以:

- 对能力有 RequiresCompetence 依赖项（来自 ResourceRole）
- 与 使用”函数有一组关联（来自 ResourceRole）

支持运营活动

断言服务以某种方式有助于或协助执行 OperationalActivity。

扩展：

- 依赖

约束：

- 客户端必须是 ServiceInterface
- 供应商必须是 OperationalActivity

系统

任何有组织的资源和程序的集合，通过相互依存的相互作用进行联合和调节，以完成一组特定的功能。

扩展：

- 类

概括：

- 资源工件
- 操作交换项目
- 制造资源类型
- 资源
- 预测主题
- 资源交互项
- 演员
- 资源约束的主题

使用: 可以:

- 由 MaterielExchange 传送（来自 ResourceArtifact）
- 成为 OperationalParameter 的类型（来自 OperationalExchangeItem）
- 自己的托管软件（来自 ResourceArtifact）
- 成为一个元件的类和类型（来自部件）
- 成为 ResourceComponent 的类型（来自 ResourceArtifact）
- 成为设备的类型（来自 ResourceArtifact）
- 成为控制流的目标（来自 ManfacturedResourceType）
- 有一组相关的里程碑，定型的 ActualProjectMilestone（来自 Resource）
- 成为实现能力的客户（来自资源）
- 成为 Competence 的 ProvidesCompetence 依赖项的客户端（来自 Resource）

- 有一个附加的 ResourceConstraint（来自 Resource，SubjectOfResourceConstraint）
- 成为 Forecast 依赖项的供应商或客户（两者必须具有相同的构造型）（来自 SubjectOfForecast）
- 拥有一个 ServicePoint（来自 Resource）
- 拥有一个 RequestPoint（来自 Resource）
- 拥有一个 ResourcePort（来自 Resource）
- ResourceInteraction 的源和目标（来自 Resource）
- 拥有一个 ServiceOperation（来自 Resource）
- 成为 KnownResource 的类型（来自 Resource）
- 成为 ResourceRole 的类型（来自 Resource）
- 对 PerformedActivity（函数或 OperationalActivity）有 Performs 依赖（来自 Performer）

系统连接器

两个系统之间A链接。

扩展：

- 关联
- 连接器

概括：

- 资源接口
- 系统元素

专长：

- 系统连接器

约束：

- 结束角色必须是 ResourceRole（来自 ResourceInterface）
- 结束类型必须是 Resource（来自 ResourceInterface）

使用：

- 可以实现一个ResourceInteraction（来自ResourceInterface）

系统功能

函数A DoDAF 别名。

扩展：

- 活动

概括：

- 函数
- 执行活动
- 系统元素
- 资源约束的主题

约束：

- 拥有的参数是 FunctionParameter（来自函数）

使用: 可以:

- 成为 Performs 依赖项的供应商（来自 PerformedActivity）
- 自己的 ServiceOperationAction、FunctionAction 或 FunctionEdge（来自函数）
- 成为 OperationalActivity 的 ImplementsOperational 依赖项的客户端（来自 SystemsElement）
- 有一个附加的 ResourceConstraint（来自 SubjectOfResourceConstraint）

SystemFunction动作

FunctionAction A DoDAF 别名。

扩展：

- 呼叫行为动作

概括：

- 功能动作

约束：

- 活动是定型函数（来自 FunctionAction）

使用：

- 按 Ctrl+L 设置函数（来自 FunctionAction）

系统功能边缘

FunctionEdge 的别名。

扩展：

- A控制流

概括：

- 功能边缘
- 系统元素

约束：

- 源必须是 ServiceOperationAction（来自 FunctionEdge）
- 目标必须是 ServiceOperationAction（来自 FunctionEdge）

使用：

- 可以实现一个ResourceInteraction（右键，高级> Information Flows Realized）（来自FunctionEdge）

系统节点

CapabilityConfiguration A DoDAF 别名。

扩展：

- 类

概括：

- 能力配置
- 资源、概念项
- 演员

- 资源交互项
- 资源约束的主题
- 预测主题
- 系统元素
- 资源状态机的主题
- 资源交互项

使用: 可以:

- 有一组相关的已部署里程碑，原型 DeployedMilestone（来自 CapabilityConfiguration）
- 有一个可选的关联不再使用的里程碑，原型 NoLongerUsedMilestone（来自 CapabilityConfiguration）
- 有一组相关的增量里程碑，定型的 IncrementMilestone（来自 CapabilityConfiguration）
- 有一个可选的关联停止服务里程碑，原型 OutOfServiceMilestone（来自 CapabilityConfiguration）
- 由 StandardConfiguration注记注释（来自 CapabilityConfiguration）
- 成为 ConceptRole 的类型（来自 ConceptItem）
- 有一组相关的里程碑，定型的 ActualProjectMilestone（来自 Resource）
- 成为实现能力的客户（来自资源）
- 成为 Competence 的 ProvidesCompetence 依赖项的客户端（来自 Resource）
- 有一个附加的 ResourceConstraint（来自 Resource，SubjectOfResourceConstraint）
- 成为 Forecast 依赖项的供应商或客户（两者必须具有相同的构造型）（来自 SubjectOfForecast）
- 拥有一个 ServicePoint（来自 Resource）
- 拥有一个 RequestPoint（来自 Resource）
- 拥有一个 ResourcePort（来自 Resource）
- ResourceInteraction 的源和目标（来自 Resource）
- 拥有一个 ServiceOperation（来自 Resource）
- 成为 KnownResource 的类型（来自 Resource）
- 成为 ResourceRole 的类型（来自 Resource）
- 对 PerformedActivity（函数，OperationalActivity）有 Performs 依赖（来自 Performer）

技术预测

关于A或多种标准的未来状态的声明。

扩展：

- 预报
- 依赖

约束：

- 客户和供应商都是预测的主题（标准、能力、能力、能力配置、组织、职位、资源工件或软件）（来自预测）
- 客户和供应商必须是 SubjectOfForecast 的同一专业化（来自 Forecast）

时间部分

EnterprisePhase 元素具有基于时间的性质；用于 AV-1。

扩展：
- 部件

约束：
- 类型必须是 EnterprisePhase
- 类必须是 EnterprisePhase

使用的配置

在另一个 CapabilityConfiguration 中使用 CapabilityConfiguration；用于 SV-1。

扩展：
- 部件

概括：
- 资源角色

约束：
- 类型必须是 CapabilityConfiguration
- 类必须是 CapabilityConfiguration

使用: 可以:
- 对能力有 RequiresCompetence 依赖（来自 ResourceRole）
- 有一组关联（usedFunctions）到函数（来自 ResourceRole）

愿景声明

EnterpriseVision A高级文本描述。

扩展：
- 注记

终身企业

涉及人员、组织和支持系统的任何规模A目的的努力；用于 AV-1 和 StV-1。

扩展：
- 类

概括：
- 企业阶段

使用: 可以:
- 有一组关联（statementTasks）到 EnduringTask类（来自 EnterprisePhase）
- 与 Capability类（来自 EnterprisePhase）有一组关联（展览）
- 有一组关联（居住）到环境类（来自 EnterprisePhase）
- 与 EnterpriseGoal类（来自 EnterprisePhase）有一组关联（目标）
- 与 EnterpriseVision类（来自 EnterprisePhase）有一组关联（愿景）
- 是 StructuralPart 或 TemporalPart 的类型（来自 EnterprisePhase）

- 使命用例（来自EnterprisePhase）
- 成为 DefinesArchitecture 实现的供应商（来自 EnterprisePhase）

抽象构造型

构造型专业

构造型	描述
实际组织资源	一个实际的组织或职位。 专长： ● 实际组织 ● 实际邮政
概念项目	可能具有高级操作概念特征的项目。 专长： ● **能力配置** ● 节点 ● **推荐位置** ● 资源
数据模型	数据A结构化规范，显示数据元素的分类和它们之间的关系。 专长： ● 逻辑数据模型 ● 物理数据模型
环境类型	A环境。 专长： ● 光照条件 ● **地点** ● **物理位置** ● 气候
制造资源类型	工件或软件A 概括： ● 资源 专长： ● 资源工件 ● 软件
节点子	一种抽象元素，用于支持节点和逻辑架构等操作元素的复合结构。 专长： ● 节点角色 ● 问题域 ● 已知资源
父节点	代表运营上下文的复合结构的所有者/时间。 专长：

	• 节点 • 外部节点 • 操作节点 • 逻辑架构
操作元素	与运营模式相关的元素。 专长： • 运营活动 • 标准作业活动 • 操作信息 • 节点 • 外部节点 • 操作节点 • 针线 • 操作交换 • 信息元素 • OperationalActivityEdge
操作交换	描述交换项目的特征，例如内容、格式（语音、图像、文本和消息格式）、吞吐量要求、安全或分类级别、及时性要求和互操作性程度。 概括： • 操作元素 专长： • 配置交换 • 能源交换 • 信息交换 • 物资交换 • 组织交流
操作交换项目	参与操作交换的项目。 专长： • 邮政 • 组织 • 资源工件 • 系统
组织资源	无论是组织还是职位。 概括： • 资源 • 操作交换项目 专长： • 邮政 • 组织
组织角色	代表由另一个组织或帖子键入的组织中的属性。 概括：

	• 资源角色 专长： • 子组织 • 岗位角色
执行活动	可以由执行者执行A行为。 专长： • 运营活动 • 函数
演员	可以执行行为A结构元素（例如 PerformedActivity） 专长： • 节点 • 资源
协议实施	实现特定协议的元素。 专长： • 资源端口 • 资源交互 • 控件 • 命令 • 数据交换 • 资源连接器
推荐位置	可以进行操作的实际位置或位置类型（即环境）。 概括： • 概念项目 • 环境类型 专长： • 地点 • 物理位置
资源	有助于实现某项能力A有形资产、组织资源或功能资源。 概括： • 系统元素 • 资源状态机的主题 • 资源交互项 • 演员 • 资源约束的主题 • 概念项目 • 预测主题 专长： • 邮政 • 组织 • 能力配置 • 系统节点

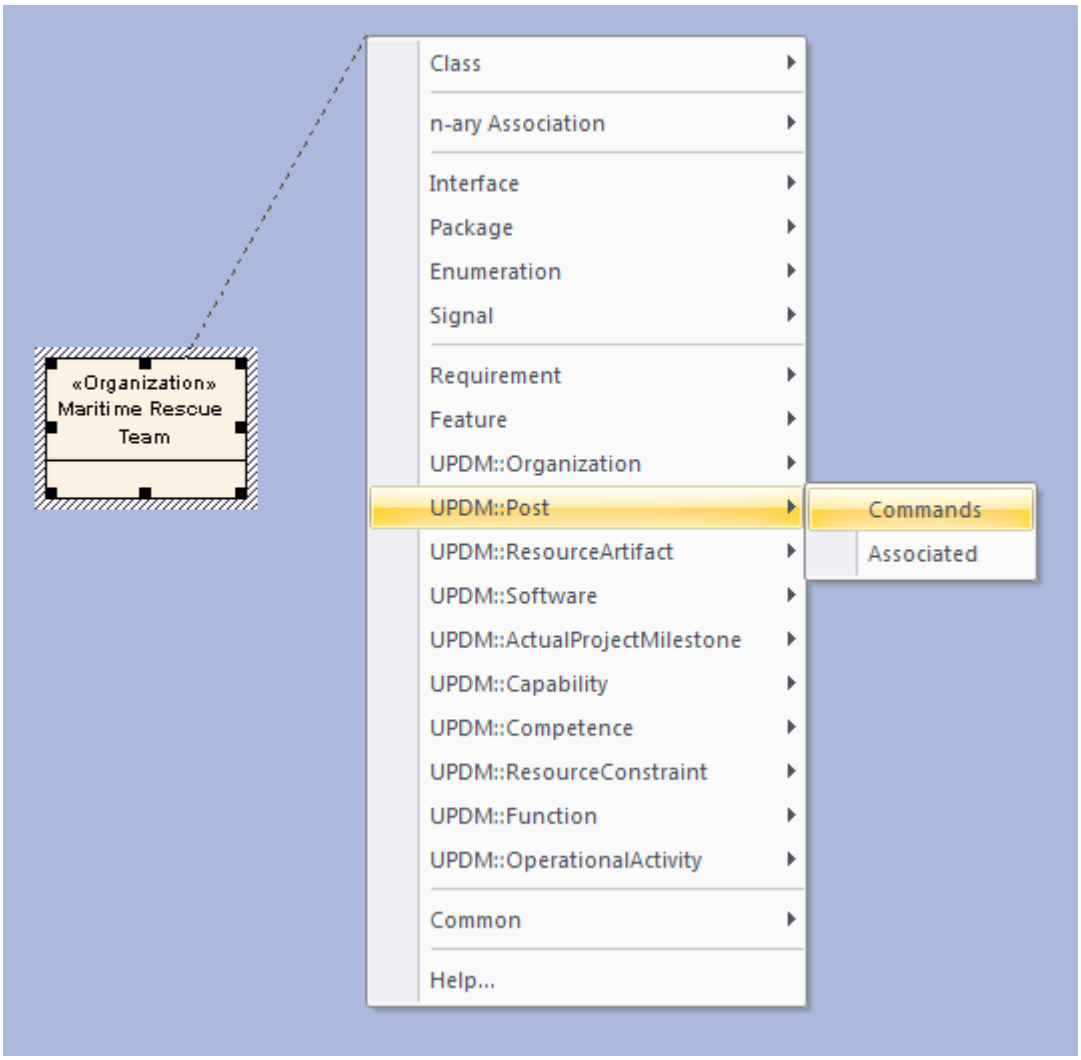
	• 软件 • 资源工件 • 系统
资源交互项	代表资源之间通过资源交互交换的物品。 专长： • 活力 • 资源 • 数据元素
资源角色	定义系统中任何资源的使用。 专长： • 使用的配置 • 设备 • 子组织 • 岗位角色 • 部件 • 子系统部分 • 人力资源 • 资源组件 • 平台 • 托管软件
预测主题	任何可以预测的元素。 专长： • 标准 • 协议 • 能力 • 权限 • 邮政 • 组织 • 能力配置 • 系统节点 • 软件 • 资源工件 • 系统
操作约束的主题	可以受 OperationalConstraint 或 OperationalStateDescription 约束的架构元素。 专长： • 运营活动 • 信息元素 • 节点 • 使命
SubjectOfOperationalStateMachine	状态机描述的元素。 专长：

	• 运营活动 • 信息元素 • 节点 • 使命
资源约束的主题	任何可以受 ResourceConstraint 约束的东西。 专长： • 邮政 • 组织 • 能力配置 • 系统节点 • 软件 • 资源工件 • 系统 • 数据元素 • 函数
资源状态机的主题	状态机描述的元素。 专长： • 邮政 • 组织 • 能力配置 • 系统节点 • 软件 • 资源工件 • 系统 • 数据元素
系统元素	与系统模型有关的元素。 专长： • 资源 • 资源交互 • 资源消息 • 资源交互 • 数据元素 • 资源状态机 • 功能边缘 • 函数
UPDM 元素	所有 UPDM 元素A超类型，提供以通用方式扩展 UPDM 元素的方法。 专长： • 所有 UPDM 刻板印象

快速链接

UPDM 配置文件利用Enterprise Architect的 快速链接”特征来更快、更轻松地创建正确和一致的 UPDM 模型。
当您选择一个元素时，快速链接箭头会显示在该元素的右上角旁边。

将箭头拖离元素并将其释放到空白图表空间。快速链接时间菜单显示，列出了所有通常可以附加到元素的上下文元素，如图所示。



选择 'UPDM::Post |上下文菜单中的命令选项创建一个新的帖子元素，该元素通过命令关系连接到组织元素。

UPDM标记值

UPDM 是UML的扩展，它通过将构造型应用于元素来扩展。构造型依次适用于为通常与UML关联的元素提供附加信息的标记值。由于 UPDM 经常使用标记值，因此建议保持属性窗口随时可见，并展开 "UPDM"部分。

同步标记值

元素拥有的标记值列表可能会过期。 A版本的UML配置文件可能会为元素类型定义新的或修改的标记值，或者用户可能会删除一些。此外，您可以使用不添加标记值的原型组合框来应用原型。如果要刷新单个元素的标记值列表，可以将原型从图表工具箱到元素上，然后选择 应用"选项。这仅适用于单个图表对象，不适用于连接器。

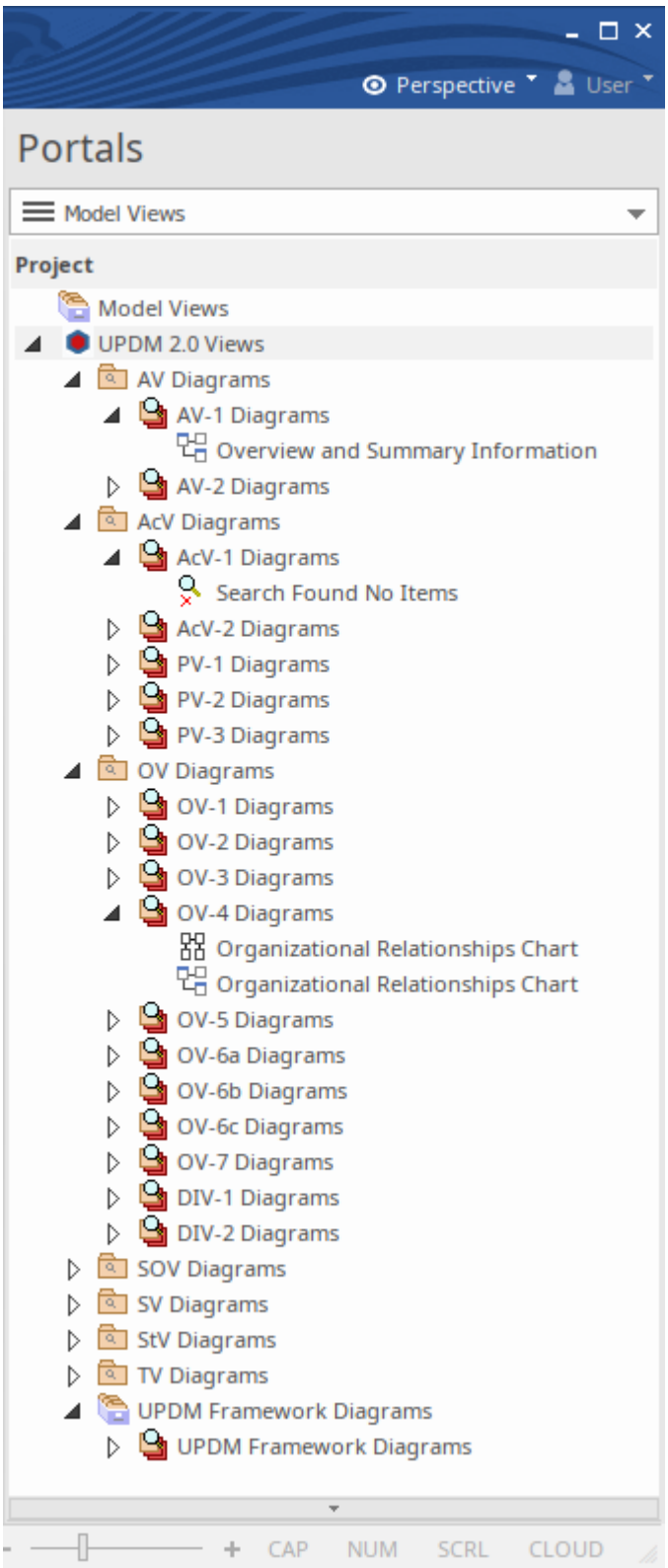
如果要刷新每个元素的标记值模型值列表，请选择 特定>技术> UPDM > 同步标记值"菜单选项。

URL/URI标记值

在 UPDM配置文件规范中，构造型 «UPDMElement» - 从中派生所有配置文件元素 - 提供标记值URL/URI。在 Enterprise Architect中，该标记值已被省略，您必须使用标准Enterprise Architect功能来获得相同的结果：即打开元素的 属性"对话框，选择 文件"选项卡或页面，然后键入网站位置。

视图中的模型视图

Focus 窗口的 模型视图”选项卡显示模型数据的各种不同视图，提供了浏览器窗口的替代方案。您可以使用此选项卡作为在当前模型中定位所有 UPDM 图表的快速简便的方法。



要打开 模型视图”选项卡，请选择 开始>所有窗口>设计>焦点>模型视图”。展开相应的文件夹并双击所需的图表将其打开。

词汇表

UPDM 提供了将所有 UPDM 原型的描述导入Enterprise Architect Glossary 的能力。这使您可以快速参考每个构造型的含义，列出构造型可能出现的视图，并且对于抽象构造型，列出从抽象构造型继承的具体构造型。

导入词汇表

您将词汇表定义分别导入到每个模型中。为此，请选择 发布>模型交换>导入>其它工具/格式”功能区选项。

视图词汇表

要查看词汇表，请选择以下选项之一：

- '设计> 词典 > 词汇表 > 词汇表视图以显示项目词汇视图
- '设计> Dictionary > Glossary > Edit' 打开'Glossary' 对话框
- 在任何对话框的 注记”字段中，一个词汇表超链接（下划线和蓝色）

使用Enterprise Architect元素

从类创建实例

UPDM 具有分类器/实例对，其中分类器描述一类元素，而实例表示该类的单个成员。 UPDM 中的分类器/实例对是：

- 测量集/实际测量集
- 组织/实际组织
- Person /实际人
- 过帐/实际过帐
- 项目/实际项目
- 项目里程碑类型/实际项目里程碑
- 能力配置/现场能力

如果您有一个元素，它是这些分类器/实例对之一的分类器部分，您可以在两种主要方法之间进行选择来创建实例：

1. 设置现有实例的分类器- 单击图表中的实例元素，然后按 Ctrl+L 或右键单击并选择 高级|"实例分类器';相同的命令设置端口或部件的类型。
2. 从现有分类器创建一个实例 - 按住 Ctrl 的同时将分类器元素从浏览器窗口拖到图表上。 '粘贴元素' 对话框显示；选择 粘贴作为元素的实例"选项。使用适当的构造型创建一个匿名实例；选择实例，按 F2 并为其命名。

设置object的运行状态

在一个分类器可以拥有一组属性的情况下，该分类器的一个实例可以为每个属性拥有一个 Slot。这些插槽的赋值集合称为运行状态。要在图表上设置object的运行状态，请右键单击它并选择'特征|设置运行状态'或按 Ctrl+Shift+R。

UPDM 将一些构造型定义为扩展 Slot元类。每个运行状态属性代表一个 Slot，但在Enterprise Architect中无法构造 Slot，因此Enterprise Architect的实现中没有 UPDM 的插槽扩展构造型。扩展 Slot 的 UPDM 原型是：

- 实际测量 (ActualMeasurementSet)
- 实际组织角色（实际组织）
- MeasureOfPerformance（实际测量集）
- 项目状态（实际项目里程碑）

属性

UPDM 中的一些构造型被定义为扩展UML属性元类。这使您可以在模型中为这些元素选择多种不同的表示形式。如果将其中一个属性从工具箱拖到图表上的分类器元素上，系统会提示您选择创建属性、部件或端口器。这些都是UML属性元类的不同表示；你选择哪一个取决于你想在你的模型中看到的属性的渲染。

UML属性元类的另一种表示形式是关联结束；将 UPDM 的属性原型之一应用于关联End：

1. 双击元素以显示 属性"对话框。
2. 选择 角色"选项卡。
3. 单击适当的 构造型"字段旁边的  按钮。
4. 在 关联的构造型"对话框中，从 配置文件"字段中选择 "UPDM"。

5. 选择每个适用的刻板印象。

扩展属性的构造型是：

- 概念角色
- 实体属性
- 环境属性
- 设备
- 托管软件
- 人力资源
- 已知资源
- 测量
- 节点角色
- 部件
- 性能参数
- 平台
- 岗位角色
- 问题域
- 项目主题
- 协议层
- 资源组件
- 服务属性
- 结构件
- 子组织
- 子系统部分
- 时间部分
- 使用的配置

UPDM 中的模型验证

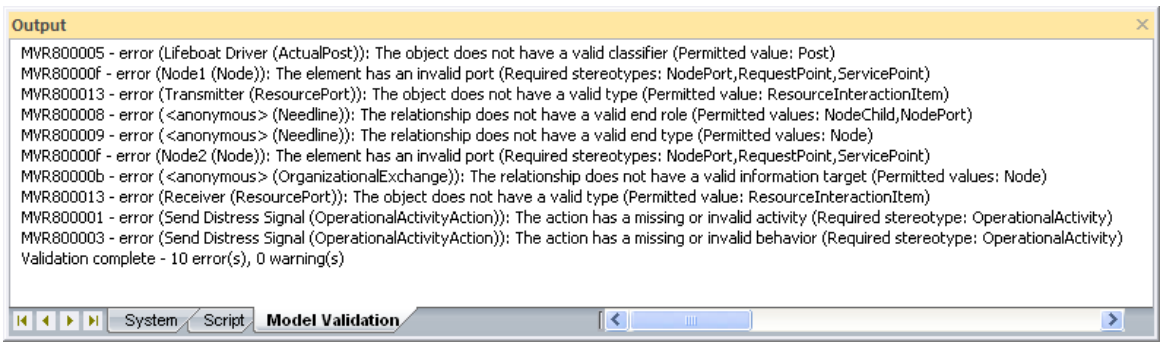
Enterprise Architect支持 UPDM 模型的模型验证，针对 160 多个不同的规则验证和报告错误。

配置模型验证

在能够验证模型之前，您首先必须选择要验证的规则。选择 设计>包>管理>验证>配置验证规则”并取消选中除UPDM集之外的所有验证规则的复选框。

执行模型验证

打开图表或在浏览器窗口中选择一个包或多个元素，然后选择 设计>包>管理>验证>验证当前包”功能区选项（或按Ctrl+Alt+V）。验证结果显示在系统输出窗口中，如果尚未显示，则打开该窗口。要转到导致验证错误的元素，请双击系统输出窗口中的错误消息。



模型验证规则

错误由格式为 MVRxxnnnn 的错误代码指示，其中：

- xx 默认为 80（如果MDG 技术for UPDM 是唯一的插件
  您已安装）但可能是其他数字，并且
- nnnn 是从 0001 到 0013 的十六进制数，如此处所述

MVRxx0001 - 活动

错误信息：该操作缺少或无效活动（需要原型：<stereotypeList>）

验证规则检查构造型行动元素是否由具有所需构造型的活动拥有。

解决方案：在浏览器窗口中找到行动，找到具有命名的刻板印象（或其专业化）之一的活动，或者创建一个新的，然后将行动拖到活动中。

行动构造型	活动构造型
功能动作	函数
OperationalActivityAction	运营活动
服务操作动作	函数
服务操作动作	服务功能

MVRxx0002 - 注释元素

错误信息：注记有一个无效的注释元素（需要stereotype：<stereotype>）

此验证规则检查构造型注记元素是否（通过 NoteLink 连接器）附加到具有所需构造型的元素。

解决方案：将注记附加到具有命名构造型（或其专业化之一）的元素上。您可以通过拖动 NoteLink 连接器的另一端或删除 NoteLink 连接器并使用快速链接器创建一个新连接器来执行此操作。

注记构造型	带注释的元素构造型
别名	UPDM 元素
架构元数据	架构描述
定义	UPDM 元素
标准配置	能力配置
刻板印象扩展	UPDM 元素

MVRxx0003 - 行为

错误信息：动作有缺失或无效行为（需要stereotype：<stereotype>）

此验证规则检查构造型行为元素是否调用具有所需构造型的行为。

解决方法：右击行动，选择高级|设置行为分类器，或按 Ctrl+L，然后选择具有命名构造型（或其特化之一）的行为元素。

行动构造型	行为构造型
OperationalActivityAction	运营活动
服务功能动作	服务功能

MVRxx0004 - 类

错误信息： object没有有效的拥有类（允许的值：<stereotypeList>）

此验证规则检查构造型属性元素（部件或属性）是否由具有所需构造型的类拥有。

解决方案：在浏览器窗口中找到该属性，找到具有命名构造型（或其特化）之一的类或创建一个新类，然后将该属性拖到属性中。

属性构造型	类构造型
设备	组织资源
托管软件	资源工件
人力资源	能力配置
节点子	父节点
节点角色	节点
部件	资源工件
岗位角色	组织
问题域	逻辑架构
协议层	协议
资源组件	能力配置
资源角色	资源
结构件	企业阶段
子组织	组织
时间部分	企业阶段

使用的配置	能力配置

MVRxx0005 -分类器

错误信息： object没有有效的分类器（允许的值：<stereotype>）

此验证规则检查构造型实例元素（对象）是否按具有所需构造型的元素分类。

解决方法：选中object ，右键选择高级|实例分类器，或按 Ctrl+L，然后选择具有命名构造型（或其特化之一）的分类器元素。

物件构造型	分类器构造型
实际测量集	测量集
实际组织	组织
实际人	Person
实际邮政	邮政
实际项目	项目
实际项目里程碑	项目里程碑类型
现场能力	能力配置

MVRxx0006 - 客户端

错误信息：关系没有有效的客户端（允许的值：<stereotypeList>）

此验证规则检查，对于构造型依赖或实现关系，它们的客户端（源）元素是否具有所需的构造型。

解决方案：将不带箭头的关系的末尾拖到具有命名构造型（或其特化之一）的元素上。

关系构造型	客户端元素构造型
任意关系	高级操作概念
架构参考	架构描述
兼容	节点
定义架构	架构描述
展品能力	节点
暴露	服务接口
填报	实际人

预报	预测主题
实现操作	系统元素
MapsToCapability	标准作业活动
里程碑序列	实际项目里程碑
拥有进程	实际组织资源
施行	演员
项目序列	实际项目
提供能力	资源
实现能力	资源
实现能力	服务接口
需要能力	资源角色
如同	UPDM 元素
支持运营活动	服务接口

MVRxx0007 - 受限元素

错误信息：约束具有无效的约束元素（需要原型：%s）

此验证规则检查构造型约束元素是否（通过 NoteLink）附加到具有所需构造型的元素。

解决方案：将约束附加到具有命名构造型（或其特化之一）的元素。您可以通过拖动 NoteLink 连接器的另一端或删除 NoteLink 连接器并使用快速链接器创建一个新连接器来执行此操作。

约束构造型	约束元素构造型
操作约束	操作约束的主题
资源约束	资源约束的主题

MVRxx0008 - endRoles

错误信息：关系没有有效的结束角色（允许的值：<stereotypeList>）

此验证规则检查，对于构造型关联或连接器关系，关系两端的元素是否具有所需的构造型。

解决方案：将关系的一端或两端拖动到具有命名构造型（或其特化之一）的元素。

关系构造型	Endstate构造型元素
针线	节点子
针线	节点端口
资源连接器	资源端口
资源接口	资源角色

MVRxx0009 - endType

错误信息：关系没有有效的结束类型（允许的值：<stereotypeList>）
此验证规则检查，对于构造型连接器，关系两端的元素（对象或部件）是否由所需构造型键入。
解决方案：将关系的一端或两端拖动到具有命名构造型（或其特化之一）的类型的元素。

连接器构造型	End类型构造型
实体关系	实体项
针线	节点
资源接口	资源

MVRxx000a - 信息源

错误信息：关系没有有效信息源（允许的值：<stereotypeList>）
此验证规则检查构造型信息流关系源元素是否具有所需的构造型。
解决方案：将不带箭头的信息流的末尾拖到具有命名构造型（或其特化之一）的元素。

InformationFlow构造型	源构造型元素
实际组织关系	实际组织资源
命令	组织资源
控件	组织资源
操作交换	节点
资源交互	资源

MVRxx000b - 信息目标

错误信息：关系没有有效的信息目标（允许的值：<stereotypeList>）
此验证规则检查构造型 InformationFlow关系目标元素是否具有所需的构造型。
解决方案：将带有箭头的信息流的末端拖到具有命名构造型（或其特化之一）的元素。

InformationFlow构造型	目标构造型元素
实际组织关系	实际组织资源
命令	组织资源
控件	组织资源
操作交换	节点
资源交互	资源

MVRxx000c - 拥有属性

错误信息：元素具有无效属性（需要原型：<stereotype>）
此验证规则检查，对于构造型类元素，它们拥有的任何属性是否具有所需的构造型。
解决方案：将属性替换为具有命名构造型（或其特化之一）的属性。

类构造型	属性构造型
实体项	实体属性
环境	环境属性
高级操作概念	概念角色
测量集	测量
项目里程碑类型	项目主题
服务接口	服务属性

MVRxx000d - 拥有操作

错误信息：元素的操作无效（需要型：%s）
此验证规则检查，对于构造型类元素，它们拥有的任何操作是否具有所需的构造型。
解决方案：将操作替换为具有命名构造型（或其特化之一）的操作。

类构造型	操作构造型
服务接口	服务运营

MVRxx000e - 拥有参数

错误信息：元素具有无效的活动参数（需要原型：%s）

此验证规则检查，对于构造型活动元素，它们拥有的任何活动参数元素是否具有所需的构造型。

解决方案：在浏览器窗口中找到活动参数并将其拖放到具有适当构造型的元素上，和/或将其当前所有者中的活动参数活动参数为具有命名构造型的活动参数。

活动构造型	活动参数构造型
函数	功能参数
运营活动	操作参数

MVRxx000f - 拥有端口

错误信息：该元素的端口无效（需要stereotypes：<stereotypeList>）

此验证规则检查，对于立体类元素，它们拥有的任何端口的原型都具有。

解决方案：在浏览器窗口中找到端口并将其拖放到具有适当构造型的元素上，和/或将其当前端口中的端口替换为具有命名构造型之一的端口。

类构造型	端口构造型
节点	节点端口
节点	请求点
节点	服务点
资源	请求点
资源	资源端口
资源	服务点

MVRxx0010 -源

错误信息：关系没有有效源（允许的值：<stereotypeList>）

此验证规则检查构造型 ActivityEdge 连接器源元素是否具有所需的构造型。

解决方案：将不带箭头的关系的末尾拖到具有命名构造型（或其特化之一）的元素上。

ActivityEdge构造型	源构造型元素
功能边缘	服务操作动作

OperationalActivityEdge	OperationalActivityAction

MVRxx0011 - 供应商

错误信息：关系没有有效的供应商（允许的值：<stereotypeList>）
此验证规则检查构造型依赖或实现关系供应商（目标）元素是否具有所需的构造型。
解决方案：将带箭头的关系的末尾拖到具有命名构造型（或其特化之一）的元素上。

关系构造型	供应商当前的元素构造型
任意关系	高级操作概念
架构参考	架构描述
兼容	推荐位置
定义架构	企业阶段
展品能力	能力
暴露	能力
填报	实际邮政
预报	预测主题
实现操作	操作元素
MapsToCapability	能力
里程碑序列	实际项目里程碑
拥有进程	运营活动
施行	执行活动
项目序列	实际项目
提供能力	权限
实现能力	能力
实现能力	权限
需要能力	外部个人
如同	外部类型

支持运营活动	运营活动

MVRxx0012 - 目标

错误信息：关系没有有效的目标（允许的值：<stereotypeList>）

此验证规则检查构造型 ActivityEdge 连接器目标元素是否具有所需的构造型。

解决方案：将带箭头的关系的末尾拖到具有命名构造型（或其特化之一）的元素上。

ActivityEdge构造型	目标构造型元素
功能边缘	服务操作动作
OperationalActivityEdge	OperationalActivityAction

MVRxx0013 - 类型

错误信息： object没有有效类型（允许的值：<stereotype>）

此验证规则检查构造型属性元素（部件或属性）是否具有具有所需构造型的类型元素。

解决方案：对于零件，右键单击部件并选择'高级|设置属性类型'，或按 Ctrl+L，然后选择具有命名构造型（或其特化之一）的类型元素。对于属性，打开属性的特征窗口并在 ‴字段中选择具有命名构造型（或其专业化之一）的元素类型。

属性构造型	类型元素构造型
概念角色	概念项目
环境属性	环境类型
设备	资源工件
功能参数	资源交互项
托管软件	软件
人力资源	组织资源
已知资源	资源
节点端口	操作交换项目
节点角色	节点
操作参数	操作交换项目
部件	资源工件

岗位角色	邮政
项目主题	项目主题状态
协议层	协议
请求点	服务接口
资源组件	资源工件
资源端口	资源交互项
服务点	服务接口
结构件	企业阶段
子组织	组织
时间部分	企业阶段
使用的配置	能力配置

ArchiMate 框架

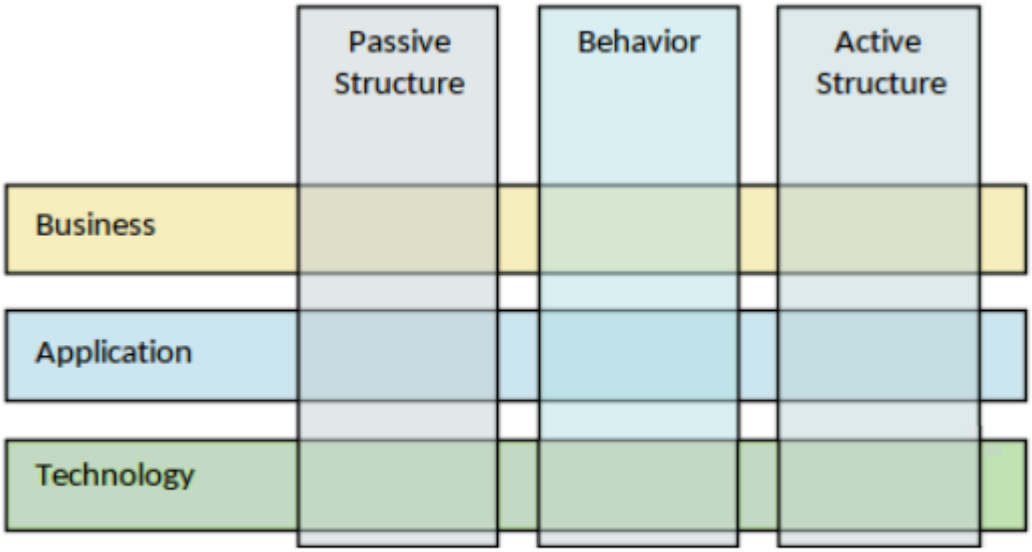
ArchiMate 既是一种语言又是一种框架，使其成为对企业或其部门架构进行描述性和规范性表示的非常有用的范例。结合语言和框架，企业架构师和其他人可以创建将业务、应用程序和技术架构结合在一起的清晰模型。反过来，这些层又细分为跨越语言语法的各个方面，从主动和被动结构到行为和动机。这些模型可以用其他框架内容进行修饰或补充，例如可重用的内容库、标准、治理日志等。

Archimate 框架以定义这些层和概念的元模型表示。核心框架定义了三层和三个方面，并在完整框架中扩展了两个附加层和一个和一个更多方面。这允许架构师模型动机、策略、实施和迁移工件。

Enterprise Architect支持 ArchiMate 语言和 Core Framework 将元素及其关系存储在一个强大而灵活的存储库中，可以使用强大的工具套件进行设计、可视化、查询和查看。行业最佳实践模型、图表和观点在一组模型模式中可用，您可以使用这些模型将预构建的内容注入到您的架构中。每个模式都附有描述性文本，并详细说明了如何使用该模式，包括工具、想法和后续步骤。

ArchiMate 核心框架

用于对 ArchiMate 核心语言元素进行分类的参考结构在企业架构的核心内部元模型中实现。它由三层和三个方面（包括物理）组成，扩展框架包含许多额外的层和方面作为扩展。这些核心和扩展语言元素图表和工具箱页面的形式在工具中可见，并在包括图标格式在内的各种形式的表示中可见。该图是在Enterprise Architect中创建的，用于演示语言的结构。



图：显示核心框架的层次和方面

方面

主动结构方面定义了可以直接表现行为的结构概念，例如业务层中的业务Actor和应用层中的应用部件等元素。这些就像我们自然语言中的主题。

行为方面定义了分配给结构元素的行为，包括服务、功能、流程等元素。结构元素定义了表现行为的元素。这些就像我们自然语言中的动词。

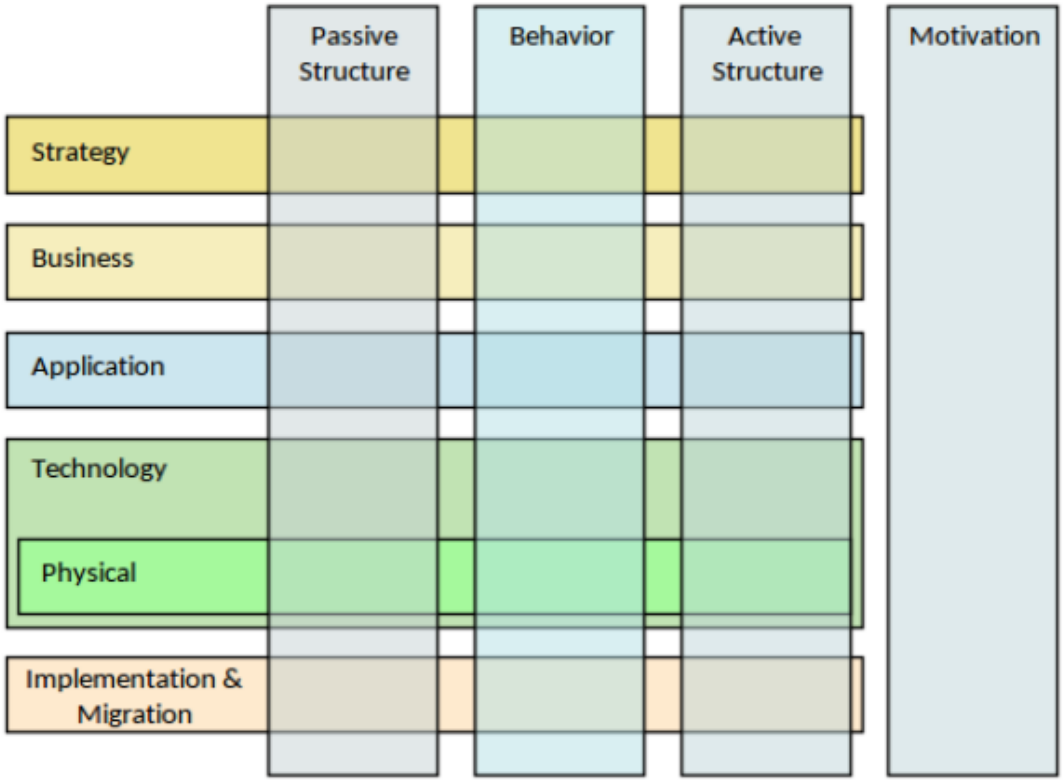
被动结构方面定义了作为行为接受者的元素。这些元素通常是信息、数据和物理对象。这些就像我们自然语言中的对象（或谓词）。

图层

ArchiMate 为架构模型定义了一个分层的和面向服务的范例。较高层利用较低层提供的服务。这些层是抽象概念，架构实践或团队可以根据任何结构自由组织他们的模型。这些层提供了从业务层中定义的利益相关者和用户到为中间技术层提供平台的虚拟或物理技术元素的连续统一体。

完整的框架

ArchiMate 核心框架已得到扩展，允许架构师对一系列附加概念进行模型，包括跨越所有层的横切动机方面和用于模型策略、实施和迁移的两个新层。另一层包括在技术层中，架构师用于模型物理项目，例如机器和设备。



图：显示整个框架的层次和方面

扎克曼框架

Zachman 框架是一种广泛使用的工程企业架构方法。该框架是一个简单的逻辑结构，有助于组织企业的信息基础设施，并在帮助使技术与业务需求保持一致方面提供许多好处。

讨论

此处描述的主题提供了在Enterprise Architect中使用 Zachman 框架的介绍和程序说明。

部分	内容
欢迎 	本节介绍 Zachman 框架，并包含定义其与Enterprise Architect一起使用的正式文档。
使用 Zachman 框架 	开始使用 Zachman 框架，了解模型结构、模板、图表类型等。
验证模型 	了解如何为 Zachman 框架开发和配置模型验证。

简单的介绍

欢迎使用Enterprise Architect中的 Zachman 框架。

将这项技术与Enterprise Architect一起使用，您可以使用 Zachman 框架以及多功能、开放标准建模系统的相关优势。 Zachman 框架已与终极版和统一版集成；它可以单独购买，与Enterprise Architect专业版或企业版一起使用。

关于 Zachman 框架

Zachman 框架是一种广泛使用的工程企业架构方法。该框架是一个简单的逻辑结构，有助于组织企业的信息基础设施。

虽然概念上很简单，但 Zachman 框架在帮助使技术与业务需求保持一致方面提供了许多好处。它已成为定义企业架构的流行方法，因为它：

- 平台中立
- 是一种多功能的规划设备
- 既全面又易于非技术人员理解
- 协助解决问题
- 帮助记录企业范围的信息系统架构

在 Zachman 框架下，企业通过回答六个问题来建模：什么？如何？在哪里？谁？什么时候？为什么？关于六个角色视角：Planner、拥有者、设计师、Builder、拥有着和Functioning企业。

如需更多信息，请访问 Zachman 框架网站。

开始

有关如何使用 Zachman 框架的说明，请参阅以下主题：

- 开始使用 *Zachman* 框架和
- 使用 *Zachman* 框架

支持 Zachman 框架

Enterprise Architect的注册用户可以通过与Enterprise Architect本身相同的渠道获得 Zachman 框架的技术支持。

Zachman Framework系统需求

Zachman 框架版本1 。 1 .4 在此处确定的环境下运行。

支持的 Microsoft® 操作系统

- 窗口10
- 窗口8
- 窗口7
- 窗口2008服务器
- 窗口2003服务器
- 窗口XP Service Pack 2

支持的Enterprise Architect版本

- Enterprise Architect版本7。 1或更高版本

注记

- 支持 32 位和 64 位操作系统

与开始一起开始

当您安装Enterprise Architect的统一或终极版时，Zachman 框架已完全启用并可以使用。

如果您有Enterprise Architect的企业版或专业版，您可以单独购买和安装MDG 技术for Zachman Framework；一旦您输入了 Zachman 框架的MDG 技术注册密钥，它就会自动在Enterprise Architect中可用并集成到统一和终极技术版本中。

访问MDG 技术For Zachman Framework

1. 创建一个新的Enterprise Architect项目文件，然后单击顶层包。
2. 选择功能区选项 设计>包>模型构建器”。
3. 在 模型构建器”对话框中，选择 企业架构> Zachman”蓝图和 “Zachman框架”模式。
4. 单击创建模型按钮。

在浏览器窗口中创建了A新的基础 Zachman模型，其中包含Zachman框架图和规划器、拥有着、设计师、构建器、分包商和功能企业包。

许可版权和商标

Zachman 框架版权声明

版权所有 © 2007-2022 Sparx Systems Pty. Ltd. 保留所有权利。

Sparx Systems Pty. Ltd.

99 Albert St,

Creswick, Victoria 3363,

澳大利亚

电话：+61 (3) 5345 1140

传真：+61 (3) 5345 1104

支持电子邮件： support@sparxsystems.com

销售电子邮件： sales@sparxsystems.com

网站： sparxsystems.com

Zachman 框架软件产品许可协议的MDG 技术

本软件产品许可协议涉及单独购买的 Zachman 框架的MDG 技术，用于Sparx Systems Enterprise Architect的专业版和企业版。与 Enterprise Architect统一版和终极版集成的MDG 技术受Enterprise Architect Sparx Systems Enterprise Architect Modelling Tool。

MDG 技术for Zachman Framework - Enterprise Architect MDG插件

,版本1 . 1

重要 - 请仔细阅读：本最终用户许可协议（“EULA”）是您作为被许可人与 SPARX 之间就上述软件产品达成的法律协议。通过安装、复制或以其他方式使用软件产品，您同意受本 EULA 条款的约束。如果您不同意本 EULA 的条款，请立即删除未使用的软件产品。

软件产品及其文档的版权归Sparx Systems Pty Ltd, A所有。 B 。 N 38 085 034 546. 根据本 EULA 的条款，您被授予在 EULA 有效期内使用软件产品的非排他性权利。您不会根据本 EULA 获得软件产品任何部分的版权或其他知识产权的属性。

您使用本软件即表示您接受本 EULA 和保修。

定义

在本最终用户许可协议中，除非出现相反的意图，

- “EULA”是指本最终用户许可协议
- A SPARX”是指Sparx Systems Pty Ltd A.C.电话号码N 034 546
- 被许可人”是指您或您代表其接受 EULA 的组织（如果有）
- “Zachman 框架的MDG 技术注册版”是指可从以下网站购买的软件产品版本： https ://sparxsystems.com/products/mdg/tech/zachman/purchase.html
- 软件产品”或 软件”是指 Zachman 框架的MDG 技术，包括计算机软件和相关媒体和印刷材料，可能包括在线或电子文档
- 支持服务”是指 SPARX 提供的基于电子邮件的支持，包括有关Enterprise Architect使用的建议、错误调查、修复、模型维修（如果适用）以及一般产品支持
- “SPARX 支持工程师”是指提供在线支持服务的 SPARX 员工

授予许可

根据本 EULA 的条款，您被授予以下权利：

- 在单台计算机上安装和使用软件产品的一个副本，或替代同一操作系统的任何先前版本；作为安装本软件产品的计算机的主要用户，您可以制作第二份副本，供您在家或便携式计算机上独家使用
- 在存储设备（例如网络服务器）上存储或安装软件产品的副本，仅用于通过内部网络安装或运行软件产品
- 为备份、存档和指导目的制作软件产品的副本

评估许可证

Zachman 框架的MDG 技术Trial Edition不是免费软件。根据本协议的条款，特此授权您在三十 (30) 天内免费使用本软件进行评估。

三十 (30) 天到期后，必须从计算机中删除软件产品。在 30 天评估期后未注册使用 Zachman 框架的MDG 技术违

反了澳大利亚、 U。 S 。和国际版权法。

SPARX 可根据要求自行决定延长评估期。

如果您选择在 30 天评估期后使用此软件，则必须购买许可证（如https中所述）。支付许可费后，您将收到有关在何处下载 Zachman 框架的MDG 技术注册版的详细信息，并将通过电子邮件向您提供合适的软件 密钥”。

附加权利和限制

您在此承诺，除非本 EULA 明确授权，否则不会出售或分许可软件产品。

没有保修。软件产品按 原样”提供，不提供任何形式的保证，SPARX 明确否认与软件产品有关的所有明示、暗示或法定保证和/或条件，包括但不限于暗示保证和/或适销性、令人满意的质量、适用于特定目的、准确性、安静享受和不侵犯第三方权利的条件。

局限性

在任何情况下，SPARX 均不对因本许可或您使用、复制、修改、分发软件产品或其任何部分而引起或与之相关的任何偶然、特殊、间接或后果性损害承担责任，无论是否根据合同理论、保证、严格责任或其他方式，即使版权持有人已被告知此类损害的可能性，尽管任何补救措施未能达到基本目的。

商标

本 EULA、软件产品或随附文档中使用的所有产品和公司名称可能是其相应所有者的商标。它们在本 EULA 中的使用旨在遵守相应的指南和许可。

Zachman企业架构框架是 John A的商 标 。扎克曼和扎克曼国际。

适用法律

本协议应根据维多利亚州的状态联邦法律解释。

商标确认

Sparx Systems承认这些商标在整个MDG for Zachman 框架文档中使用。

微软的商标

- Microsoft Word
- 微软办公软件
- 视窗®

物件管理集团的商标

- 物件管理组™
- 天 哪
- UML ™
- Unified Modeling Language语 言

John A商标。扎克曼和扎克曼国际

- Zachman Framework For企业架构™

使用 Zachman 框架

Zachman 框架提供了一个基于模型的框架，用于为企业规划、设计和实施架构。随技术提供的启动器模型可作为您构建企业架构的基础。您可以使用支持 Zachman 分类框架每个单元的工具箱页面，从扩展的Enterprise Architect UML图集创建适当的图。

该技术还为战略项目计划提供模型验证和报告功能。

在Enterprise Architect中，您可以在图表视图和元素列表视图之间进行选择。元素列表视图可以在您喜欢仅定义模型工件的单元中使用。

您还可以通过Enterprise Architect矣系矩阵跨框架（水平和垂直）对齐单元。

您可以在Sparx Systems网站上观看正在使用的MDG 技术For Zachman Framework 的演示视频。

帮助框架帮助主题提供了对 Zachman 框架工具和特征的详细探索，例如。

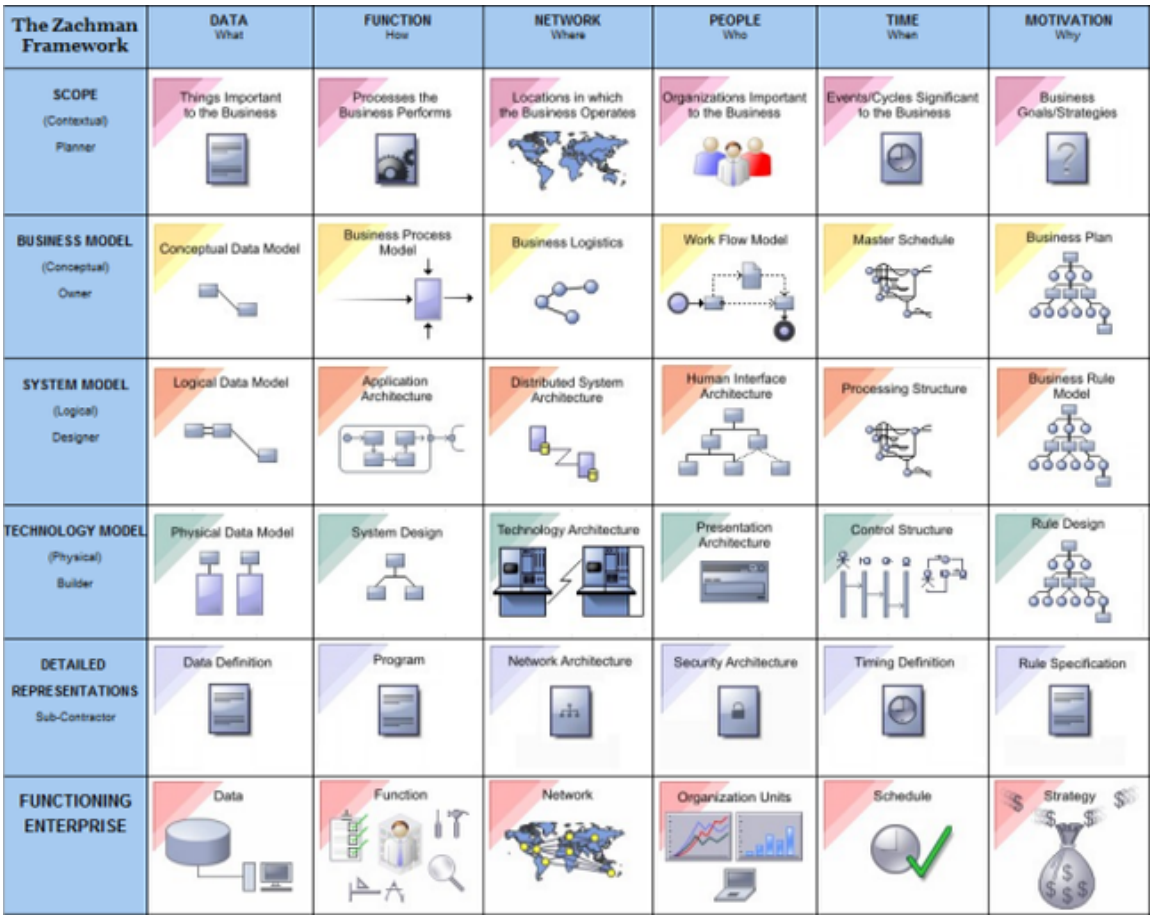
- Zachman 框架的Enterprise Architect模型示例
- 用于特定 Zachman 框架单元的UML配置文件（工具箱页面）
- Zachman 框架A图表界面
- 特定于 Zachman 框架的新图表类型
- 灵活A模型启动器结构
- 战略项目计划的报告生成功能

Zachman 框架的MDG 技术与Enterprise Architect的特征集成。

Zachman 框架接口图表

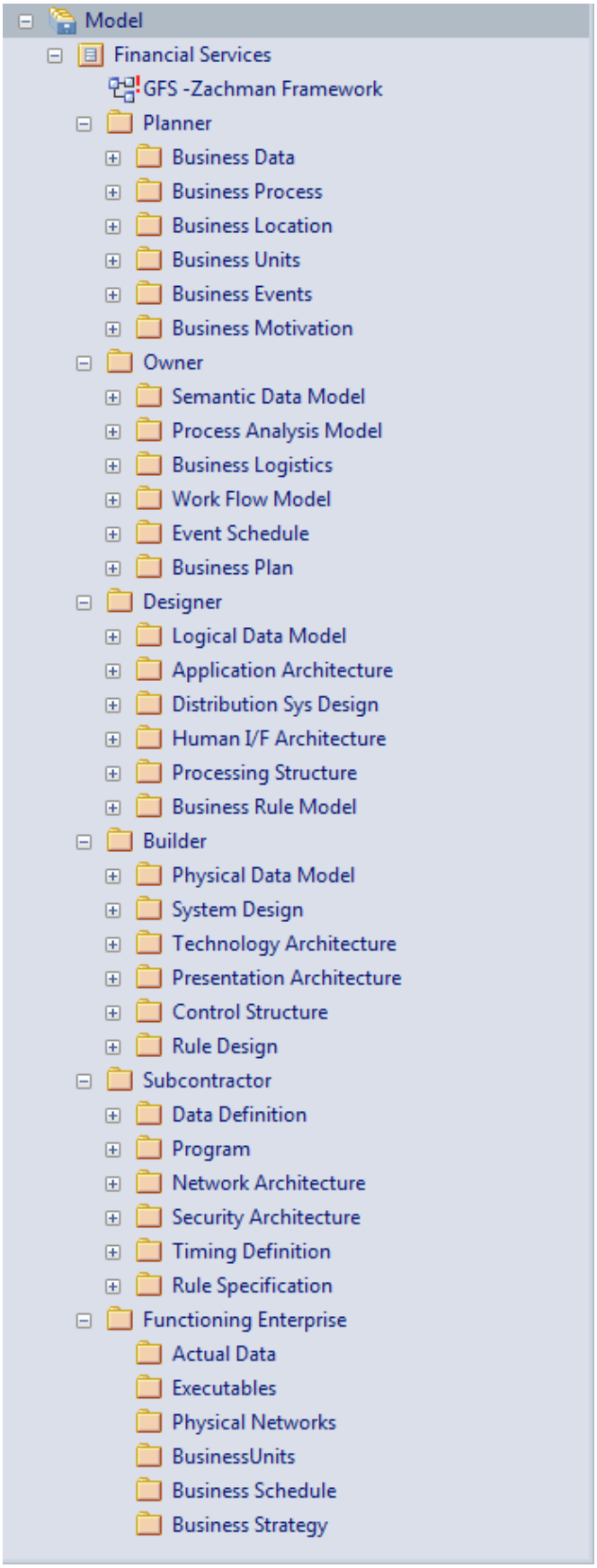
Zachman 框架是Enterprise Architect中的预定义模型。模型结构的模型级图为Zachman框架接口图，作为基于Zachman分类框架开发企业架构的模板。

每个单元链接到基础模型包中的相关 Zachman 框架图。



Zachman Framework模型Structure

Zachman 框架提供了一个框架模型模板，其中每个 Zachman蓝图（或行）都被建模为模型内部的最高级别包。属于蓝图的单元格被建模为相应行包包


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

Zachman 框架模型模板

Zachman 框架模型模板提供了模型骨架，您可以从中开发您的企业定义。

将新的 Zachman Framework模型添加到项目中

1. 右键单击根并选择 模型构建器（模式库）”。将显示 模型构建器”对话框。
2. 在 模型构建器”对话框中，单击按钮，然后从列表中选择 企业架构> Zachman”。
3. 展开组节点 “Zachman框架”，然后选择 “Zachman框架”模式。
4. 单击创建模型按钮。

Zachman 框架图表

Zachman 框架引入了支持 Zachman 分类框架建模的新图表类型。 Zachman 框架图A创建方式与Enterprise Architect中的任何其他图相同。

该技术通过 新图表"对话框提供对这些类别的图表的访问：

- 规划师
- 拥有着
- 设计师
- 建造者
- 分包商
- Zachman 框架接口

Zachman 框架图表类型

Zachman 框架进一步扩展了Enterprise Architect单元图表集以支持该框架，图表类型适用于每个单元框架。

ZFI Zachman Framework

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

Zachman 框架工具箱

工具箱的 Zachman 框架页面为图表MDG 技术支持的所有 Zachman 框架图提供了元素和关系。可以通过单击工具箱并在 查找工具箱项”对话框  指定“Zachman”来访问 Zachman 框架工具页面。图表工具箱可以停靠在图表的任一侧，或自由浮动在图表的顶部以暴露更多的表面以供编辑。

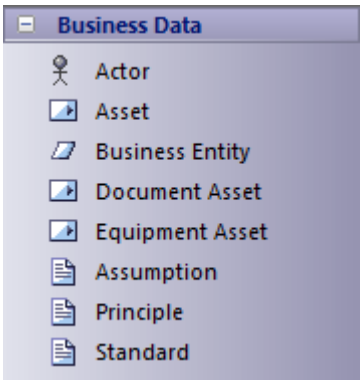
图表图形

此表显示了每个 Zachman 框架单元可以使用的图表。

扎克曼细胞	图表
规划师 - 数据	业务数据
规划师-函数	业务流程
规划师 - 位置	业务地点
规划师 - 人	组织图表
规划师 - 时间	业务事件
规划师 - 动机	业务动机
拥有着-数据	数据图和 进程映射（由插件 生成）插件)
拥有着-函数	进程分析
拥有着-位置	业务物流
拥有着-人	BPMN
拥有着- 时间	事件
拥有着-动机	Enterprise Architect思维导图和 策略地图
设计师- 数据	类
设计师——函数	活动
设计师- 位置	数据分发架构
设计师- 人	用例
设计师——时间	状态转移

设计师- 动机	企业规则模型
生成器 - 数据	物理数据模型
Builder -函数	类和 部件
生成器 - 位置	部署
建设者 - 人	用户接口
生成器 - 时间	通讯和 交互
建设者 - 动机	规则设计
分包商 - 数据	数据定义；图表的默认工具箱是自定义。
分包商——函数	未定义图表 - 在此单元中完成代码生成。
分包商 - 地点	网络架构
分包商 - 人	安全架构
分包商 - 时间	定时
分包商 - 动机	规则规范

业务数据页面



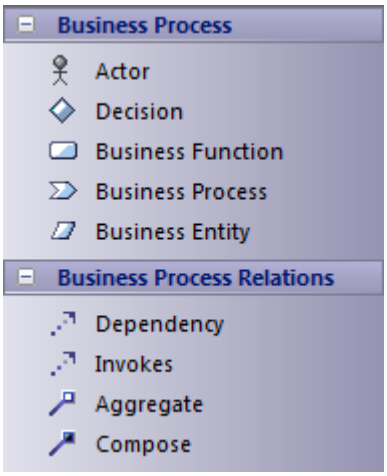
业务数据工具箱

物品	描述
参与者	为企业的利益相关者或任何其他人力资源建模。
资产	代表可以估价的企业资源。
实体业务	代表通用企业资源。
文件资产	捕获企业重要文档A资产子类型。
设备资产	捕获企业设备资源A资产子类型。
假设	捕获在信息操作中所做的假设。 应用标记值类型=企业/业务/系统/应用/技术/数据。
原则	定义在企业中制定和遵循的原则。 应用标记值类型=企业/业务/系统/应用/技术/数据。
标准	定义企业遵循的标准。 应用标记值类型=企业/业务/系统/应用/技术/数据。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在 "Object Toolbox"部分

业务流程页面



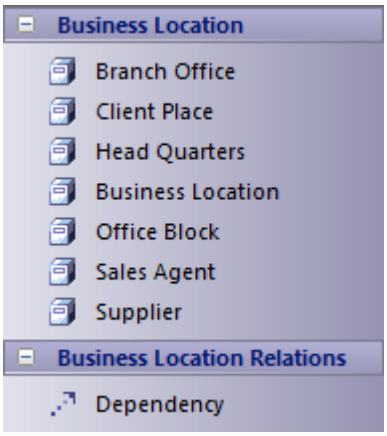
业务流程工具箱

物品	描述
参与者	为企业的利益相关者或任何其他人力资源建模。
决策	指示做出业务决策的条件进展点。
函数业务	代表由企业或企业的一部分执行的主要函数。
业务流程	代表企业或企业的一部分的函数或行为。
实体业务	代表通用企业资源。
调用	A关系定义业务流程的调用。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

业务定位页面



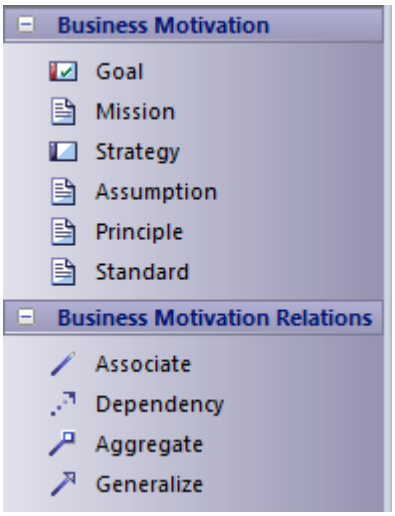
业务定位工具箱

物品	描述
分支机构	将业务位置建模为分支机构。
客户场所	将业务位置建模为客户场所。
总部	将业务位置建模为总部。
业务地点	对业务运营的位置进行建模。
办公块	将业务位置建模为 Office块。
销售代理人	将业务地点建模为销售代理人。
供应商	将业务地点建模为供应商。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

业务动机页面



业务动机工具箱

物品	描述
目标	用标记值定义的规范对企业要实现的目标进行建模。
使命	对企业的使命宣言、政策和价值观进行建模。
战略	为商业计划的战略陈述建模。
假设	对信息操作中的假设进行建模。 标记值类型=企业/业务/系统/应用/技术/数据。
原则	定义在企业中制定和遵循的原则。 标记值类型=企业/业务/系统/应用/技术/数据。
标准	定义企业遵循的标准。 标记值类型=企业/业务/系统/应用/技术/数据。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

组织图表页面

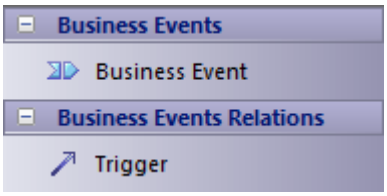
组织图表工具箱

物品	描述
董事会	捕获董事会的详细信息。
持股人	定义企业的利益相关者。
外部组织	定义不受企业直接控制但与企业有关系的任何外部业务单位。
组织单位	定义受企业直接控制的任何业务单位。
人员	捕获企业中人员的详细信息。
合同中	表示业务单位之间基于合同的关系A连接器。
效劳于	捕获团队链接详细信息A连接器；例如，利益相关者1为组织单位1工作。
监督	捕获流程监督详细信息A连接器。
控件	捕获单位负责人或负责Person信息A连接器。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

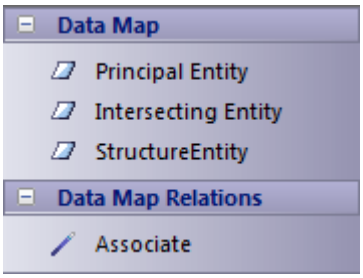
业务事件页面



业务事件工具箱

物品	描述
业务事件	捕捉企业的重大业务事件。
触发器	表示一个业务事件触发了另一个事件或业务流程。

数据映射页



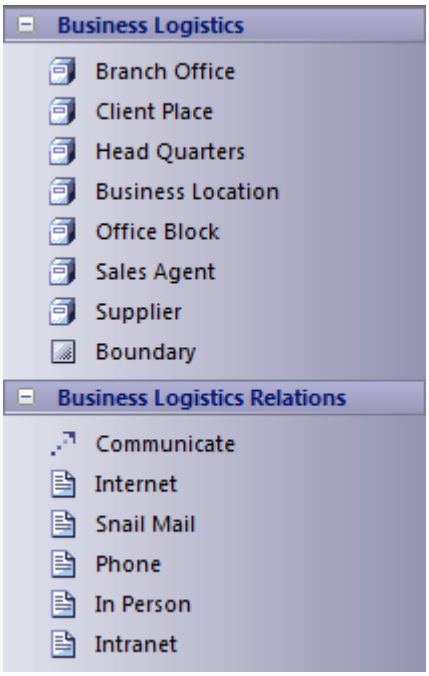
数据图工具箱

物品	描述
校长实体	代表形成企业资源的商业实体。
相交实体	规范主体实体之间的多对多关系。
结构实体	捕获潜在的基于知识的实体。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

业务物流页面



业务物流项与关系

物品	描述
分支机构	将业务位置建模为分支机构。
客户场所	将业务位置建模为客户位置
总部	将业务地点建模为总部。
业务地点	对业务运营的位置进行建模。
办公块	将业务位置建模为 Office块。
销售代理人	将业务地点建模为销售代理人。
供应商	将业务地点建模为供应商。
交流	表示一个营业地点直接与另一个营业地点通信。
互联网	表示通信方式是网络。
蜗牛邮件	表示通讯方式是邮政系统或快递服务。
电话	表示通讯方式是电话。
Person	表示沟通方式是直接的人对人。

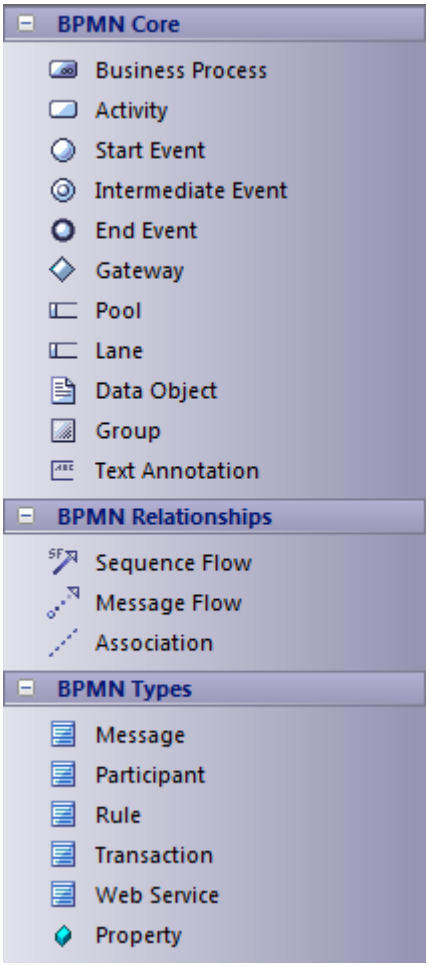
内联网	表示通信方式是本地 Intranet 或 WAN。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在 "Object Toolbox"部分

BPMN 页面

BPMN工具箱页面提供图形（核心）和非图形（类型）业务流程模型和表示法 (BPMN) 元素，用于通过 Zachman 框架技术在业务流程流程图中使用。这些元素的规格和关系由标记值定义。



BPMN工具箱

物品	描述
业务流程	定义业务流程；复合活动的延伸。
活动	定义业务流程中的活动。
开始事件	定义流程中的启动事件。
中间事件	定义流程中的中间事件。
结束事件	定义流程中的终止事件。
网关	定义业务流程中的决策点。如果条件为真，则处理以一种方式继续；如果没有，那么另一个。

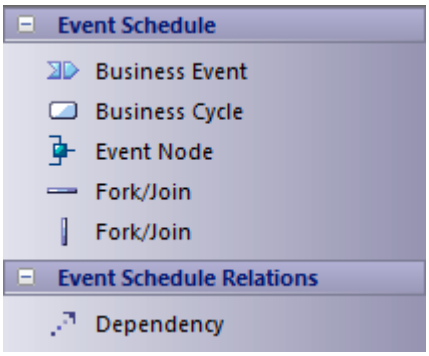
泳池	有逻辑地组织活动；元素的分区。
泳道	细分一个泳池；元素的分区。
数据物件	定义系统使用或产生的物理信息；一种工件的元素。
团体	对许多其他元素进行分组；一个边界的元素。
文本注释	A评论。
序列	定义活动的流程；控件流关系的扩展。
信息流	定义流程中的通信流程；控件流关系的扩展。
关联	将信息和工件与流对象相关联。
信息	定义消息；类元素的扩展。
参与者	定义活动的参与者；类元素的扩展。
规则	定义业务规则语句；类元素的扩展。
交易	定义活动中的事务；类元素的扩展。
网络服务	定义一个网络服务；类元素的扩展。
属性	将属性分配给元素；属性的扩展。

注记

- Enterprise Architect交付时自动安装了 BPMN技术（适用于 BPMN 1 .0. 1 . 1和 2.0），提供与此 Zachman 版本分开的 BPMN 配置文件和工具箱；要进一步使用 BPMN功能，请下载 BPMN插件从：

  https

事件计划页面



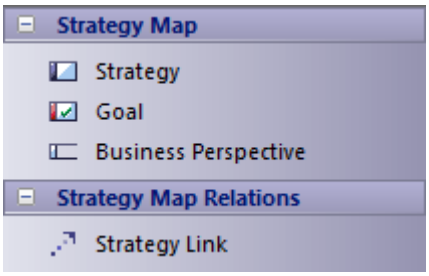
活动事件工具箱

物品	描述
业务事件	捕捉企业的重大业务事件。
业务周期	捕捉企业的主要业务周期。
事件节点	捕获业务周期中的事件点。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

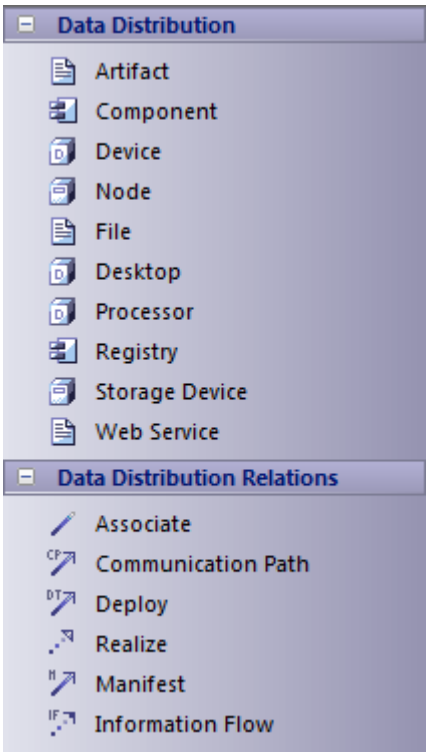
策略地图页面



策略地图工具箱

物品	描述
战略	捕获业务计划的战略声明。
目标	捕捉企业要实现的目标，以标记值定义的规范。
业务蓝图	将策略与特定类别相关联。
战略链接	表示一个策略与另一个策略或目标相关联。

数据分发架构页面



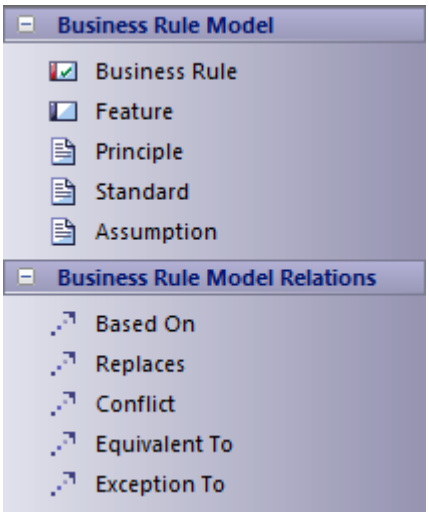
数据分发架构工具箱

物品	描述
文件	代表一个文件。
桌面	代表桌面。
处理器	代表处理器。
登记处	代表一个注册表。
储存装置	代表存储设备。
网络服务	代表一个网络服务。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

企业规则模型页面



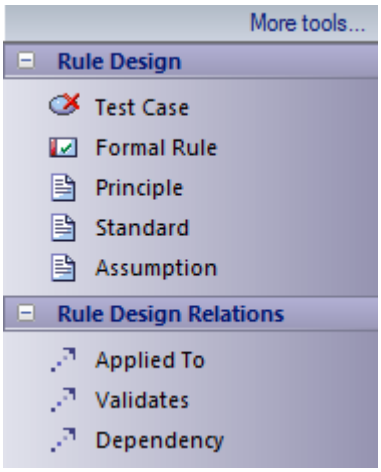
企业规则模型工具箱

物品	描述
企业规则	捕获企业规则语句。
原则	定义企业制定和遵循的原则。 标签值类型=企业/业务/系统/应用/技术/数据。
标准	定义企业遵循的标准。 标签值类型=企业/业务/系统/应用/技术/数据。
假设	捕获在信息操作中所做的假设。 标签值类型=企业/业务/系统/应用/技术/数据。
基于	表示规则基于另一个模型元素，这构成了规则的基本原理。
替换	表示一个新规则替换另一个规则。
冲突	表示一个规则与另一个定义的规则冲突。
相当于	表示一个规则等同于另一个规则。
例外	表示规则的例外。

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

规则设计页面



规则设计工具箱

物品	描述
正式规则	代表将业务规则转换为特定于技术的逻辑规则或约束语句。
原则	定义企业制定和遵循的原则。 标签值类型=企业/业务/系统/应用/技术/数据。
标准	用于定义企业遵循的标准。 标签值类型=企业/业务/系统/应用/技术/数据。
假设	用于捕捉在信息操纵中所做的假设。 标签值类型=企业/业务/系统/应用/技术/数据。
应用于	表示将正式规则应用于其他模型工件，例如场景或活动。
验证	表示一个规则模型工件一个形式

注记

- Enterprise Architect UML和扩展图通用的元素和连接器记录在“Object Toolbox”部分

网络架构页面

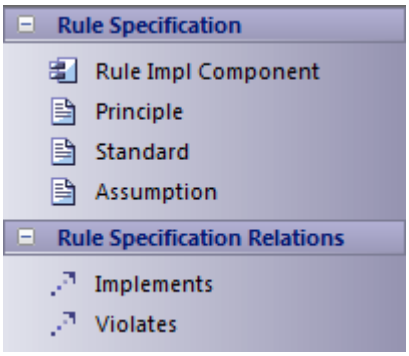
网络架构工具箱

物品	描述
工件	用于捕获信息的通用图形元素。
文档工件	通用图形元素，用于捕获网络配置细节等详细信息。

注记

- 有关工件元素的完整描述，请工件主题

规则规范页面



规则规范工具箱

物品	描述
规则部件	捕获实现规则的组件。
原则	定义在企业中制定和遵循的原则。 标签值类型=企业/业务/系统/应用/技术/数据。
标准	定义企业遵循的标准。 标签值类型=企业/业务/系统/应用/技术/数据。
假设	捕获在信息操作中所做的假设。 标签值类型=企业/业务/系统/应用/技术/数据。
工具	表示一个 Rule部件实现了一个规则。
违规	表示连接模型元素违反规则。

Zachman 框架标记值

Zachman 框架广泛使用标记值将自定义属性分配给各种 Zachman 框架元素。创建或查看 Zachman 框架模型时，建议您始终保持属性窗口停靠且可见，并展开 ŹF"部分。

访问

功能区	开始> 所有窗口>属性> 常规>标记值 浏览>门户>>窗口属性>标记值
键盘快捷键	Ctrl+2

同步标记值

有时您可能需要在模型中需要它们的所有元素中添加缺失的标记值，例如：

- 每当您通过任何方式创建新元素时，除了直接从 Zachman 框架工具箱页面中删除元素之外
- 在使用新版本技术之前，将现有模型中元素的标记值更新为最新版本的 Zachman Framework 配置文件

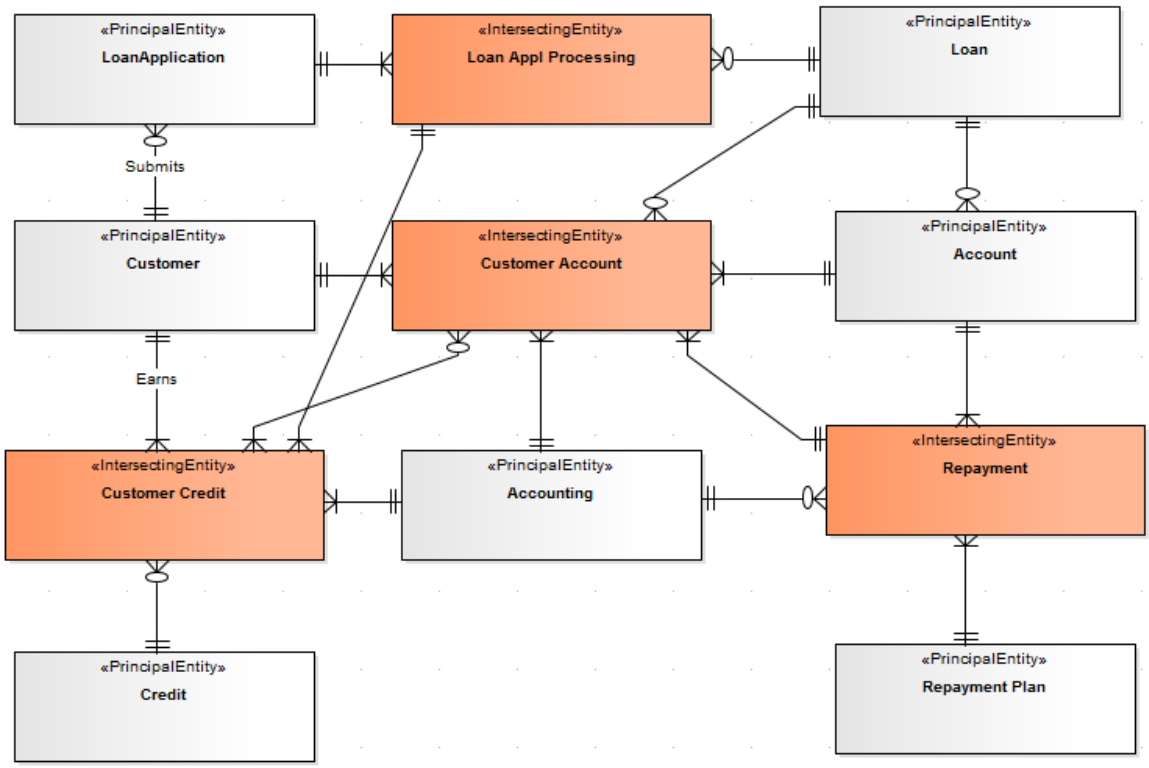
您可以使用图形工具箱的图表框架页面中的图标上的 冏步构造型"选项来执行此工具箱。

数据图分析

有效A数据映射图基本上是使用主体实体、结构实体和相交实体元素构造的实体关系图。它们之间的关系由业务规则定义。

- 主要实体是从范围内的业务实体中识别出来的
- 交叉实体用于打破主要实体之间的多对多关联，从而形成潜在的业务流程
- 结构实体代表潜在知识库的存在

这是有效数据映射图的示例：



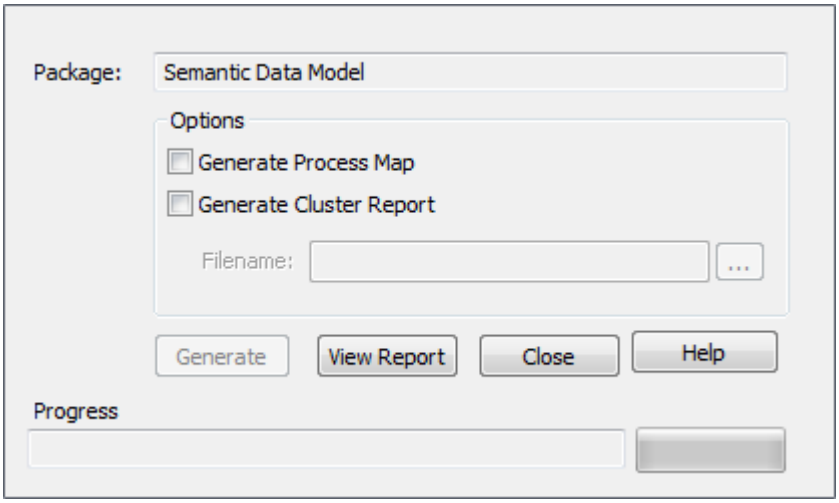
Cluster Reports 和进程是有效数据映射图分析的可交付成果。

执行数据映射图分析

在要分析的数据映射图打开并处于活动状态时，可以：

- 选择 特定>插件> Zachman Framework > Do Data-Map分析”功能区选项，或
- 右键单击浏览器窗口中的数据映射图，然后选择 特定|扎克曼框架 |做Data-Map分析的上下文菜单选项

将显示 数据图分析”对话框。

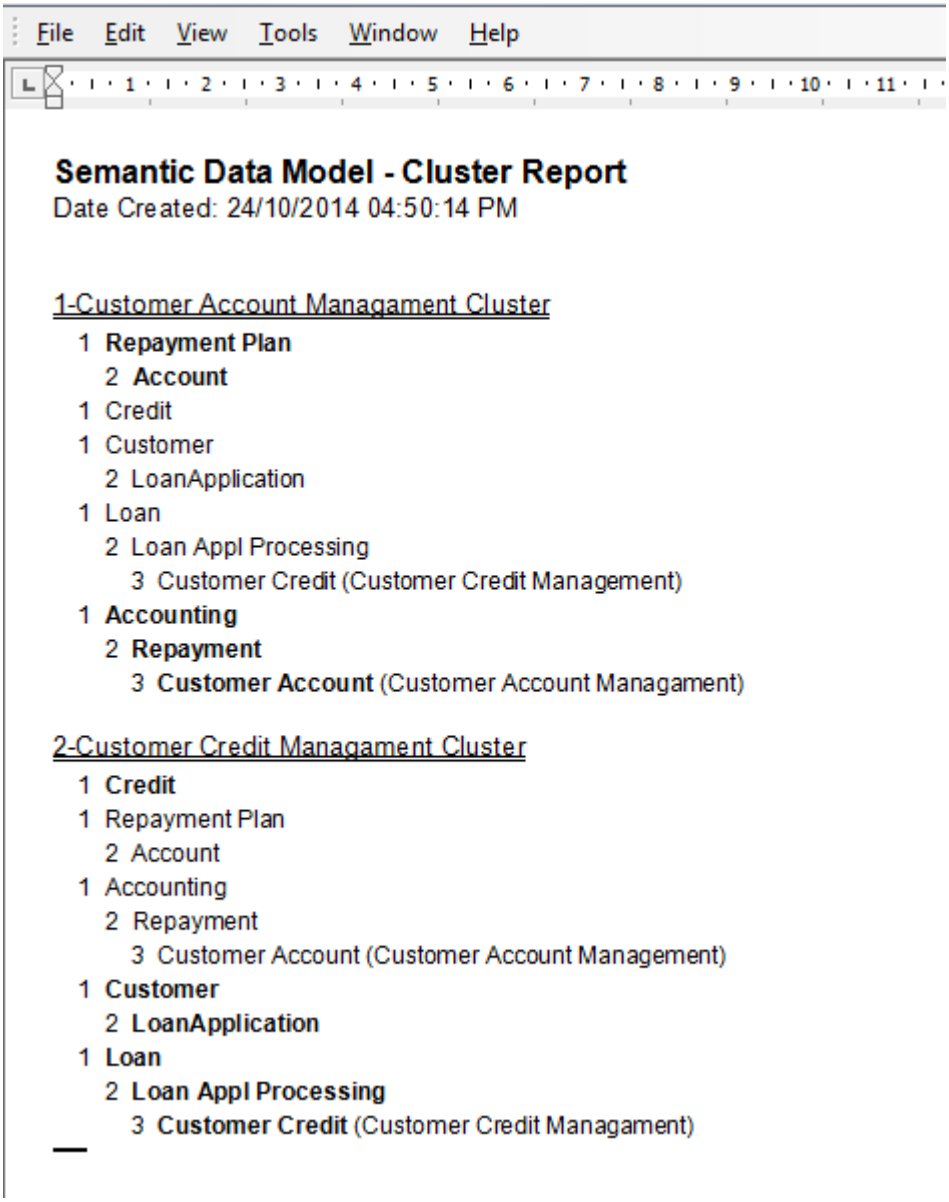


单击每个所需交付物对应的复选框。如果您选择了 生成集群报告”，还请输入保存报告的文件路径名。
点击生成按钮。

集群报告

集群是按顺序排列的逻辑相关的A组序列，这是进程执行顺序的计划。

此集群报告是为示例数据映射图生成的，格式为 .rtf。


File Edit View Tools Window Help

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)

该报告显示每个集群如何成为形成主要业务流程的流程或任务的逻辑组。

每个实体名称前面的数字是实体的相号。相对于实体的相1意味着该实体形成了潜在的资源/元素，必须在进行业务流程之前进行采购/框架。

相数大于1的实体是潜在进程，序列的执行顺序是在集群中采购/成帧相1实体后设置的。

成功完成Data Map分析后，Data Map图中每个实体的相属性都相应设置。

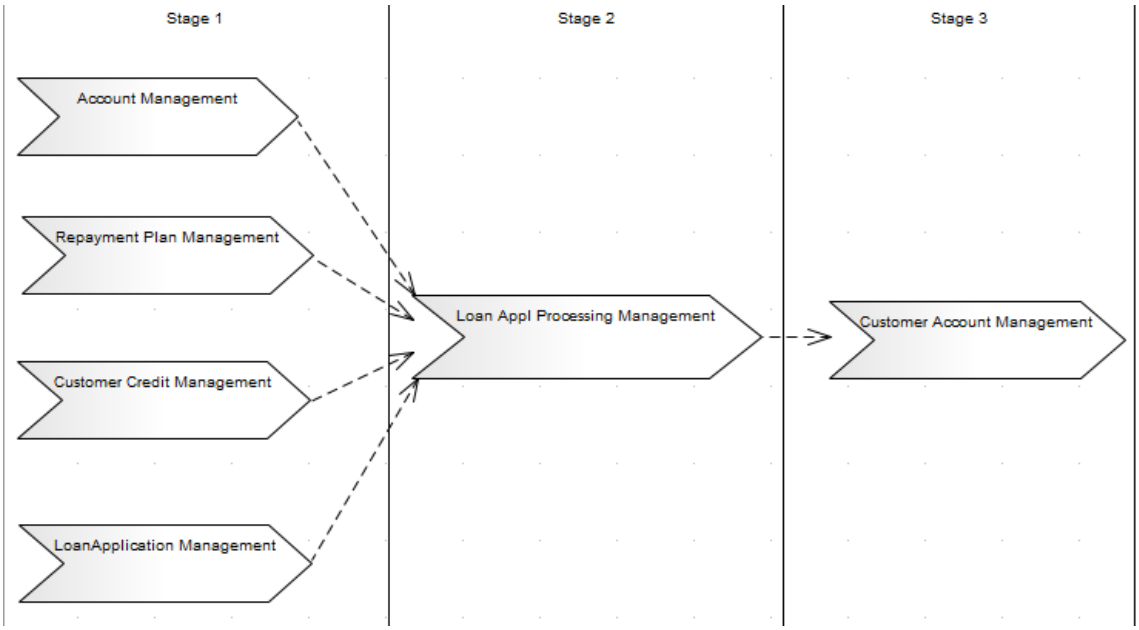
致谢

生成集群报告的算法源自《企业架构集成：快速交付方法和技术》一书（Clive Finkelstein；2006 年 4 月）。

进程映射

进程图是集群报表A可视化模型；但是，集群报告中的相1实体没有显示。进程将识别出的业务流程分组到项目的各个阶段中，作为项目的指南。

这是为示例数据映射图生成的进程映射。



业务记分卡报告模板

为了帮助您的战略管理方法，Zachman 框架提供了用于创建业务记分卡的报告模板。

生成一个业务记分卡

节	行动
1	在浏览器窗口中，单击包含您的业务蓝图和战略的包（拥有着|业务计划|战略计划包）。 业务蓝图必须拥有各自的策略。
2	任何一个： • 按 F8，或 • 选择 发布 >模型报告 > 报告生成器 >生成文档"菜单选项 将显示 生成文档"对话框。
3	在"使用模板"字段中，单击下拉箭头并选择 平衡计分卡"。
4	点击生成按钮。

验证模型

Zachman 框架向Enterprise Architect注册以接收来自用户的模型验证请求。

配置模型验证

要配置Enterprise Architect以执行 Zachman Framework模型验证，请选择：

- '设计>包>管理>验证>配置验证规则'

将显示 模型验证配置”对话框。

要仅对 Zachman 框架模型执行验证，请单击 选择无”按钮，然后单击 Zachman 框架 (ZF) 规则”复选框。点击确定按钮。

验证 Zachman 框架模型

您可以根据 Zachman 框架规则进行验证：

- 一个元素和任何连接到它的连接器
- 图表及其所有元素A或
- A包及其所有图表和元素

为此，请单击元素、图表或包，然后选择：

- '设计>包>管理>验证>验证当前包'

将显示 模型验证状态”对话框，显示验证进度。

元素的验证消息

这些错误消息可以通过 Zachman 框架元素的验证来输出。

留言

元素	图表和信息
事件节点	事件 信息：事件节点只能与业务循环一起使用 含义：事件节点已与业务Cycle以外的元素一起使用。
事件节点	事件 信息：信息触发事件节点必须定义消息 含义： 简单”标记值设置为 信息”的事件节点没有 触发器”标记值设置。
事件节点	事件 信息：规则触发事件节点必须定义规则 含义：设置了 触发器”的标记值设置为 规则”的事件节点没有设置 规则”标记值。
事件节点	事件 信息：Error触发的事件节点必须有Error定义 含义：一个 触发器”标记值设置为 节点”的事件节点没有设置 错误”标记值。
事件节点	事件 信息：多个触发事件节点必须有一个定义的触发器 含义： 简单”触发器设置为 多个”的事件节点没有 触发器”标记值标记值。
业务周期	事件 信息：业务周期必须定义事件节点 含义： A业务周期元素没有定义任何事件节点。
目标	业务动机/策略地图 信息：目标未实现 含义A目标与其他模型工件没有定义关系。
战略	业务动机/策略地图 信息：战略未实现 含义A策略与其他模型工件没有定义关系。

连接器的验证消息

这些错误消息可以通过 Zachman 框架连接器的验证来输出。

留言

连接器	图表和信息
关联	资料图 信息：DataMap关联必须有一个有效的源元素 含义：关联具有除主体实体、结构实体或相交实体之外的源元素。
关联	资料图 信息：DataMap关联必须有一个有效的目标元素 含义：关联具有除主体实体、结构实体或相交实体之外的目标元素。
关联	资料图 信息：存在可能代表潜在业务流程的交叉实体<名称>的可能性——这是一条警告消息。 含义：一个关联有一个多对多的关系，告知该关系可以被规范化。
战略链接	策略地图 信息：策略地图关联必须有有效的源元素 含义： A战略链接除了战略和目标之外还有一个源元素。
战略链接	策略地图 信息：StrategyMap关联必须有一个有效的目标元素 含义：战略链接有A目标元素，而不是战略和目标。

图表的验证消息

这些错误消息可以通过 Zachman 框架图的验证来输出。

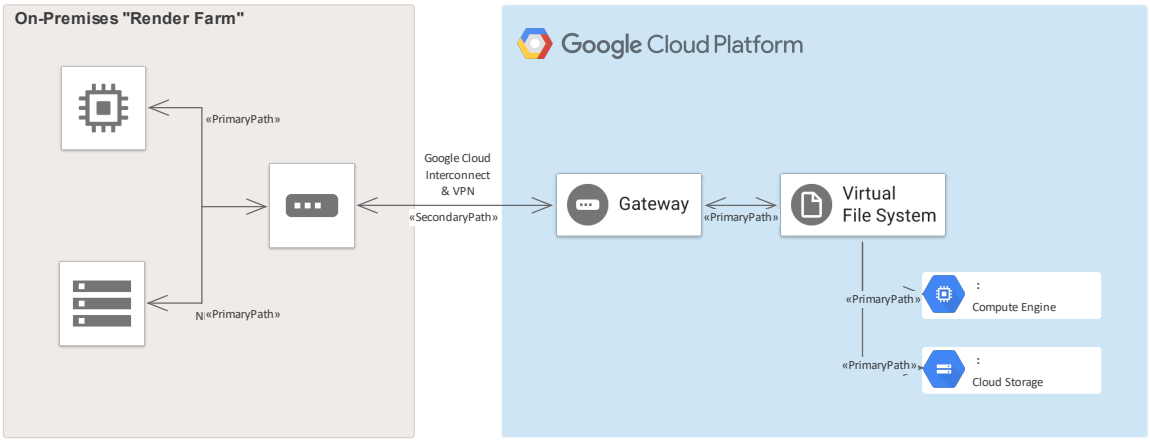
留言

图表	信息
资料图	实体必须在 DataMap 中有关系 含义：在数据映射图中，存在未定义关系的实体。

Google Cloud Platform (GCP)图标

创建指定和记录GCP虚拟基础设施的谷歌云平台图表

Google Cloud Platform (GCP)继最初的 Google App 产品之后，提供了一套云计算服务。除了一套管理工具外，它还提供一系列模块化云服务，包括计算、数据存储、数据分析和机器学习。谷歌云平台提供基础设施即服务 IaaS、平台即服务 PaaS 和无服务器计算环境。Enterprise Enterprise Architect提供建模结构，允许您创建富有表现力的 GPC 图表，以指定新的云基础设施和平台或记录现有的基础设施和平台。您还可以模型其他云基础设施和平台提供商，例如亚马逊的AWS和微软的Azure 。



GPC 图显示了本地渲染农场

虽然 Google 提供了创建图表的工具，但Enterprise Architect的强大之处在于，您可以创建可视化效果，显示与内部部署平台的关系，并且元素和服务可以与其他系统生命周期工件（如策略、业务规则相关联、需求、约束、应用程序 XML 和数据库模式，只是举几个例子。

Google Cloud Platform (GCP) UML配置文件提供了模型GCP架构图所需的所有图形（图标和图像）。图标和图像由模型生成器框架模式提供，必须先将其导入模型然后才能开始创建GCP架构图。Google网络图像模式包含 250 多种图像资产，可以将其拖放到图表上。

开始

在本主题中，您将学习如何使用支持每个部分中概述的谷歌云平台图表的特征。

选择蓝图

Enterprise Architect将工具的广泛特征划分为蓝图，确保您可以聚焦于特定任务并使用您需要的工具，而不会分散其他特征的注意力。要使用谷歌云平台特征，您首先需要选择此蓝图：

 <透视名称> >分析>谷歌云平台

设置蓝图可确保默认情况下谷歌云平台图表、工具箱和蓝图的其他特征可用。

示例图表

示例图提供了对该主题的可视化介绍，并允许您查看在指定或描述定义云架构的方式时创建的一些重要元素和连接器，包括：可用区、VPC、子网、EC2、RDS 等.

导入谷歌云平台模式

在您开始创建 GCP 图表以指定或记录您的云服务之前，您需要首先从模式中导入图形。这会将所有 GCP 图标作为组件注入浏览器窗口中的选定位置。

创建谷歌云平台图表

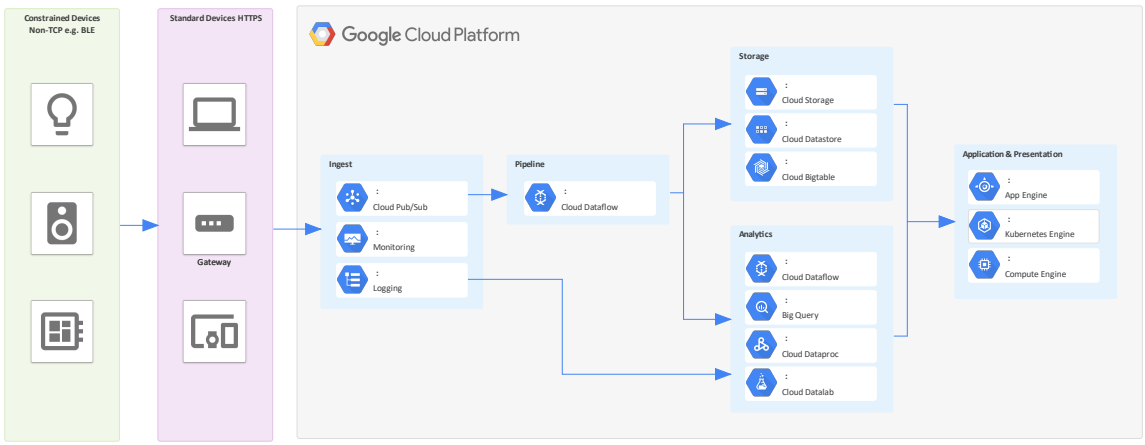
导入 GCP 图像后，创建 GCP 图表就很简单了，因为浏览器窗口和工具箱中的所有图标，包括 App Engine、计算引擎、虚拟文件系统和网关都可用。您只需创建一个图表，然后从 GCP浏览器包或工具箱中拖放元素。

更多信息

本部分提供了指向其他主题和资源的有用链接，您在使用谷歌云平台工具特征时可能会发现它们很有用。

示例图表

使用 GCP 图表，您可以模型云架构。您可以从导入的 GCP 图标、GCP 工具箱或从浏览器拖动的现有元素向图表添加新元素。此示例是传感器流摄取和处理图。



显示传感器流摄取和处理的 GCP 图

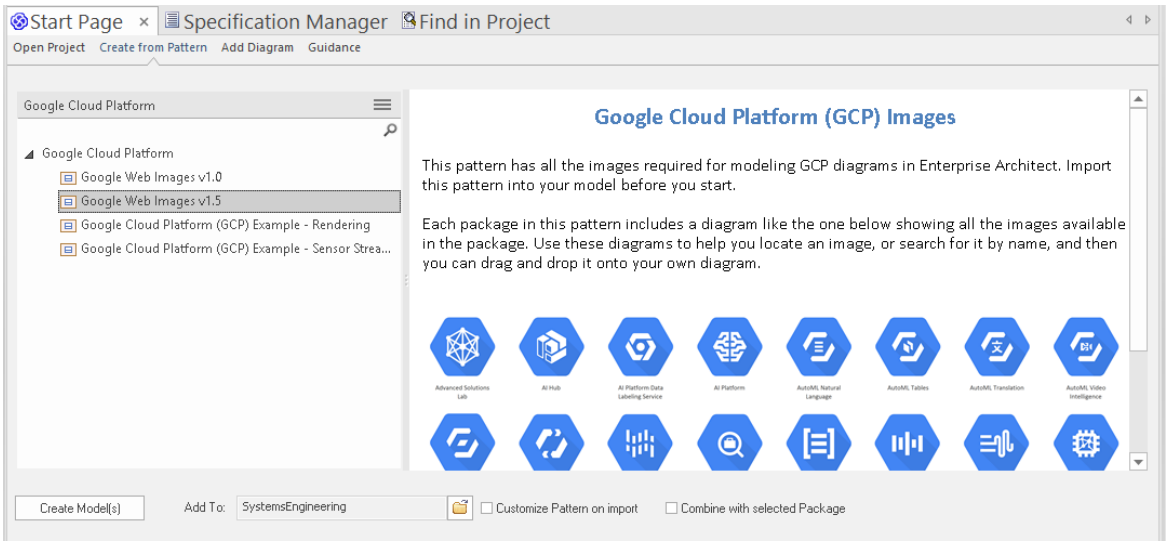
导入谷歌云平台模式

在将 "Google网络"模式导入模型之前，单击  图标并选择 分析>谷歌云平台"蓝图。

这会自动在 谷歌云平台"页面打开模型构建器。

单击浏览器窗口中的目标包，然后单击 "Google网络"模式并单击 创建模型按钮。

注记：当您的模型中有网络图像包时，请勿将其复制到模型中的其他位置或将其另存为 *XMI*；始终使用模型构建器将模式导入到新模型中。其原因是所提供的图表工具箱模式（此处描述）通过 *GUID* 引用图像资产。复制图像资产将为它们提供新的 *GUID*，并且图表工具箱模式将不起作用。



模型构建器中有一些示例模式，展示了图表中图像的典型用法，这些示例模式是从 谷歌云平台" Powerpoint 复制而来的。

创建谷歌云平台图表

您可以通过右键单击其父包并选择 新建图表”菜单选项来创建图表，以显示 模型生成器”对话框的 图表生成器”页面。

如果您没有选择谷歌云平台蓝图，请单击 类型”字段中的下拉箭头，然后选择 分析>谷歌云平台”。

在 图表”字段中为图表输入适当的名称，在 选择面板”中单击 谷歌云平台”，在 图表类型”面板中单击 “Google”，然后单击确定按钮。打开图表工具箱的 谷歌云平台”页面，包括：

- 区域
- 源
- 人工智能与机器学习
- API 管理
- 计算
- 数据分析
- 数据库
- 开发人员工具
- 通用卡
- 混合和云
- 物联网
- 管理工具
- 迁移
- 联网
- 无服务器计算产品卡（展开的）
- 安全
- 无服务器计算
- 储存
- 路径

注记， GCP图会自动设置为自定义样式，当您右键单击图中的元素时，您可以使用格式工具栏上的自定义样式图标。

Google网络图像模型生成器模式中的每个包都有一个图表，显示包中包含的每个图像。

要将其中一个图像添加到您的图表中，请通过以下方式在浏览器窗口中找到它：

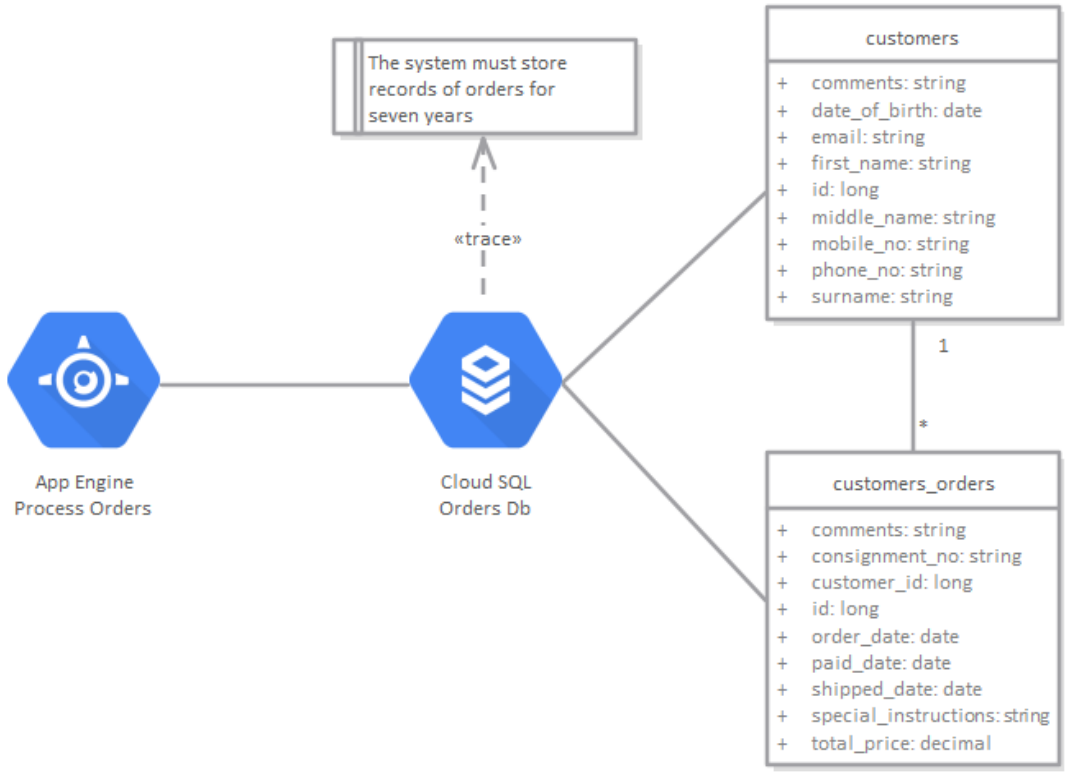
- 按名称搜索或
- 打开您认为应该位于的包的图表，在图表中找到它并按 Alt+G 在浏览器窗口中突出显示图像资产

现在将图像资产拖放到您的图表上。您可以选择：

- 将其添加为带有图标的元素
- 将其添加为带有图像的元素，或
- （如果您已经从图标创建了元素）添加为链接

跟踪工件

您可以创建富有表现力的图表，以显示 GCP 元素如何与项目中的其他工件相关联。这是通过将任何 GCP元素放入图表并在 AWS 元素和其他元素（如需求、用户案例、概念、逻辑和物理数据库库表）之间创建跟踪、依赖、关联或其他关系来实现的。



GCP 图显示了一个需求的跟踪和两个数据库库表。

更多信息

Sparx Systems Enterprise Architect是一款出色的Google Cloud Platform (GCP)图表绘制工具，为可视化GCP特征提供广泛支持。其详细的图表绘制功能使用户能够创建GCP架构的全面可视化表示，包括组件、服务及其关系。

版信息

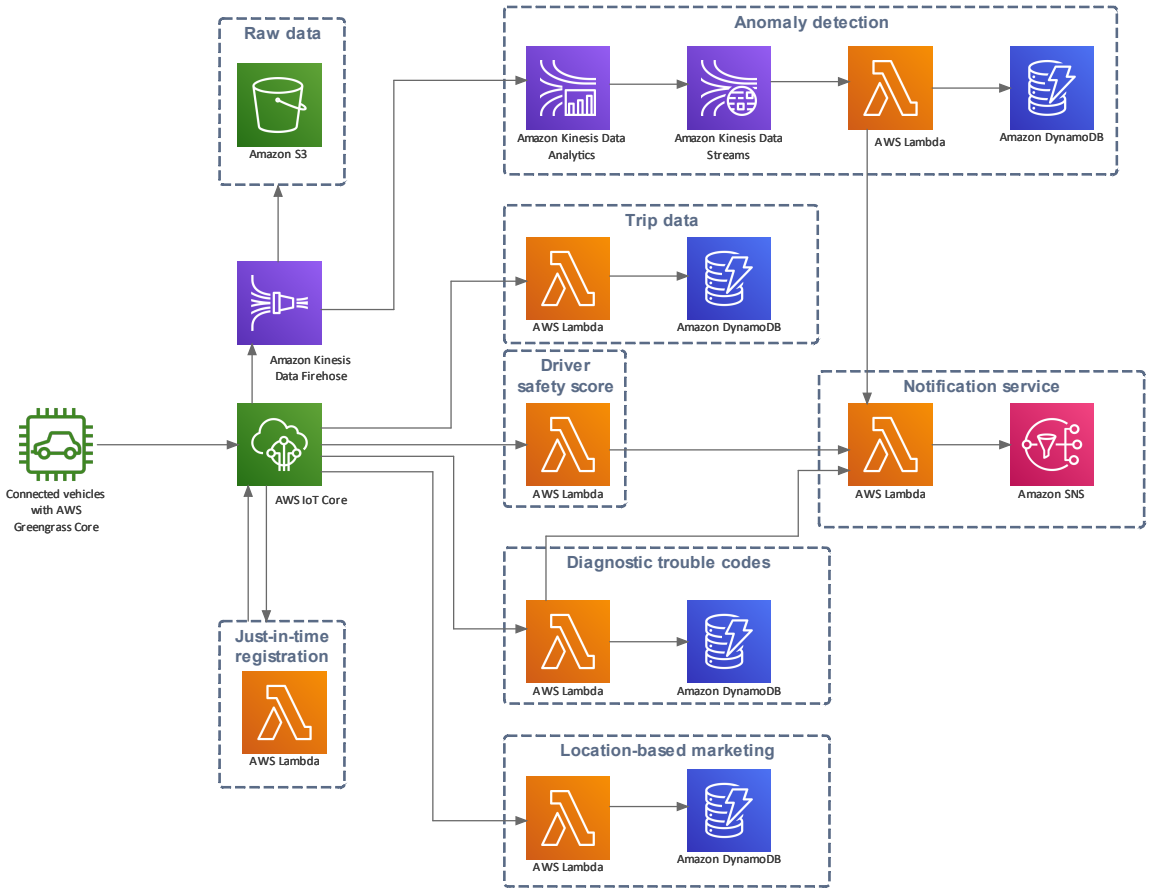
此特征在Enterprise Architect的企业统一版和终极版中可用，从 15.0 版开始。

Enterprise Architect版本 15.2 支持 GCP 图形文件的版本1 .5。

Amazon Web Services (AWS)

创建指定和记录 AWS 虚拟基础设施的Amazon Web Services图表

Amazon Web Services (AWS) 是在云环境中定义 IaaS（基础设施即服务）和 PaaS（平台即服务）的服务市场领导者之一。这些服务可以单独使用，但更常见的是组合使用以创建可扩展的云应用程序和服务，从而减少与基础设施配置和设备管理（如计算、储存和网络设备）相关的任何延迟和问题。 Enterprise Architect提供建模构造，允许您创建富有表现力的 AWS 图表，指定新的云基础设施和平台或记录现有的。您还可以模型其他云基础设施和平台提供商进行建模，例如谷歌云平台和微软的 Azure。



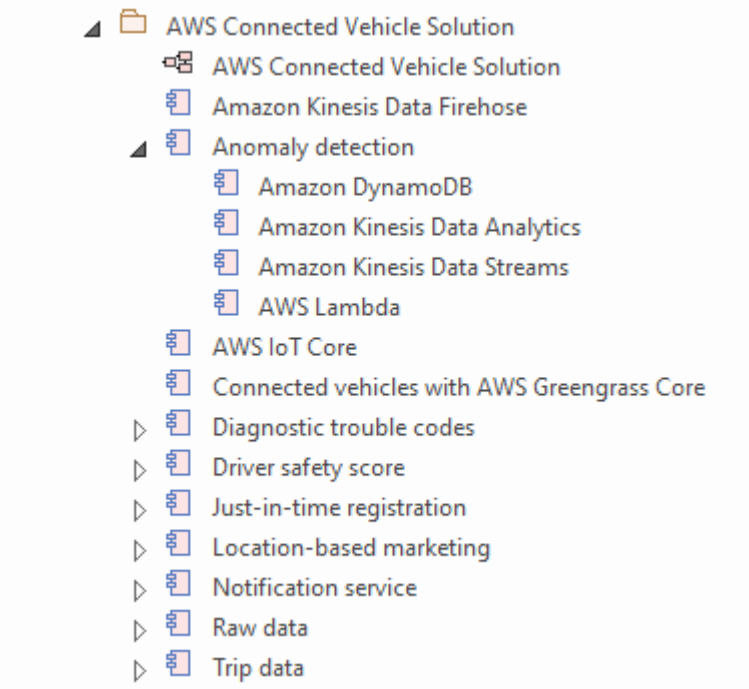
图表联网汽车解决方案图表

虽然 AWS 有执行此操作的工具，但Enterprise Architect的强大之处在于您可以创建可视化来显示与本地平台的关系，并且元素和服务可以与其他系统生命周期工件相关联，例如战略、业务规则、需求、约束、XML 和数据库模式，仅举几例。

开始

创建AWS平台图表非常简单 - 所有AWS服务构造都可从AWS包中的工具箱或浏览器获取。这允许您创建包含元素项（例如 EC2 计算和 RDS 数据库）以及容器项（例如 VPC 和子网）的富有表现力的图表。

Amazon Web Services (AWS)架构提供构建AWS架构图所需的所有图形（图标和图像）。图标和图像由模型生成器框架模式提供，必须先将其导入模型然后才能开始创建AWS架构图。Amazon AWS网络图像模式包含 350 多种图像资产，可将其拖放到图表上。



使用AWS图表非常简单；本主题将指导您在Enterprise Architect中设置AWS建模、创建图表并追踪到其他项目工件。

选择蓝图

Enterprise Architect将工具的广泛特征划分为蓝图；这确保您可以聚焦于特定任务并使用您需要的工具，而不会分散其他特征的注意力。要使用Amazon Web Services (AWS)特征，您首先需要选择 AWS架构蓝图：

<透视名称>>>分析> AWS架构

设置蓝图可确保默认情况下可以使用Amazon Web Services图表、它们的工具箱和蓝图的其他特征。

示例图表

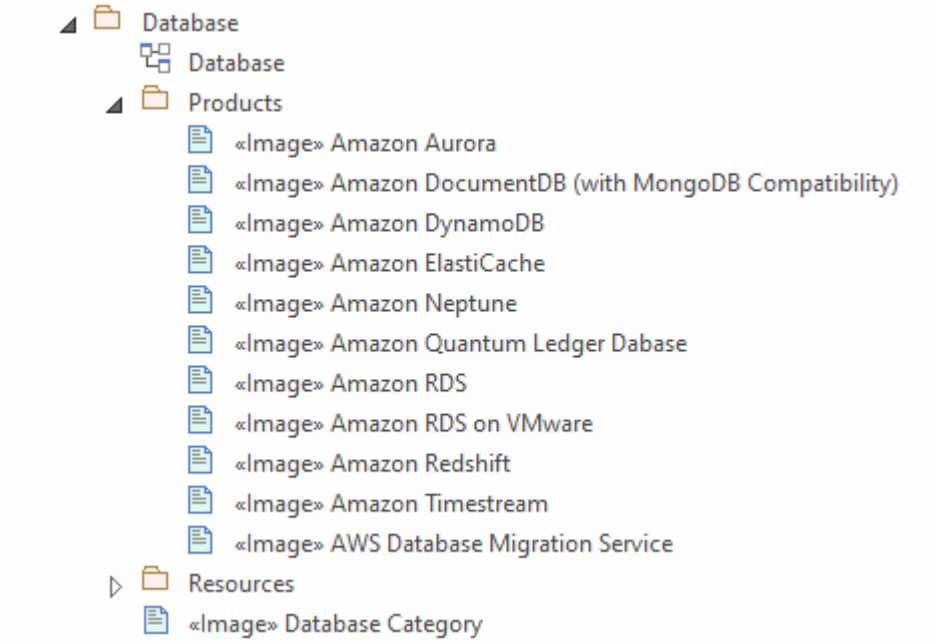
示例图提供了对该主题的可视化介绍，并允许您查看在指定或描述云架构定义方式时创建的一些重要元素和连接器，包括：可用区、VPC、子网、EC2、RDS 等.

导入Amazon Web Services模式

在开始创建 AWS 图表以指定或记录您的云服务之前，您必须首先从模式中导入图形。这会将所有 AWS 图标作为组件注入浏览器窗口中的选定位置。

创建一个Amazon Web Services图表

导入 AWS 图像后，创建 AWS 图表就很简单了，因为所有图标包括产品和资源（例如 EC2 和 RDS）以及容器（例如 VPC 和可用区）都可以分别从浏览器窗口和工具箱中获得。您只需创建一个图表，然后从 AWS浏览器包或工具箱中拖放元素。



显示 AWS 数据库产品图像的浏览器窗口。

追踪工件

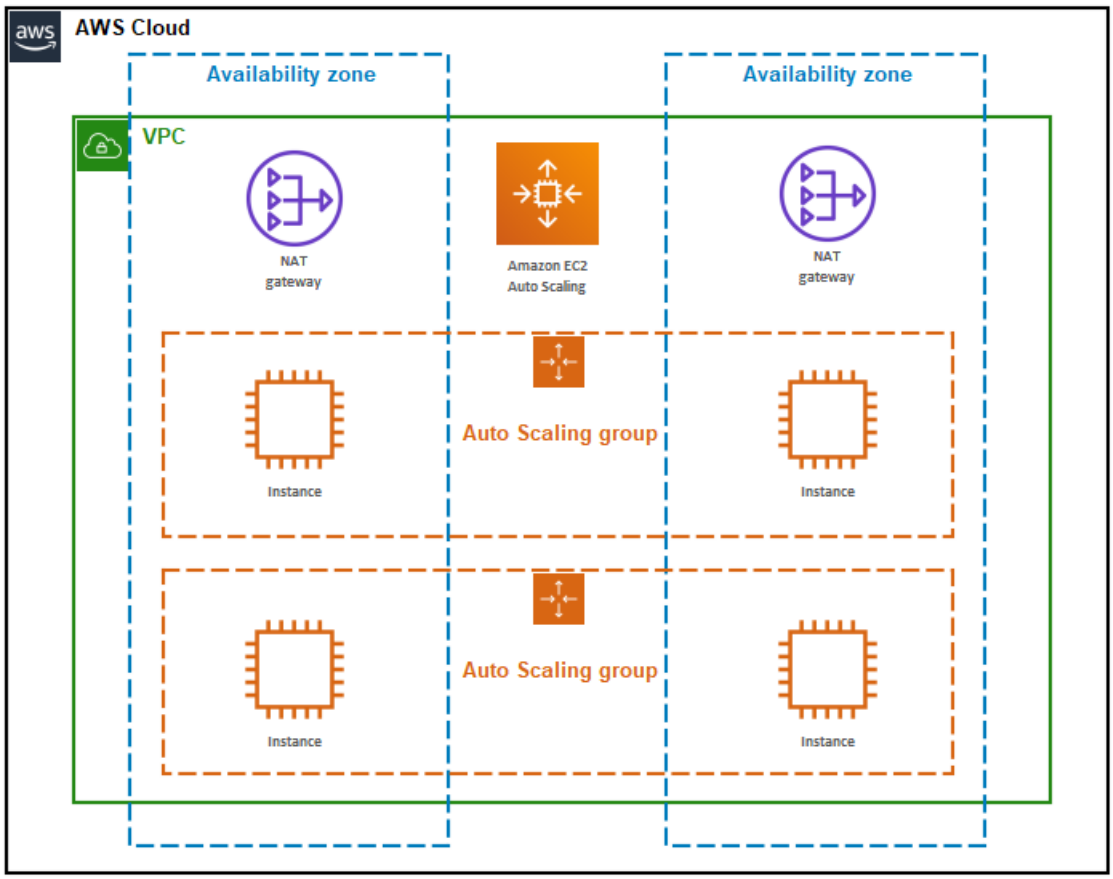
Enterprise Architect是一个适用于所有学科的协作平台，亚马逊网络服务建模的一大优势是基于云的基础设施的一部分可以与您项目中的其他域相关联。您可以将 AWS 图表中的元素跟踪到各种其他工件，包括：需求、业务规则、数据库架构、本地基础设施等。

更多信息

本部分提供了指向其他主题和资源的有用链接，您在使用Amazon Web Services工具特征时可能会发现它们很有用。

示例图表

使用 AWS 图表，您可以模型云架构。您可以从导入的 AWS 图标、AWS 工具箱或从浏览器拖动的现有元素向图表添加新元素。此示例是 Instances on AWS 示例，其中包含两个可用区和自动扩展组。



显示两个可用区和 Auto Scaling 组的 AWS 图表。

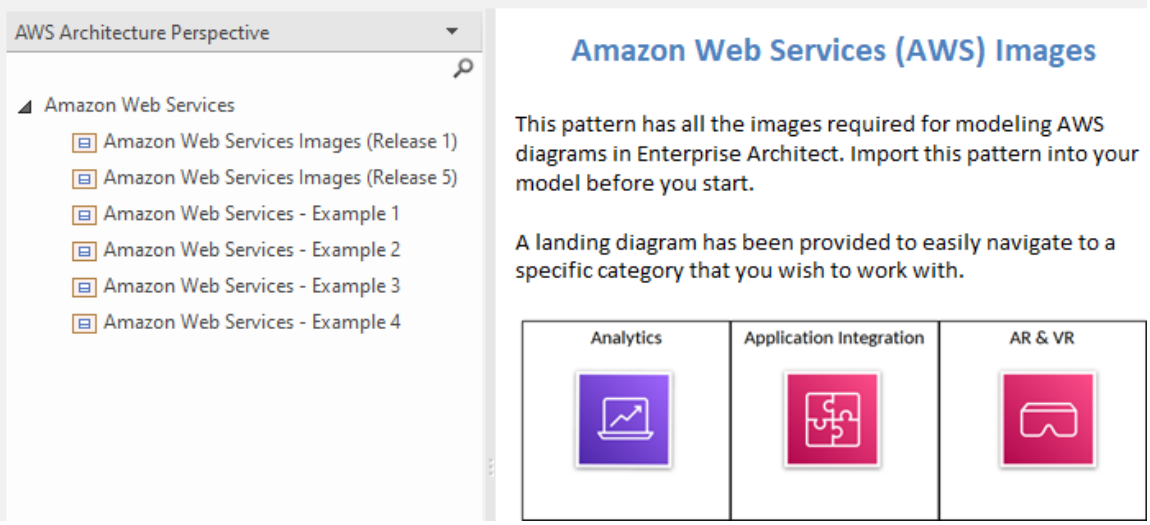
导入Amazon Web Services模式

在将 "Amazon/ AWS网络"模式导入模型之前，单击图标并选择 分析> AWS架构"蓝图。

这会自动打开AWS架构蓝图页面上的模型构建器对话框。

单击浏览器窗口中的目标包，然后单击 "Amazon/ AWS网络"模式并单击 创建模型按钮。

在模型构建器中，有三个示例模式展示了图表中图像的典型用法，这些示例模式是从 "AWS架构图标"Powerpoint 中复制的。



显示导入AWS模式的模式窗口。

注记：当您的模型中有网络图像包时，请勿将其复制到模型中的其他位置或将其另存为 *XMI*；始终使用模型构建器将模式导入到新模型中。其原因是所提供的图表工具箱模式（此处描述）通过 *GUID* 引用图像资产。复制图像资产将为它们提供新的 *GUID*，并且图表工具箱模式将不起作用。

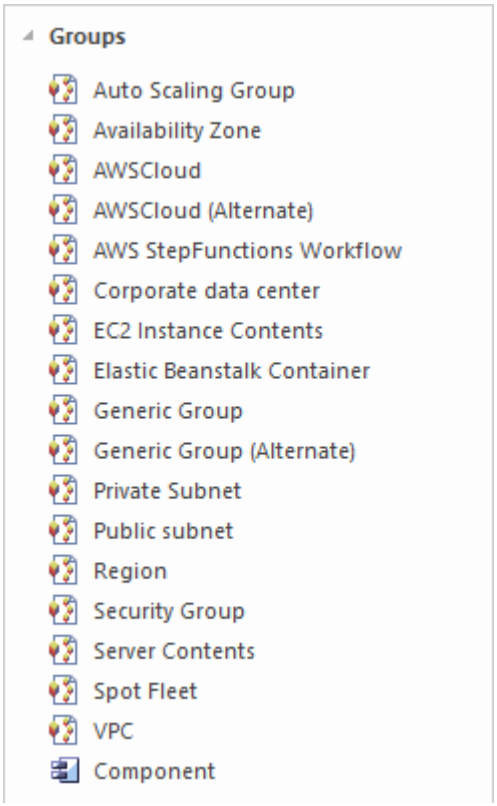
创建一个Amazon Web Services图表

您可以通过右键单击其父包并选择 新建图表”菜单选项来创建图表，以显示 模型生成器”对话框的 图表生成器”选项卡。

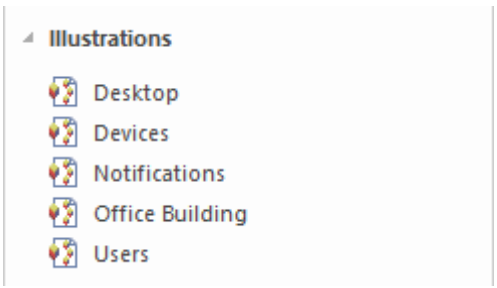
如果您没有选择AWS架构蓝图，请单击类型字段中的下拉箭头并选择 分析> AWS架构”。

在 图表”字段中为图表输入适当的名称，在 选择自”面板中单击 “AWS ”，在 图表类型”面板中单击 “AWS ”，然后单击确定按钮。图表工具箱包的AWS页面打开，包括：

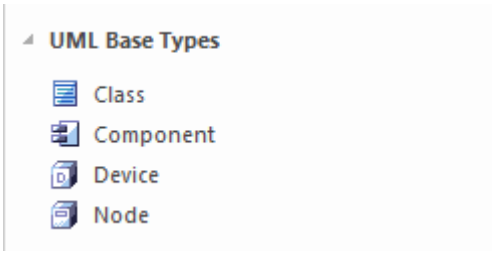
- AWS组 - 此页面提供了许多模式，这些模式将创建一个组，其图标（来自图像资产）位于左上角，名称在顶部左对齐；例外是自动扩展组和弹性负载平衡，它们的图标位于顶部中央，通用组和突出显示没有图标



- AWS插图 - 此页面提供五种说明性图案，包含用户、通知、设备、桌面和办公楼的图像



- UML基本类型 - 此页面提供了少量可在AWS图表中使用的UML元素



注记， AWS图表会自动设置为自定义样式，当您右键单击图表中的元素时，您可以使用自定义样式工具栏上的自定义样式图标。

图表工具箱栏中的所有图标都生成AWS模型生成器模式的 _General包中列出的图像资产。AWS AWS生成器模式中的其他 22 个包包含所有其他图像。每个包都有一个图表，显示包中包含的每个图像，以及两个子包 产品"和 资源"，其中包含图像资产。 模型"图像为深灰色底白色， 资源"图像为白色底深灰色。

要将其中一个图像添加到您的图表中，请通过以下方式在浏览器窗口中找到它：

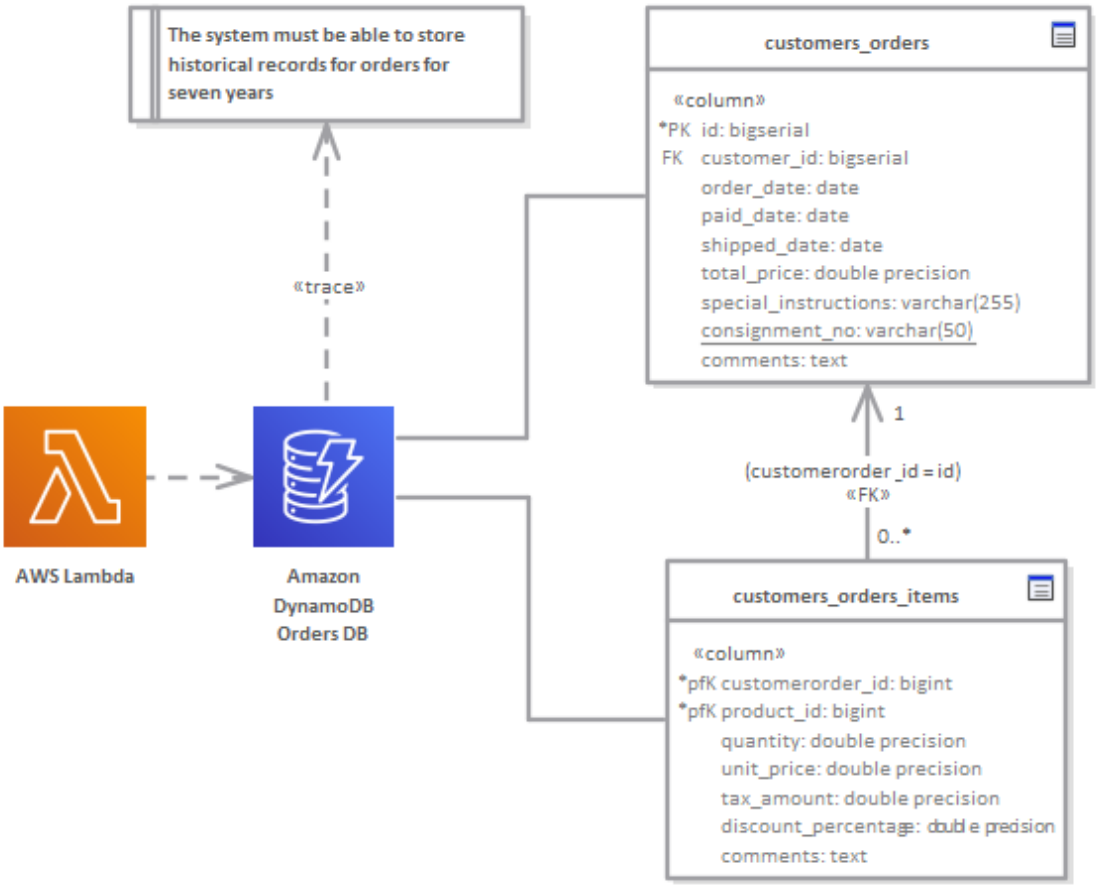
- 按名称搜索或
- 打开您认为应该位于的包的图表，在图表中找到它并按 Alt+G 在浏览器窗口中突出显示图像资产

现在将图像资产拖放到您的图表上。您可以选择：

- 将其添加为带有图标的元素
- 将其添加为带有图像的元素，或
- （如果您已经从图标中创建了元素）添加为链接

跟踪工件

您可以创建富有表现力的图表，以显示 AWS 元素如何与您项目中的其他工件相关联。这是通过将任何 AWS元素放入图表并在 AWS 元素和其他元素（例如需求、用户案例、概念、逻辑和物理数据库库表）之间创建跟踪、依赖、关联或其他关系来实现的。



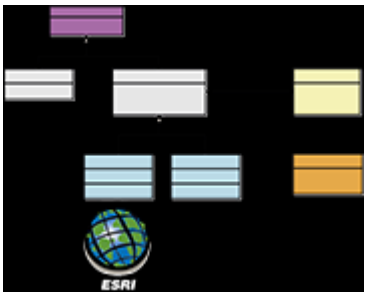
AWS 图显示了一个需求的跟踪和两个数据库库表。

更多信息

版信息

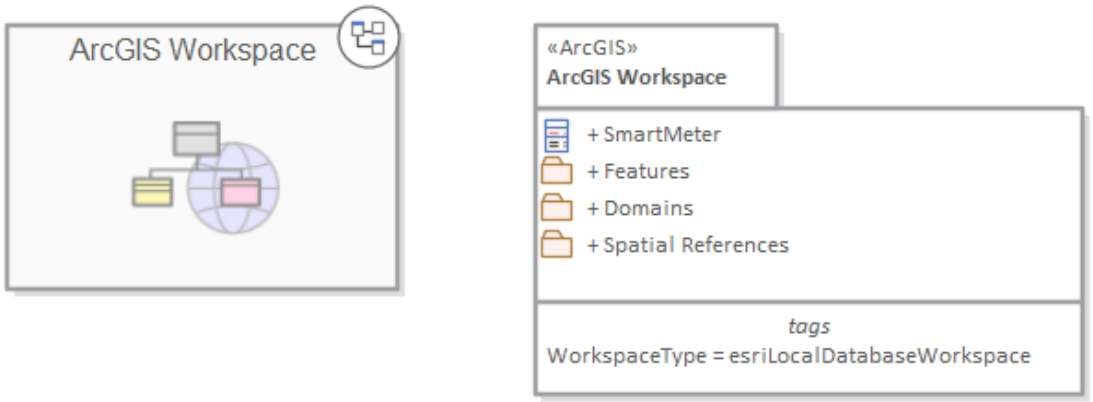
此特征在Enterprise Architect的企业统一版和终极版中可用，从 15.0 版开始。
Enterprise Architect版本 15.2.1559 支持 AWS架构图形文件的版本 7。

ArcGIS 地理数据库



交换、模型和可视化 ArcGIS 地理数据库

Enterprise Architect支持 ArcGIS 地理数据库的导入和导出，允许您在这个多功能协作平台中可视化特征和域。在最近的过去，系统软件开发和地理空间开发之间的学科出现了分离。在这个社会架构和数字颠覆的时代，几乎每个项目和努力都需要位置信息的某些方面，从简单的交付服务到农业、采矿、勘探、天气、房地产和灾难恢复系统。

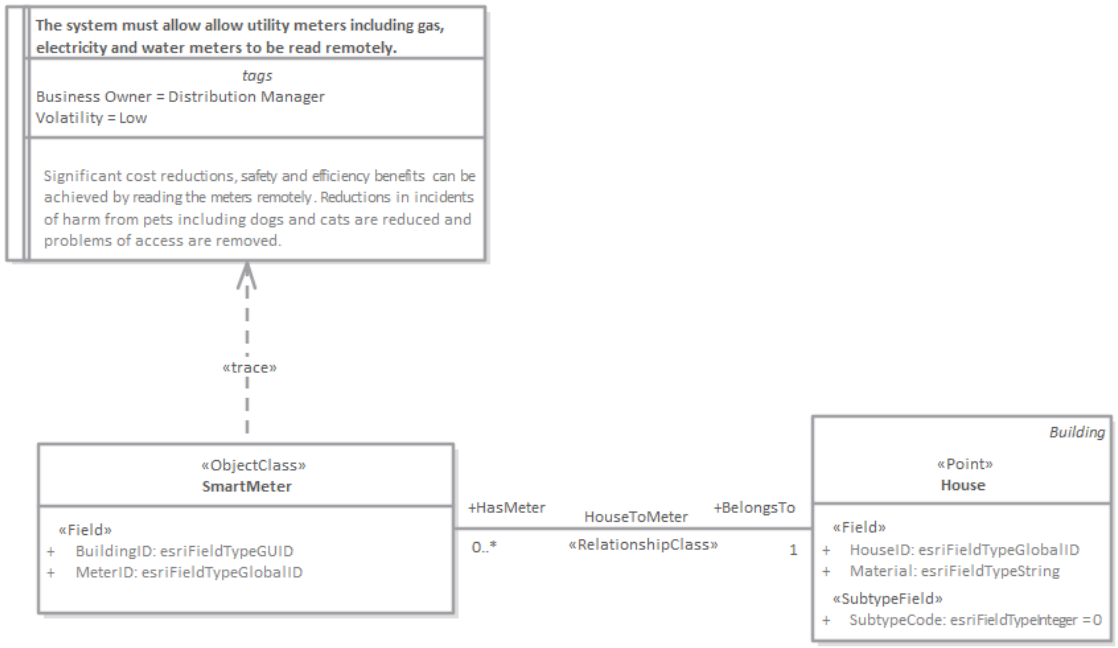


显示导航单元的包图和包含特征域和地理空间参考的包

由 Esri 开发的 ArcGIS 系统支持地理数据库的开发和管理。与其他数据库一样，使用UML等标准符号对地理数据库的设计进行模型很有用。您可以使用 ArcGIS 的UML配置文件在Enterprise Architect中执行此类建模。在Enterprise Architect中为 ArcGIS 模式建模后，您可以将模型作为 XML工作空间文档导出到 ArcGIS。您还可以通过将 ArcGIS XML工作空间文档导入Enterprise Architect来可视化现有的 ArcGIS 地理数据库方案。

开始

使用 ArcGIS特征Enterprise Architect ，您可以可视化此系统和协作平台内的地理数据库。这允许您将在传统以软件为中心的工程系统中工作的团队与定义特征和域的地理空间团队统一起来。为系统或提供系统功能的组件定义战略业务规则和要求的团队可以与地理空间团队共享模型，从而创建有助于集成和降低风险的集成模型。多学科团队可以使用包括聊天、讨论和评论在内的协作特征进行交流和协作，确保在整个系统的战略、规范、分析、设计、实施和支持过程中充分考虑地理空间组件。



ArcGIS 图表显示了智能仪表和正式系统要求之间的跟踪。

在本主题中，您将了解如何使用各节中概述的支持 ArcGIS 地理数据库的特征。

选择蓝图

Enterprise Architect将工具的广泛特征划分为蓝图，确保您可以聚焦于特定任务并使用您需要的工具，而不会分散其他特征的注意力。要使用 ArcGIS 地理数据库特征，您首先需要选择此蓝图：

<透视名称> >数据库工程 > ArcGIS

设置蓝图可确保默认情况下可以使用 ArcGIS 图表、工具箱和蓝图的其他特征。

示例图表

示例图提供了对该主题的可视化介绍，并允许您查看在指定或描述 ArcGIS 地理数据库方案（包括特征和域）时创建的一些重要元素和连接器。其他主题中的图表将显示空间参考、几何网络拓扑可以在工具中建模。

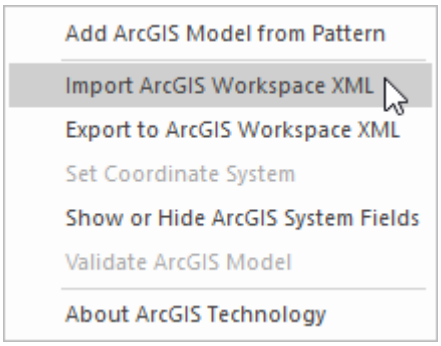
用 ArcGIS建模

本主题介绍 ArcGIS 配置文件，该配置文件提供了您将使用的图表、工具箱页面和元素，包括连通性规则和拓

扑。您可以从地理空间组中选择 ArcGIS 透视图，这将为地理数据库建模设置工具。

导入 ArcGIS XML工作空间

此功能允许您导入 ArcGIS 工作空间，该工作空间是包含地理数据库方案的 XML 文档。可以导入任意数量的模式，然后可以使用布局工具来显示或隐藏特征，包括显示或隐藏 ArcGIS系统特征的能力。



将特定功能区上的特定菜单选项导入ArcGIS工作空间。

导出 ArcGIS XML工作空间

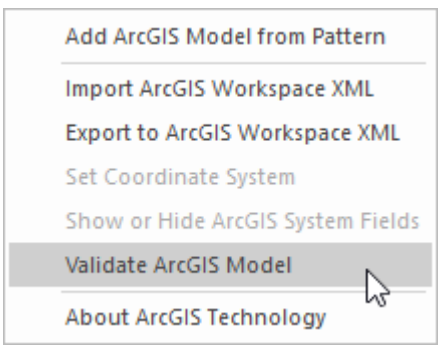
您可以在Enterprise Architect中建立模型ArcGIS 地理数据库模式，当您对它们已正确阐述感到满意时，包括添加特征和编码值和范围域，可以将它们导出到 XML 文档，然后可以将其导入 Esri 工具。

导出Modular ArcGIS Schemas

在Enterprise Architect中，您可以导出模式的谨慎部分。如果您有一个大型地理数据库方案，例如定义为行业参考模型一部分的方案，这将非常有用。可以使用此功能导出单个特征（元素），从而允许您逐步建立地理数据库

验证 ArcGIS工作空间

Enterprise Architect提供了一种验证服务，允许您检查您开发的模型是否符合格式良好模型的一组内置系统规则。



用于验证 ArcGIS模型的特定功能区上的特定菜单选项。

更多信息

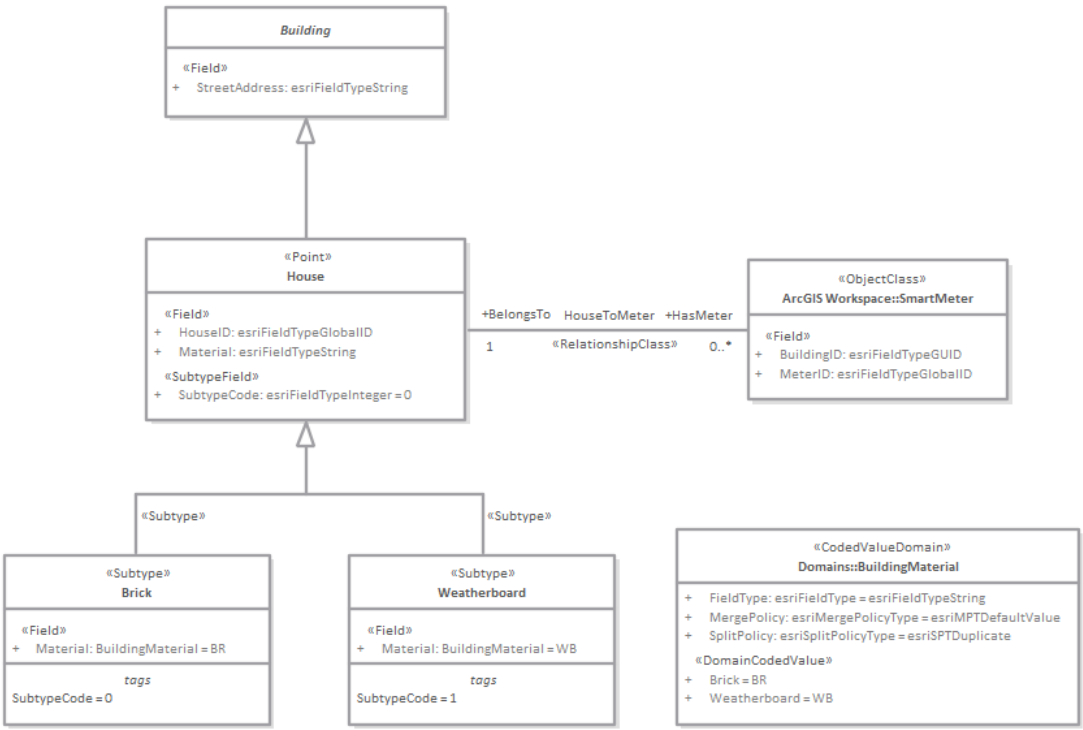
本部分提供了指向其他主题和资源的有用链接，您在使用 ArcGIS 地理数据库工具特征时可能会发现它们很有用。

示例图表

ArcGIS 图表允许您可视化构成地理数据库方案的地理特征、域和其他元素。在此示例中，建造建筑子类型化为房屋，而房屋又根据材料类型进行子类型化。房屋的子类型引用了一个编码值域，图中也显示了两个域编码值：

- 砖
- 气象板

智能电表与房屋相关A 。该房屋属于建造类型，而建造包含Street Address的属性



用 ArcGIS建模

集成ArcGIS 内置于Enterprise Architect安装程序中。该技术的A关键组件是 ArcGIS 的UML配置文件。

访问

功能区	特定>技术> ArcGIS
上下文菜单	右键包|特定| ArcGIS

特征

特征	细节
配置文件支持	ArcGIS 提供： • 将 ArcGIS 概念映射到适当构造型UML元素的 ArcGIS工具箱页面 • A模型模式，可帮助您快速开始设计地理数据库并在Enterprise Architect中使用所需的包结构 • 特定于 ArcGIS 平台的数据类型 • 快速链接器功能可帮助您在元素之间建立有效连接
ArcGIS工具箱页面	ArcGIS工具箱包含五个核心页面： • 域 - 用于编码值和范围域 • 特征和表- 用于自定义特征类型和库表 • 网络特征-用于几何网络和拓扑包 • 栅格 - 用于栅格数据集 • 工作空间- 用于 ArcGIS 工作空间和空间参考信息 另外两个工具箱将专门用于创建几何网络和拓扑图的对象分组。
空间参考	Enterprise Architect可帮助您为 ArcGIS 模式提供模型空间参考信息，包括选择预定义的坐标系和相关值。
显示/隐藏系统属性字段	通过工具箱页面提供的 ArcGIS 元素包含许多系统分配的属性，这些属性定义了 «AttributeIndex»、«SpatialIndex» 和 «RequiredField» 原型。当您将元素从工具箱拖到图表上时，这些属性在新创建的结构中不可见。 如果要显示这些系统属性，请右键单击元素并选择 特定>技术> ArcGIS > 显示或隐藏 ArcGIS系统字段”功能区选项。同样，如果您已经公开了属性并想要隐藏它们，请选择元素并再次选择菜单选项。 此选项不适用于您添加到选定元素的属性或构造型，也不适用于您尚未选择的元素。 如果您不选择任何元素，该选项将显示为灰色。

注记

- ArcGIS 在Enterprise Architect的专业版、企业统一版和终极版中可用

ArcGIS工具箱页面

ArcGIS工具箱页面提供了可用于模型ArcGIS 地理数据库概念和关系的元素和连接器。 ArcGIS工具箱由五个核心页面组成：

- 域 - 用于编码值和范围域
- 特征和表- 用于自定义特征类型和库表
- 网络特征——识别几何网络和拓扑包
- 栅格 - 用于栅格数据集
- 工作空间- 用于 ArcGIS 工作空间和空间参考信息

另外两个工具箱将专门用于创建几何网络和拓扑图的对象分组。

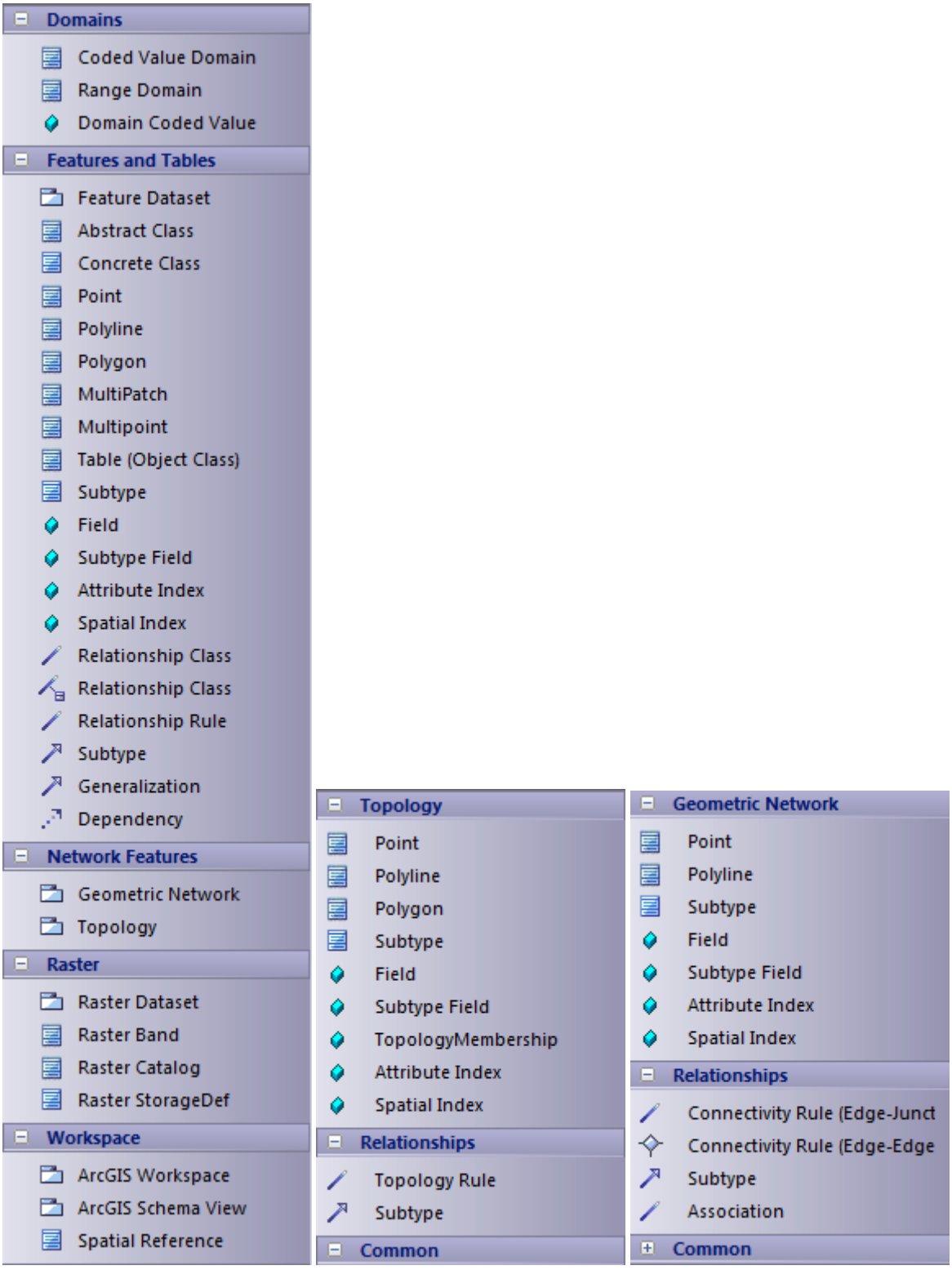
访问

在  工具箱图表 查找工具箱项”对话框并指定 ArcGIS:

- 核'
- 几何网络'或
- 拓扑'

功能区	设计>图表>工具箱
键盘快捷键	Ctrl+Shift+3

ArcGIS工具箱页面



图表图标

工具箱图标	描述
包	

ArcGIS工作空间	地理数据库工作空间包，其中包含所有 ArcGIS 建模元素。 导出该包的内容，生成 Geodatabase XML工作空间文档，该文档可导入 Esri ArcCatalog。
ArcGIS架构视图	A原型包，代表 ArcGIS工作空间包中定义的地理数据库方案的子集。如果您需要根据完成的地理数据库设计创建部分或模块化模式，ArcGIS架构视图包非常有用。您可以在 ArcGIS工作空间包下创建任意数量的 ArcGIS架构视图包。 将此元素添加到您的工作空间下的图表中，然后从它创建一个UML依赖连接器到每个包以包含在生成的 XML工作空间文档中。例如，您可以通过将UML依赖连接器绘制到适当的包来仅包含您的特征数据集和域的子集。 要导出您的 ArcGIS架构视图以用于 ArcCatalog，请右键单击它并选择 特定|ArcGIS |导出到ArcGIS工作空间XML'选项。系统会生成一个工作空间XML 文档，其中仅包含与 ArcGIS架构视图包相关的元素。 请参阅导出的导出*ArcGIS* 模式主题。
特征数据集	包含或组织具有相同空间参考、几何类型和属性字段（即特征类）的点、折线、多边形或多面体元素A原型包。 特征数据集仅在 ArcGIS工作空间包下创建；它不能在另一个特征数据集包下创建。特征数据集包可以包含其他类型的子包，但是，这对于组织大型特征数据集很有用。当导出到 XML工作空间文档时，包含任何子包的元素而忽略子包本身，从而形成 扁平化"模型层次结构。 尽管 ArcGIS 阻止在特征数据集下定义表（对象类），但Enterprise Architect为方便起见允许您在特征数据集下定义模型表。在导出时，表被放置在根级别以创建一个有效的模式。
几何网络	一个扩展的UML包，表示网络系统中特征之间的逻辑关系——在 ArcGIS 中作为几何网络实现。
栅格数据集	保存或组织栅格数据（作为栅格波段元素） A原型包。
拓扑	一个扩展的UML包，它表示来自特征数据集的一组特征类的共享几何。
元素（按字母顺序）	
抽象类	A标准的UML抽象类，表示一个概念和一组字段，可以由多个特征类共享。通过继承关系连接到抽象类的特征类获得其所有字段。由于地理数据库不直接支持抽象类，因此在从模型生成模式时将继承的字段导出到每个子特征类的定义中。
编码值域	一个扩展的UML类，表示一组可能适用于任何类型属性的有效值。
具体类	A标准的UML类，可以表示 ArcGIS 中的特征类或库表，具体取决于其父类的构造型设置。如果元素没有原型父类，则默认将其视为 ArcGIS库表（物件类）。
多补丁	一个扩展的UML类，代表 ArcGIS MultiPatch。
多点	一个扩展的UML类，代表 ArcGIS Multipoint。
观点	一个扩展的UML类，代表 ArcGIS 点。

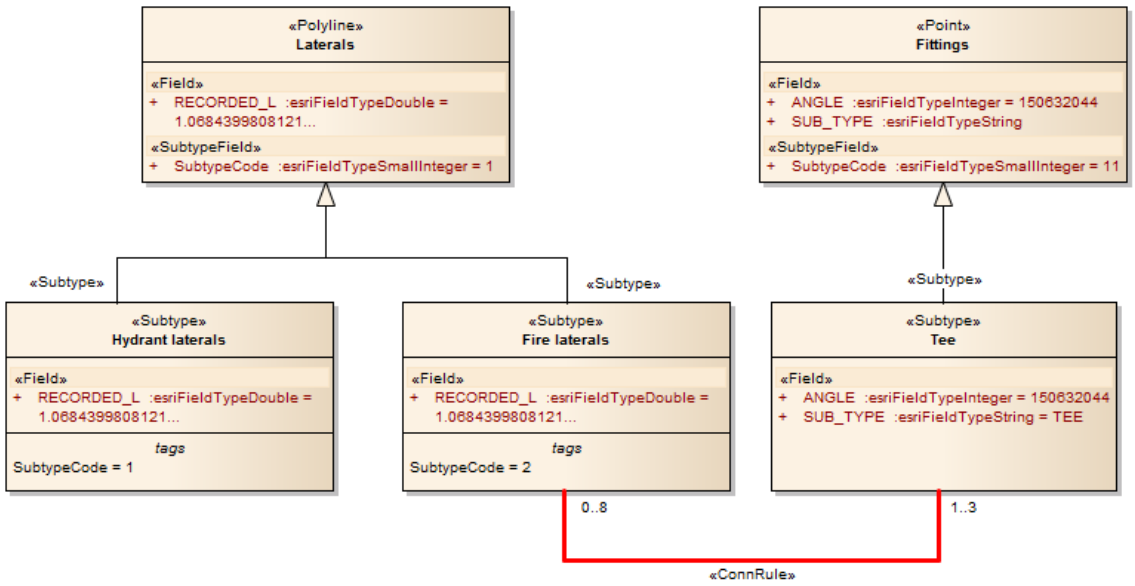
多边形	一个扩展的UML类，代表 ArcGIS 多边形。
折线	一个扩展的UML类，表示 ArcGIS 折线。
范围域	扩展的UML类，表示可能适用于数值类型属性的有效数值范围。
光栅带	扩展的UML类，表示单元值矩阵的一层。 每个栅格数据集都包含一个或多个栅格波段。
栅格目录	一个扩展的UML类，表示地理数据库中的栅格数据集的集合。
栅格存储定义	扩展的UML类，表示地理数据库中 Raster 值的存储属性；创建栅格数据集包元素时需要此信息。
空间参考	定义模式的空间参考信息的扩展UML类，例如坐标系和 XYTolerance。 您可以定义一个或多个空间参考元素，通过它们的空间参考标记值链接到其他 ArcGIS 元素。
亚型	一个扩展的UML类，包含特征数据集中元素的属性子集。
库表(物件类)	一个扩展的UML类，表示相同类型或类的非空间数据的集合。
关系（按字母顺序）	
关联	A普通的UML关联连接器。
连通性规则（边-边）	一个扩展的UML N-ary关联，它对几何网络中边缘元素之间的有效关系进行建模。 有关示例，请参阅连接规则示例。
连通性规则（边-交汇点）	一个扩展的UML关联，它对几何网络中边和连接元素之间的有效关系进行建模。 有关示例，请参阅连接规则示例。
依赖	A普通的UML依赖连接器。
概括	表示从特定分类器到一般分类器的继承。
关系类	扩展的UML关联，提供以下关系： ● 特征数据集中的两个元素，或 ● 特征数据集中的一个元素和一个物件类元素
关系类	扩展的UML关联类，提供以下之间的属性关系： ● 特征数据集中的两个元素，或 ● 特征数据集中的一个元素和一个物件类元素
关系规则	一个扩展的UML关联，确定可以在地理数据库中关联哪些子类型。
亚型	扩展的UML关联，提供特征类元素和子类型元素之间的关系。
拓扑规则	连接地理数据库中特征类和子类型元素的扩展UML关联。

属性（按字母顺序）	
属性索引	表示 ArcGIS 属性索引的扩展UML属性。
域编码值	一个扩展的UML属性，用于指定 ArcGIS 编码值域的值。
字段	在库表或特征类中表示地理数据库的 ArcGIS 字段的扩展UML属性。
空间索引	表示 ArcGIS 空间索引的扩展UML属性。
子类型字段	表示 ArcGIS库表或特征类的 孚类型”字段的扩展UML属性。
拓扑会员	一个扩展的UML属性，表示特征类的准确度等级。

连接规则示例

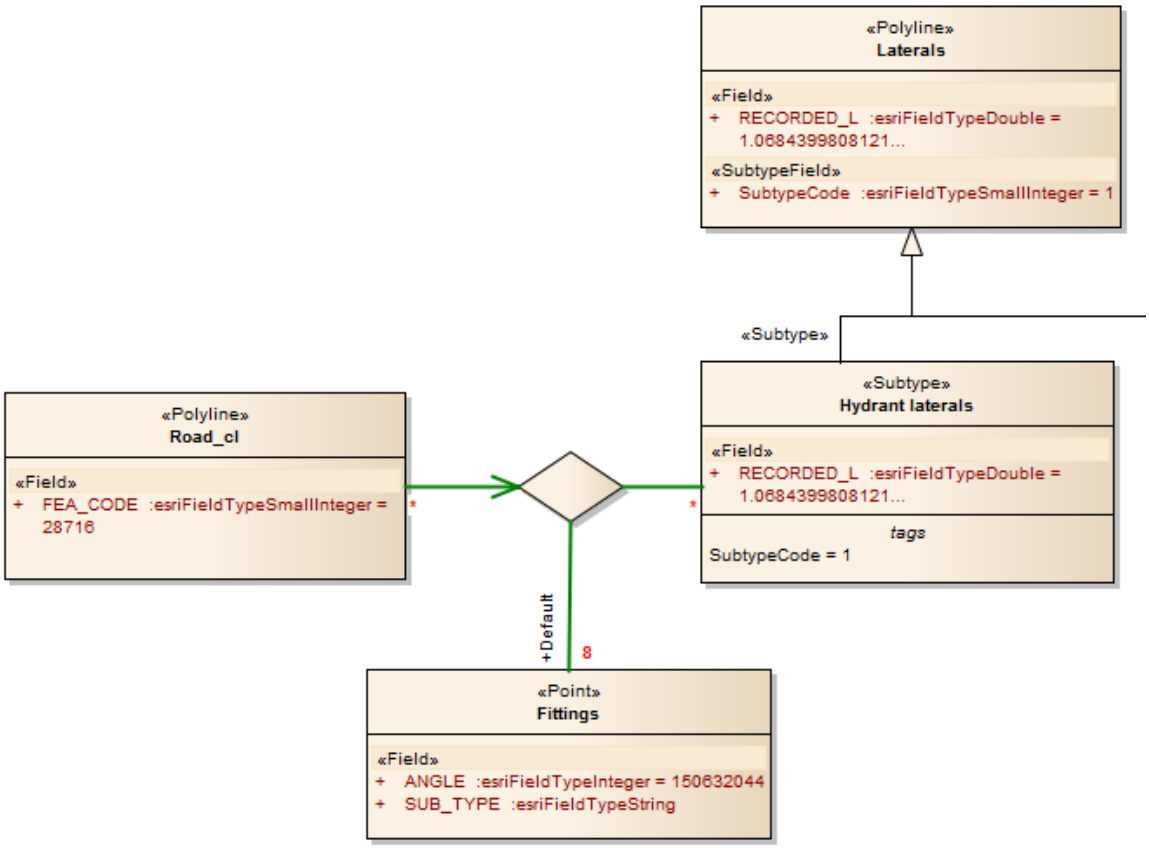
在 ArcGIS 几何网络图中，您可以使用两种连通性规则关系（边-交汇点或边-边）中的一种或另一种。这些示例说明了每种类型的使用。

边-结连通性规则



- Connectivity-Rule (Edge-Junction) 连接器是一个UML二进制关联连接器
- 连接包括一个边元素（«Point»，或以«Point»为父的«Subtype»）和一个连接元素（«Polyline»，或以«Polyline»为父的«Subtype»）
- 可以从连接器 属性"对话框上的源和目标 多重性"字段设置基数
- 您可以在连接器 属性"对话框中将 源角色"或 目标角色"字段设置为 默认"
- 此 Edge-Junction 规则中的所有元素都必须保存在 «GeometricNetwork»包中

边-边连通性规则



- Connectivity-Rule (Edge-Edge) 连接器是一个UML N-ary关联连接器
- 连接应包括两个边元素（«Polyline»，或以«Polyline»为父项的«Subtype»）和任意数量的连接元素（«Point»，或以«Point»为父项的«Subtype»）
- 建议您使用直接关联连接器，从边缘元素之一绘制到N-ary元素，以指示'from'类- 在图中，Road_cl 是设置为'from'的边缘元素类；对于其余的连接，您可以使用关联连接器连接边或结元素和N-ary ，从边缘或结元素绘制到N-ary元素，或从N-ary元素绘制到边缘或连接元素
- 基数可以从连接器 属性"对话框上的源或目标元素 多重性"字段中设置；您只需要设置连接器一端的重数 - 如果设置了两端，则仅使用目标端的重数
- 您必须使用连接器 属性"对话框中的 源角色"或 目标角色"字段将其中一个 Junction-N 元连接标记为默认值
- 此 Edge-Edge 规则中的所有元素都必须保存在 «GeometricNetwork»包中

拓扑示例

在地理数据库中，拓扑定义了地理特征之间的空间关系；也就是说，点、折线和多边形特征如何共享重合几何。拓扑是 GIS 数据库中数据完整性的基础。在Enterprise Architect ArcGIS 配置文件中，您使用 «Topology»包来模型特征类之间的数据完整性。

Enterprise Architect ArcGIS模型中的建模拓扑很简单：

1. 选择一个«FeatureDataset»包，在其中创建拓扑关系。
2. 打开«FeatureDataset»包下的图表。
3. 从工具箱图表将一个«特征»包图标拖放到图表上；这将创建一个包含拓扑定义所需的所有元素和关系的包。

Enterprise Architect中定义A拓扑具有以下特征：

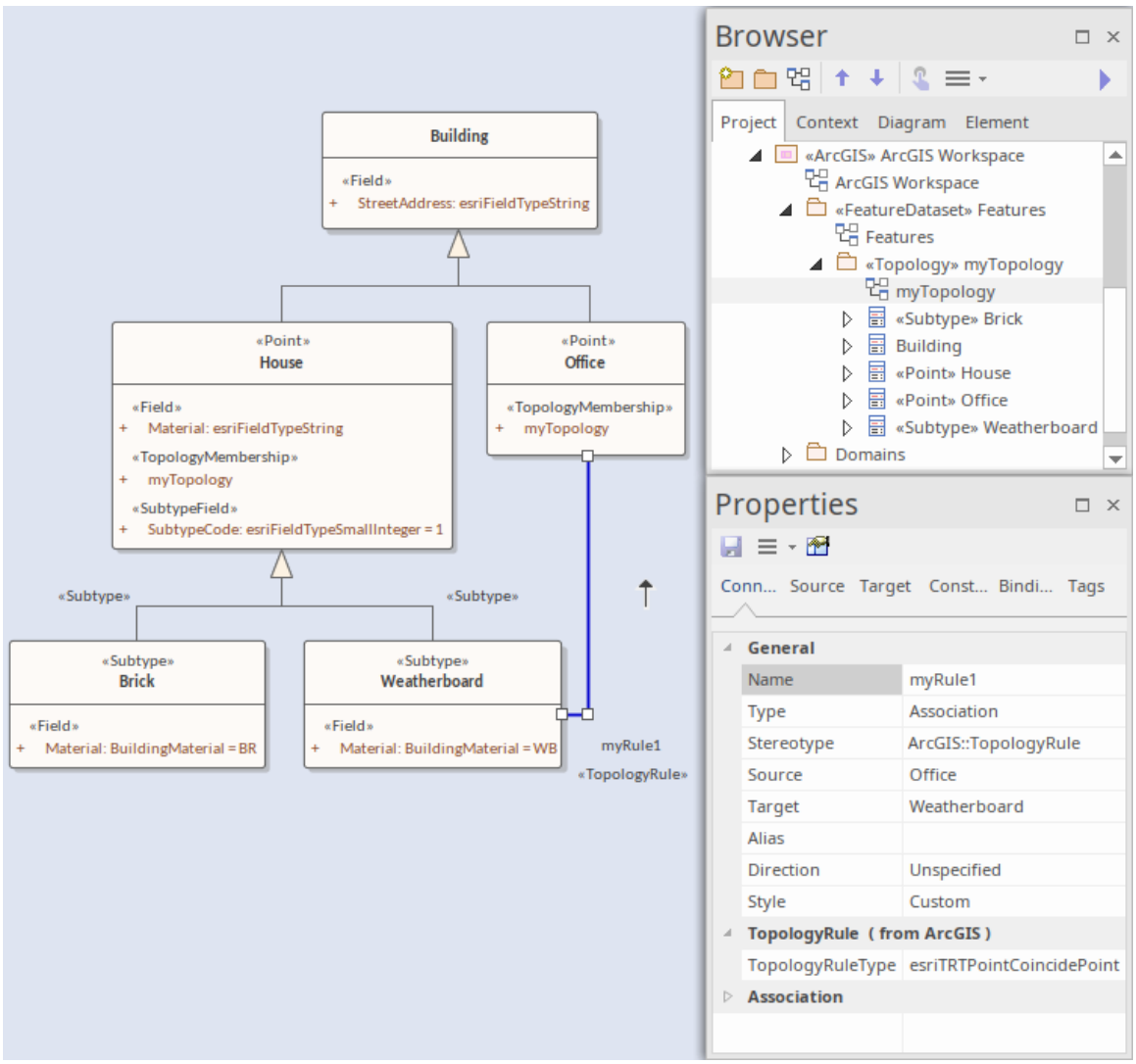
- «Topology»包不能在 « FeatureDataset»包之外创建
- 在一个«FeatureDataset»包中，可以创建多个«Topology»包
- A特征类（点、折线或多边形）只能参与一个«拓扑»包
- 特征A不能类参与«Topology»包和«GeometricNetwork»包

拓扑元素

元素	描述
名称	您可以将 Topology 的名称定义为 «Topology»包名。
特征类列表	任何一个： • 从图表工具箱或创建新的特征类 • 将现有的特征类从浏览器窗口拖到 «Topology»包中
X,Y 集群容差和 Z 簇容差	您可以使用«Topology»包的ClusterTolerance 和ZClusterTolerance标记值定义集群容差值。
准确度等级	精度等级是使用 TopologyMembership 属性的标记值定义的，您可以使用工具箱的 拓扑"页面上的 图表成员"图标创建该属性。 将此定型属性添加到每个特征类元素中，然后为每个等级设置一个值。 • 属性名称应该是«Topology»包的名称 • 您不需要设置属性的类型 每个特征类只有一个 TopologyMembership 属性。如果您未将 TopologyMembership 属性添加到特征类，ArcGIS 导出器将为您生成一组默认排名值。 XYRank 和 ZRank 的值介于1和 50 之间。
拓扑规则	拓扑规则由具有 «TopologyRule» 原型的UML关联连接器表示。您可以使用工具箱的 拓扑"页面上的 图表规则"图标来创建连接器。 使用此连接器链接： • 两个特征类（ 点"、 折线"或 多边形"）元素 • 两个 子类型"元素 • 特征A （«Point»、«类» 或 «Polygon»）元素到 «Subtype»元素 • 特征A （«Point»、«类» 或 «Polygon»）本身，或 • A子类型"元素本身

	TopologyRuleType 标签用于定义拓扑规则的类型。

示例拓扑规则连接



关系规则示例

在 ArcGIS 建模中，您可以使用关系规则来细化源特征类或库表与目标特征类或库表之间的“RelationshipClass”连接器的基数；关系类连接器仅定义初始基数，例如一对多或多对多。

Enterprise Architect中A关系规则由 «RelationshipRule» 连接器表示，这是一个原型UML关联连接器，您可以使用工具箱的 图表Core”页面上的关系规则图标创建它。您可以从连接器 属性”对话框上的源和目标 多重性”字段设置基数。

在两个对象之间创建 «RelationshipRule» 连接器时，您必须具有：

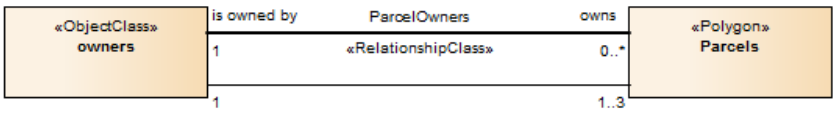
- 您要为其定义关系规则的两个对象之间的现有«RelationshipClass»连接器；如果没有连接器，则您创建的 «RelationshipRule» 在 ArcGIS 模式生成期间将被忽略
- 每端A基数范围与父 «RelationshipClass» 的基数兼容；例如，如果您在 «RelationshipClass» 连接器中定义 1-M 的基数，则 «RelationshipRule» 连接器的源端必须为1 ，而您可以将 «RelationshipRule» 的目标端设置为特定数字，例如3（请参阅本主题中的示例图）

特征关系规则还可以限制特征源类或库表中可以与目的地特征类或库表中的某种object相关的object类型。例如，如果源类没有子类型元素，则关系规则适用于所有特征。如果源类有子类型元素并且«RelationshipRule»链接到子类型元素之一，这意味着只有关联的子类型元素与«RelationshipRule»相关。同样的限制也适用于目的地特征类或库表。

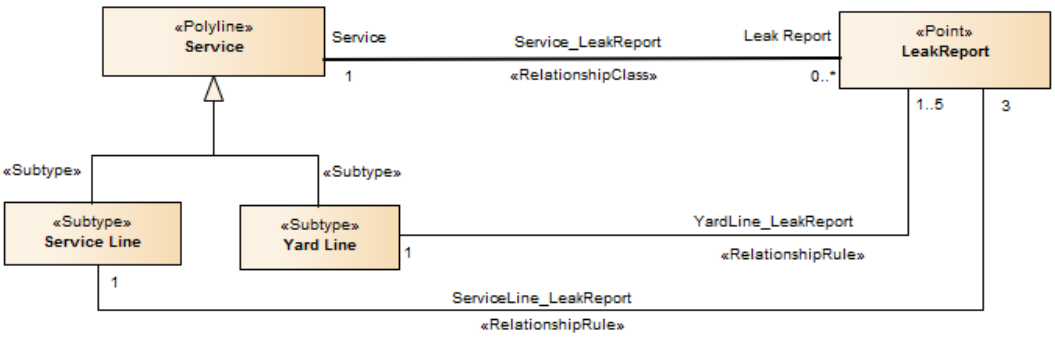
例子

该图提供了 ArcGIS模型中可能的 «RelationshipRule» 连接的三个示例。已应用A定义线粗来突出关系类连接器，并且在适当的地方隐藏了 «RelationshipRule» 构造型标签：

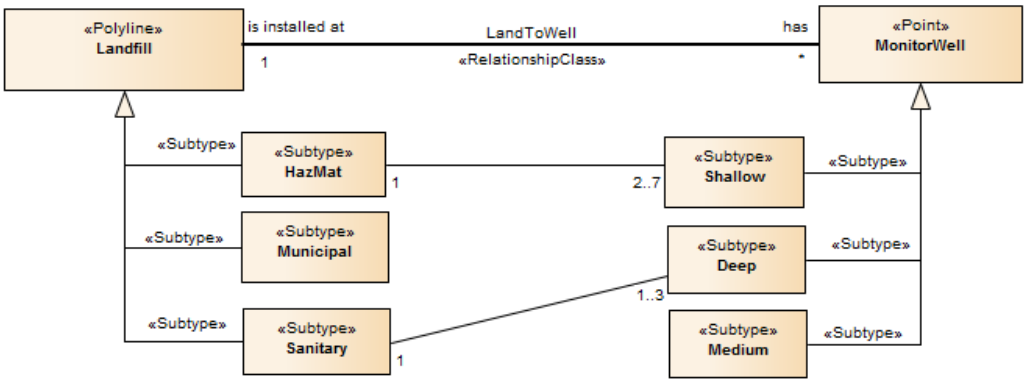
1) Relationship Rule Links feature class and object class/table.



2) Relationship Rule links feature class and subtype.



3) Relationship Rule links subtype and subtype.



设置 ArcGIS 坐标系

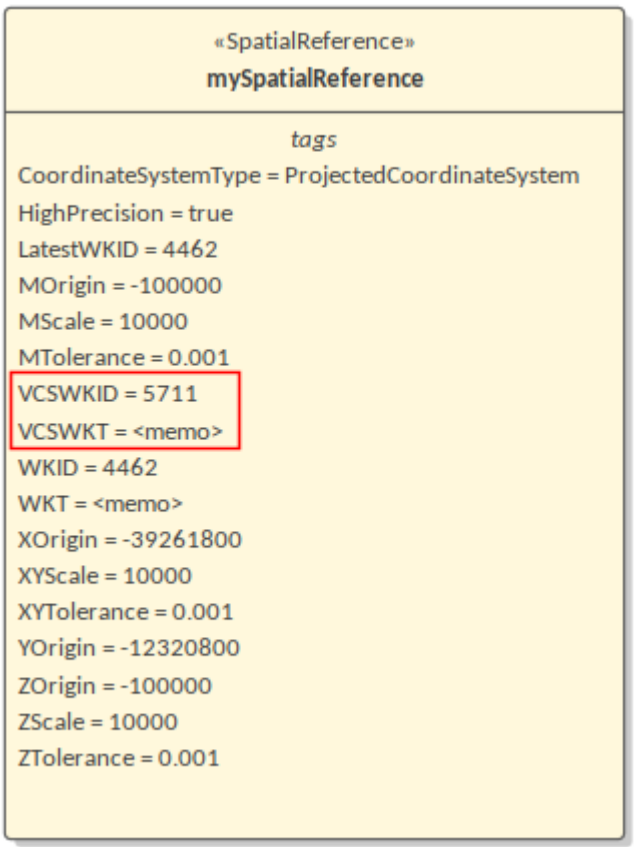
ArcGIS特征类和特征数据集使用空间参考，它由坐标系和相关值（例如 XY 分辨率和各种容差值）组成。

您可以使用定型为 «SpatialReference» 的类来捕获空间参考属性，该类可从 ArcGIS工具箱页面获得。 ArcGIS模型模式包括一个名为 Spatial References 的包，作为创建此类元素的占位符。

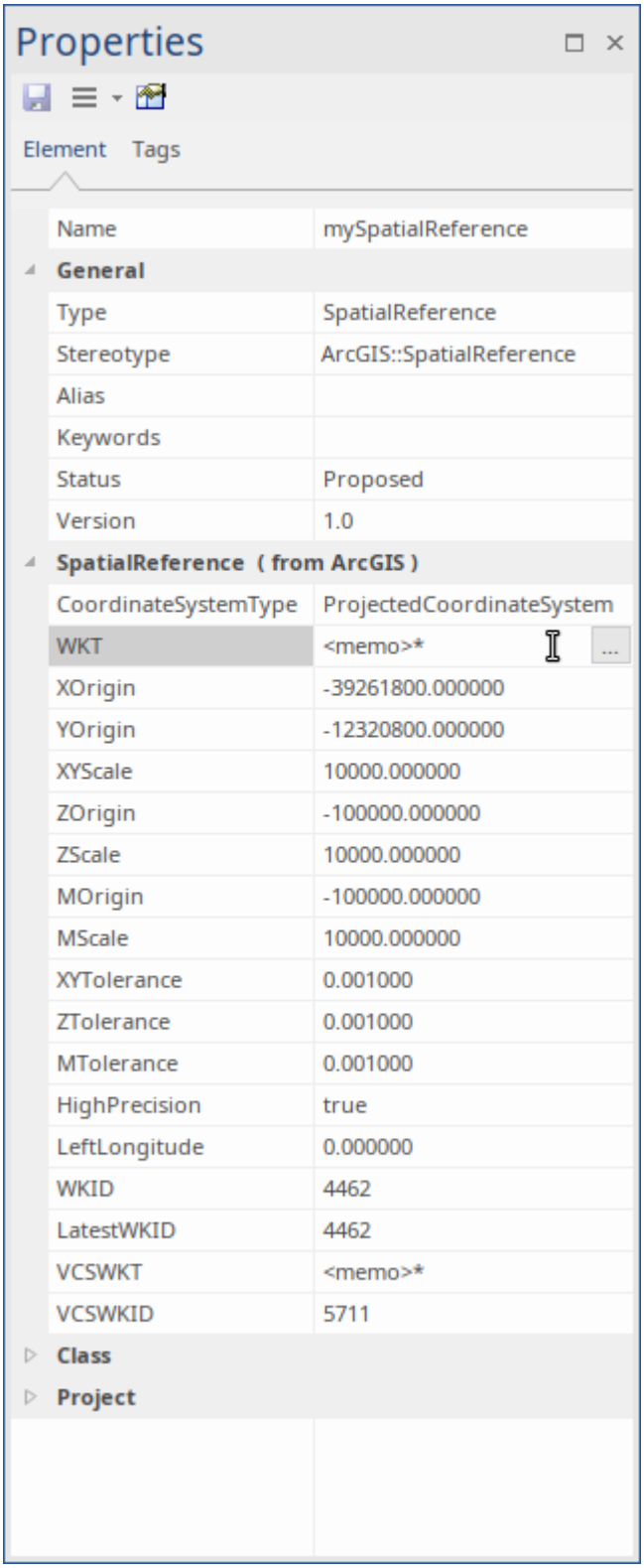
为了帮助您模型空间参考属性， Enterprise Architect提供了一个对话框，用于选择 ArcGIS 支持的预定义坐标系之一。当您选择地理或投影坐标系时， Enterprise Architect会自动插入相关属性的默认值，例如众所周知的文本(WKT)、分辨率、精度或公差。这些值作为标记值保存在«SpatialReference»元素上。

您还可以将垂直坐标添加到选定的地理或投影坐标系；垂直坐标加载到 «SpatialReference»元素上的 VCSWKT 和 VCSWKID标记值。

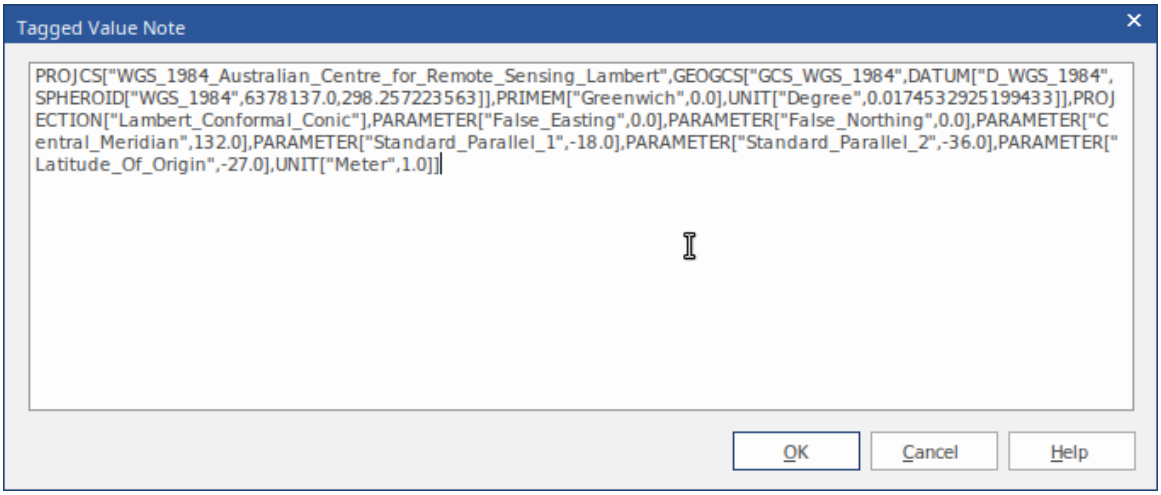
这是一个示例 «SpatialReference»元素：



查看该属性窗口 标签”选项卡中的 WKT标记值元素，您可以看到已选择 ‘WGS 1984 澳大利亚遥感兰伯特投影坐标中心”系统。



您可以通过查看其标记值注记来扩展该标记值中的信息。



定义一个空间参考元素

节	行动
1	打开 ArcGIS模型的 Spatial References包下的图表。 （您实际上可以使用模型中的任何 ArcGIS 图来定义空间参考元素；但是，此图是Enterprise Architect的 ArcGIS模型模式创建的方便占位符。）
2	将空间元素从 ArcGIS Core工具箱的 工作空间参考”页面拖到图表上。
3	右键单击空间参考元素，然后选择 特定| ArcGIS |设置坐标系统的菜单选项。 将显示 设置坐标系统”对话框。
4	根据需要展开地理坐标系或投影坐标系层次结构，然后单击列表中所需的坐标系。
5	如果您还想应用垂直坐标系，请单击 垂直坐标”字段右侧的 ... 按钮。 将显示 设置垂直坐标系统”对话框，其中包含您再次展开并从中选择列出的垂直坐标系的层次结构。 点击确定按钮返回 设置坐标系统”对话框； 垂直坐标”字段现在显示您选择的系统。
6	单击确定按钮关闭对话框并返回图表。 空间参考元素的标记值将使用您选择的坐标系统信息进行更新。

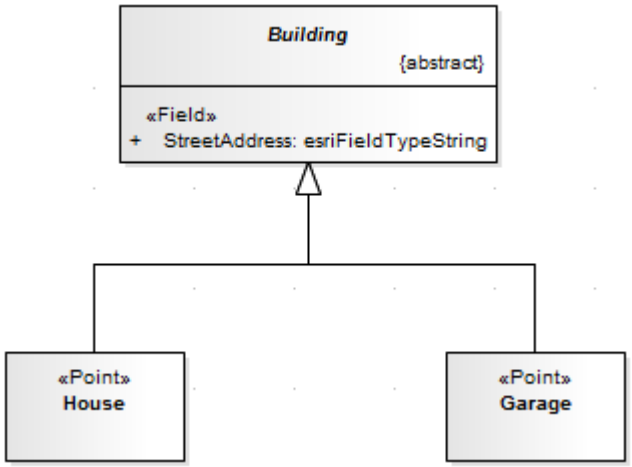
注记

- 您可以使用 SpatialReference标记值从任何其他特征数据集或模型中的特征类中引用 SpatialReference”类；«SpatialReference»类因此提供了单点控制，如果您以后需要更改参考信息
- 如果一个特征类元素引用了一个包含垂直坐标的«SpatialReference»类，如果您希望这个特征类元素存储三维数据，则将该特征类元素上的特征标记值设置为 true
- 如果您在 ArcGIS模型中没有从任何特征数据集或特征类中引用 «SpatialReference»类，系统将为这些元素生成一个具有未知空间参考类型的 XML 模式

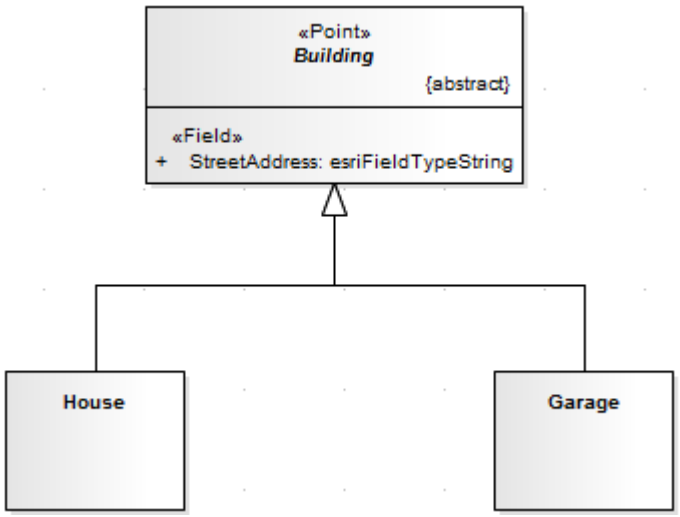
将 ArcGIS构造型应用于抽象类

使用适用于 ArcGIS 的Enterprise Architect UML配置文件，您可以在地理数据库方案中的特征类上指定几何构造型。几何刻板印象包括«点»、«折线»、«多边形»和«多点»等。 ArcGIS工具箱为每个几何图形提供了方便的图标，以便您可以将原型类拖放到您的地理数据库设计模型中，这些模型可以立即导出。这些类被称为具体类；由于它们的UML属性的值为False ，它们将直接在地理数据库模式中实现。

但是，有时在抽象UML类上指定几何构造型很有用，这样多个具体类可以继承几何，以及标记值和抽象类中定义的任何字段。此示例将房屋和车库建模为点特征类。 'House' 和'Garage' 类都从名为'建造' 的抽象类继承 'StreetAddress' 字段。



您可以通过在抽象类上指定构造型并为 "House"和"Garage"使用未构造型的具体类来创建等效模型，如下所示：



对抽象类而不是每个具体类（尤其是当你有许多这样的派生特征类）进行刻板印象的优点包括：

- 在设计时更容易改变几何形状；您只需对抽象类原型进行一次更改，然后它会自动应用于每个具体类
- 模型的创建速度更快，因为您只需编辑一组与原型相关的标记值；具体类可能不必复制（或覆盖）与其继承的几何构造型关联的任何标记值
- 同理，整体模型更小更简单

在模型中创建一个带有几何的抽象类

A类的UML属性'Abstract' 设置为True时，它被认为是抽象的。当您使用 ArcGIS 工具箱中的 抽象类”图标创建类时， 抽象类”属性会自动设置为True 。您还可以在类 属性”对话框的 详细信息”选项卡上为任何类手动设置属性。

节	行动
1	在您的模型中打开相关图表。
2	选择ArcGIS 工具箱'Find工具箱'对话框并指定'ArcGIS Core'）并将'类'图标拖到图表上以创建元素。
3	如果 属性”对话框没有自动显示，请双击抽象类元素。
4	在 属性”对话框的 常规”页面上，单击 构造型”字段 按钮，然后在 类构造型”对话框中，将 配置文件”设置为 ArcGIS”并单击复选框反对所需的几何刻板印象。

创建一个继承抽象类的几何构造型的具体类

当您将模型导出为地理数据库方案时，系统会将抽象类中的几何构造型应用到其派生的任何具体类。此外，导出器将添加任何缺少的 系统级别”字段。例如，类不需要指定或继承名为 OBJECTID”的字段。对 形状”、“Shape_Length”和”Shape_Area”字段也是如此。尽管导出器将使用这些字段，如果它们在继承层次结构中的某处建模，它会根据需要自动生成它们的有效实例。

节	行动
1	打开包含抽象类的图表。
2	选择 ArcGIS 工具箱 查找工具箱项”对话框并指定 ArcGIS Core”）并将 类”图标拖到图表上以创建元素。
3	单击概括中的工具箱类，然后将光标从具体类拖动到抽象类。
4	保存您的图表。

注记

- 任何没有构造型且不继承构造型的具体类都被导出为库表（ObjectClass）；如果模型中没有定义它，它的 OBJECTID 字段也会被插入
- 具体类只能从抽象祖先类继承几何构造型或«ObjectClass»构造型；目前， Enterprise Architect不支持从其他具体类继承原型
- 除了继承原型之外，具体类还从祖先抽象类继承字段
- 您可以从继承层次结构中任何级别的抽象类继承构造型；例如，指定几何的抽象类可以是具体类的祖父类，而不是父类
- ArcGIS 不支持单个特征类的多个形状， Enterprise Architect的 ArcGIS 配置文件也不支持；因此，如果一个具体类继承自多个几何刻板抽象类，将是一个建模错误
- 如果您指定的具体类已经存在于其父抽象类之一中的给定标签，则具体类具有优先权，并且其标记值将被

导出到模式

- Enterprise Architect不需要您在图表上显示物件和特征Esri 类，甚至不需要将它们包含在您的模型中，因为当您将几何或 ObjectClass 构造型应用于类时，系统会隐式应用它们的特征
- 然而，将物件类和特征与模型的概括包括在内并不是错误，即使通常没有标记它们

导入 ArcGIS XML工作空间

如果您有一个地理数据库工作空间XML 文档（包含 ArcGIS 模式），您可以将它作为UML模型导入到您的Enterprise Architect项目中。

在运行导入之前，取消选中 首选项"窗口（开始>外观>首选项>首选项）的 对象"页面上的 按字母顺序排列特征"复选框。这可确保在Enterprise Architect中以与源中相同的顺序导入和组织字段。

访问

在浏览器窗口中单击目标包。

功能区	发布 >模型交换>导入> ArcGIS
上下文菜单	右键包|特定| ArcGIS |导入ArcGIS工作空间XML

导入地理数据库工作空间XML 文档

选项	行动
文件名	类型在或浏览要导入的 ArcGIS XML 文件的名称。
创建图表	选中复选框以在导入的包下创建类图。
在图表上隐藏系统级 ArcGIS 字段	选中复选框以隐藏这些原型属性： • 必填项目 • 属性索引 • 空间索引 在这些刻板印象的类上： • 观点 • 折线 • 多边形 • 多补丁 库表（物件类）类的 'RequiredField' 和 'AttributeIndex' 属性也是隐藏的。 This option is enabled only when the 'Create图表' checkbox is selected.
移 除 GUIDs	"移 除 GUIDs "特征目前对 ArcGIS 导入是强制性的，这意味着每次导入 ArcGIS 模式时都会将元素创建为 新的"。
写入日志文件	选中复选框以写入导入活动log （推荐）。 log文件保存在要导入文件的目录中，与导入文件同名加后缀_import。 log 。
视图XML	单击此按钮可在导入前查看 XML。
导入	单击此按钮可导入 ArcGIS XML 文件。

关	单击此按钮可关闭此对话框。
帮助	单击此按钮可显示此帮助页面。
导入进度	此字段指示导入的进度。

注记

- ArcGIS 在Enterprise Architect的专业版、企业统一版和终极版中可用

导出 ArcGIS XML工作空间

为地理数据库工作空间XML 文档（包含 ArcGIS 模式）建模后，可以将其导出到外部目录（使用 Publish模型包功能），然后可以从该目录将其导入 Esri ArcCatalog。

访问

单击浏览器窗口中的 ArcGIS 原型包（您的 ArcGIS工作空间包）。

功能区	特定>工作技术>ArcGIS>导出到ArcGIS工作空间XML或 发布 >模型交换 > 发布为...
上下文菜单	右键包|特定| ArcGIS |导出到ArcGIS工作空间XML
键盘快捷键	Ctrl+Alt+E：发布

导出工作空间

选项	行动
根包	显示选中的 ArcGIS工作空间包的名称。
文件名	类型输入或浏览要生成 XML 文件的文件路径。
XML类型	选择 ArcGIS"作为要导出包的 XML/XMI 版本。
格式XML输出	将输出格式化为可读的 XML（在运行结束时这需要几运行）。
写入日志文件	写一份出口活动的log（推荐）。 log文件保存到 XML 文件导出到的目录中。
视图XML	单击此按钮可查看导出的 XML 文件。
导出	单击此按钮以启动 XML 导出。
关	单击此按钮可关闭此对话框。
进步	观察 XML 导出的进度。

注记

- ArcGIS 在Enterprise Architect的专业版、企业统一版和终极版中可用
- 在Enterprise Architect的企业统一版和终极版中，如果启用了安全性，则必须具有 导出XMI"权限才能导出

为XML

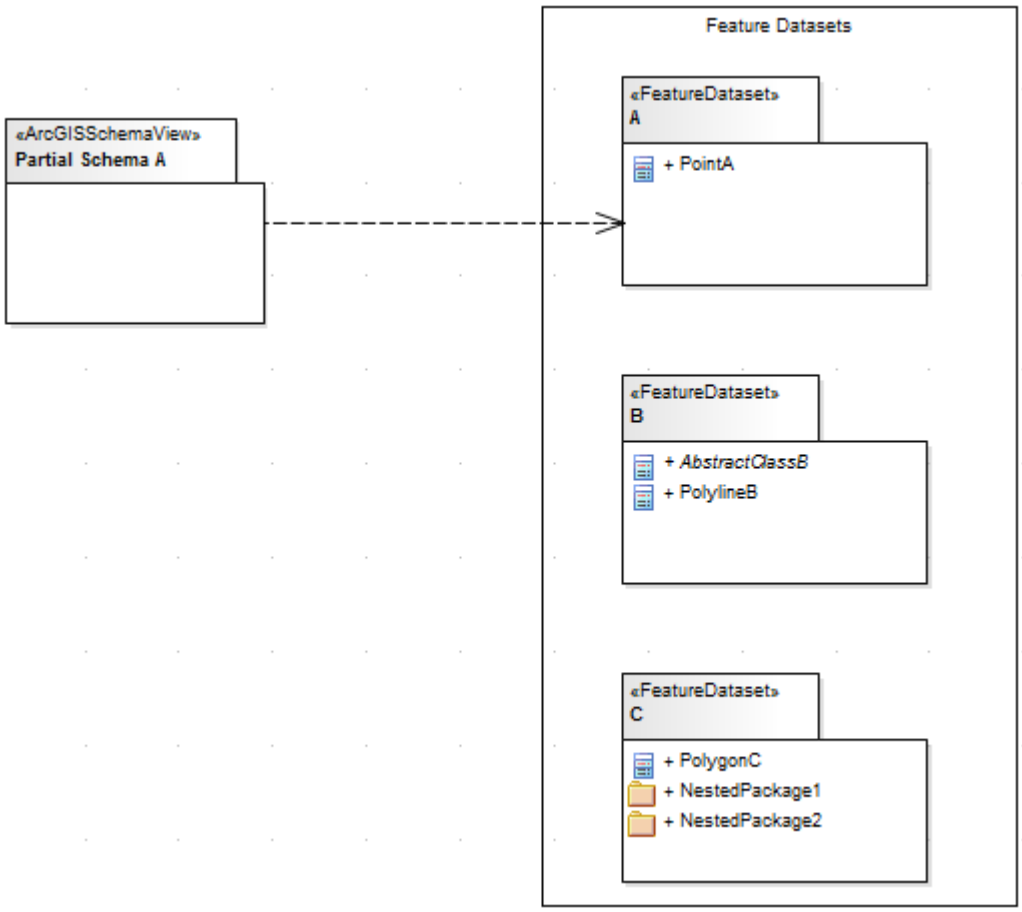
- 在将您的模型导出到 ArcGIS 模式之前，您必须至少定义一个空间参考元素；空间参考元素被其他模式元素通过动态链接的标记值引用，名为 SpatialReference
- ArcGIS包上的 DefaultSpatialReference 标签用于指定可应用于工作空间中所有特征数据集和特征类的空间参考；因此，您不需要对每个特征数据集或特征类应用空间参考元素
- 如果您没有从任何特征数据集或 ArcGIS模型中的特征类中引用空间参考类， Enterprise Architect默认会为这些元素生成一个未知类型为空间参考的 XML 模式

导出Modular ArcGIS Schemas

在Enterprise Architect中，除了导出整个 ArcGIS 工作空间之外，您还可以导出部分模式。如果您有一个大型地理数据库方案，这将很有用，如行业参考模型中定义的那样。在某些情况下，您可能需要整个模式，但只需要其中的一小部分用于特定的空间应用程序，例如字段数据收集。在这种情况下，您可能希望导出仅包含字段数据应用程序使用的特征表和域的模式，而不复制原始模式模型的部分内容。为此，您使用 «ArcGISSchemaView» 原型包。

«ArcGISSchemaView»包被建模为 ArcGIS工作空间包的子包包。您可以定义任意数量的 «ArcGISSchemaView»包-每个包代表地理数据库方案的不同子集。您可以通过从 «ArcGISSchemaView»包到每个包含的包中绘制一个 UML依赖连接器来指定包含模式的哪些部分。当您导出 «ArcGISSchemaView»包时，系统会包含您包含的包所依赖的任何其他包（通过依赖关系连接器）。

此图显示了一个部分模式，其中仅包含完成模式中的三个特征数据集之一。



创建 ArcGISSchema包

节	行动
1	在您的 ArcGIS工作空间中创建或打开 ArcGIS 图表。
2	将 ArcGIS架构视图图标从 Core图表工具箱您的图表上。 将显示A提示以输入包名。
3	类型一个有意义的包名，然后点击确定按钮。

4	将要包含在导出模式中的任何其他包拖到图表上。 （您可以使用«ArcGISSchema View»包的子图来绘制包含的包来获得相同的结果）。
5	从 «划View»包中绘制一个依赖连接器到每个其他包。

注记

- 在图上定义依赖关系很方便，但不是必需的；只要在模型中定义了依赖关系——无论它们是否存在于图表中——ArcGIS 模式导出器都会使用它们
- 您可以将依赖关系图安排在 ArcGIS工作空间中任何合适的部分——这些图可以位于 视图"包本身或 ArcGIS工作空间内的任何其他元素下

导出一个 ArcGIS架构视图，用于 ArcCatalog

节	行动
1	在图表或浏览器窗口中选择 ArcGIS架构视图包。
2	右键单击并选择 特定| ArcGIS |导出到ArcGIS工作空间XML'菜单选项。
3	识别目标文件并点击导出按钮。 系统会生成一个工作空间XML 文档，其中仅包含与 ArcGIS架构视图包相关的元素。

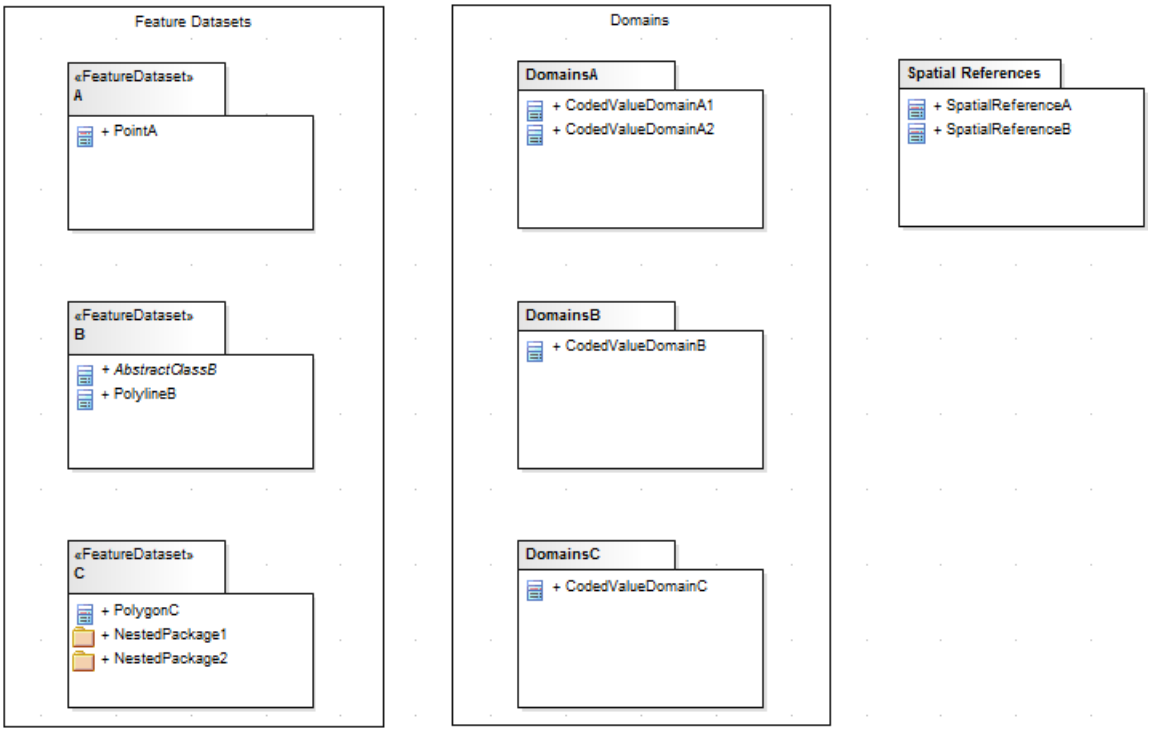
导出 ArcGIS架构视图包时包含哪些相关元素？

这些规则在您导出 ArcGIS架构视图包时适用：

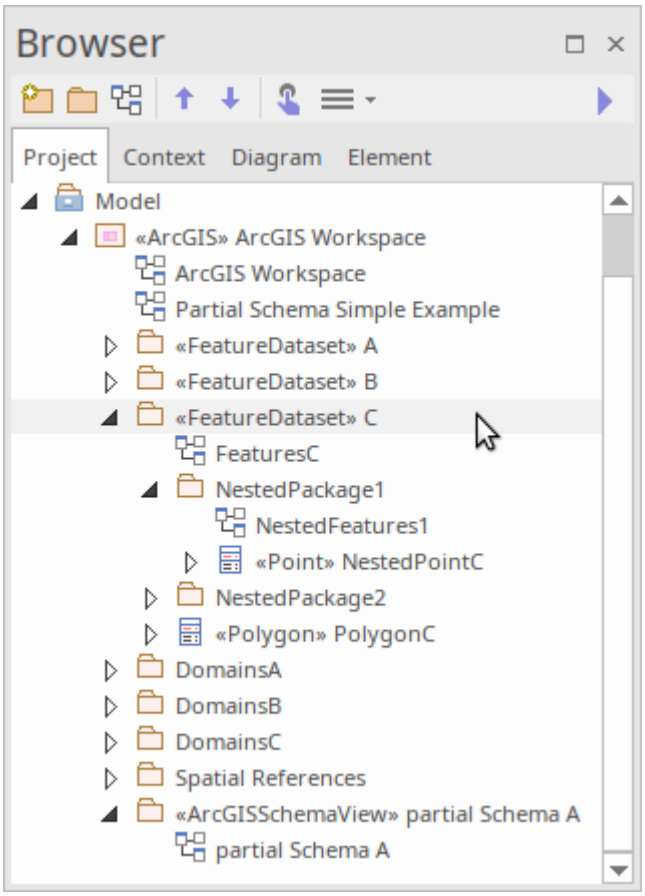
- 依赖关系是使用UML依赖关系连接器建模的
- ArcGIS架构视图依赖（直接或间接）的包的所有元素都包含在生成的模式中
- 导出包含元素从抽象类继承的所有字段，而不管抽象类所在的包
- 包含的元素所引用的所有编码值域元素都将被导出，而不管编码值域元素所在的包
- 如果ArcGIS架构视图包依赖于特征数据集包的一个或多个子包，则特征数据集仅导出链接子包中包含的元素 - 不导出特征数据集包中直接包含的特征类、域和表，因为对其子包之一的依赖关系；因此，如果要导出整个特征数据集，则必须使用对特征数据集包本身的依赖关系
- 如果包含元素的字段引用编码值域元素，则导出编码值域元素，无论 ArcGIS架构视图包是否对编码值域元素包有显式依赖
- 如果包含的元素具有与另一个元素X 的关系类连接器，并且元素X 尚未包含在 ArcGIS架构视图中，则不会元素X 和关系类连接器； log文件将保存关系类连接器的名称列表，因此不会导出这些连接器

建模部分模式的示例

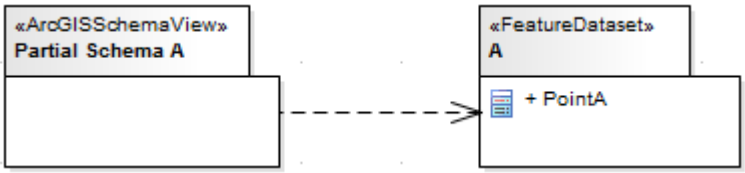
考虑这个完成工作空间，其中包括三个名为A 、 B和 C 的特征数据集，以及三个名为 DomainsA、DomainsB 和 DomainsC 的编码值域包：



浏览器窗口中对应的模型如下所示：

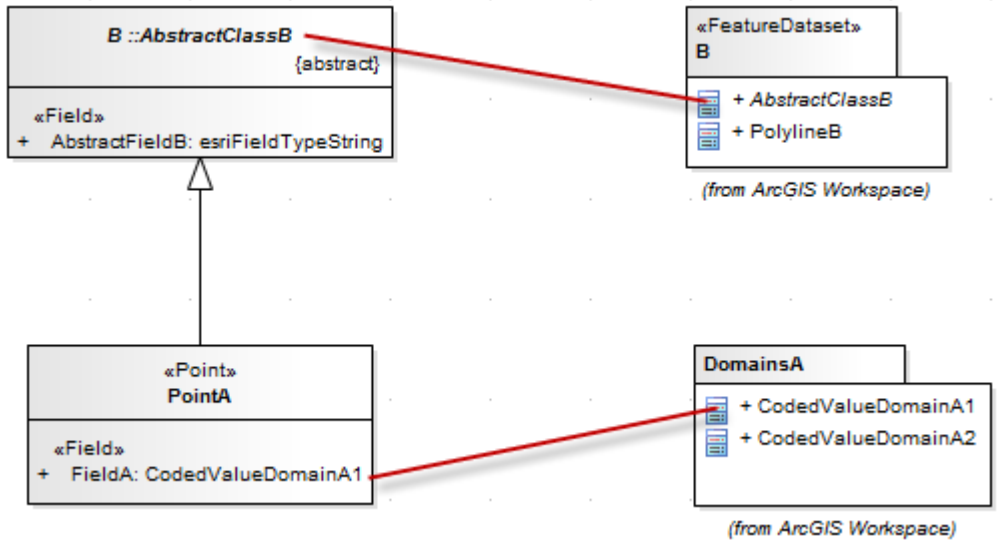


如果您只想导出特征数据集A及其所需元素，您可以将模型架构为包含单个特征数据集的部分架构，如下所示：

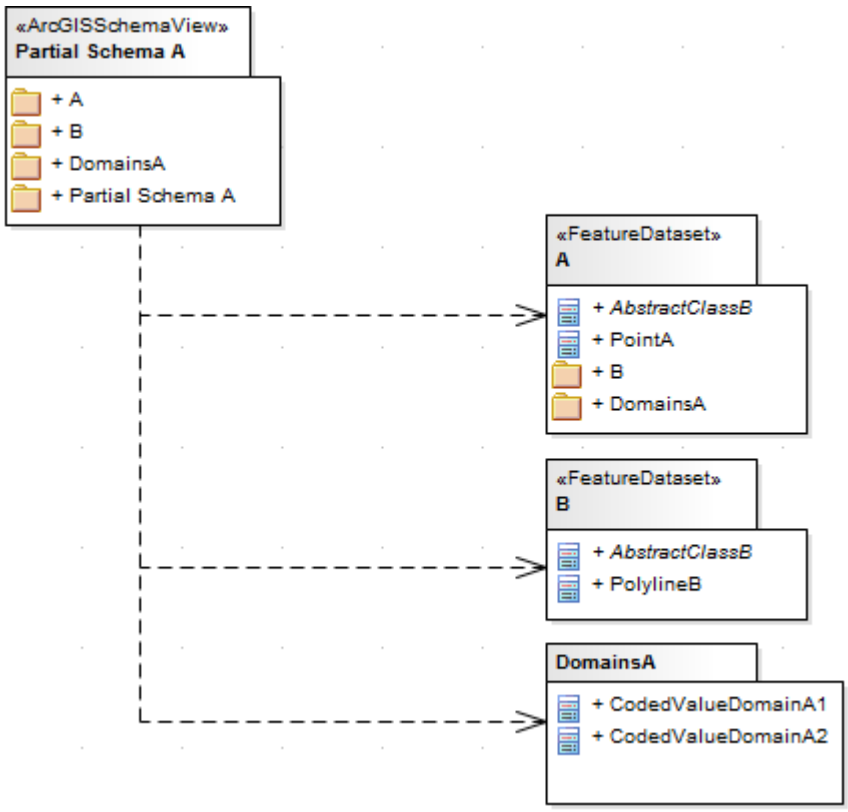


（此图等效于本主题开头提供的第一个图。）假设A点不依赖于其他元素，则生成的模式将仅包含FeatureDataset A及其特征类Point A。

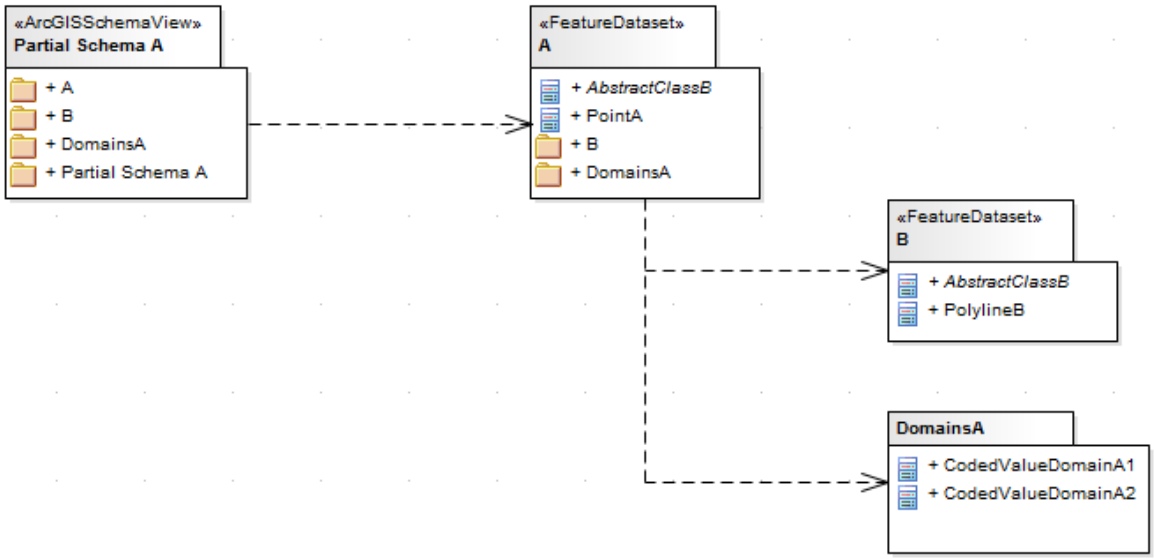
现在假设A点继承自 AbstractclassAbstractClassB（在 FeatureDataset包B中定义），并且A的一个字段具有在类包中定义的模型类型（如下图所示）。现在，相同的部分架构模型将导致导出的架构包含 AbstractClassB 和 CodedValueDomainA1 字段，即使部分架构A没有明确依赖包B或包域A，因为部分架构自动包含通过继承或相关的元素由字段类型引用。因此，导出器通过包含此类必需元素来帮助您生成有效的 ArcGIS 模式。



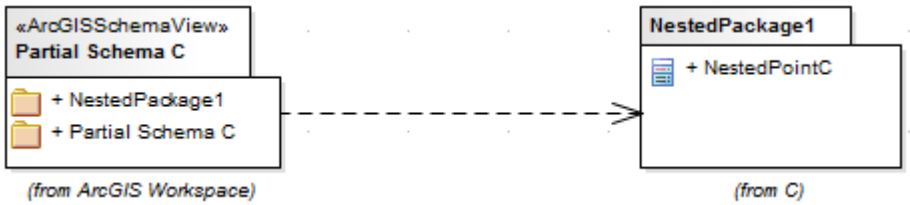
如果您想包含 DomainsA 中的所有 CodedValueDomains 和 FeatureDataset B中的所有特征类（包括它们所依赖的任何域），您可以模型这种情况，通过直接和间接UML依赖连接器在部分模式中包含整个元素包，如图所示.



您还可以通过间接依赖连接器包含包。例如，您可以通过将包链接到特征数据集A而不是将它们直接链接到ArcGIS架构视图来获得与上一个示例相同的结果。



最后，如果要创建仅包含嵌套包 1 等元素的部分模式，则可以将模型建模为部分模式包，该部分模式包指特征数据集中的嵌套包。



生成的模式将包括一个名为 C 的特征数据集，其中包含 NestedPackage1 中的所有元素。 NestedPackage2 中的元素将被排除，PolygonC 也将被排除（假设与 NestedPackage1 的元素不存在显式关系）。

验证 ArcGIS工作空间

开发或导入 ArcGIS模型后，可以根据系统提供的 ArcGIS 验证表中的一组规则对其进行验证。

访问

功能区	特定>技术> ArcGIS > Validate ArcGIS模型
上下文菜单	浏览器窗口 |右击«ArcGIS»工作空间包|特定| ArcGIS |验证 ArcGIS模型

进程

该选项在工作区上启动验证脚本。运行时，脚本会将信息记录到系统输出窗口的 ArcGIS模型验证”选项卡中。选择脚本输出以获取错误和警告。

有两种方法可以调查模型验证脚本报告的错误：

- 展开系统输出窗口，直接审阅错误和警告；您可以在浏览器窗口中双击警告或错误行以突出显示消息相关的元素或属性
  或者
- 将所有输出复制到一个文本文件并使用您喜欢的文本编辑器打开该文件；这可能会提供更清晰的脚本输出格式

更多信息

Sparx Systems Enterprise Architect是 ArcGIS 建模的理想工具，因为它具有集成地理数据库和促进团队间协作的强大功能。

地理数据库可视化： Enterprise Architect允许您在系统和协作平台内可视化地理数据库。此特征可实现传统以软件为中心的团队和地理空间团队之间的无缝集成和通信。

统一团队协作：平台将负责战略、业务规则、需求和系统组件的团队统一起来。这种集成确保模型可以在团队之间共享，从而形成一个集成模型以增强集成工作并降低风险。

全面的协作特征： Enterprise Architect提供一系列协作特征，如聊天、讨论和评论。这些特征使多学科团队能够在整个系统生命周期阶段（从策略和规范到实施和支持）进行有效沟通。

考虑地理空间组件：通过利用Enterprise Architect的协作工具，团队可以确保地理空间组件得到充分考虑并集成到整个系统设计中。这种整体方法有助于做出更好的决策并建立更具凝聚力的系统架构。

总之， Sparx Systems Enterprise Architect通过提供强大的可视化功能、无缝的团队协作以及将地理空间组件有效地集成到系统设计中的全面特征，成为 ArcGIS 建模的强大工具。

版信息

- ArcGIS 在Enterprise Architect的专业版、企业统一版和终极版中可用

确认通知：

Enterprise Architect中对 ArcGIS 数据库建模的支持是与联邦科学与工业研究组织 (CSIRO) 合作开发的，CSIRO定义了UML 2 和 ArcGIS 概念之间的映射，并为UML中表示的 ArcGIS 地理数据库模式的自动导入和导出功能建立了原型。

微软天青

创建Microsoft Azure图表以指定和记录Azure虚拟基础架构

Microsoft Azure （或简称Azure ）是公共云计算平台的市场领导者之一，提供定义 IaaS（基础设施即服务）、PaaS（平台即服务）和 SaaS（软件即服务）云环境的服务。Azure 于 2010 年进入市场，作为企业云解决方案的重要性稳步增长。该Azure用于通过 Microsoft 管理的数据中心构建、测试、部署和管理应用程序和服务，并支持多种编程语言、工具和框架，包括 Microsoft 特定和第三方软件和系统。

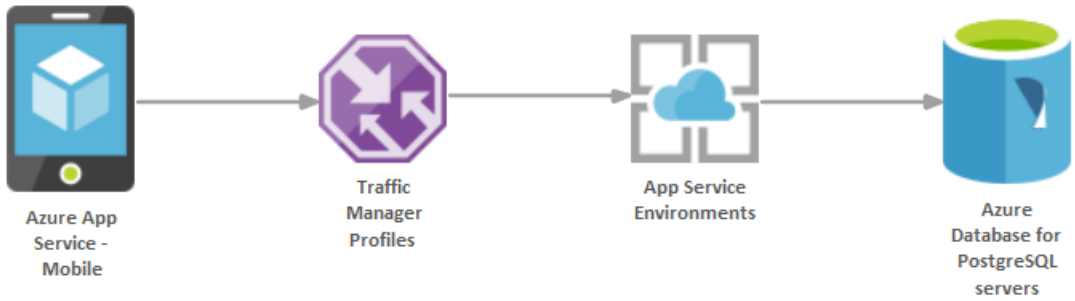
Enterprise Architect提供建模结构，可让您创建富有表现力的Azure图表，以指定新的云基础架构和平台或记录现有的云基础架构和平台。您还可以对其他云基础架构和平台提供商模型，例如Amazon Web Services (AWS)和谷歌云平台(GPC)。

Microsoft Azure UML配置文件提供了模型Microsoft Azure部署所需的图形（图标和图像）的新图像库。图标和图像由模型生成器框架模式提供，必须先将其导入到模型中，然后才能开始创建描述Microsoft Azure部署的图表。Azure Azure包含超过 350 个图像资产，可以拖放到图表上。

开始

创建Microsoft Azure图表很简单；所有AWS服务结构都可以从工具箱页面或 MS Azure包中的浏览器获得。这允许您创建包含元素项目的富有表现力的图表，例如虚拟机、MS #

数据库，用于大型计算机集群的 Cycle云，用于代码协作的 DevOps，用于数据存储的储存存储和托管磁盘，以及用于网络的负载均衡器和流量管理器。



显示使用智能手机流量管理器和图表数据库的移动应用程序的数据库

在本主题中，您将学习如何使用主题部分中概述的支持 MS Azure图表的特征。

选择蓝图

Enterprise Architect将工具的广泛特征划分为蓝图，确保您可以聚焦于特定任务并使用您需要的工具，而不会分散其他特征的注意力。要使用 Microsoft Azure特征，您首先需要选择此蓝图：

<透视名称> >分析> Microsoft Azure

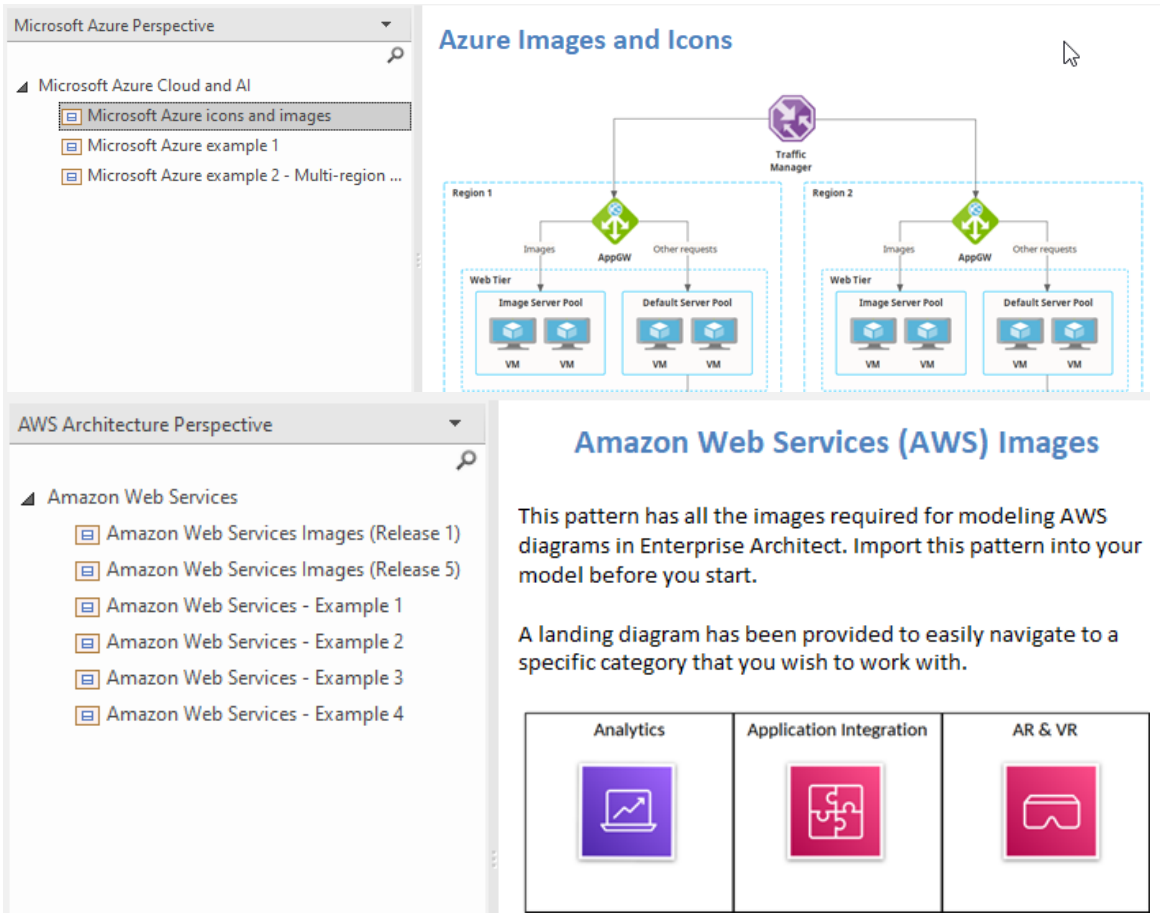
设置蓝图可确保 Microsoft Azure 图表、其工具箱和蓝图的其他特征默认可用。

示例图表

示例图提供了对该主题的可视化介绍，并允许您查看在指定或描述 AWS云基础设施时使用的一些重要元素和连接器。

导入Microsoft Azure模式

在开始创建 Microsoft Azure 图表以指定或记录您的云服务之前，您首先需要从模式中导入图形。这会将所有 Azure 图标作为组件注入浏览器窗口中的选定位置。



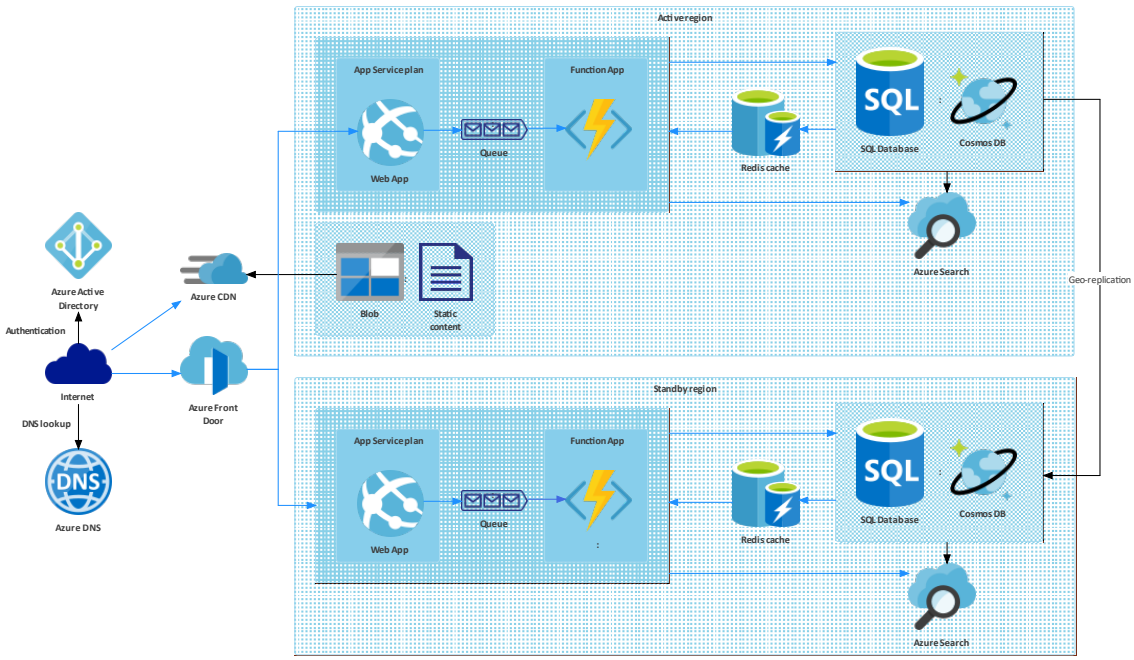
显示用于导入模式的 Azure模式的导入。

创建 Azure图表

更多信息

本部分提供了指向其他主题和资源的有用链接，您在使用 Microsoft Azure 工具特征时可能会发现它们很有用。

示例图表


Activeregion
App Serviceplan
Function App
Web App
Queue
Redis cache
SQL
SQL Database
Cosmos DB
Azure Search
Blob
Static content
Azure Active Directory
Authentication
Azure CDN
Internet
Azure Front Door
DNS lookup
DNS
Azure DNS
Geo-replication
Standbyregion
App Serviceplan
Function App
Web App
Queue
Redis cache
SQL
SQL Database
Cosmos DB
Azure Search

导入Microsoft Azure模式

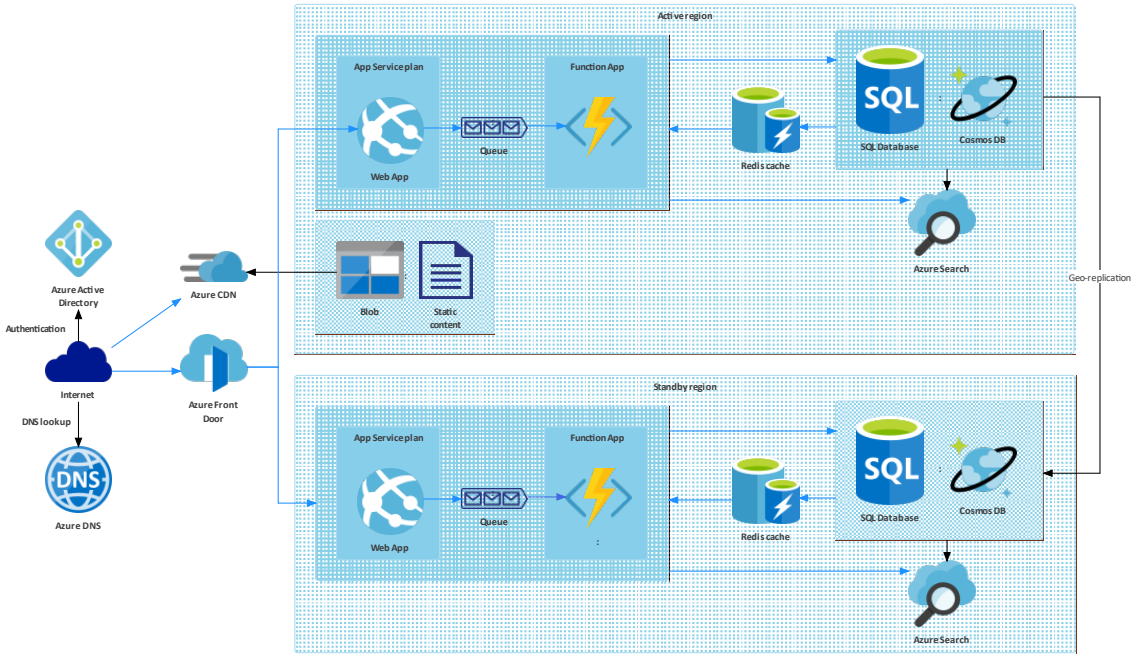
在将 "Microsoft Azure图标和图像"模式导入模型之前，请单击 图标并选择 分析> Microsoft Azure "蓝图。

这会自动打开模型生成器对话框，显示 "Microsoft Azure蓝图"页面。

在浏览器窗口中点击目标包，然后点击 "Microsoft Azure图标和图像"模式并点击创建模型按钮。

注记：当您的模型中有图像包时，不要将其复制到模型中的其他位置或将其保存为 *XMI*；请始终使用模型生成器将模式导入新模型。这样做的原因是复制图像资产将为它们提供新的 *GUID*，而您不想这样做。

在模型生成器中，有两个示例模式展示了图表中图像的典型用法。此示例是Azure Multi-region网络应用程序。

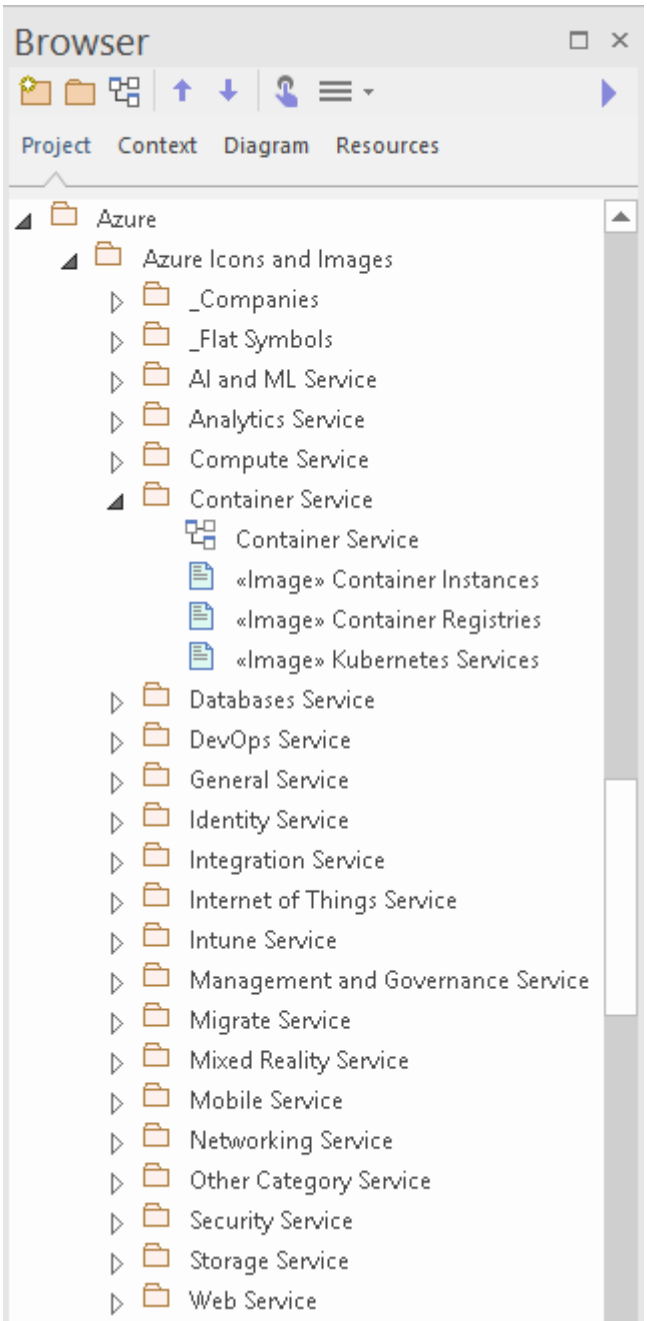


创建 Azure图表

您可以通过右键单击其父包并选择 添加图表"菜单选项以显示 新图表"对话框来创建图表。

如果您没有选择 Microsoft Azure蓝图，请单击类型字段中的下拉箭头并选择 分析> Microsoft Azure"。

在 图表"字段中为图表键入适当的名称，在"Select From panel"中单击 Microsoft Azure"，在 图表"面板中单击 "Azure"，然后单击 确定"按钮。工具箱的 Azure"页面打开，显示了图表的少量UML基本元素类型。但是，您可以使用从您导入的 Azure 库包中拖到图表上的图像来创建图表，如下所示：



注记Azure 图表会自动设置为自定义样式，当您右键单击图表中的元素时，您可以使用自定义样式工具栏上的自定义样式图标。

下载的 Microsoft Azure 图标和图像"模式中的每个包都有一个图表，显示包中包含的每个图像。

要将这些图像之一添加到图表中，请通过以下任一方式在浏览器窗口中找到它：

- 按名称搜索或

- 打开您认为应该在其中的包的图表，在图表中找到它并按 Alt+G 突出显示浏览器窗口中的图像资产

现在将图像资产拖放到您的图表上。您可以选择：

- 将其添加为带有图标的元素
- 将其添加为带有图像的元素，或
- （如果您已经从图标中制作了元素）添加为链接

更多信息

使用AWS Sparx Systems Enterprise Architect可以轻松创建Microsoft Azure图表。AWS 服务构造可从工具箱页面或浏览器中的 MS Azure包中轻松获取。这种可访问性允许创建具有虚拟机、MS #

等元素的富有表现力的图表#

数据库、用于大规模计算集群的Cycle云、用于代码协作的DevOps、用于数据存储的Blob储存和托管磁盘，以及用于网络目的的负载均衡器和流量管理器。

版信息

此特征在Enterprise Architect的企业统一版和终极版中可用，从 15.2 版开始。

MDG 技术

MDG 技术是一种工具，用于提供对商业可用技术或您自己创建的技术资源的访问。这些资源包括范围广泛的功能和工具，例如UML Profiles 、代码模块、脚本、模式、图像、标记值类型、报告模板、链接文档模板和工具箱页面。

使用Enterprise Architect ，您可以开发基于标准UML规范的模型，并且可以使用 UML 支持的机制（如标记值、构造型、 Profiles和设计模式）扩展核心UML结构。这些功能在Enterprise Architect核心技术中，您可以激活和使用进一步的模型驱动生成 (MDG)技术，这些技术要么与系统集成，要么可从外部位置获得。

如果您的系统或工作领域需要进一步的专业化，作为技术开发人员，您可以使用Enterprise Architect开发您自己的定制建模语言和解决方案。

获取和使用技术

技术源
核心技术Enterprise Architect本身包含： • 基础 UML 2 技术作为UML 2.5 结构和行为建模的实现，以及 • 核心扩展技术，应用配置文件和刻板印象来提供需求、用户界面和数据建模等方面的扩展数据建模
Enterprise Architect安装目录 MDGTechnologies 子文件夹中包含其他技术。
您可以将外部来源的技术导入 APPDATA 文件夹 (%APPDATA%\ Sparx Systems \EA\MDGTechnologies) 以供您自己使用，或导入浏览器窗口的 资源”选项卡以供其他项目用户访问。
您可以将技术转移到 MDGTechnologies 子文件夹中；这些技术在您重新启动Enterprise Architect时可用（在 Vista/窗口系统上，您可能需要增加访问权限才能执行此操作）。
您可以从Enterprise Architect访问和激活远程系统文件夹或网站中的MDG 技术。
技术开发人员可以通过 MDGTechnologies 子文件夹或远程文件夹或网站创建新的MDG 技术并将其部署到项目团队。
要查看Enterprise Architect中可用的技术并激活您需要的技术，请使用 MDG 技术”对话框（”特定>技术> 管理技术”功能区选项）。 使MDG 技术可用后，您可以管理它们对用户的可用性，并且可以与他们一起工作。 您还可以功能或禁用Enterprise Architect 基础 UML 2”和 核心扩展”技术和功能，以便您可以将Enterprise Architect功能和特征专门应用于一个或多个选定的MDG 技术。

指定需要MDG 技术

当您有必须使用某些MDG 技术的模型时，模型管理员可以配置系统以在加载过程中检查这些技术是否可用且处于活动状态，然后才能真正打开模型。您可以在 管理模型选项"对话框的 'MDG 技术"部分中识别技术。如果技术是：

- 需要且未安装在用户的机器上，该用户将无法打开模型
- 需要且可用，但未启用，系统可配置为自动启用该技术
- 具体不在本模型中使用，但可用并启用，系统可配置自动禁用该技术

管理员因此可以确保正确的操作环境在模型中工作，以便所有用户具有相同的视图并使用相同的模型功能（或者，至少没有使用错误的工具和创建结构其他用户无法使用）。

您可以在需要某些技术但其他技术可由用户自行决定使用的情况下使用 宽松"模型，或者在需要某些技术而所有其他技术被阻止的情况下使用 受限"模型。

访问

功能区	设置 >模型> 选项 > MDG 技术

选择需要技术

选项	行动
技术	审阅您当前可以访问的MDG 技术，按字母顺序列出。这些技术可能内置在Enterprise Architect中，由插件 提供插件 或从导入的目录或 URL。
需要	对于模型管理员，请针对打开模型之前必须可用的每项技术选中此复选框。 下次用户尝试打开模型时， Enterprise Architect将在允许访问模型之前检查所选技术在用户系统上是否可用。如果没有安装所需的技术， Enterprise Architect将不会打开模型。 此外，如果标记为需要的技术可用但未启用，系统将自动启用该模型；在用户可能访问的任何其他模型中，该技术仍将被禁用。
禁用	所有复选框默认为未选中，允许使用该技术。 选中模型中明确不得使用的每种技术的复选框。如果该技术可用并启用，系统会在模型中自动禁用它。它仍将在用户可能访问的其他模型中启用。
全部	单击此按钮以选中列表中每个技术的 需要"复选框。
没有任何	单击此按钮可清除列表中所有选中的 需要"复选框。

注记

- 在Enterprise Architect的企业版、统一版和终极版中，如果启用了安全性，您必须具有 配置项目要求”权限才能选中或清除技术对应的 需要”和 禁用”复选框

与MDG 技术合作

可以启用 "MDG 技术"对话框中列出的任何MDG 技术，这使得它们的界面配置文件和工具箱页面可供您使用。

当您启用MDG 技术时，任何特定于技术的图表类型都将添加到 "新图表"对话框列表中，并且该技术的图表工具箱页面将添加到通过工具箱的搜索功能可用的工具箱中。

如果将MDG 技术设置为 "Active"，它将成为模型的主要技术。一次只能激活一个技术。该技术的验证配置已设置，虽然常见的工具箱页面始终可见，但该技术的工具箱页面会覆盖任何并行的Enterprise Architect工具箱页面；例如，ICONIX 的 "类"页面会覆盖Enterprise Architect的 "类"页面。

您可以创建特定于技术的图表，并以与标准Enterprise Architect图表相同的方式使用元素和连接器填充它们。

管理MDG 技术

您使用 管理MDG 技术”对话框来管理项目可访问且可供项目用户使用的MDG 技术。该对话框列出了项目访问的多个位置中保存的技术，例如 APPDATA 文件夹和Enterprise Architect安装目录。您可以根据需要将这些技术设置为可用或禁用。 MDG 技术部署为 .xml 文件。

访问

功能区	特定>技术> 管理技术

配置技术可用性

选项	行动
技术	按字母顺序列出项目当前可访问的所有MDG 技术。 如果单击技术名称，对话框的右上方面板将显示该技术： • 名称 • 版本号 • 徽标（如果已定义），以及 • 部署的 XML 文件的位置，可以是： - Enterprise Architect内部 - 一个扩展 - 在安装目录中（只是文件名） - 在 APPDATA 文件夹中（文件名后跟（在 APPDATA 中）） - 在模型中 右下方面板显示技术描述，在许多情况下提供制造商的网站地址和支持联系人。
启用	针对您希望在项目中使用的每种技术选中此复选框。当启用MDG 技术时： • 该技术已添加到默认工具工具栏的 配置文件”字段中的可用选项列表中，以便您可以应用MDG 技术的接口配置文件 • 至少一组MDG 技术的工具箱页面会自动添加到图表工具箱；您可以通过 查找工具箱项”对话框访问添加的工具箱页面 • 任何MDG技术特定的图表模板都添加到 新图表”对话框中以供选择；选择后，这些显示特定于图表的工具箱页面 清除针对 a技术的复选框以使其对项目用户不可用。 如果您禁用正在使用的MDG 技术，其工具箱页面、图表类型和快速链接将在用户界面中的图表工具箱、默认工具工具栏、图表和 新图表”对话框中省略。
全部	单击此按钮以选中对话框中列出的每个技术的 启用”复选框。
没有任何	单击此按钮可清除对话框中列出的每个技术的 启用”复选框。 如果单击此按钮，滚动到列表顶部并选择 基础 UML 2技术”和 核心扩展”复选框以重新启用 UML ”和 扩展”工具箱页面和图表类型。

设置为活动	将技术设置为 Active 使该技术成为Enterprise Architect的默认界面，并且可以： • 使用特定于活动技术的页面覆盖各种工具箱页面（包括来自其他技术的页面） • 在另一个配置文件中重新定义原型，添加新标签并删除或修改现有标签，而原型在所有其他方面的行为就好像它是原始原型一样 如果您的首选技术不使用覆盖和重新定义，则无需将其设置为 Active。 选择并突出显示您的首选技术，然后单击设置活动按钮。这会在 技术"面板中的技术名称旁显示一个星号，并选择默认工具工具栏的 配置文件"字段中的技术。如果MDG 技术尚未启用，此按钮也将启用它。
高级	单击此按钮可将MDG 技术添加到Enterprise Architect远程的文件夹和网站中。
消除	（仅对直接导入模型的技术启用。） 单击此按钮可从列表、浏览器窗口的 资源"选项卡和模型中删除选定的技术。
确定	单击此按钮关闭对话框，保存更改并使其生效。
取消	单击此按钮可关闭对话框并中止您所做的更改。

注记

- 如果您更改了MDG 技术的 启用"设置，或者如果您更改了外部路径列表，请单击确定按钮以重新加载所有启用的技术；您无需重新启动Enterprise Architect即可使更改生效
- 要专门使用选定的MDG 技术或少数技术，您可以仅启用这些技术（可能将其中一项设置为 Active），然后取消选中 基础 UML 2技术"复选框（如果需要，还可以 核心扩展"复选框）
- 对已添加新标记值且定型名称一致的配置文件的更新，则可以同步这些新的或更改的标记值；更多详情请参见同步标记值和约束帮助主题

访问远程MDG 技术

当您处理您的模型时，您可以使用系统本地的MDG 技术技术或者您可以访问您在远离系统的文件夹和网站中识别的技术。您基本上将这些远程技术为 书签”以供继续使用，然后在您不想再使用它们时删除链接。

访问

功能区	特定>技术> 管理技术：高级

注记

- 要删除 MDG 技术MDG 技术-高级”对话框中列出的 MDG 技术，请单击文件夹路径或 URL，然后单击 删除”按钮；路径或 URL 被删除

指定远程MDG 技术

节	行动
1	在” MDG 技术-高级”对话框中，单击 添加”按钮。 A简短的上下文菜单显示，提供： • '添加路径' • '添加网址'
2	要在目录文件夹中指定MDG 技术，请选择 添加路径”选项。 将显示 浏览文件夹”对话框。 浏览MDG 技术文件夹，点击它，然后点击确定按钮；转到第 4 步。
3	要在网站上指定MDG 技术，请选择 添加 URL”选项。 将显示 输入”对话框。 在 输入值”字段中，键入或复制并粘贴MDG 技术URL，然后单击确定按钮。
4	MDG 技术的文件夹路径或 URL 显示在路径面板中。 该技术可用

导入MDG 技术模型

如果您找到或创建了对您的项目有用的MDG 技术，您可以将其导入到项目中：

- 仅供您自己使用；也就是说，将技术导入工作站上的 %APPDATA%\ Sparx Systems \EA\MDGTechnologies 文件夹，或者
- 对模型的所有用户可用，通过模型浏览器窗口的 资源”选项卡

要导入MDG 技术，您必须有一个合适的MDG 技术XML 文件。如果MDG 技术包括对任何元文件的引用，它们应该与MDG 技术XML 文件位于同一目录中。

与模式MDG 技术一起提供的模型必须在Enterprise Architect安装目录的模型模式目录中都有相关的模式XML 文件和一个包含模型描述的相同文件名的模式文件。

启动时， Enterprise Architect扫描 APPDATA 文件夹和Enterprise Architect安装目录模型子文件夹中的技术文件，通过 “MDG 技术”对话框和浏览器窗口的 资源”选项卡使它们可用导入到 APPDATA 文件夹的.技术由文本 “Location:技术.xml”指示。模型模 需要单独导入到用户系统的模型模式目录中。

访问

功能区	特定>技术>发布技术>导入MDG 技术
上下文菜单	在浏览器窗口的 资源”选项卡中 |右键单击MDG 技术文件夹|导入技术

导入一项技术

节	行动
1	在 导入MDG 技术”对话框的 文件名”字段中，键入要导入的MDG 技术文件的路径和文件名，或使用 按钮进行浏览。 当您输入文件名时， MDG 技术名称和版本会显示在 技术”和 版本”字段中，任何注记都会显示在 注记”字段中。
2	为您要执行的导入类型选择适当的单选按钮： • 导入模型 • 导入用户
3	点击确定按钮。 • （如果您选择了 导入到用户”选项）如果 APPDATA 文件夹还不存在， Enterprise Architect会创建它 • 如果MDG 技术已经存在， Enterprise Architect会显示一个覆盖现有版本并导入新版本的提示 一旦导入到 APPDATA完成，您必须重新启动Enterprise Architect ；然后在” MDG 技术MDG 技术”对话框中列出 MDG 技术。

注记

- 要删除已添加到 APPDATA 的MDG 技术，请在 %APPDATA%\ Sparx Systems \EA\MDGTechnologies 文件夹中找到相应的 XML 文件并将其删除
- 考虑到一些MDG 技术可能很大并且可能在工作站上施加一些延迟，因为它们在每次用户连接到模型时加载
- 要从浏览器窗口和模型的 资源”选项卡中删除MDG 技术，可以：
  - 右键单击技术名称并选择 删除技术”菜单选项，或
  - 单击 管理MDG 技术”对话框中的技术名称，然后单击
  删除按钮

扩展 - MDG 技术

Enterprise Architect是其建模功能的一系列模型驱动生成 (MDG) 扩展的核心，使用更专业的利基框架和配置文件。

扩展功能

扩展
许多技术已经与Enterprise Architect A程序集成，包括： • ArchiMate • BPEL • BPMN • 数据流图 • Eriksson-Penker 扩展 • ICONIX • 思维导图 • SoaML • SOMF 2. 1 • 策略建模 • 系统建模语言(SysML) • Eclipse 的MDG链接 • Visual Studio.NET 的MDG链接
Enterprise Architect提供以下支持： • 从外部系统文件或网站下载MDG 技术，或 • 使用Enterprise Architect MDG 技术向导轻松创建您自己的
Sparx Systems还销售一些MDG产品： MDG 技术为： • 扎克曼框架 • The Open Group架构框架(TOGAF) • 统一架构框架（UAF），原为 DoDAF 和 MODAF 的统一配置文件（UPDM） • 数据分布服务(DDS) • Python（ Enterprise Architect版本 4.5 到 5.0；集成在更高版本中）（*免费产品！*） • CORBA（* 免费产品！*） • Java Beans（* 免费产品！*） • 测试（*免费产品！*） MDG集成对于： • 日食 3.3 • 视觉工作室 2005、2008 和 2012 MDG链接 • Microsoft Visio（* 免费产品！*） • IBM Rational软件架构师（原 Telelogic） DOORS

随着时间的推移，此列表正在扩展以包括更多产品。
Sparx Systems提供Enterprise Architect的扩展版本，为系统工程和业务工程提供更大的支持。 这些版本包含几个列出的MDG 技术和其他插件。
有关可用插件的最新列表和每个产品的介绍，包括定价、购买和下载选项的详细信息，请参阅Sparx Systems网站。 当您购买其中一个插件时，您会收到一个或多个许可证密钥以及有关获取、安装和注册产品的说明。

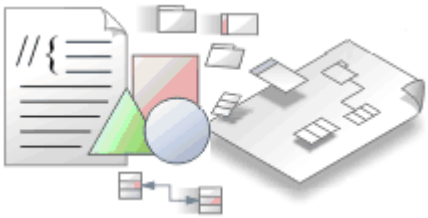
MDG 技术SDK

Enterprise Architect是一个多功能工具，具有数百个内置特征，并支持开箱即用的各种建模标准。它还提供了一系列有用的扩展机制。 Enterprise Architect软件开发工具包 (SDK) 包含扩展核心UML以支持特定领域、平台或方法的建模的机制。 Enterprise Architect和其他合作伙伴组织提供商业上可用的模型驱动生成 (MDG)技术，但任何人都可以免费使用 SDK 创建新的配置文件并将其作为MDG 技术分发。例如，您可能在安全工程领域工作，并使用特定的结构来模型您的领域和使用的方法。例如，您可以使用Enterprise Architect创建新元素来表示故障事件、故障模式和任何其他特定于域的实体。一旦配置文件完成，就可以将其捆绑到MDG 技术中，然后在您的组织内本地使用或分发给整个行业。

注记

- 在开发您的技术时，您需要熟悉核心系统和扩展机制的建模结构和概念，因为它们影响并被您为其设计技术的人使用；即本用户指南的建模部分中描述的系统

定义建模语言



如果您想执行更专业的建模，您可以扩展基本的UML建模元素及其使用来开发您自己的建模语言或解决方案。A简单的方法是开发和部署MDG 技术，它可以包含许多专门的Profiles和一系列其他机制，为您的定制解决方案提供最广泛的范围。

扩展功能

功能	描述
MDG 技术	MDG 技术是一种工具，用于提供对商业可用技术或您自己创建的技术资源的访问。这些资源包括范围广泛的功能和工具，例如UML Profiles 、代码模块、脚本、模式、图像、标记值类型、报告模板、链接文档模板和工具箱页面。
Profiles	Profiles是一种扩展UML的方法；您可以使用它们来构建特定领域的模型。 配置文件是扩展或应用于元素、属性、方法和连接器的附加构造型和标记值A集合，它们一起描述一些特定的建模问题并促进该域中的建模构造。
构造型	构造型是一种内在的机制，用于在逻辑上扩展或改变元素模型的含义、显示和句法。不同的模型元素具有与之相关的不同标准构造型。 当您自定义自己的构造型时，同样的原则也适用，或者通过“UML类型”对话框来限定现有类型的元素，或者作为扩展特定元类的元素来定义新的元素类型。
设计模式	模式是可以从一组一般建模场景（即参数化协作）中抽象出来的协作对象/类的组。 它们通常描述如何解决抽象问题，并且是实现重用和构建健壮性的极好方法。
形状脚本	形状脚本脚本是A脚本，它将自定义形状和方向应用于元素或连接器，以代替该object的标准UML表示法。每个脚本都与一个特定的原型相关联，并为每个具有该原型的object绘制。 在您重新定义标准UML object的属性以创建新object的地方，您也可以将新形状应用于object 。
标记值类型	您可以使用标记值向模型元素添加更多属性。您可以在三个级别应用它们： • 作为与元素模型相关的标准标记值 • 作为基于标准标记值类型的定制标记值 • 作为基于自定义标记值标记值类型的自定义标记值
代码模板框架	在Enterprise Architect中，您可以通过自定义控制这些操作的模板来修改生成或转换代码的方式，包括为行为模型生成代码。您还可以将这些模板合并到一项技术中，以将自定义生成和转换添加到该技术的功能中。

开发Profiles

Profiles是扩展的集合，基于应用于UML元素、连接器和特征的构造型。构造型可以具有专门定义标记值的属性，从而进一步扩展构造型元素或连接器的特性。 Profiles存储为具有特定格式的 XML 文件；要应用配置文件的扩展，您将其 XML 文件添加为MDG 技术的组件，并部署该技术；那是：

1. 创建一个模型来开发MDG 技术，并在其中创建一个配置文件包，您可以在其中定义您的配置文件
2. 将配置文件另存为具有特定格式的 XML 文件。
3. 使用MDG 技术创建向导将 XML 文件调用为MDG 技术技术。
4. 在您的系统上部署MDG 技术（以及配置文件）。

创建构造型Profiles

当您创建一个配置文件来定义一个新的建模解决方案时，您最初会创建一个带有 «profile» 原型的包。然后考虑需要创建的模型元素（以及因此的构造型元素）的数量。如果您要创建：

- A的构造型元素，可以在配置文件包内的单个子图上进行管理，并将图保存为配置文件
- 大量A构造型元素，在方便的情况下在尽可能多的子图上创建它们（如果您愿意，每个图一个构造型）并将包保存为配置文件

每个构造型元素至少扩展一个元类元素。构造型元素使用配置文件名作为它们的命名空间。创建配置文件后，您可以将其合并到MDG 技术中。

创建配置文件并将其应用到模型的过程包括多个步骤。仅当您希望配置文件将特定含义、显示、外观或语法应用于模型元素类型时，这些步骤中的某些步骤才是必需的。

创建配置文件

节	描述
1	在技术开发模型中创建配置文件包。
2	将构造型和元类元素添加到配置文件包的子图中。
3	为构造型标记值。
4	定义构造型元素的约束。
5	添加枚举元素以定义构造型上标记值的下拉元素。
6	为构造型元素添加形状脚本。
7	为每个原型模型元素设置默认外观。
8	在配置文件中包含快速链接器定义。
9	将包或图表另存为配置文件，然后导出。
10	将配置文件合并到MDG 技术中并部署该技术。

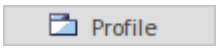
注记

- A配置文件包可以包含多个图表和许多元素和连接器，但不能包含其他包；不要在配置文件中使用嵌套包
- 如果您正在创建配置文件以构成MDG 技术的一部分，请注记您在单独的Profiles中为该技术定义了特殊的工具箱页面和图表

创建配置文件包

创建UML配置文件以定义新模型元素的第一步是创建一个包，该包在您的技术开发模型中具有构造型«profile»。

工具箱图标



访问

创建一个新的包图，然后显示图表工具箱打开 配置文件"页面。
使用此处列出的方法之一来访问工具箱的 配置文件"图表。

功能区	设计>  工具箱图表查找工具箱项"对话框并指定 配置文件"
键盘快捷键	Ctrl+Shift+3 :  显示 查找工具箱项"对话框并指定 配置文件"
其它	您可以通过单击  图表工具箱显示或隐藏图形图表视图。

创建配置文件包

节	描述
1	在"New图表"对话框中，单击 'Select From"字段中的" UML Structural"和 包"字段中的 图表"。 点击确定按钮。新图表在 图形"图表视图中打开。
2	打开工具箱的 配置文件"页面（点击图表显示 查找  工具箱"对话框并指定 配置文件"）。
3	将 配置文件"项拖到包图上。 将显示 新模型包"对话框。
4	在 包名称"字段中，输入配置文件的名称并选中 自动添加新图表"复选框。 点击确定按钮。将显示 新图表"对话框。
5	在 名称"字段中，输入图表名称，然后在"Select From"字段中单击 'UML Structural"，在 类"字段中单击 图表"。
6	点击确定按钮。 系统创建一个带有原型«profile»的包和一个子类图。 根据您的系统设置，可能会显示包的 属性"对话框。如有必要，您可以添加要分配给包的任何基

	本包详细信息，例如版本、相或注记。
7	在图上，双击配置文件包打开子图。 您现在使用此子图将构造型元素添加到配置文件中。

添加构造型和元类

当您扩展UML以开发特定于域的工具集时，您首先为要定制的构造型创建一个配置文件包。此包至少有一个子类图，您在此子图上指定：

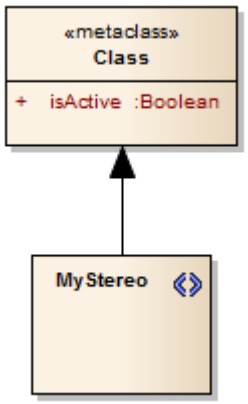
- 您正在扩展的object类型，由 Metaclass 元素表示，以及
- 每个object的扩展方式，由构造型元素表示

您可以使用一系列其他工具来限定构造型对元类的影响，包括：

- 形状类型中的构造型
- 由标记构造型元素中的属性定义的标记值
- 结构化标记值类，使用构造型元素标记中的属性定义
- 枚举，使用构造型元素中的属性定义
- 标记值连接器，用于识别以构造型生成的元素中的标记值的可能值
- 关于构造型元素的约束
- 特殊属性，定义原型元素的特定默认行为，例如元素的初始大小和颜色
- 修改构造型元素的默认外观

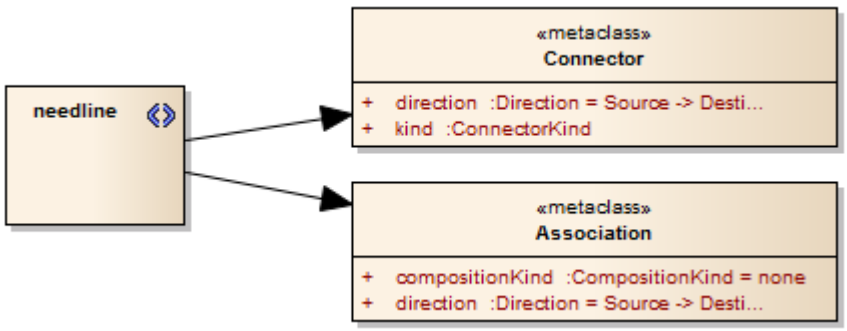
将元类和构造型添加到配置文件

节	描述
1	打开配置文件包的子图。
2	将元类元素从工具箱的 配置文件"页面工具箱图表上。 将显示 扩展元类"对话框，列出您可以扩展的object类型，即： ● 核心UML元素、属性和操作 ● 核心连接器 ● 抽象元类型，例如行动类型、ConnectorEnd 和门，以及 ● 构造型 在 核心元素"选项卡上，您可以通过选中 包含扩展"复选框来包含系统定义的扩展元素集，例如 ActivityRegion 和更改用户。 在 构造型"选项卡上，要指定包含要扩展的构造型的技术，请单击顶部字段中的下拉箭头并选择技术名称。
3	滚动所选列表并勾选一种或多种object类型以进行扩展。 如果要选择选项卡上的所有对象，请单击全部按钮。
4	点击确定按钮。 对于您选择的每个复选框，都会在图表上创建一个新的元类元素。
5	将构造型元素从工具箱拖到图表上。 如果没有显示 属性"对话框，请双击图表上的元素。
6	在名称字段中，输入构造型的名称。
7	点击确定按钮。

8	单击工具箱中的扩展关系并将连接从构造型元素拖动到它将扩展的元类元素。
9	您的图表现在类似于此示例： 
10	或者，您可以现在添加到您的构造型元素： • 构造型标签 • 枚举标签 • 结构化标记值 • 标记值连接器 • 特殊属性 • 约束和/或 • 形状脚本 您还可以根据需要定义元素或连接器的默认外观。

注记

- 如果您打算扩展大量的模型元素，而不是将它们全部放在一个图表上，您可以在配置文件包下创建额外的子类图表，并将不同类型的元类元素添加到不同的图表中；在这种情况下，您将包保存为配置文件，而不是单个图表
- 如果您想要一个构造型扩展多个元类，请创建一个构造型元素，并为每个元类元素创建一个扩展连接器，如下所示：



- 构造型元素必须具有唯一的名称，但元类元素可以具有相同的名称（例如，可以有多个行动元类，每个具有不同的行动属性）

创建构造型扩展非 UML 对象

配置文件通常A扩展核心UML object类型来创建您自己的建模语言或技术来定义；但是，您也可以扩展由其他现有技术（如 ArchiMate、BPMN 或 SysML）定义的非 UML 对象。

扩展非 UML object允许从现有构造型继承属性，即：

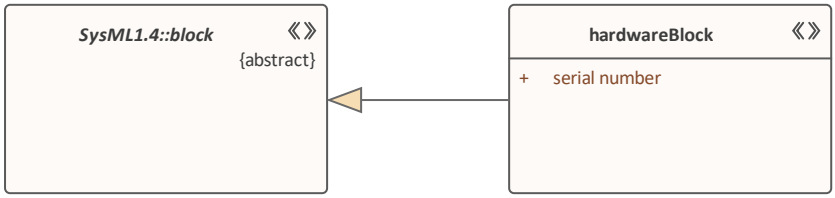
- 标记值
- 形状脚本
- 构造型颜色和
- 元类型属性

创建一个构造型扩展一个非 UML物件

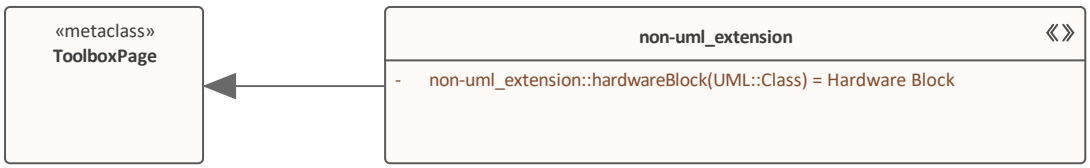
节	描述
1	在浏览器窗口中，找到带有<<profile>>构造型包并打开其子图。 如果您没有现有的<<profile>>包，请使用模型生成器中的 管理 | MDG 技术生成器”选项来创建新技术，然后从新创建的<<profile>>包中打开图表。
2	将 无类”图标从 工具箱”的 配置文件”页面图表图表上。 将显示 扩展元类”对话框。
3	选择 构造型”选项卡。
4	从下拉列表中，选择要扩展的配置文件（例如，“SysML1.4”）并选中要扩展的非 UML构造型旁边的复选框（例如， 块”）。 点击确定按钮。 适当的构造型元素被添加到配置文件图中。
5	通过从图表工具箱中拖动 添加构造型配置文件帮助器”来添加新的构造型工具箱 这将是扩展在步骤 4 中添加到图表中的非 UML 类型的自定义构造型类型。 完成后，构造型元素和元类元素将显示在配置文件图上。
6	从第 5划中添加的自定义概括连接或在第 4 步中添加的非 UML构造型构造型元素
7	将图表另存为配置文件。
8	定义一个工具箱配置文件，其中包含每个构造型的项目。
9	将保存的Profiles合并到MDG 技术中。

示例构造型配置文件

此示例显示定义构造型<<hardwareBlock>> 的构造型配置文件。 <<hardwareBlock>> 原型与 SysML块具有泛化关系，来自 SysML MDG 技术。这意味着在任何允许使用 SysML块的地方都可以使用 hardwareBlock。



这个例子展示了一个工具箱如何允许创建一个hardwareBlock。注记扩展总是原始原型扩展的UML类型。它绝不是对刻板印象的参考。



注记

- 当使用形状脚本来自定义构造型的外观时，您可以使用形状脚本() 方法来呈现为正在扩展的非 UML object定义的形状
- 如果您将任何元类元素属性添加到您的构造型，或者如果您想使用配置文件帮助器来创建一个工具箱配置文件，那么您的构造型类必须扩展一个元类以及专门化一个构造型

在另一个配置文件中重新定义构造型

如果您想在另一个配置文件中重新定义构造型，添加新标签并删除或修改现有标签，而构造型在所有其他方面的行为就好像它是原始构造型一样，您可以使用此处描述的重定义关系。

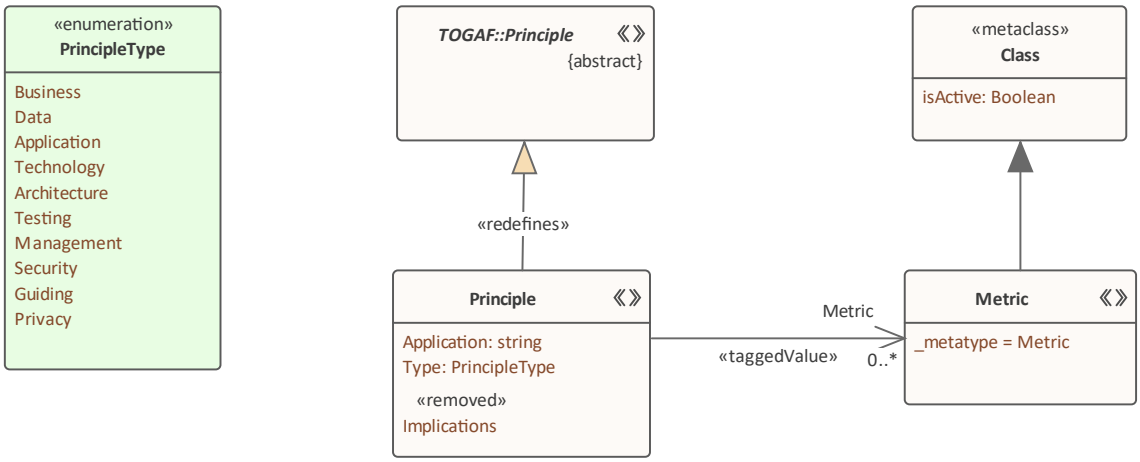
应用重新定义关系

节	行动
1	创建一个构造型元素，其名称与您正在重新定义的构造型的完全限定名称相同。请参阅示例中的“TOGAF::Principle”。 将此构造型元素设置为 抽象”。
2	创建具有相同名称的构造型元素，而不是完全限定的。请参阅示例中的 原理”。 划a 将关系从重新定义的刻板印象重新定义为重新定义的刻板印象。
3	到： • 从重新定义的原型中**删除**一个标签，将图表工具箱的 配置文件”页面中的 已删除属性”属性拖放到重新定义的原型上。使用与要删除的标签相同的名称该属性。这将创建一个具有原型 <<removed>> 的新属性，从而防止从原始原型继承标记值；请参阅示例中的 含义” 使用我们的配置文件创建的 “A ”元素将以各种方 充当 TOGAF 原则元素，但不会有通常的 含义”标签 • 向重新定义的构造型**添加**新标签，只需为重新定义的构造型提供标签即可；在示例中，新的 应用程 ”标签不是由 TOGAF 配置文件提供的，但会显示为 • **修改**重新定义的构造型中的现有标记值类型，为重新定义的构造型提供一个与要修改的标签同名但类型不同的标签；在示例中， 类型”是来自 TOGAF 配置文件的枚举，但我们已为其提供了一组修改后的枚举文字，“Metric”是 TOGAF 配置文件中的纯文本标签，但我们已将其重新定义为引用新 Metric”构造型的 RefGUIDList 标签
4	在将配置文件保存并部署在MDG 技术中之后，用户可以通过将技术设置为 活动”来指定创建的任何重新定义的元素都应该使用活动技术中的重新定义来创建。

示例图表

This diagram demonstrates a more complex scenario for extending a non-UML type. It demonstrates the capability of using a Redefines relationship to declare that this should behave like it is the original Principle element from TOGAF.

It also demonstrates how to remove and override Tagged Values defined in the original profile. Elements created with this stereotype will:
1. Not contain an Implications tag
2. Provide different options for Type from the base profile.
3. Allow metric definitions to be re-used by replacing the plain text with a RefGUIDList to a new Metric stereotype
4. Add a new simple tag "Application" that appears in the TOGAF::Principle group



Optionally, an end user can specify that creating a TOGAF::Principle should actually create an instance of Principle from this profile. They do that by setting this profile as Active (which can only be done for a single technology.)

定义构造型标记值

您可以通过添加各种类型的元素标记值来为构造型定义额外的元信息，您将其标识为构造型的属性。最简单的标记值是那些在 值”字段中键入纯文本的值。

对于更复杂的标记值，例如枚举和结构化标记值，请参阅以下主题：

- 将枚举添加到构造型
- 定义结构化标记值
- 使用预定义的标签类型定义构造型标签

访问

使用此处概述的方法之一显示特征窗口的 属性”页面。

功能区	设计>元素>编辑>特征>属性
上下文菜单	在浏览器窗口或图表中|右键单击元素|特征|属性
键盘快捷键	F9 或 Ctrl+5 > 属性

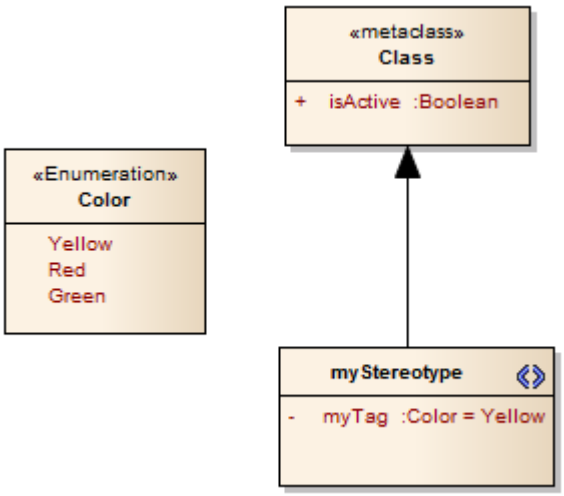
为构造型元素标记值

字段	行动
名称	用新属性/标签的名称改写新属性文本。
类型	默认为int 。 如有必要，单击下拉箭头并选择不同的属性类型。
范围	默认为私人。如有必要，单击下拉箭头并选择不同的范围值。
构造型	如果需要属性构造型，请单击 ... 图标并从 构造型选择器”对话框中搜索和/或选择构造型。
别名	如有必要，输入属性/标签的别名。
初始值	（可选。）类型属性/标签的初始值。

将枚举添加到构造型

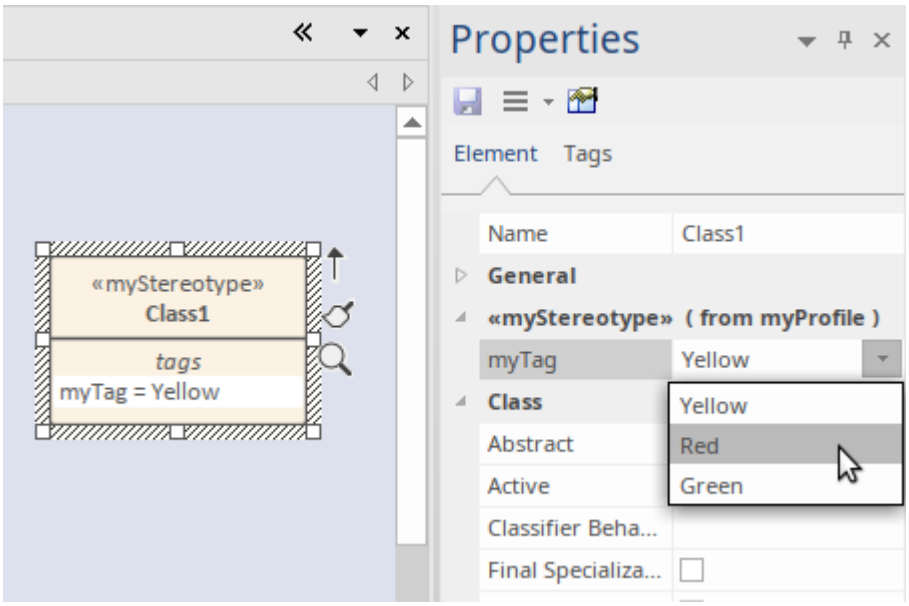
枚举元素可用于生成与构造型元素关联的标记值的下拉列表。在属性窗口的 标签”选项卡中显示列表和选定的值。

继主题定义排序标记值构造型之后，此示例说明如何使用枚举 颜色”为 ḿyTag”提供值的下拉列表（ 黄色”、 红色”、 绿色”）对元素的价值'我的标记值印象'。



给构造型添加一个枚举

节	描述
1	打开配置文件包图。 在这张图上，我们应该已经有了元素<<metaclass>>类和原型元素'myStereotype'。
2	在工具箱中，找到并选择 配置文件”页面。
3	将 枚举”图标从工具箱拖到图表上。
4	如果尚未显示，请打开 属性”对话框。 功能区：'设计>元素>属性>常规>属性'（或按对话框+2）
5	在 名称”字段中，输入新枚举元素的名称。
6	如果尚未显示，请在 属性”页面打开特征窗口： 功能区：'开始>所有窗口>>属性>元素特征>属性
7	在 名称”字段中，输入枚举属性的名称（例如， 黄色”），然后按“Enter”。
8	单击*New Attribute*文本并输入下一个枚举属性的名称。对其他属性重复此步骤，以定义下拉列表的其他值。
9	右键单击构造型元素 ḿyStereotype”并选择 特征>属性”选项。 原型的特征窗口显示在 属性”页面上。

10	在 名称”字段中输入属性的名称。
11	在 类型”字段中，单击下拉箭头并单击 类型”选项，然后从 选择 <Item>”对话框中浏览并选择枚举元素的名称。
12	在 初始”字段中输入定义默认值的必需枚举属性的名称。
13	单击关闭按钮。 您现在已经生成了一个下拉列表，用于在属性窗口的 标签”选项卡中设置标签的值。使用配置文件时，使用构造型创建的元素的标记值可能如下所示： 

定义结构化标记值

如果要定义一个包含多个组成部分的属性，例如地址，可以使用结构化标记值。这由一组相关的简单标记值序列，它们一起定义了属性。例如，街道地址的结构化标记值标记值有以下组成部分：

物业编号 - 448

街道 - 我的街道

城镇 - 克雷斯维克

区号 - 3363

当您最初在属性窗口或元素的标签隔间中显示它时，标签的值显示在一个string中，例如：

448，我的街，克雷斯维克，3363

然后，您可以展开结构标记值以列出组件标记名称和值。

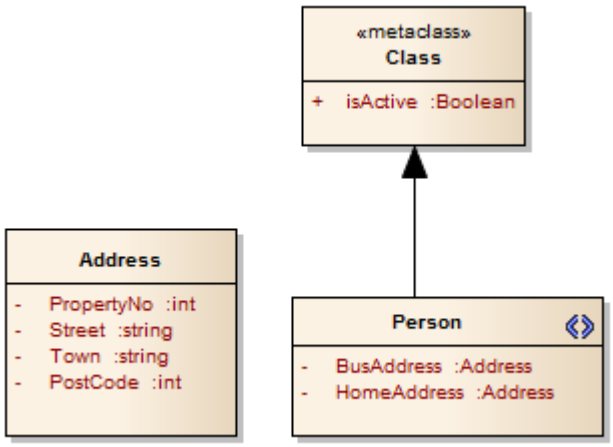
您使用非定型类在配置文件中创建结构化标记值。任何由构造型元素拥有的由此类类型键入的配置文件中的属性都将定义结构化标记值。

创建结构化的标记值类

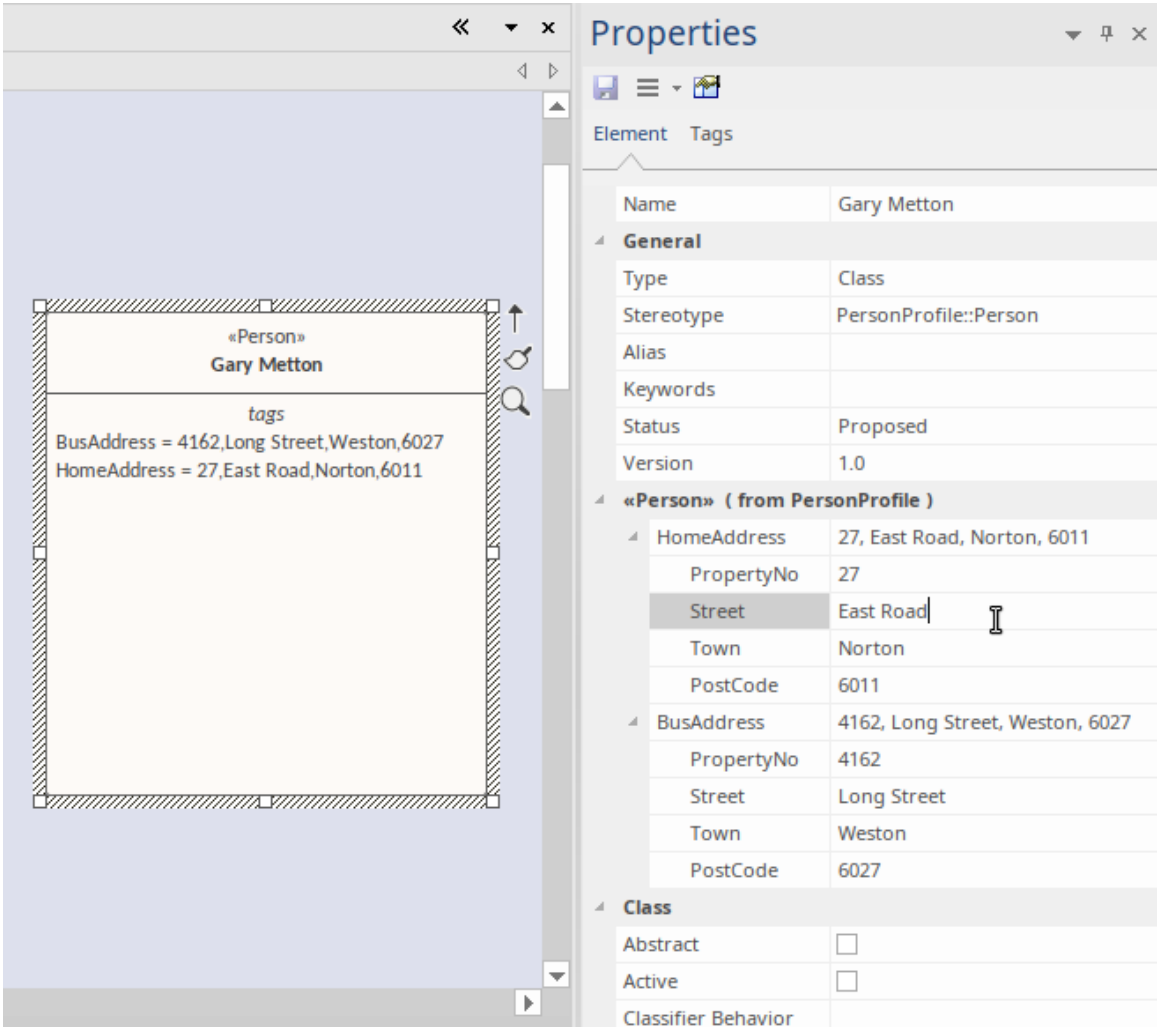
节	描述
1	在您的配置文件包中，打开子类图。
2	在工具箱中，找到并选择 类”页面。
3	将一个类项从工具箱拖到图表上。 如果没有显示 属性”对话框，请双击图表上的元素。
4	在 名称”字段中，输入新类元素的名称。
5	单击 详细信息”选项卡和 属性”按钮。 特征窗口显示，显示 属性”页面。
6	在 名称”字段中，输入结构化标签属性的名称（例如*PropertyNo* ）。
7	在 类型”字段中，单击下拉箭头并选择适当的类型（例如” int ”或” string ”）。
8	单击*New Attribute*文本，然后对每个剩余的组件标记属性（例如：Street、Town、AreaCode）重复步骤 6 到 8。
9	定义完所有组件标签后，点击构造型元素；特征窗口显示在 属性”页面上，用于构造型。
10	在 名称”字段中输入属性的名称（例如：“HomeAddress”）。
11	在 类型”字段中单击 ... 按钮并从 选择<项目>”对话框中选择结构化标记值类元素的名称，作为属性的分类器。 您现在已经为从配置文件的这一部分派生的任何元素生成了要在属性窗口中维护的结构化标记值的组件。
12	继续定义配置文件，然后将图表或包保存为配置文件，然后将其导出以供使用或将其添加到MDG

	技术文件中。

示例



当这些元素作为配置文件导入模型时，定义了一个可以应用于类元素的 'Person "原型。此构造型允许您在应用构造型的元素中以结构化标记值的形式输入家和业务地址详细信息。



注记

- 通过配置文件应用结构标记值的过程是通过插件
  应用标记值的替代方法插件
  播送;查看了解更多主题
- 构成一个结构的标记值标记值必须简单；备注标记值不能包含在结构化标记值中

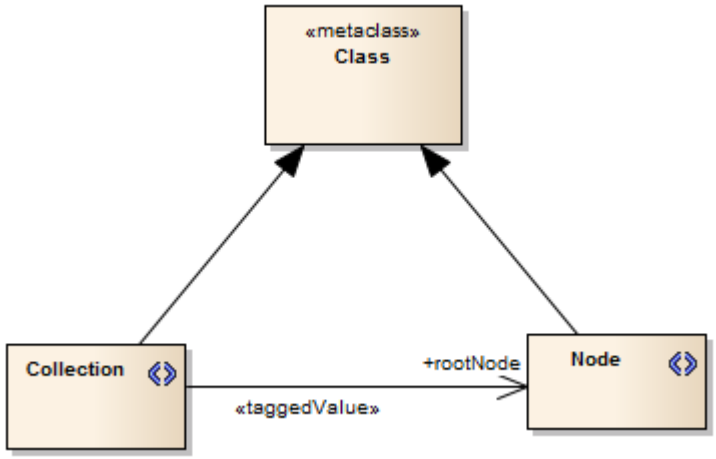
使用标记值连接器

创建配置文件时A常见情况是一个原型的实例需要引用应用了另一个原型的元素。例如，一个定义了 Collection 的元素可能有一个名为 rootNode 的标记值来标识该 Collection 的根，这将是一个具有构造型 <<Node>> 的类。

在属性窗口中，用户点击根节点标记值的选择按钮（ ... ）；当 选择<Item>”对话框出现时，用户可以定位当前模型中的所有节点，并选择其中一个元素作为标签的值。

为此，您可以使用工具箱 配置文件”页面中的标记值连接器。 A标记值连接器定义了一个标记类型（即 RefGUID ）属于源原型的标记值；标记值名称是该连接器的目标角色的名称，并且标记值仅限于引用具有目标元素的构造型的元素。

此图演示了如何使用连接器来表示示例。 A配置文件定义了两个原型：«Collection» 和 «Node»（它们都扩展了 Metaclass类）。 «Collection»构造型拥有一个标记值连接器，其目标角色为rootNode，指向«Node»构造型。您在连接器 属性”对话框的 角色”页面上输入目标角色名称。



注记

- 标记值连接器也可以直接与元类元素链接，以识别基本UML元素类型；例如：如果目标是参与者元类，当您选择识别特定目标时， 选择 <item>”对话框将列出基 参与者元的所有元素
- 此外，连接器可以链接到元素类型组的元类，即分类器和属性；如果连接器目标是元类：
  -分类器，当您选择识别特定目标元素时， 选择<item>”对话框将
  列出所有企业架构师定义的分类器类型，例如类和部件
  -属性，当您选择识别特定目标元素时， 选择 <item>”对话框将列出
  列出端口、部件和属性元素

使用预定义的标签类型

标记值定义了一个模型元素具有的范围广泛的属性和特征，而这些属性中的一些是复杂的或结构化的值。例如，您可能希望您的用户在上限和下限之间选择一个值（使用 旋转"箭头）、设置日期和时间、从调色板中选择一种颜色，或者通过核对表进行操作。

您可以从许多预定义的简单标记值类型和过滤器中的任何一个创建这些复杂的标记值，其中一些可能是您自己创建的，使用工具箱的 配置文件"页面中的 图表类型"元素。

使用数据类型元素的巨大优势在于它可以帮助您定义特定于profile的标记标记值，因此您可以在不同的profile中创建具有相同名称的不同类型的标记值，而不会在运行MDG 技术时产生冲突那些配置文件。

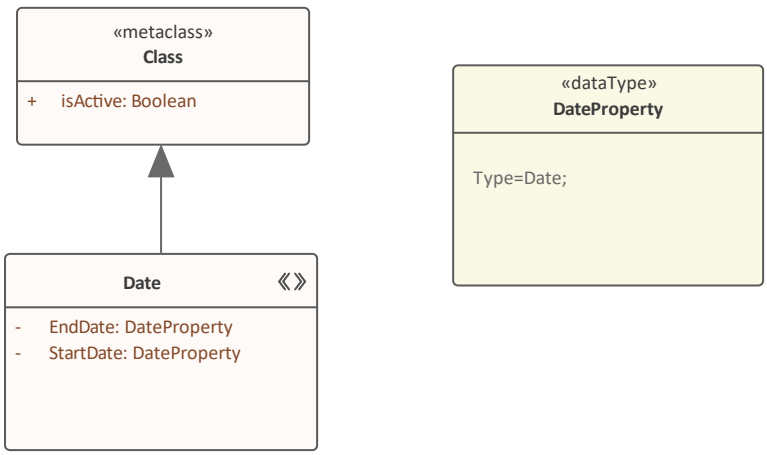
建议使用此方法从预定义的简单标记值类型和过滤器创建复杂的标记值配置文件。出于维护现有配置文件的目的，仍支持在UML类型对话框中定义全局标记值类型的原始方法；请参阅*With Predefined Tag Types (Legacy Profiles)*帮助主题。

将值分配给构造型标记值

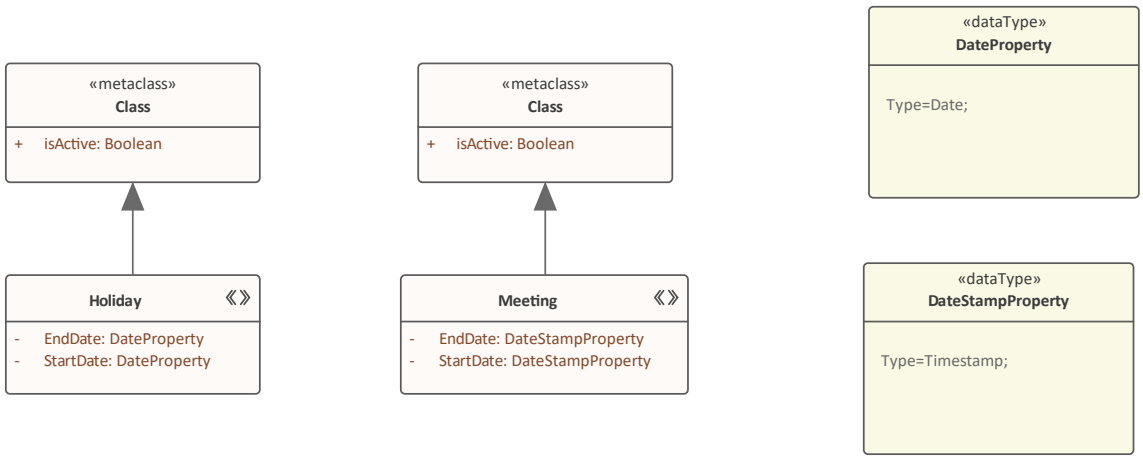
创建结构化标记值后，您可以将其分配给构造型元素，方法与简单标记值相同，方法是在构造型元素中使用标记值类型的名称创建一个属性。

示例

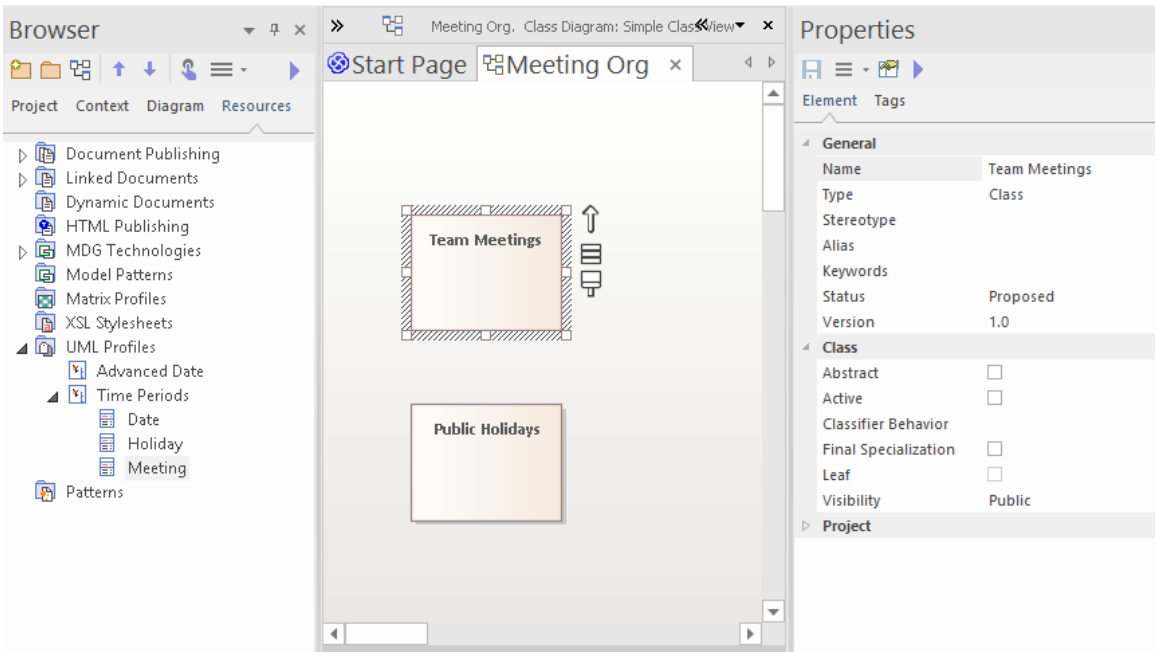
考虑 开始日期"和 结束日期"标记值的例子。使用 DataType元素，您将在标记值的注记中定义一个类型- '标记值' 帮助标记值类型（使用*Predefined Structured Types*帮助中列出的定义），并从任意数量引用该类型构造型元素中的属性 - 例如 开始日期"和 结束日期"。此定义及其引用对象与父包之外的任何其他定义无关。



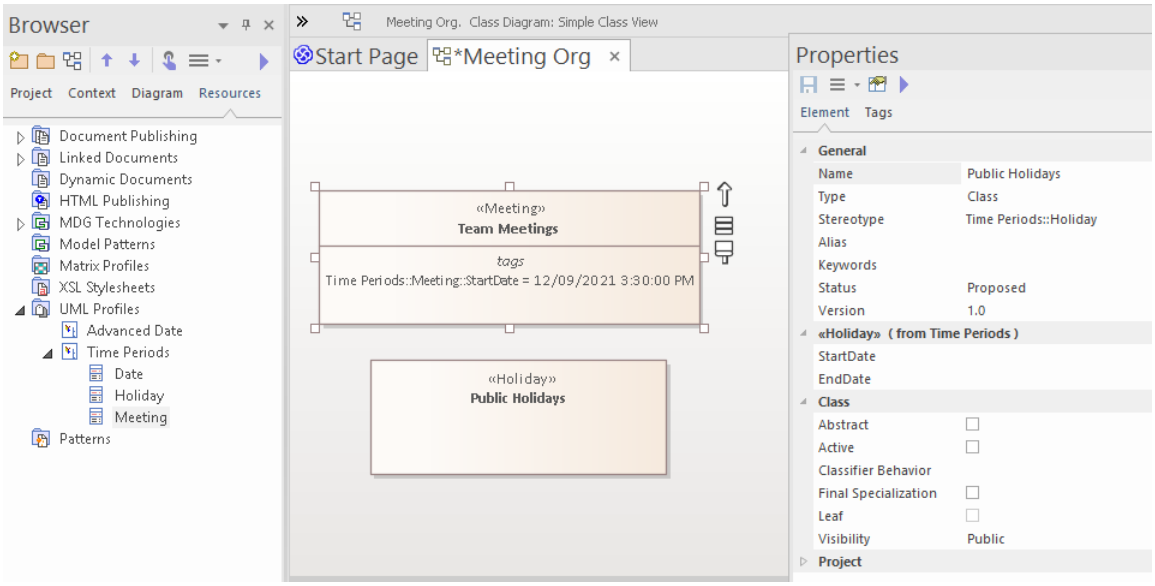
扩展此示例，假设您有名为 'Holiday' 和 'Meeting' 的构造型，并且两者都有 StartDate 和结束日期属性。但是，“Holiday"使用"DateProperty”，定义为 Type=Date;”，而"Meeting"可以使用"DateTimeProperty”，定义为“Type=Timestamp;"。



将配置文件导入用户模型时（作为独立配置文件或作为MDG 技术的组件），可以将构造型应用于新元素或现有元素，并添加标记值类型（并在创建时可用）'标记值'对话框下拉菜单中的更多标签）。在此插图中，时间周期配置文件已导入模型的 资源”选项卡，将用于定制图表上存在的两个标准类元素。

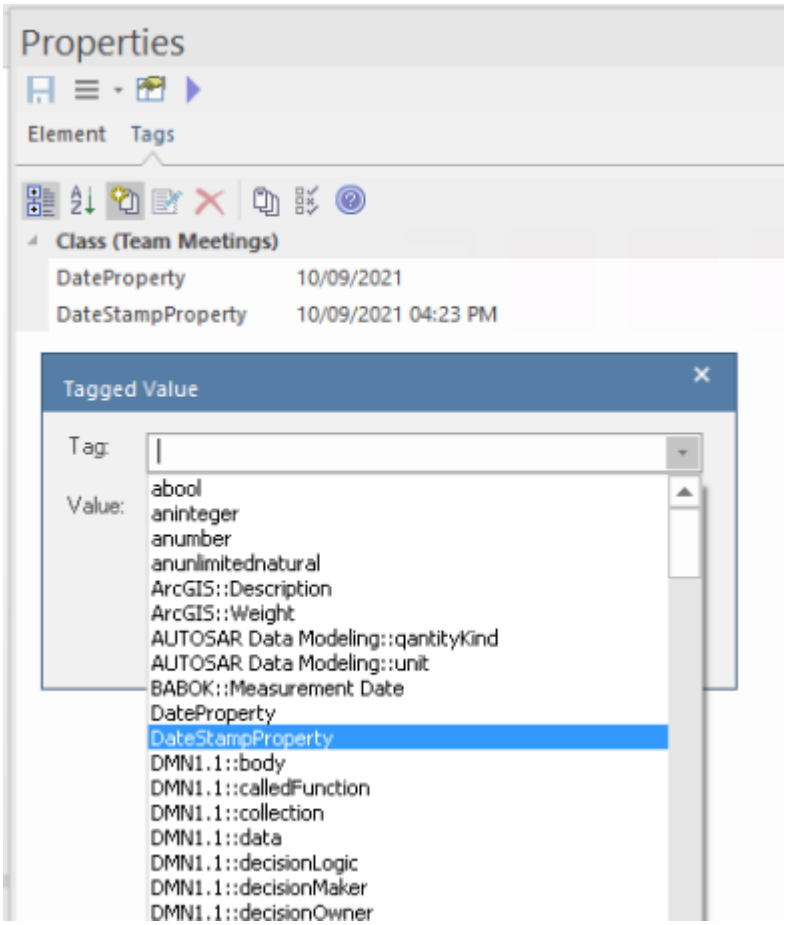


'Meeting' 配置文件元素现在是 Ctrl+拖动到 Team Meetings类，而 'Holiday' 配置元素是 Ctrl+拖动到 Public Holidays类。结果是两个类元素都采用了适当的构造型，这些构造型显示在图表中的元素上以及属性窗口的 元素”选项卡上的 构造型”字段中。另请注意， 元素”选项卡上有一个构造型组，列出了为构造型定义的标签。



对于团队会议元素，我们刚刚在元素的 开始日期”字段中输入了一个值，该值立即显示在图表上元素的 标签”隔间中。

如果您需要将构造型标签添加到其他类元素，一旦导入配置文件，您可以通过每个元素的属性窗口的 标签”选项卡执行此操作，选择最适合该元素的格式。注记如何此示例中两种格式不同，其中插入了两种数据类型。



使用预定义的标签类型（传统Profiles ）

标记值定义了一个模型元素具有的范围广泛的属性和特征，而这些属性中的一些是复杂的或结构化的值。例如，您可能希望您的用户在上限和下限之间选择一个值（使用 旋转”箭头）、设置日期和时间、从调色板中选择一种颜色，或者通过核对表进行操作。

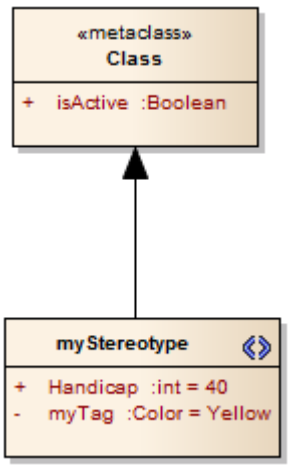
您可以使用 “UML类型”对话框的 标记值类型”页面 - '设置>参考，从许多预定义的简单标记值类型和过滤器中的任何一个创建这些复杂的标记值，其中一些可能是您自己创建的> UML类型 >标记值类型'。您在标记值的注记中指定标记值标记值类型（使用*Predefined Structured Types*帮助主题中列出的定义）。

请注记，此方法支持维护现有配置文件和MDG 技术，但对于新的或不完整的配置文件，我们建议您使用类型-请参阅*With Predefined Tag Types*帮助主题。

使用数据类型元素可以帮助您定义特定于配置文件的标记标记值，因此您可以在不同的配置文件中创建不同类型的同名标记值标记值而不会在运行从这些配置文件派生的MDG 技术时发生冲突。 标记值类型”页面适用整个模型，所有同名标签必须为同一类型。这可能会妨碍您在模型中创建多个配置文件，当他们使用这些全局标签并且其中一个或多个发生冲突时，无论是在技术开发模型中还是在启用该技术的任何模型中。

将值分配给构造型标记值

创建结构化标记值后，您可以将其分配给构造型元素，方法与简单标记值相同，方法是在构造构造型元素中创建一个具有标记值类型名称的属性。例如，要使标记值'Handicap' 出现在构造型中，请创建名为'Handicap' 的属性。根据标签类型，您可以通过为属性赋予初始值来设置标签的默认值。



定义构造型约束

如果需要定义构造型元素运行和存在的条件和规则，可以通过在元素上设置约束来实现。典型的约束是前置条件和后置条件，它们指示在创建或访问元素之前必须为真的事物以及在元素被销毁或其动作完成之后必须为真的事物。

您可以使用 分隔可见性”函数直接在图表上显示元素的约束。

访问

选择构造型元素，然后使用本表中列出的任何方法显示 属性”对话框的 约束”页面。

功能区	设计>元素>职责>约束
上下文菜单	右键单击元素|属性|职责 >约束
键盘快捷键	Shift+Alt+C
其它	双击构造型元素>约束

为构造型定义约束

字段/按钮	描述
新的	单击此按钮可清除准备创建新约束的字段。
约束	类型约束的值。
类型	单击下拉箭头并选择适当的类型（前置条件、后置条件或不变量）。
状态	单击下拉箭头并选择适当的状态。
注记	类型所需的任何附加信息。
节省	单击此按钮以保存约束数据。
确定	单击此按钮可关闭对话框。

添加形状脚本

UML元素和连接器在形状、颜色和标签方面都具有标准外观。可以通过多种方式更改元素类型或连接器的外观，使用形状脚本来定义要施加在默认或主形状上的确切特征。如果要标准化外观以应用于许多元素，请将形状脚本脚本附加到UML配置文件（例如 MDG元素MDG 技术UML配置文件）中的构造型属性。

访问

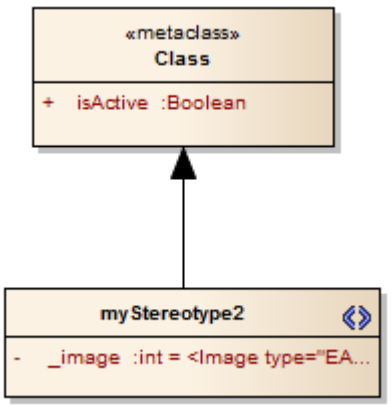
对于在UML配置文件中定义构造型的元素，定义一个名为 形状脚本”的属性，该属性将指定形状脚本。

通过单击 形状脚本”属性的 初始值”字段中的浏览图标来显示形状脚本编辑器。

功能区	设计>元素>特征>属性>[定义或选择属性'_image']>单击 初始值”字段中的 ...。
上下文菜单	右键单击构造型元素|特征|属性 | <定义或选择属性'_image'> |单击 初始值”字段中的 ...
键盘快捷键	F9 | <定义或选择属性'_image'>] |单击 初始值”字段中的 ...

将形状脚本脚本添加到构造型元素

构造型元素现在类似于这个例子：



节	描述
1	在 名称”字段中，输入 _“image”。
2	单击 初始值”字段旁边的 ... 按钮。 将显示 形状编辑器”对话框。
3	在 形状编辑器”对话框中输入形状脚本。 写完形状脚本后，单击确定按钮，然后单击关闭按钮。

注记

- 你的形状脚本可能包含外部定义的图像；在这种情况下，形状脚本将包含 image 方法，指定以技术名称为前缀的图像文件名
- 如果您正在为关联类创建形状脚本，请注记该形状脚本适用于类部分和关联部分；因此，您可能必须在 shape main 中包含测试元素类型的逻辑，以便您可以为类和关联提供单独的绘图说明

  这种逻辑在以下情况下是不必要的：
  - 形状源或形状目标，它们被类忽略，或者
  - 装饰形状，被关联忽略
- 您还可以临时将形状脚本应用于元素，将形状脚本附加到“UML类型”对话框（ 设置 >参考> UML类型”）上定义的形状脚本型

设置默认外观

如果您想为构造型元素或连接器定义一个简单的默认外观，您可以选择定义它的构造型元素并设置任何或全部：

- 背景/填充颜色
- 边框颜色
- 边框线宽，或
- 字体颜色

要设置这些，请使用 默认外观”对话框。

访问

上下文菜单	元素右键|外观|默认外观
键盘快捷键	F4

注记

- 当您保存定义原型元素和连接器的配置文件时，选择 保存UML配置文件”对话框上的 颜色和外观”复选框

特殊属性

可以定义原型元素模型的许多特殊特征和行为，例如在浏览器窗口和图表工具箱中表示它的图标，与原型关联的任何图像文件的默认位置，图表中的元素，或者外观是否由形状脚本定义。您可以在您的配置文件中定义这些特征，使用可应用于以下任一项的特殊属性：

- 构造型元素或
- 元类元素，指的是扩展它们的原型

访问

功能区	设计>元素>特征>属性
上下文菜单	右键单击元素|特征|属性
键盘快捷键	F9

设置属性

字段/按钮	描述
名称	类型属性的名称（在这些库表中列出）。
最初的	类型或选择属性的初始值。
关	单击此按钮可关闭对话框。

构造型元素属性

属性	意义
_defaultAttributeType	定义从图表工具箱创建的新属性的默认类型。在扩展属性元类的构造型元素中使用this，并将 初始值”字段设置为所需的属性。 如果您不提供此选项，系统将创建默认类型为**int**的属性。
图标	包含在浏览器窗口中由构造型定义的所有元素旁边显示的 16x16 像素图标的位图文件位置。这不适用于包元素。该图标也自动用作图表工具箱图像，只要列出了原型元素。 对于透明背景，您可以使用浅灰色 - RGB (192,192,192)。 要使该属性正常工作，还要设置 _metatype 属性。
_图片	标识一个形状脚本脚本定义，在 初始值”字段中为其创建脚本。

	要使该属性生效，您需要在保存配置文件时设置 图像"选项。
_instanceMode	已弃用。
_instanceOwner	已弃用。
_instanceType	将对话框中 粘贴as"字段的第二个选项修改为： • 作为元素的实例(ProfileName::<<stereotype>>) <<stereotype>> 值在属性的 初始值"字段中定义，并对应于使用 _"metatype"属性赋予原型元素的元类型。
_元类型	将原型定义为元类型，以便隐藏元素作为自定义的原型元素的身份。
_scriptlet	在显示图表之前定义一个脚本来设置和自定义此原型的元素。
_sizeY	设置元素的初始高度，以像素为单位，在 100% 缩放。 要使此属性生效，您需要在保存配置文件时设置 无素大小"选项。注记：不能将元素设置为小于其默认最小高度。
_sizeX	设置元素的初始宽度，以像素为单位，在 100% 缩放。 要使此属性生效，您需要在保存配置文件时设置 无素大小"选项。注记：不能将元素设置为小于其默认最小宽度。
_严格	定义一个原型元素可以应用多个原型的程度。

元类元素属性

属性	意义
_AttInh	如果设置为1 ，则在每个新的定型元素模型上将 继承特征：显示属性"复选框设置为选中。
_AttPkg	如果设置为1 ，则在每个新的原型模型元素上将 属性可见性：包"复选框设置为选中。
_AttPri	如果设置为1 ，则在每个新的原型模型元素上将 属性可见性：私有"复选框设置为选中。
_AttPro	如果设置为1 ，则在每个新的原型模型元素上将 属性可见性：受保护"复选框设置为选中。
_AttPub	如果设置为1 ，则在每个新的原型模型元素上将 属性可见性：公共"复选框设置为选中。
组成种类	当应用于关联时，定义源端或目标端是聚合还是复合。允许的值为： • 没有任何 • 聚合源 • 聚合在目标

	• 复合源 • 复合目标
_ConInh	如果设置为1 ，则在每个新的定型模型元素上将 显示元素隔间：继承约束”复选框设置为选中。
_约束	如果设置为1 ，则在每个新的定型模型上将 显示元素隔间：元素约束”复选框设置为选中。
_dbtype	当与扩展数据库建模原型（例如<<EAUML::table>>）的原型一起使用时，将为新元素设置默认数据库语言。
_defaultDiagramType	定义元素组合时创建的子图的类型。
方向	当任何类型的连接器元类元素从 配置文件”工具箱页面拖到图表上时自动创建。您可以为此属性设置一个值，而不是使用**_SourceNavigability**或**_TargetNavigability**属性。
_gentype	设置非数据库元素的默认代码生成语言。
_HideMetaclassIcon	如果您要扩展一个将以矩形表示法显示的元素并且您不希望它在右上角显示 无类”图标，则设置为True 。影响需求、组件和使用案例等元素。
_HideStype	通过为每个新的刻板模型元素设置 隐藏刻板特征”过滤器，将 初始值”字段设置为以逗号分隔的刻板印象列表以隐藏这些刻板印象。
_HideUmlLinks	如果您使用元模型来创建您的快速链接器定义并且您想从您的原型源元素的快速链接器中排除UML快速链接器定义，则设置为True。 （否则， UML快速链接器定义将从基本的UML元类继承。）
_常见	如果您不希望在创建目标元素时在快速链接器中提供关系，请将其设置为True 。
_isVertical	为构造型 ActivityPartition 设置为True以使默认活动分区方向垂直。
_lineStyle	设置原型连接器的线型；属性的 初始值”可以是以下之一： • 直接的 • 汽车 • 风俗 • 贝塞尔曲线 • 树H（水平） • 树V（垂直） • treeLH（横向横向） • treeLV（横向垂直） • 正交S（正交、方角） • 正交R（正交、圆角）
_makeComposite	使每个原型元素在创建时成为复合元素。
_含义向后	从目标读到源时，关系A自然语言含义。例如，<<Flow>>关系可能将 _MeaningBackwards 设置为 Flows from”。用 可追溯性窗口和其他地方。

_意义转发	从源读到目标时关系A自然语言含义。例如，<<Flow>>关系可能将 _MeaningForwards 设置为 "Flows to"。用 可追溯性窗口和其他地方。
_OpInh	如果设置为1 ，则在每个新的定型模型元素上将 继承的特征：显示操作"复选框设置为选中。
_OpPkg	如果设置为1 ，则在每个新的原型模型上将 操作可见性：元素包"复选框设置为选中。
_OpPri	如果设置为1 ，则在每个新的定型模型元素上将 操作可见性：私有"复选框设置为选中。
_OpPro	如果设置为1 ，则在每个新的定型模型元素上将 操作可见性：受保护"复选框设置为选中。
_OpPub	如果设置为1 ，则在每个新的原型模型元素上将 Operation Visibility: Public"复选框设置为选中。
_PType	如果设置为1 ，则将在每个新立体元素上选中的 显示元素类型（仅端口模型或部件）"复选框设置为。
_树脂	如果设置为1 ，则在每个新的定型模型元素上将 显示元素隔间：继承的职责"复选框设置为选中。
_责任	如果设置为1 ，则在每个新的原型模型上将 显示元素分隔：元素需求"复选框设置为选中。
_运行状态	如果设置为任何非空白值，则在每个新的定型模型元素上将 隐藏当前图中的物件运行状态"复选框设置为选中。 要显示运行状态，请省略此属性或给它一个空白值。
_SourceAggregation	已弃用。请参阅**compositionKind** 。
_SourceMultiplicity	设置源元素的多重性，例如1 ..* 或 0.. 1 。
_SourceNavigability	如果连接器不可导航，则将此属性设置为 不可导航"。 如果其他值更合适，请使用**方向**属性。
_subtype 属性	指定每次从工具箱创建具有构造型的元素时用于生成弹出子菜单的标记值的完全限定名称。 标记值是一个枚举，子菜单由每个枚举文字的命令组成。使用子菜单中选择的任何命令初始化标记值；如果没有选择（例如，如果用户单击子菜单），则默认值将照常使用。 例如，如果您创建 BPMN 2活动元素，则会显示一个子菜单，其中列出了任务类型，例如"BusinessRule"、"Manual"和 接收"。选择其中一个值将其设置为taskType标记值。 值实际上是活动标记值的子类型；在 BPMN 2 配置文件中，在格式 profile::stereotype::tag 中，活动原型的 subtypeProperty 将是： BPMN2.0::活动::taskType。
_标签	如果设置为1 ，则在每个新的原型模型元素 显示元素分隔：标签"复选框设置为选中。

_tagGroupings	将标记值映射到属性窗口的 标记”选项卡中显示的标记组中，格式为： 标记名1=组名1；标记名2=组名2； 此功能目前仅适用于object类型，不适用于属性等其他类型。
_tagGroups	定义所需组的逗号分隔列表，按照它们在属性窗口的 标签”选项卡中的显示顺序。例如： 组名1、组名2、组名3 此功能目前仅适用于object类型，不适用于属性等其他类型。
_tagGroupStates	将属性窗口 Tags”选项卡中显示的属性映射到打开或关闭状态，形式为： 组名1=打开；组名2=关闭； 此功能目前仅适用于object类型，不适用于属性等其他类型。
_TagInh	如果设置为1 ，则在每个新的原型模型元素上将 显示元素分隔：继承的标签”复选框设置为选中。
_TargetAggregation	已弃用。请参阅**compositionKind** 。
_TargetMultiplicity	设置目标元素的多重性，例如1 ..* 或 0.. 1 。
_TargetNavigability	如果连接器不可导航，请将此属性设置为不可导航。 如果其他值更合适，请使用**方向**属性。
_UCRect	（仅适用于具有不同矩形符号的元素类型，或具有评估 'rectanglenotation'属性的形状脚本的元素，其中可以包括通常不具有矩形符号的元素类型。 ） 如果设置为1 ，最初显示矩形表示法中的元素。如果设置为 0，则最初以标准符号显示元素。

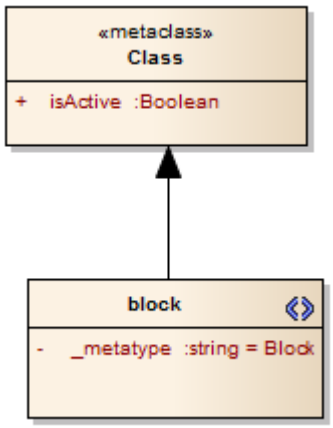
注记

- 如果属性设置为1以打开特征，则将其设置为 0 会关闭特征

将构造型定义为元类型

如果您想将自定义元素的身份隐藏为原型UML元素，您可以在定义它的构造型元素中设置 _metatype 特殊属性。 _metatype 属性还使自定义元素类型出现在只有Enterprise Architect的内置类型通常会出现的上下文中；例如，在关系矩阵的元素类型列表中。

在这个来自 SysML 的示例中，块被定义为扩展UML类的构造型元素。



但是，SysML 用户对UML类不感兴趣，只对 SysML 块感兴趣。如果您将 _metatype 属性设置为块，则从该构造型创建的任何元素，虽然在大多数情况下的行为方式与构造型类相同，但将：

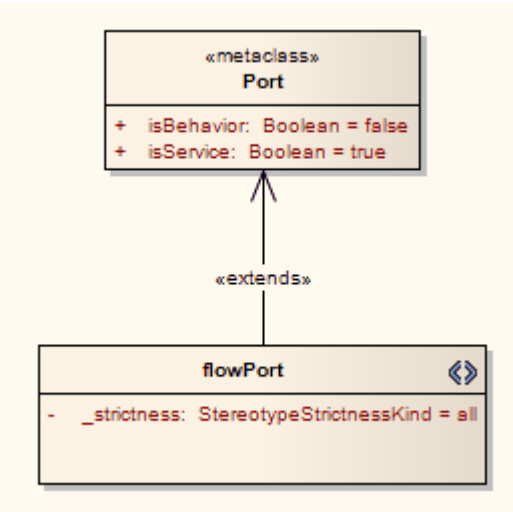
- 显示块<名称> 而不是类<名称> 作为其 属性"对话框的标题
- 在创建时自动编号为 Block1 而不是 Class1，并且
- 在整个Enterprise Architect的许多其他上下文中显示为块而不是类

定义多重刻板印象级别

一个元素可以有多个应用到它的刻板印象。您可以通过在定义构造型元素中创建 _strictness 特殊属性来定义可以应用多个构造型的级别。属性的类型是 StereotypeStrictnessKind，在 初始值"字段中具有四个值之一：

- profile，它指出不能从同一个配置文件中给这个类型的元素一个以上的构造型
- 技术，它指出这种类型的元素不能从同一技术中获得多个刻板印象
- all，它表明这种类型的元素根本不能有多个构造型，或者
- none，这是默认行为，并声明对使用多个构造型没有限制

此示例来自 SysML，并表明 «flowPort» 不能应用任何其他构造型。

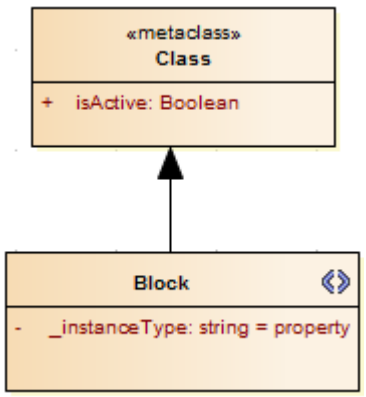


定义实例的创建

A原型元素可以是从它创建的实例的分类器。您可以通过向定义构造型添加特殊属性来定义如何从该构造型元素创建实例。属性修改 粘贴As"对话框上的文本，当一个原型元素被拖出浏览器窗口到图表上时，该对话框会显示。

属性

这个来自 SysML 的示例显示了可能创建的 SysML块元素的任何实例的定义。



当用户将 SysML块元素从浏览器窗口拖到图表上时，系统会检查 _instanceType 属性值并在 SysML配置文件中搜索具有匹配 _metatype 属性值的元素模板，并从中生成实例。通过示例定义，您将获得具有 属性"构造型的块元素。

属性	意义
_instanceMode	已弃用 将对话框中 粘贴as"字段的第二个选项修改为： • 实例（<元素类型>）或 • 属性（物件） 文本由属性的 初始值"字段的值（ 实例"或 属性"）确定。 如果未应用该属性，则该选项默认为 实例"。
_instanceOwner	已弃用 将对话框中 粘贴as"字段的第二个选项修改为： • 作为 <元素> 的实例 文本由属性的 初始值"字段的值确定，例如 块"。 如果未应用该属性，则该选项默认为 元素"。
_instanceType	将 粘贴as"对话框中 粘贴as"字段的第二个选项修改为： • 作为元素的实例(ProfileName::<<stereotype>>) <<stereotype>> 值在属性的 初始值"字段中定义，并对应于使用 "metatype"属性赋予原型元素的元类型。 注记您可以使用 _instanceType 属性或元约束来定义实例的创建。区别在于： 1. 在 粘贴As"对话框中，元约束允许您定义多个实例原型，而 _instanceType 则没有。多个实例都列出来了；这是一个非常有用的特征。 2. 元约束（当前）对 _instanceType 所做的 转换为实例"命令没有任何影

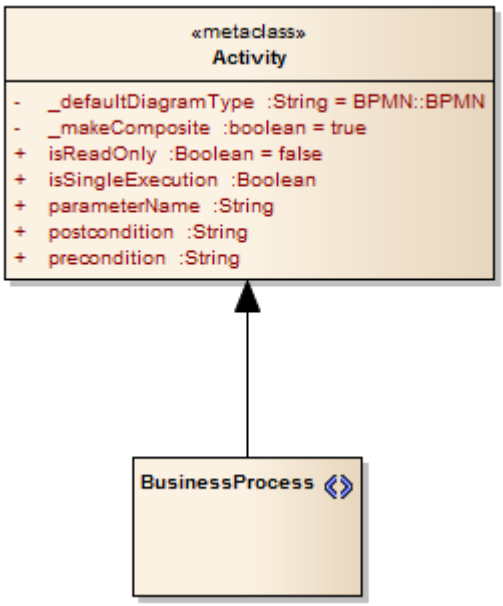
	响。

定义复合元素

原型元素A自动创建为复合元素。您可以使用特殊属性定义这一点，以及组合的子图是否属于特定类型。

要定义元素是否总是在创建时复合，您将 _makeComposite 特殊属性应用于适当的元类元素（而不是原型元素）。构造A类在创建时默认没有子图，因此您使用 _makeComposite 属性来触发子图的创建。对于原型组合，子图是元类通常的默认图类型；您可以使用 _defaultDiagramType 特殊属性更改子图表类型，以识别首选图表类型，

来自 BPMN 的这个示例表明，BusinessProcess元素始终创建为具有 BPMN 自定义子图的复合元素。

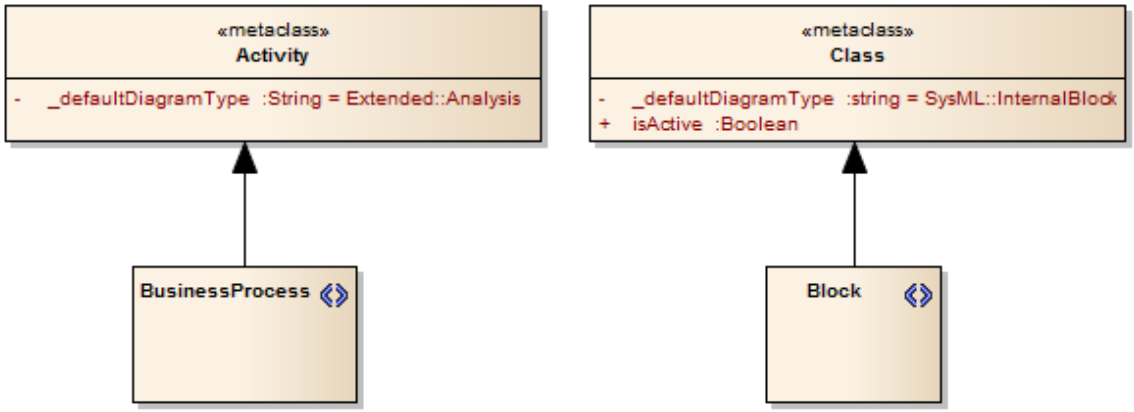


定义子图表类型

如果您将构造型元素类型定义为组合，则其子图类型最初与您扩展的元类元素的默认类型相同。您可以使用 _defaultDiagramType 特殊属性将图表类型更改为任何其他内置UML或扩展类型，或任何您自己的自定义图表类型。由于图类型默认来自 Metaclass元素，因此您可以在该 Metaclass元素（而不是构造型元素）上设置属性以更改默认值。

您在属性的 初始值”字段中标识子图表类型。内置UML和扩展图类型的实际值列在*Initial Values*部分中。如果要设置自定义图表类型，请在图表类型名称前加上图表配置文件名称和 ::”。图表配置文件名称是您保存配置文件时为其指定的名称，默认为配置文件包或配置文件图表的名称。我们建议图表配置文件名称基于技术名称。您还可以使用包的 _defaultDiagramType 属性，扩展包元类元素。

这些示例展示了一个 «BusinessProcess»活动，当制作一个复合元素时，它会自动创建一个分析图，以及一个创建 SysML InternalBlock 自定义图的 «block» 构造型。



初始值

这些字符串可用于 _defaultDiagramType 的 初始值”字段，以识别内置UML和扩展图类型：

- UML行为::用例
- UML行为::活动
- UML行为：状态机
- UML行为::通讯
- UML行为::序列
- UML行为::时间
- UML行为：：交互
- UML结构::包
- UML结构::类
- UML结构::物件
- UML Structural::复合结构
- UML Structural::部件
- UML结构::部署
- 扩展::自定义
- 扩展::需求
- 扩展::维护
- 扩展::分析
- 扩展::用户接口

- 扩展::数据建模
- 扩展::模型文档。

注记

- 尽管我们建议自定义图表类型的图表配置文件名称基于技术名称，但属性前缀并不是对技术名称的直接引用

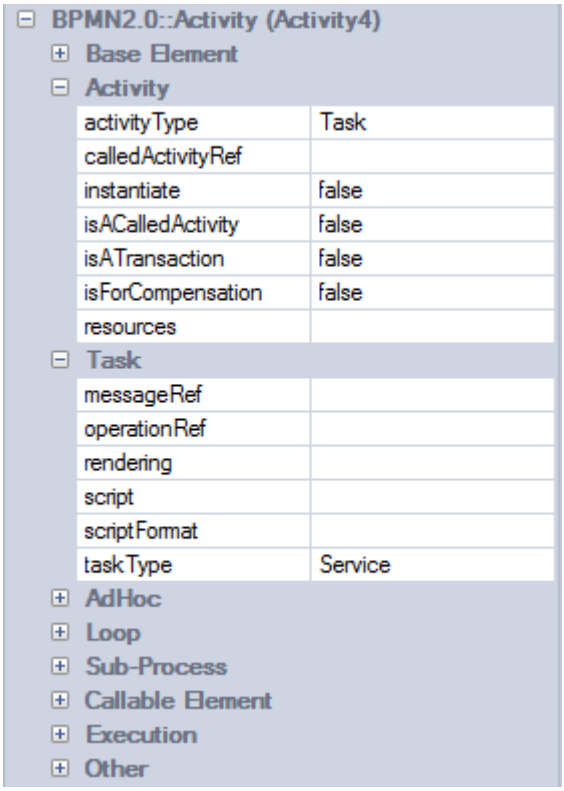
定义标签分组

在配置文件中开发一个定型元素时，可能会定义大量的标记值。例如，BPMN 2.0配置文件中的一个BPMN活动元素有30个标记值。默认情况下，在元素的属性窗口的 "Tags"选项卡中，这些标记值最初都将按字母顺序显示，如果它们恰好具有按字母顺序遥远的名称，则可能会拆分相关标签。为了将相关标签保持在一起并控制最初显示哪些标签，在 BPMN 2.0配置文件中，标记值已被分组。您可以应用相同的解决方案，使用元类中的三个标记分组特殊属性，这些属性由构造型元素元素，其中标记被定义为属性。

您可以使用以下方式应用分组：

- _tagGroups 定义组名
- _tagGroupings 定义哪些标签进入每个组
- _tagGroupStates 定义哪些标记组最初在属性窗口的 标记"选项卡中展开，哪些标记组被折叠

BPMN 2.0活动元素属性窗口的 标签"选项卡最初显示如下：



活动元类属性

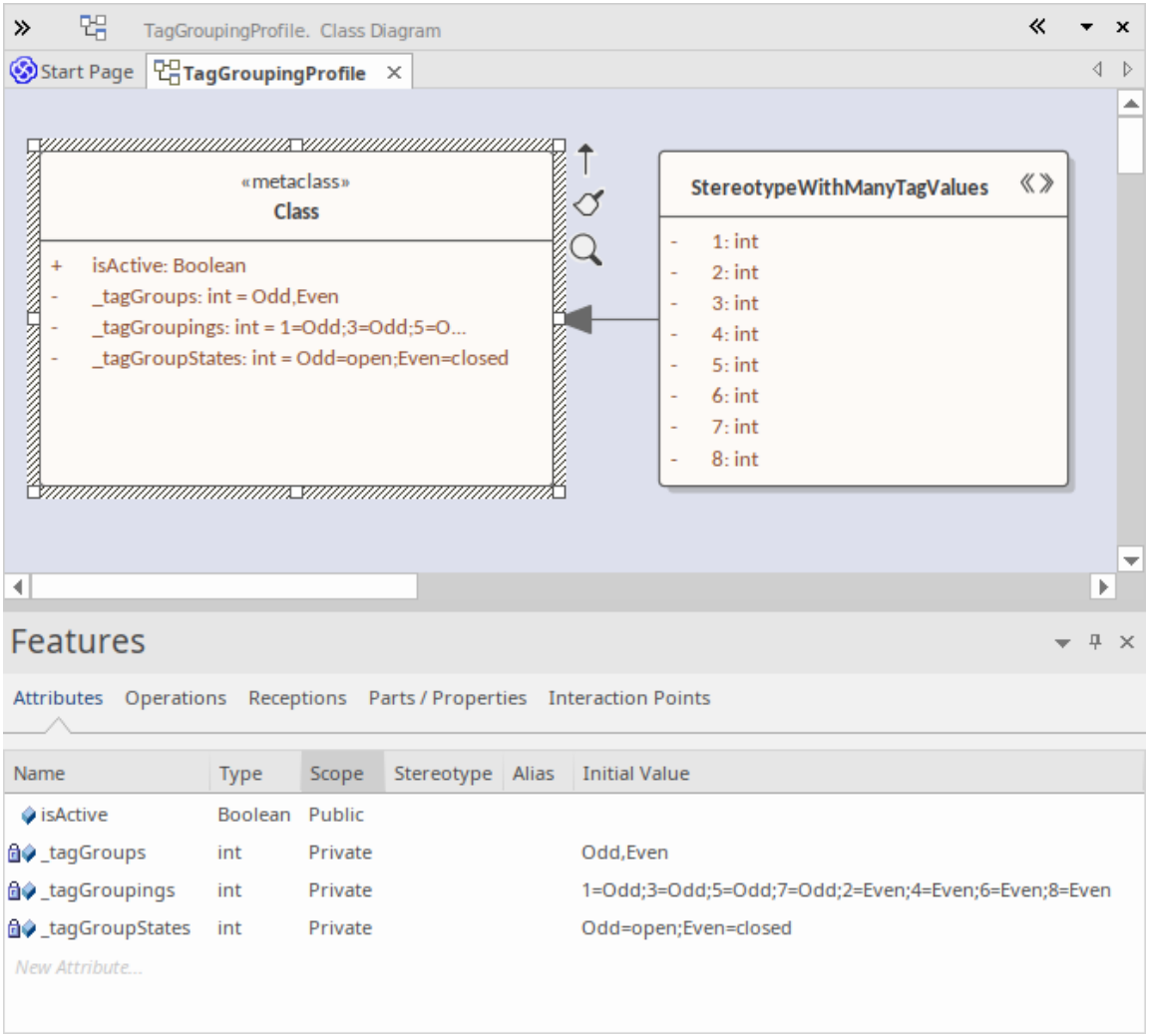
为了实现 BPMN 2.0活动标记值的显示，技术开发人员在活动元类元素中定义了特殊属性，如下所示：

属性	价值观
_tagGroups	Base元素,活动,任务,AdHoc,Loop,Sub-Process,Callable元素,其它,其他
_tagGroupings	auditing=基本元素；categoryValue=基本元素；documentation=基本元素；monitoring=基本元素；activityType=Activity;calledActivityRef=Activity;instantiate=Activity;isACalledActivity=Activity;isATransaction=Activity;isForCompensation=Activity;resources=Activity;messageRef=任务；操作引用=任务；渲染=任务；脚本=任务；scriptFormat=任务；taskType=任务；adHoc=AdHoc；adHocOrdering=AdHoc；...（等等）

_tagGroupStates	基本元素=关闭；活动=打开；任务=打开；AdHoc=关闭；循环=关闭；子流程=关闭；可调用元素=关闭；执行=关闭；其他=关闭

示例

此处显示的是如何使用标签分组属性的简单示例。



注记

- 此功能目前仅适用于object类型，不适用于属性等其他类型

介绍元模型视图

Enterprise Architect包括一个非常有效和灵活的系统定义和用户定义元模型的视图系统。视图系统提供高度集中的图表，将可用的元素和连接数量限制为仅用于完成特定任务所需的核心。例如，类图上的层次结构视图可能会将唯一可用的元素限制为 类”，并将唯一的连接器限制为 继承”。

使用视图系统来指导建模调色板和可用关系，您将构建紧凑且有目的的图表，这些图表仅使用当前建模上下文中所需的元素。消除噪音和减少可用构造集是确保设计解决预期目的并避免可能对模型的可读性和正确性产生负面影响的无关元素的好方法。

元模型视图

类别	描述
系统	Enterprise Architect提供多种内置视图，可满足众多建模场景和领域的需求。许多模型模式都预设了视图，而 图表生成器”对话框包含许多派生图表视图，可扩展和完善基本图表类型的功能。
风俗	除了在Enterprise Architect中使用系统定义的基于元模型的视图外，还可以创建自己的元模型并轻松地将它们添加到当前模型中，然后您和其他建模者可以根据需要将它们应用于各种图表。例如，您可以定义一个特定的元模型集来满足您组织中需求建模的需求，然后要求所有需求图都使用该元模型视图。

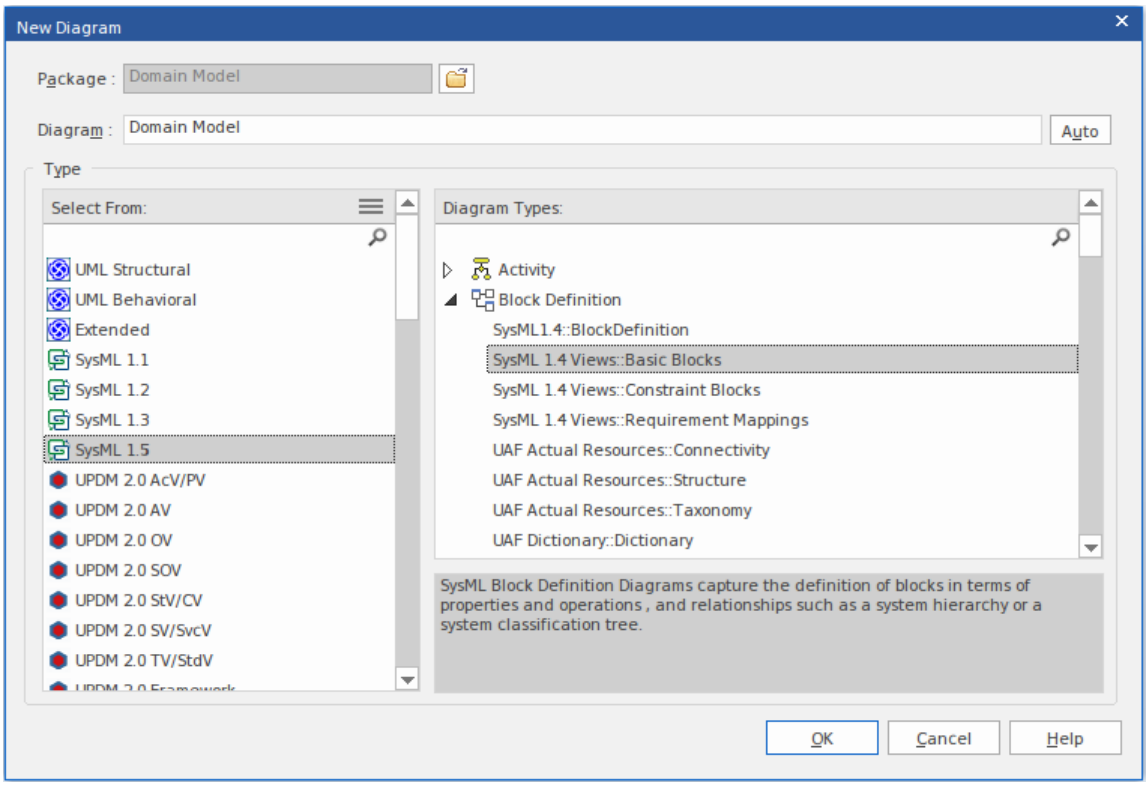
视图系统功能

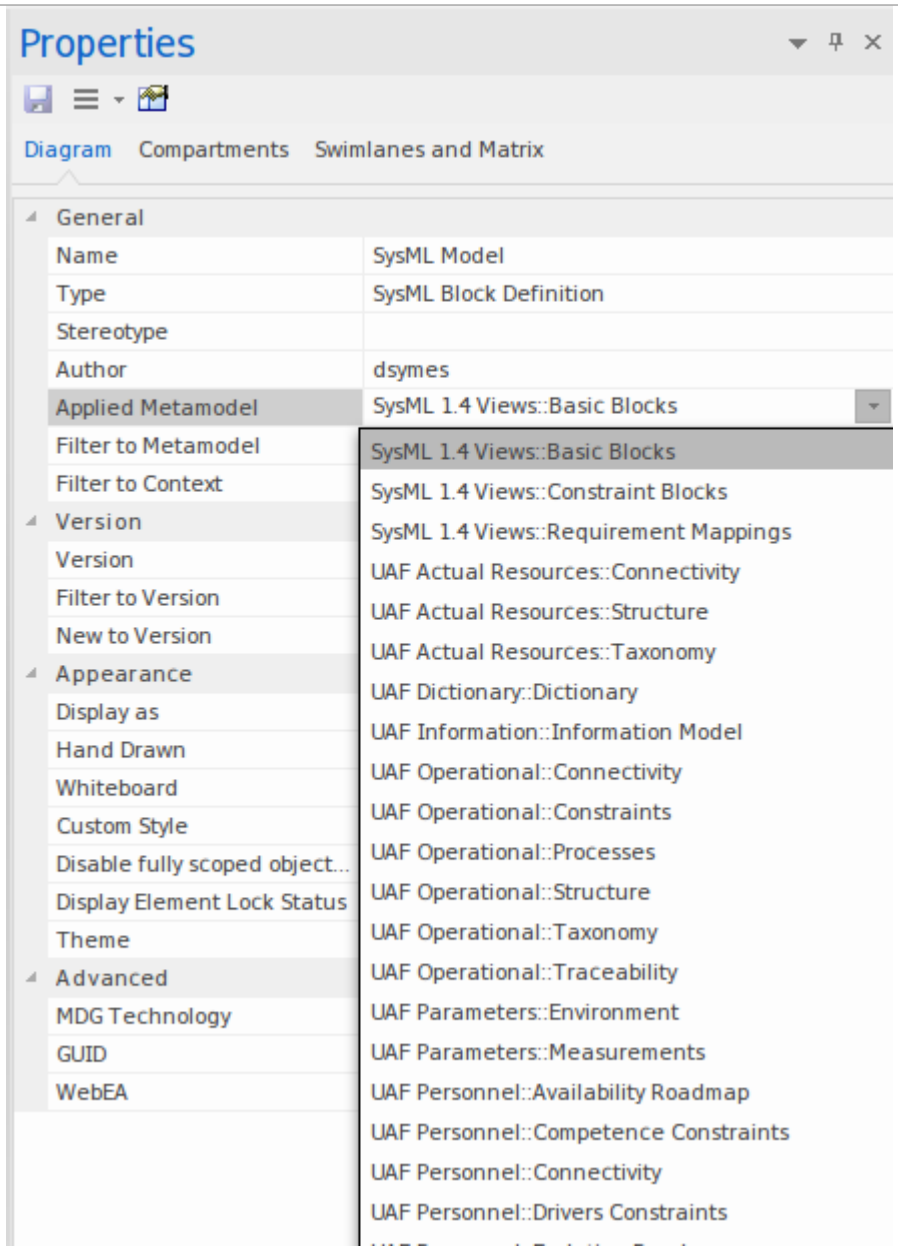
功能	描述
图表过滤器	除了限制可用的调色板外，视图系统还允许建模者启用图表过滤器，该过滤器将使不属于当前视图集的任何元素变灰。这允许建模者更正其模型中不符合所选视图目的的任何部分，或者过滤掉需要存在但不构成当前建模目标一部分的元素。
图表属性	图表的 属性”对话框包括当前所选图表类型的可用视图的下拉列表。选择其中一个视图将减少可用构造的调色板并限制快速链接器中的条目。建模者可以轻松激活视图，甚至在必要时移除视图 - 实际模型内容不会改变。
图表视图	图表构建器”对话框包括许多不同的视图，这些视图为UML 、SysML、BPMN 和 UAF 等图表类型提供不同的调色板集和聚焦目标。如果您的目标是对没有高级特征的简单活动图进行建模，那么UML活动图部分下的简单活动视图可能是比使用完整活动图集更好的选择。

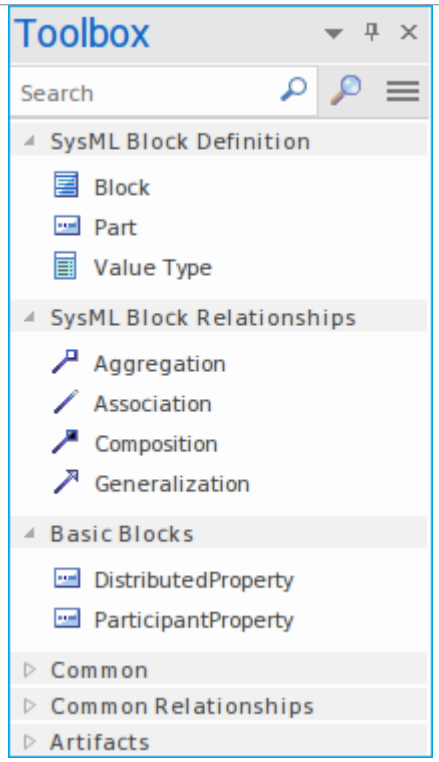
内置元模型图表视图

新图表”对话框包括许多不同的视图，为UML 、SysML、BPMN 和 UAF 等图表类型提供不同的调色板集和聚焦目标。例如，如果您的目标是对没有高级特征的简单 SysML块定义图进行建模，则 图表1 .5块定义图”部分下的 基本块视图”可能比使用完整块更好的选择定义图集。此示例用于在本主题的过程中提供值。

使用图表视图

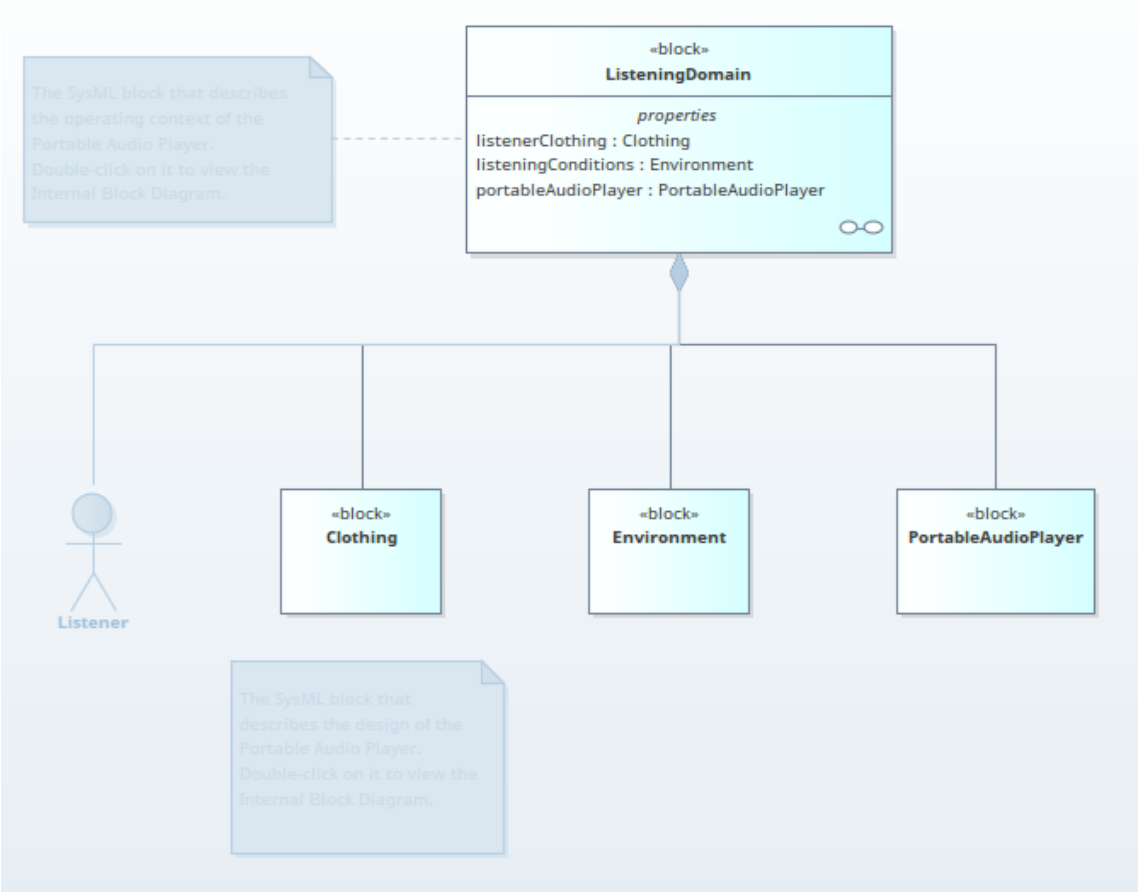
节	行动
1	在浏览器窗口中，单击要放置图表的包或元素。 打开 新图表”对话框，选择 SysML 1 .4视图:: Basic Blocks”并点击确定按钮来创建图表。 
2	在创建图表的属性窗口中， 应用元模型”字段将显示应用的图表视图。您还可以单击此字段中的下拉箭头，然后从列表中选择另一个可用的图表视图。

	
3	在图表工具箱中，与图表视图关联的受限元素和关系集将是可见的。



更改 应用元模型”选项列表中的图表视图将更改工具箱中的元素和关系。

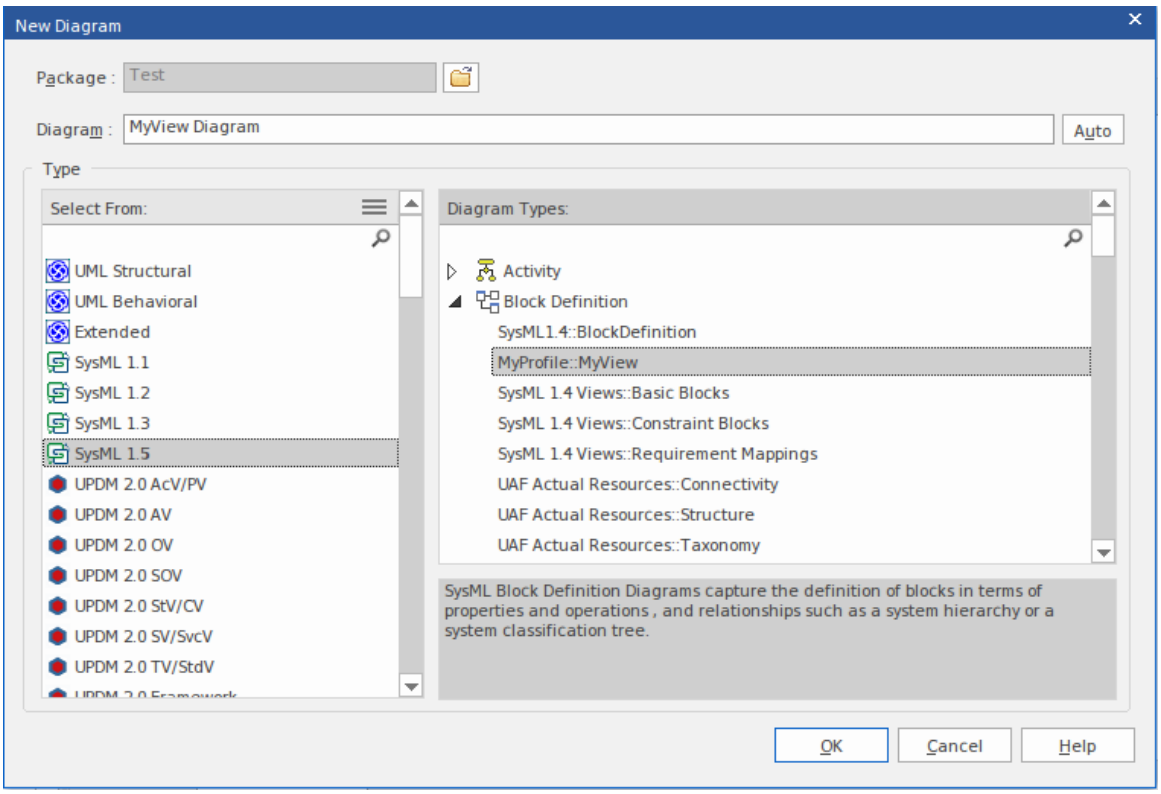
4 在属性窗口中选择 对滤器到元模型”选项将使不属于当前图表视图集的任何元素变灰。这使您可以更正模型中不符合所选视图目的的任何部分，或过滤掉可能需要存在但不构成当前建模目标一部分的元素。



自定义元模型图表视图

Enterprise Architect具有广泛的内置图表视图，但您也可以创建自己的元模型来定义自定义图表视图。例如，您可以定义一个特定的元模型来满足您组织中需求建模的需求，然后要求所有需求图表都使用该图表视图而不是内置的需求图表视图。您可以快速将图表视图添加到当前模型，您或其他建模者可以将它们应用于您的图表。

作为说明，假设您决定在您的项目中提供一个新的 SysML 1 .4块定义图视图，称为 "MyView"。用户将通过 "New块图表类型。

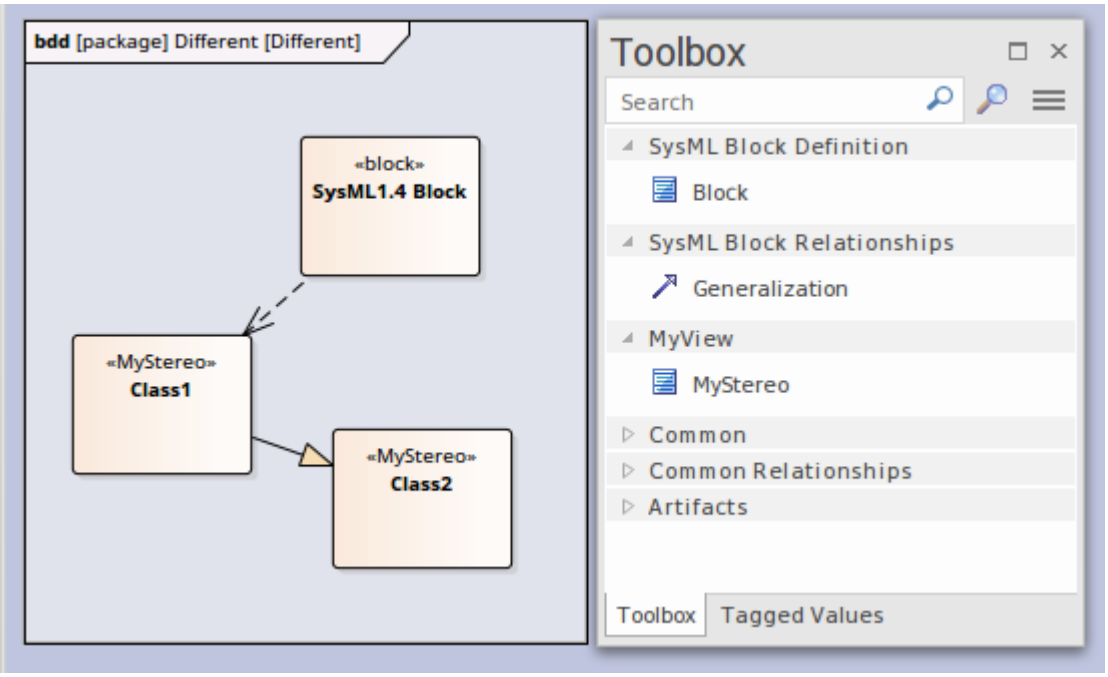


图表视图的完全扩展名称反映了父配置文件名称 (MyProfile) 和视图名称 (MyView) - 因此是 "MyProfile::MyView"。您可以调用示例视图1 1视图的视图。

如果您使用配置文件名称 "UML "扩展UML基本图类型，则等效视图名称可能是诸如 "UML ::视图"之类的名称。

用户选择示例图视图来创建一个非常简单的 SysML 1 .4块图，它可以具有：

- 两种元素：
  - SysML 1 .4块元素（ SysML 1 .4 技术的扩展类 ）
  - 您在新元模型 "MyView"中定义的 MyStereo元素
  具有原型 MyStereo 的类
- 一种标准的概括块连接器（ 与标准UML概括相同 ）



图表视图使元素和连接器可从工具箱（如图所示）和快速链接器中使用。

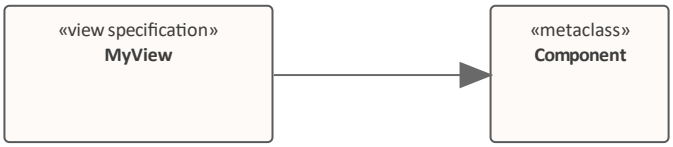
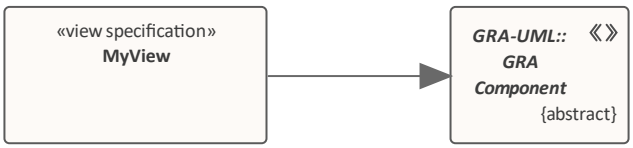
配置文件中的表配置文件自定义图表视图解释了如何创建定义新图表视图的元模型，最后以 MyView 示例结束。

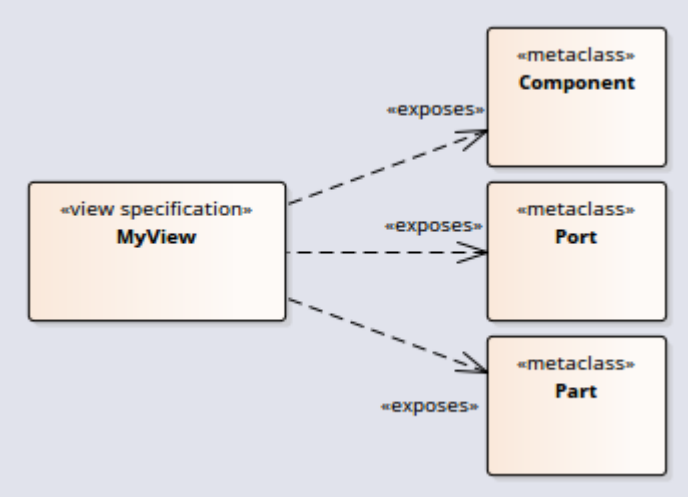
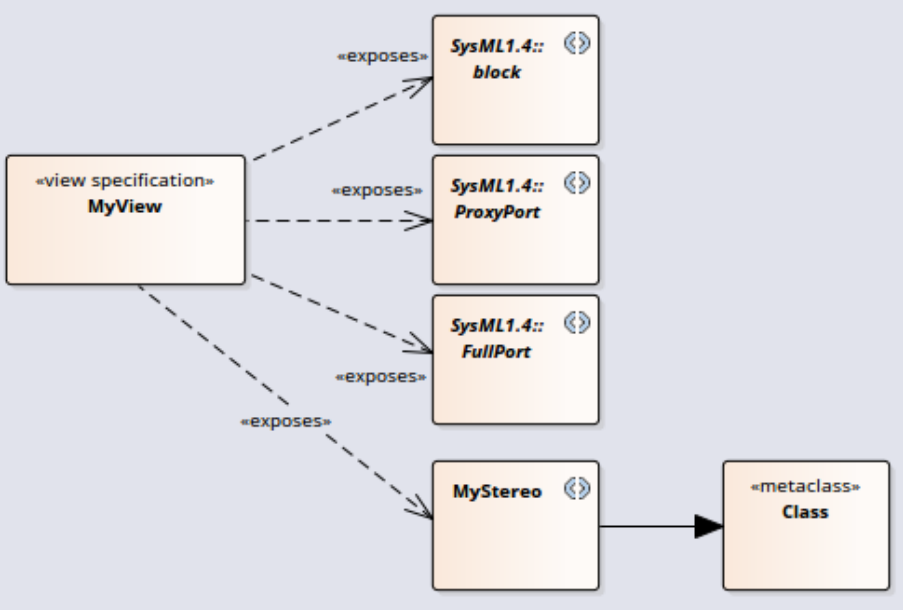
访问

功能区	设计>图表>工具箱： >配置文件>元模型
键盘快捷键	Ctrl+Shift+3： >配置文件>元模型

在配置文件中创建自定义图表视图

手术	行动
创建配置文件图	在您的配置文件包中，创建一个新包图，然后在图表工具箱中打开 配置文件”页面（选择 设计>图表>工具箱”功能区选项，然后单击 并选择 配置文件”）。 将 配置文件”图标拖到图表上并将其命名为 MyProfile”，选择添加名称为 “MyView”的子类图表，然后打开它。 展开工具箱中的 元模型”页面并注记： • '视图'元素，您可以使用它来创建自定义视图 • '工具箱' 连接器，用于指定与自定义图表关联的工具箱页面的视图
添加视图规格	在配置文件中，您使用 视图规范”原型元素将新的自定义图表视图为现有内置或原型图的扩展。 将 视图规范”图标拖到配置文件图上，并为元素命名；在我们的示例中，

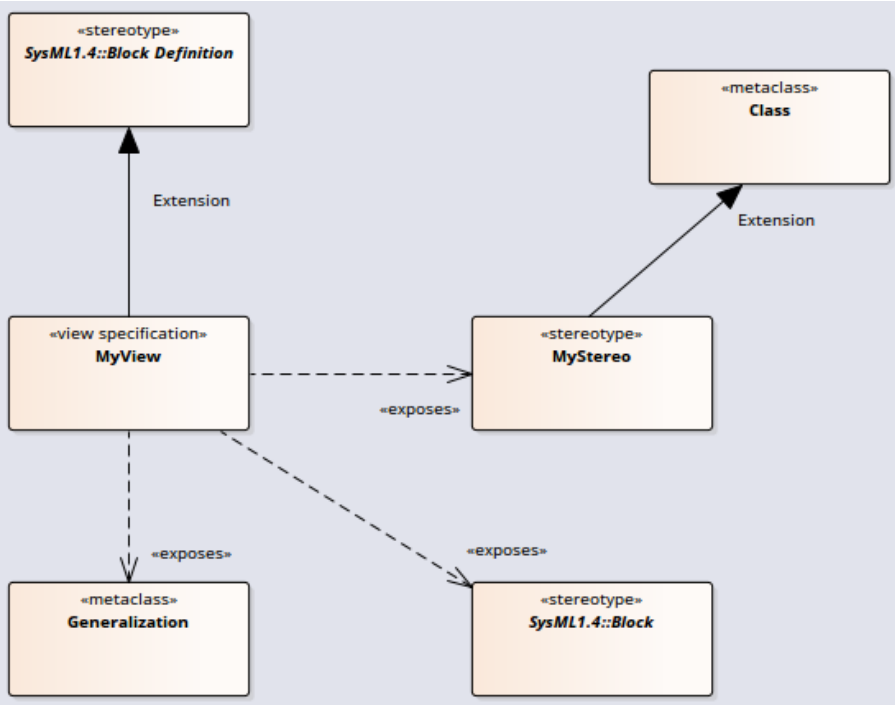
	“MyView”。 定义新视图时首先要考虑的是它应该适用于哪种图表类型。接下来的两行显示了如何为UML图和配置文件图定义视图。 在这两种情况下，单击 扩展”图标并从视图规范拖动到图表类型元素，以创建扩展连接器。
扩展UML图表类型	要扩展基本的UML图类型，将 类”图标从工具箱拖到图上，然后在属性窗口中，给出元素： • 图表类型的确切名称（在*Built-in*图表帮助主题中列出），例如 “Logical”（用于类图），以及 • 刻板印象<<元类>> 此示例显示之前创建的 MyView”，扩展了UML部件图。  结果是在 新图表”对话框中，在UML部件图表类型下添加了一个额外的视图。
扩展 Profiled图表类型	要扩展分析图表类型，例如 BPMN 或 SysML 图表类型，请将 构造型”图标拖到图表上，并为构造型元素提供图表类型的完全限定名称。 因为这是对外部原型的引用，所以它也应该被标记为 Abstract 以防止它被导出到配置文件中。为此，显示属性窗口，展开 高级”部分并选中 摘要”复选框。 此示例显示之前创建的 MyView”，扩展了 GRA-UML部件图表类型。  结果是 新建图表”对话框将显示我们在 GRA-UML 组件图下定义的视图。 注记：如果您不知道要扩展的图表类型的完全限定名称，请查询 API 以获取 “Metatype”字段。在JavaScript控制台中，您可以使用： ？获取当前图表()。元类型 或者，在浏览器中选择图表，然后在停靠的属性窗口中查看，它将在MDG 技术下列出。
在图表视图视图工具箱中工具箱对象	工具箱连接器将object添加到图表视图的工具箱页面。对于要添加到图表视图工具箱页面的每个元素和连接器，您将 定义元素”拖到图表上，然后单击工具箱 配置文件”页面中的 公开”图标并将光标从将视图规范元素添加到 定义元素”以创建连接器。 定义元素的类型取决于您是公开基本UML元素还是原型元素，如下两行所示。
暴露UML元素类型	如果您在自定义图表视图中使用基本UML元素或连接器，那么对于每个元素或连接器： 1. 将工具箱 配置文件”页面中的 元类”图标拖到图表上，并为其提供其代表的基本元素或连接器类型的名称，然后 2. 在视图规范元素和元类元素之间添加视图连接器 例如：

	
暴露已分析的元素类型	如果您在图表视图中定义一个新的原型object ，或者使用已经在其他配置文件中定义的原型元素，那么对于每个元素或连接器： 1. 将 构造型"图标从工具箱 配置文件"页面拖到图表上，并为元素指定其代表的构造型元素或连接器的名称 2. 如果构造型在另一个配置文件中定义，展开属性窗口的 高级"部分并选中 抽象"复选框 3. 如果在这里定义了构造构造型，则将构造型扩展的基本元素添加到图中，并在构造型和基本元素之间创建扩展连接器 4. 在视图规范元素和构造型元素之间添加视图连接器 例如： 
完成示例	参考表中前面的行，在 MyView类图（MyProfile 图的子图）上： 1. 创建元素规范视图。 2. 创建构造型元素SysML1.4::块并将其设置为Abstract。 3. 使用扩展连接器将视图规范连接到视图::块。 4. 创建一个元素的元概括。 5. 创建一个名为块::block 的构造型元素并将其设置为 Abstract。 6. 创建一个名为 MyStereo 的构造型元素和一个名为UML类的元类元素，并

使用扩展连接器将构造型连接到元类。

7. 将元素视图规范元素连接到元素、概括::块元素和元素，每个元素都有一个 Exposes 连接器。

此插图代表您创建的图表：



当您完成您的图表视图时，您可能会决定应该使用特定类型的连接器将一种类型的元素连接到相同类型或其他类型的元素。您可以使用元关系连接器来定义这一点，如定义元模型约束帮助主题中所述。

保存视图规格图。您现在可以将其作为其父配置文件的一部分添加到MDG 技术文件中；您将父配置文件添加到 'MDG 技术向导 -配置文件文件选择"页面。请参阅**添加配置文件**帮助主题。

定义元模型约束

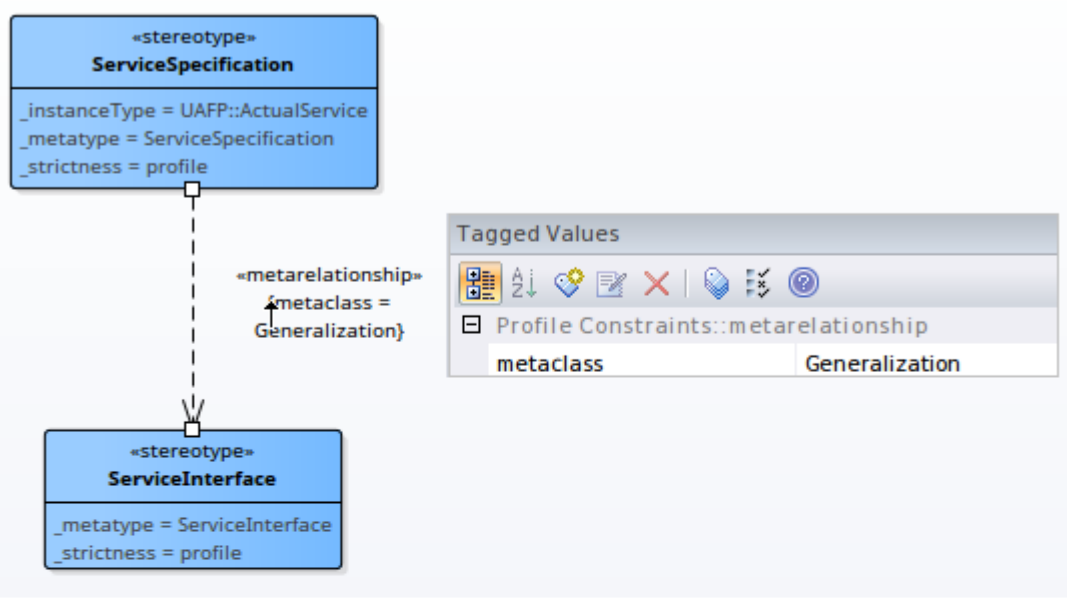
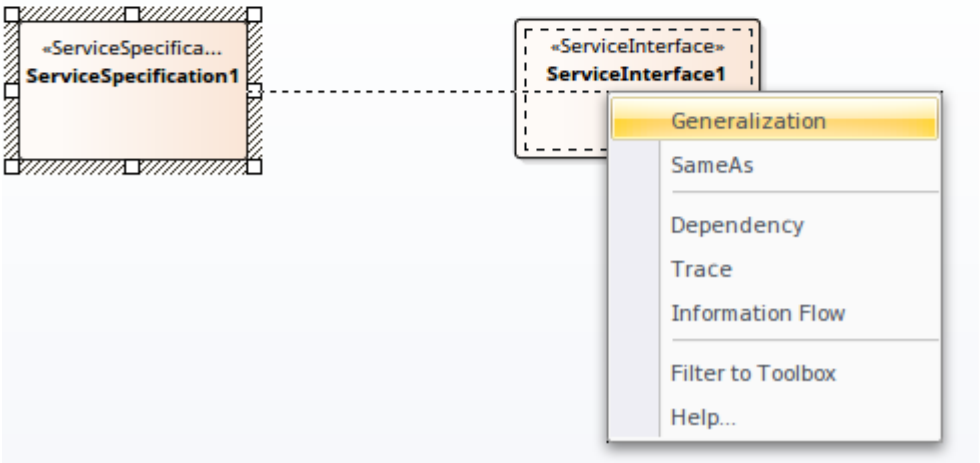
当扩展UML以开发特定于域的配置文件时， Enterprise Architect允许您指定约束以限制可以从构造型绘制的连接器，使用快速链接器或工具箱。这些约束是使用 配置文件”工具箱的 无模型”页面下的关系定义的。

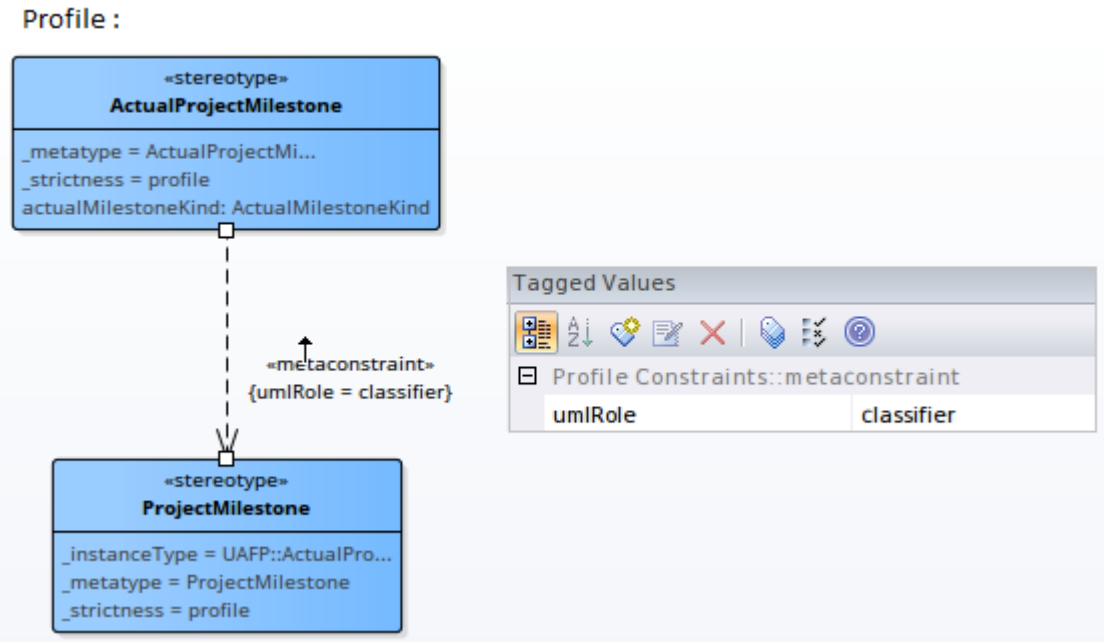
访问

功能区	设计>图表>工具箱： >配置文件
键盘快捷键	Ctrl+Shift+3

将元模型约束添加到配置文件

物品	细节
元关系	两个构造型之间A 无关系”连接器用 指定这两个构造型之间的有效UML连接器。 UML连接器的名称应该设置在«metarelationship» 连接器上的标记 无类”中。

	Profile :  Quick Linker in Model :  在配置文件示例中，从 ServiceSpecification 到 ServiceInterface 绘制了一个 «metarelationship» 连接器，并且UML连接器的名称在连接器的属性窗口的 "Tags"选项卡中指定。 将此配置文件导入模型后， Enterprise Architect将在使用 Quick Linker 绘制 ServiceSpecification 和 ServiceInterface 之间的关系时显示UML连接器。
元约束	两个构造型之间A «metaconstraint» 连接器用于指定这两个构造型之间的约束。 应在 Meta-Constraint 连接器上的标记 "umlRole"中设置约束。



在配置文件示例中，从 ActualProjectMilestone 到 ProjectMilestone 绘制了一个 «metaconstraint» 连接器，并且在连接器的标记值中的标记 "umlRole" 上将约束指定为分类器。

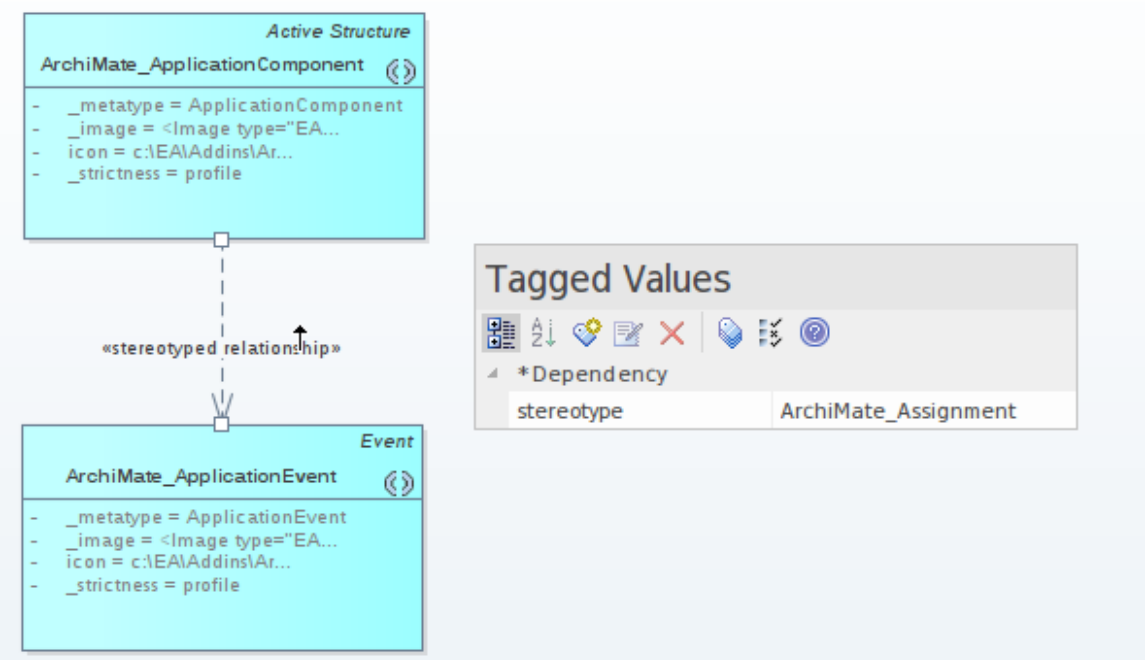
将此配置文件导入模型后， Enterprise Architect将在为 ActualProjectMilestone元素分配分类器时仅显示 ProjectMilestone 原型元素。

标签 "umlRole"的约束值包括：

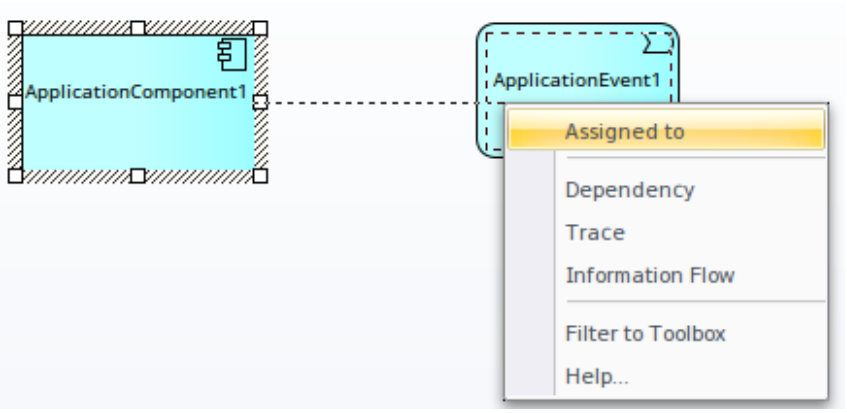
- 分类器——将源构造型元素的分类器限制为目标构造型元素
- type – 将源构造型元素的类型限制为目标构造型元素
- 行为 - 将源构造型元素的行为限制为目标构造型元素
- 传递 - 将源构造型元素的传递元素限制为目标构造型元素
- slot - 将源构造型元素的槽限制为目标构造型元素
- client/源元素- 将连接器的源限制为目标构造型
- provider/target/end[1].role/informationTarget - 将连接器的目标限制为目标构造型元素
- implementationConnector/realizingActivityEdge/realizingMessage - 限制可以实现信息流的关系
- typedElement/instanceSpecification – 当作为分类器从浏览器窗口中删除时，此约束将类型限制为目标构造型元素
- owner/class/activity/owningInstance – 将这个元素的容器限制为目标构造型元素；此约束用于创建嵌入式元素，并为快速链接器验证嵌套验证模型验证
- ownElement/ownedAttribute/ownedOperation/ownedParameter/ownedPort——限制源构造型元素可以拥有的元素/属性/操作/参数/端口；此约束通常用于验证嵌套模型验证
- annotatedElement/constrainedElement – 将注记连接器的目标限制为目标构造型元素

刻板关系	您可以在两个构造型或元类之间使用 "stereotyped relationship"连接器来指定这些元素的实例之间的有效构造型连接器。 指定关系时，如果所引用的关系是在定义规则的配置文件中定义的，则可以将属性型属性设置为仅该构造型的名称。但是，如果关系是在另一个概要文件中定义的，则必须使用与定义构造型的位置相对应的完全限定构造型名称。

Profile :



Quick Linker in Model :



在配置文件示例中，从 ApplicationComponent 到 ApplicationEvent 绘制了一个«stereotyped relationship»连接器，并且在连接器的标记值中将关系的构造型设置为 "Assignment"。

将此配置文件导入模型后， Enterprise Architect将在使用快速链接器绘制 ApplicationComponent 和 ApplicationEvent 之间的关系时显示 "分配"选项。

特殊元类

您可以将连接器的源指定为所有特殊形式的超类，并将目标指定为特殊元类，该元类在使用时指定与实际元类的关系。您可以使用这些术语之一作为具有构造型 «metaclass» 的类元素的元素名称。

物品	细节
源 .元	目标元素必须与源中定义的精确原型相匹配。

类型	
源.metatype.一般	目标元素可以匹配源中使用的确切构造型，以及任何具体的（isAbstract=false）广义构造型。
源.metatype.specific	目标元素可以匹配源中使用的确切原型，以及任何具体的 (isAbstract=false) 专门的原型。
源.metatype.both	目标元素可以匹配源中使用的确切刻板印象，以及任何具体的（isAbstract=false）概括或专门的刻板印象。
<profile_name>::*	将“profile_name>”替换为配置文件的名称；这将扩展为给定配置文件中所有具体原型的列表。
<none>	当你想防止源元素从它的超类型继承指定的连接器时使用这个元类名称。

元约束连接器上的约束

在创建特定领域的配置文件时， Enterprise Architect允许您指定相关构造型之间的约束。例如，您可以限制可以在 Stereotyped元素上设置为分类器的元素。

在 配置文件"工具箱的 无模型"页面上，两个构造型之间A元约束连接器用于指定两个构造型之间的约束。应在 Meta-Constraint 连接器上的标记 ű mlRole"中设置约束。

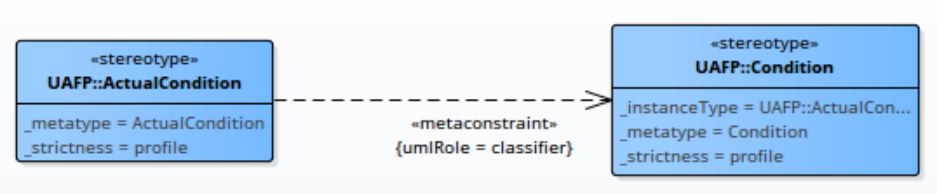
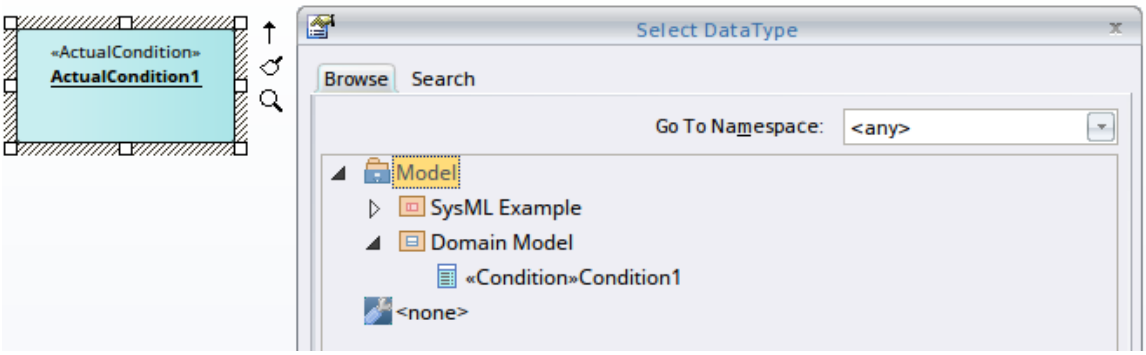
访问

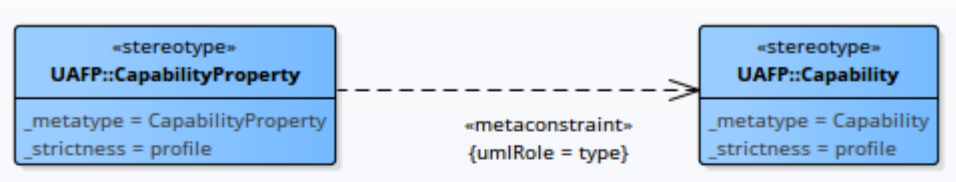
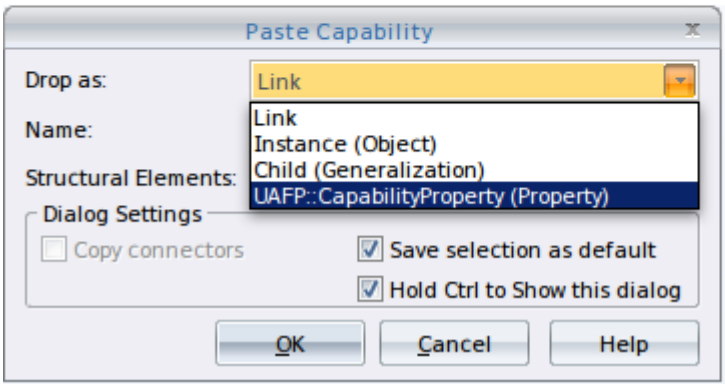
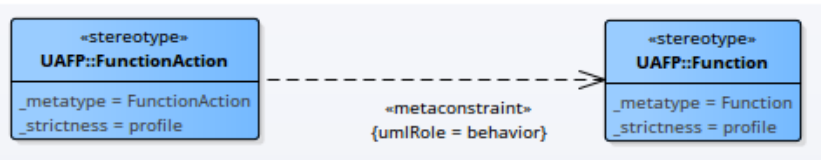
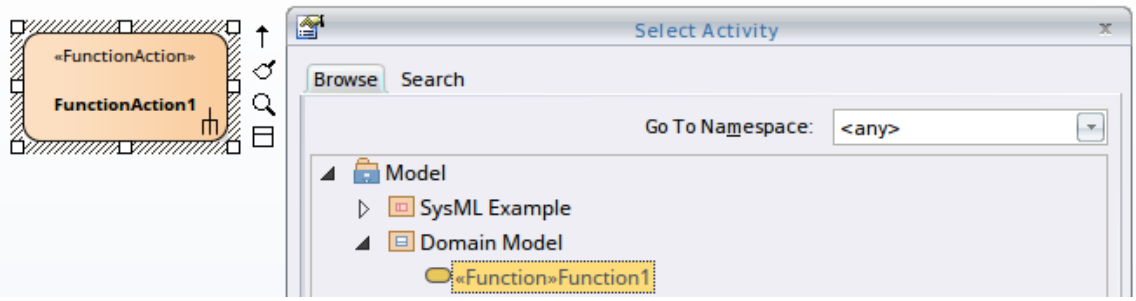
功能区	设计>图表>工具箱： >配置文件>元模型
键盘快捷键	Ctrl+Shift+3： >配置文件>元模型

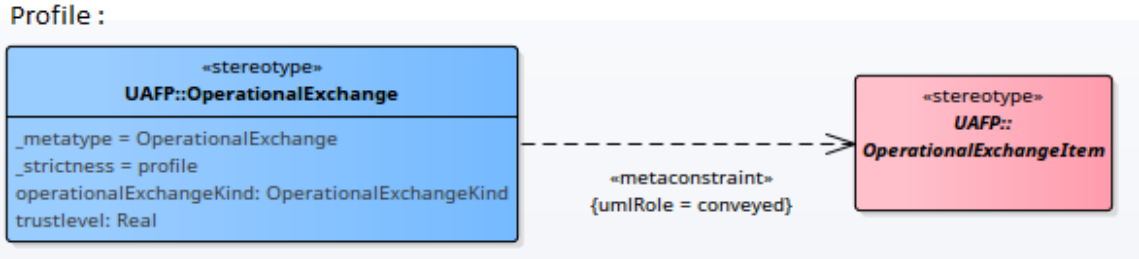
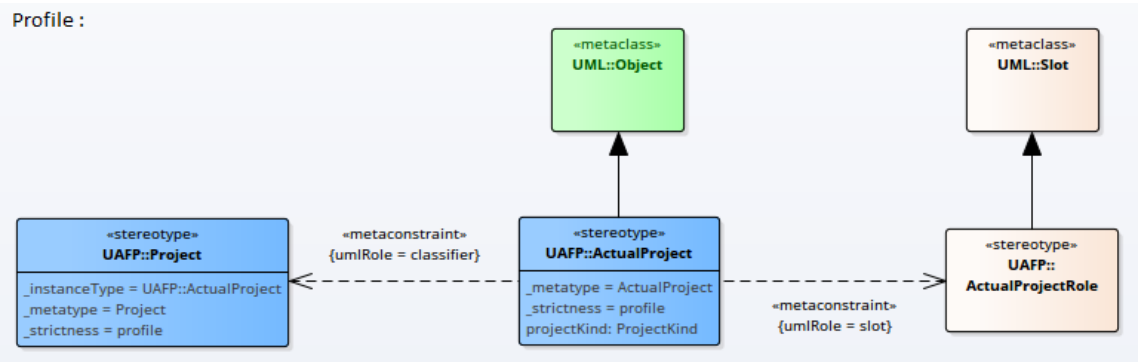
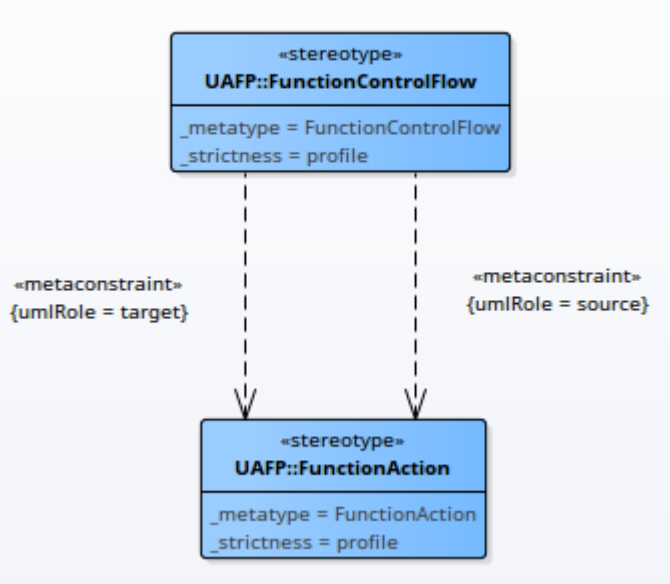
标签 ű mlRole"的约束值

（注意：下表显示了标签 ű mlRole"的所有可接受的约束值。这些值区分大 写，应按表中所示输入。）

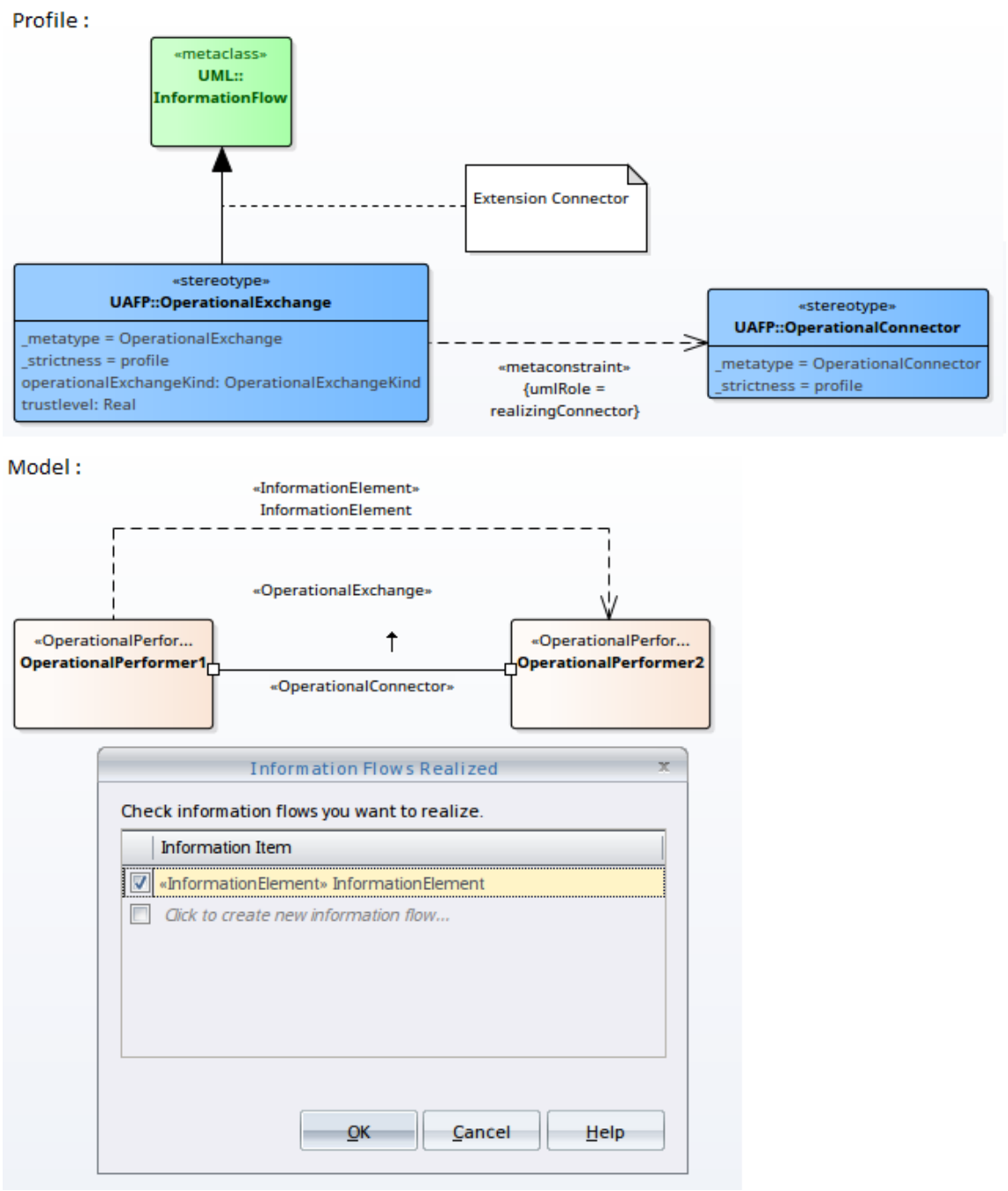
Meta-Constraint 连接器上标签 ű mlRole"的约束值为：

约束	描述
分类器	设置此约束以将源构造型元素的分类器限制为目标构造型元素。 Profile：  Model：  在配置文件示例中，Meta-Constraint 连接器从构造型 ActualCondition 绘制到条件，并且在连接器的标记值列表中的标记 ű mlRole"上将约束指定为 分类器"。这意味着只有 条件"原型元素可以设置为 ActualCondition 原型元素的分类器。 将此配置文件导入模型后， Enterprise Architect将在为条件原型元素设置 DataType 时仅在 选择数据类型"对话框中显示条件原型元素。

类型	设置此约束以在按住 Ctrl 键的同时将目标构造型元素从浏览器窗口拖放到图表中时指定其类型。 Profile :  Model :  在配置文件示例中，Meta-Constraint 连接器从原型 CapabilityProperty 绘制到 Capability，并且该约束在连接器属性窗口的 "Tags"选项卡中的标记 "umlRole"上指定为 "type"。 将此配置文件导入模型后，当按住 Ctrl 键并从浏览器窗口将 Capability 原型化元素拖放到图表中时， 粘贴<item>"对话框将显示 CapabilityProperty 作为选项之一'放下成'清单。
行为	设置此约束以将源构造型元素的行为限制为与目标构造型元素相同。 Profile :  Model :  在配置文件示例中，元约束连接器从构造型 FunctionAction 绘制到函数，并且约束在连接器的属性窗口的 标记"选项卡中的标记 "umlRole"上指定为 行为"。这意味着只有一个 函数"原型元素可以被设置为一个功能动作原型元素的分类器。 将此配置文件导入模型后， Enterprise Architect在设置 FunctionAction 原型元素的行为时，将在 选择活动"对话框中仅显示函数原型元素。
传达	设置此约束以限制可以在扩展信息项信息流连接器的构造型上传达的信息。

	Profile :  在配置文件示例中，元约束连接器从构造型 OperationalExchange 绘制到属性 OperationalExchangeItem，并且在连接器窗口的 标记"选项卡中的标记 属性"上将约束指定为 传递"。这意味着当绘制 OperationalExchange 连接器时，可以在连接器上传递的信息项被限制为 OperationalExchangeItem 原型元素。
投币口	设置此约束以将构造型元素的槽限制为目标构造型元素。  在配置文件示例中，从构造型 ActualProject 到 ActualProjectRole 绘制了一个 Meta-Constraint 连接器，并且在连接器的标记值中的标记 'umlRole' 上将约束指定为 'slot'。注记构造型 'ActualProject' 扩展了UML物件并可以对刻板印象 项目"进行分类。在模型中创建项目元素的实例规范时（通过在按住 Ctrl 键的同时将其从浏览器窗口拖放到图表中）： • 创建的实例规范将被定型为 ActualProject • 'Project' 原型元素中的任何属性都将被创建为实例规范中的原型属性'属性'
客户/源/结束[0].角色/信息来源	设置此模型验证约束以限制 Stereotyped 连接器的起始元素。 Profile :  在配置文件示例中，从构造型 "FunctionControlFlow"到"FunctionAction"绘制了一个元约束连接器，并且在连接器的标记值中的标记 "umlRole"上将约束指定为 源"。这意味着当绘制

	FunctionControlFlow 连接器时，源元素应该是 FunctionAction 原型元素。否则， Enterprise Architect将在执行模型验证时标记错误。
供应商/目标/结束[1].角色/信息目标	设置此模型验证约束以限制 Stereotyped 连接器的目标元素。
实现连接器/实现 Activity Edge/实现消息	设置此约束来限制可以实现信息流连接器的关系。

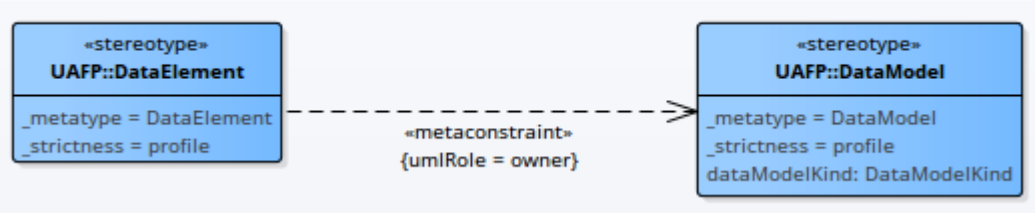


在配置文件示例中，Meta-Constraint 连接器从构造型 OperationalExchange（它扩展了UML信息流元类）绘制到 OperationalConnector，并且在连接器标记值中的标记 "umlRole" 上将约束指定为 "realizingConnector"。这意味着在绘制 OperationalConnector 连接器时，可以在此连接器上实现的信息流连接器可以是 OperationalExchange 原型连接器。

类型化元素/实例规范	当作为分类器从浏览器窗口中删除时，此约束将可用类型限制为目标构造型元素。
所有者/班级	设置此约束以将元素的容器/所有者限制为目标构造型元素。用于嵌入元素规则链接器并用于创建此快速嵌套模型验证。

/
活动
/
拥有实例

Profile :

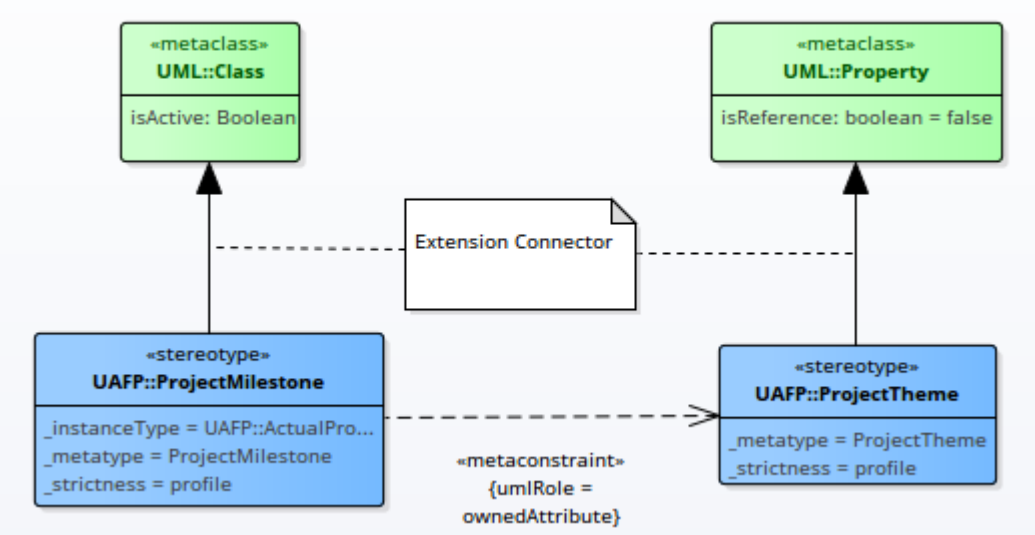


在配置文件示例中，Meta-Constraint 连接器从原型 DataElement 绘制到数据模型，并且在连接器的标记值中的标记 ŭmlRole"上将约束指定为 所有者"。这意味着 DataElement 原型元素可以是数据模型原型元素的子元素。换言之，只有数据模型可以包含/拥有模型中的数据元素。

拥有元素
/
拥有属性
/
拥有运营
/
拥有参数
/
拥有港口

设置这个约束来限制源构造型元素可以拥有的元素/属性/操作/参数/端口。此约束通常用于验证嵌套模型验证。

Profile :

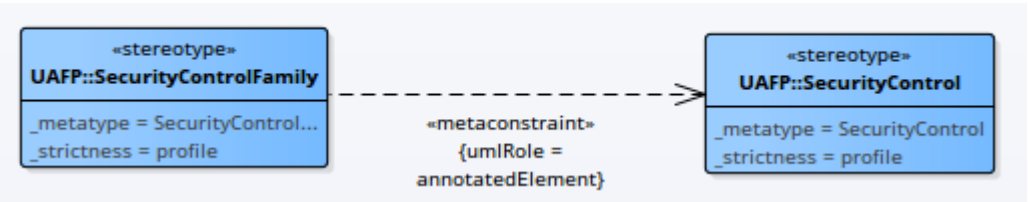


在配置文件示例中，Meta-Constraint 连接器从原型 ProjectMilestone 绘制到 ProjectTheme，并且在连接器的标记值中的标记 ŭmlRole"上将约束指定为 ŏwnedAttribute"。这意味着 ProjectMilestone 原型元素可以在模型中包含"ProjectTheme"原型属性。

注释元素
/
约束元素

设置此模型验证约束以限制 NoteLink 连接器的目标。

Profile :



在配置文件示例中，Meta-Constraint 连接器从构造型 SecurityControlFamily 绘制到 SecurityControl，并且在连接器的标记值中的标记 ŭmlRole"上将约束指定为 ăannotatedElement"。

当配置文件被导入模型时，来自SecurityControlFamily原型化元素的NoteLink连接器的目标应该是SecurityControl原型化元素。否则， Enterprise Architect将在执行模型验证时标记错误。

元模型约束和快速链接器

当您拖动快速链接器箭头以创建与另一个元素的关系时，将显示可用连接器类型的菜单以及 - 如果在图表上未选择目标元素- 将显示可用元素类型的菜单。本主题中的库表显示了连接器名称和元素类型的来源，当您提供或未提供元模型约束属性的值时。

规则过滤

元模型约束主要定义了哪些连接是有效的。快速链接器是根据这些有效关系构建的，然后以多种方式过滤，以便向用户呈现相关关系。

物品	细节
工具箱过滤	默认情况下，对于所有新图表，快速链接器提供的元素和关系被限制为与工具箱中可用的类型相匹配。 这可以由用户在图表上通过选择图表的完全视图或取消选中快速链接器菜单中的 对滤器到工具箱”选项来更改。
公共关系	当还需要创建新元素时，不会将使用关系属性定义的关系作为建议提供。 这些UML关系在与元关系一起使用时包括此行为： • 抽象 • 依赖 • 信息流 • 实现 • 用途

连接器标签

此表标识 Quick Linker 可以从中检索名称以显示在可用连接器类型的菜单中的点。

物品	细节
意义向前和意义向后	具有在 _MeaningForwards 和 _MeaningBackwards属性中定义的值的构造型将使用这些值来描述快速链接器菜单中的连接器。 注记：如果没有为构造型定义_MeaningBackwards，快速链接器将提供一个选项来创建反向或反向的关系。
元类型名称	当未定义 名称”属性时，具有在构造型属性中定义的值的构造型将使用这些值来描述快速链接器菜单中的连接器。
构造型名称	如果未定义 _MeaningForwards、_MeaningBackwards 或 _Metatype 值，则构造型名称将用作关系的菜单标签。
元类名称	当使用元关系连接器在您的原型之间包含UML关系时，您无法控制用于关系的标签。当这些关系在UML元素之间可用时，快速链接器将使用相同的标签。

元素标签

当您将快速链接器拖到空白处时，菜单会显示可用的目标元素类型。此表标识快速链接器从何处检索名称以显示在可用元素的菜单中。

物品	细节
元类型名称	具有在构造型属性中定义的值的构造型将使用这些值来描述快速链接器菜单中的元素。
构造型名称	如果没有定义 _MeaningForwards、_MeaningBackwards 或元类型值，则构造型的名称将用作元素的菜单标签。
元类名称	当使用 Metarelationship 连接器或 Stereotypedrelationship 连接器将您的原型链接到UML元素时，您无法控制用于元素的标签。快速链接器将使用与在UML下连接这些元素时使用的标签相同的标签。

快速链接器

当用户在图表上创建新元素和连接器时，他们可以使用快速链接器箭头来简化流程，该箭头显示可以从选定元素发出的常见连接器列表以及每个连接器可以连接的常见元素列表至。这些列表源自快速链接器定义，它是一个逗号分隔值 (CSV) 格式文件。

作为配置文件的一部分，您可以使用自己的定义添加或替换内置快速链接器定义。这些可以来自：

- A Quick Link Definition Format CSV 通过将 CSV 文本添加到配置文件中的配置文件中，与配置文件集成（在配置配置文件图上添加文档工件元素首选方法） - 请参阅*Quick Linker Definition Format*帮助主题
- A定义元模型图视图，包括一组元模型约束，这些约束定义哪些类型的元素器通过什么类型的连接器连接（第二个首选方法） - 请参阅引入元模型视图和定义元模型约束帮助主题）
- A库表关系配置文件，您还可以通过将元素文本添加到文档工件关系图与配置文件关系图的元数据图来集成配置文件（最好只用于实现不一定对应于配置文件的复杂规则的复杂规则） - 请参阅关系库表帮助主题

注记

- 快速链接器定义背后的理念不是提供有效或合法连接的完成列表，而是提供给定上下文的最常见连接的简短且方便的列表

快速链接器定义格式

为了替换或更改用户从图表上的一个配置文件元素中拖动快速链接器箭头时显示的快速链接器菜单，您可以创建或编辑相应的快速链接器定义。这是一个逗号分隔值 (CSV) 文本文件，由记录（行）组成，每条记录由表中定义的 23 个逗号分隔字段组成。

其中一些字段定义菜单命令，而另一些则充当过滤器，如果不满足过滤条件，则忽略该条目。

快速链接A定义可以包括注释： Enterprise Architect忽略所有前两个字符 // 的行。字段值中的引号 (" ") 不是必需的。

快速链接器定义的每条记录代表快速链接器菜单上的单个条目组合；即，对于选定的源元素，特定的连接器类型和特定的目标元素类型。从满足过滤器的所有行中填充A菜单；也就是说，第一个菜单列出了所有定义的对源元素类型合法有效的连接器，第二个菜单列出了所有对源元素和连接器类型组合合法有效的目标元素。

快速链接器定义字段

柱子	标题（作为指导意见输入）
A	源元素类型 描述：标识配置文件中的有效源元素。 如果连接器被拖离这种类型的元素，则评估该行。否则，该行将被忽略。 如果源是另一个连接器，则在连接器类型前加上单词'link：'；例如， 链接：控制流"。
B	源构造型过滤器 描述：标识源元素基本类型的构造型（例如，事件源元素可以是普通事件、开始事件、中间事件或结束事件型元素）。构造型可以是完全限定的构造型或当前配置文件中构造型的名称。 如果设置，并且如果连接器被拖离此构造型的元素，则评估该行。否则，该行将被忽略。
C	目标元素类型 描述：标识配置文件中的有效目标元素。要表明目标元素可以是抽象UML元类的任何特化，请将前缀 @"添加到元类名称；例如，"@Classifier"、"@NamedElement"。 如果设置，并且如果连接器被拖到这种类型的元素上，则评估该行。 如果为空白，并且如果将连接器拖到图表上的空白区域，则评估该行。 否则，该行将被忽略。 如果目标是另一个连接器，请在连接器类型前加上 链接："一词；例如， 链接：控制流"。
D	目标构造型过滤器 描述：标识目标元素基本类型的构造型。 如果已设置，并且目标元素类型也已设置，并且连接器被拖到此构造型的元素上，则将评估该行。否则，将忽略该行。 如果没有设置，并且如果连接器被拖到目标元素类型的非构造型元素上，则评估该行。否则，忽略该行。
E	图表过滤器 描述：包含图表类型的包含列表或独占列表，这限制了可以在其上创建指定

	连接器的图表。 • 每个图表名称都以分号结尾；例如： 协作；物件；习惯； • 可以使用完全限定的图表类型 (DiagramProfile::DiagramType) 引用来自 MDG 技术的自定义图表类型；例如： BPMN0::业务流程;BPMN2::Choreography;BPMN2::协作;协作; • 作为图表配置文件中所有图表类型的简写，您可以使用 *"通配符，它必须以图表配置文件 ID 开头；例如： BPMN2.0::*; • 每个排除的图表名称前面都有一个感叹号；例如： ！序列; 此列覆盖快速链接器的 过滤器到工具箱"设置，该设置在图表上默认启用。要强制连接器在所有图表上可见，您可以排除不存在的图表类型。例如： !TBFilter 注记：执行图表过滤器的首选机制是现在工具箱过滤器。这会根据当前图表自动显示相关连接器类型，包括图表类型，因为它们在未来由其他技术定义。
F	新元素类型 描述：定义将连接器拖入开放空间时要创建的元素类型，前提是 创建元素"字段设置为True 。 此值不能是连接器类型。
G	新元素构造型 描述：定义将连接器拖入开放空间时要创建的元素原型的类型，前提是 创建元素"字段设置为True 。这可以是完全限定的构造型，也可以是当前配置文件中构造型的名称。
H	新链接类型 描述：定义要创建的连接器类型，如果 'Create Link' 也设置为True 。
I	新的链接构造型 描述：定义创建的连接器的构造型，如果 'Create Link' 也设置为True 。将快速链接器记录添加到内置类型时需要此字段。构造型可以是完全限定的构造型，或者是当前配置文件中构造型的名称。
J	新链接方向 描述：定义连接器方向，可以是： • 定向（总是创建一个从源到目标的关联） • from（总是从目标到源创建一个关联） • 无向（总是创建一个未指定方向的关联） • 双向（始终创建双向关联），或 • to（根据 关联方向"字段的值创建有向或无向关联） 并非所有这些都适用于所有连接器类型；例如，你不能创建一个概括的你。
K	新链接标题 描述：如果正在创建新连接器但不是新元素，则定义要在 快速链接器"菜单中显示的文本。
L	新链接和元素标题

	描述：定义在创建新连接器和新元素时要在 快速链接器"菜单中显示的文本。
M	创建链接 描述：如果设置为True ，则会创建一个新的连接器；留空以停止创建连接器。
N	创建元素 描述：如果设置为True并且连接器被拖到图表上的空白区域，则会导致创建新元素。 留空以停止创建元素。这会覆盖 目标元素类型"和 目标构造型过滤器"的值。
O	禁止自连接器 描述：如果自连接器对这种连接器无效，则设置为True ；否则将此字段留空。
磷	ST过滤器+独有 没有从 Metatype 继承 描述：设置为True以指示具有构造型 源构造型过滤器"的类型 源元素类型"的元素不显示等效未构造型元素的快速链接器定义。 如果 源构造型过滤器"字段（ B列）为空，则忽略该字段。
Q	菜单组 描述：指示在其中创建菜单项的子菜单的名称。 此列仅在创建新元素时适用；也就是说，用户正在从一个元素拖动到图表上的空白区域，或者在目标元素上拖动以创建一个新的嵌入元素。
R	复杂程度 描述：包含标识复杂功能的数字位掩码值。 • **0** = 没有复杂的功能 • **4** = 强制空白源刻板印象；除非源元素没有刻板印象，否则将跳过该行 • **8** = 强制空白目标定型；除非目标元素没有刻板印象，否则将跳过该行 • **16** = 将 源构造型过滤器"列（ B列）中的值视为源名称过滤器 • **32** = 将 目标构造型过滤器"列（ D列）中的值视为目标名称过滤器，并使用 新元素构造型"列（G 列）中的值作为新的名称创建元素 • **64** = 将 源构造型过滤器"列（ B列）中的值视为源分类器名称过滤器 • **128** = 将 目标构造型过滤器"列（ D列）中的值视为目标分类器名称过滤器，并使用 新元素构造型"列（G 列）中的值作为新创建元素的分类器，如果当前模型中不存在该名称的元素，则创建一个额外的新元素 这些值可以加在一起以组合功能；例如， **192**结合了**64**和**128**的功能。
S	目标必须是父母 描述：如果菜单项只应在从子元素拖到其父元素时出现，则设置为True ；例如，从一个端口到它的包含类。否则将此字段留空。
T	嵌入元素 描述：设置为True以将正在创建的元素嵌入到目标元素中；否则将此字段留空。

U	在分隔符 LEAF 之前 描述：设置为True以在此条目下方的 快速链接器"菜单中添加菜单项分隔符；否则将此字段留空。
V	先于分隔符 GROUP 描述：设置为True以向 快速链接器"子菜单添加菜单项组分隔符；否则将此字段留空。
W	虚拟柱 描述：根据您使用的电子表格应用程序，此列可能需要每个单元中的值，以强制 CSV 导出以正确处理尾随空白值。

关系库表

指定元素之间的快速链接器链接的另一种方法是使用关系表，您最初使用 Microsoft ™ Excel 等电子表格应用程序将其创建为 CSV 文件。创建填充文件后，您将其导入到个人资料中的文档工件元素中。

此方法导致的行为等同于在您的配置文件中描述的构造型之间使用构造型关系连接器。

在大多数情况下，我们建议使用在快速链接器定义格式中定义链接的原始方法，或在元模型视图中建模关系，而不是使用这种关系库表方法。但是，为了实现不一定对应于定义的元模型的复杂关系规则，支持此方法。

格式

关系表的格式基于 ArchiMate 规范中使用的格式，增加了两行将名称映射到构造型。根据以下格式指南设置表：

部分	描述
连接器别名	定义中的第一行提供了映射到完全限定的连接器原型的单字母连接器标识符列表。例如： **a** =ArchiMate3::ArchiMate_Access; **c** =ArchiMate3::ArchiMate_Composition; 也就是说，在文件主体中， **a**表示 ArchiMate 3 ArchiMate访问连接器， **c**表示 ArchiMate 3 ArchiMate组合连接器。
元素别名	定义中的第二行提供了映射到完全限定元素原型的标识符列表。例如： 评估=ArchiMate3::ArchiMate_Assessment；约束 =ArchiMate3::ArchiMate_Constraint； 也就是说，在文件主体中，“Assessment”指的是 ArchiMate 3 ArchiMate 评估元素。
源元素	定义中的第三行列出了针对第二行中的标识符定义的所有可能的源元素。这些是表中的列标题。例如： ,评估,约束,
目标元素	第一列，从第四行开始，列出了根据第二行中的标识符定义的所有可能的目标元素。这些是表中的行标题。
链接定义	行和列交叉处的单元标识在源和目标元素之间有效的连接器，使用在第1行定义的单字母标识符。例如： **S n** o，表示A列类型的元素可以通过Specialization、 **Composition** I Aggregation、 **Influence**和Ass **o** ciation连接器连接到该行类型的元素

将库关系库表添加到配置文件

节	讨论
1	打开包含配置文件的构造型元素的配置文件配置文件子图。
2	选择 文档图表”工具箱（点击  工具箱项目”并指定 工具箱项目”，并在 文档工具”对话框中

	拖拽一个 文档工件元素"到图表上）将此元素命名为 关系表"。
3	双击元素打开链接文档编辑器；取消模板名称的提示。
4	在文本编辑器中打开您的记事本文件，然后将内容复制到文档工件中，并以链接文档的形式元素到文档中。 保存并关闭文档。
5	继续处理配置文件直到它完成，然后保存它。 QuickLink 定义与配置文件一起保存，并在将配置文件（在其MDG 技术内）导入另一个模型时进行处理和应用。 A技术可以包含多个Profiles ，因此具有多个快速链接定义，每个配置文件一个。

快速链接器示例

如果要创建快速链接器定义，最简单的方法是在电子表格中进行设置，每个菜单项定义跨行构建，如下例所示：

	A	B	C	D	E	F	G	H	I	J	K
1	//Source Element Type	Source ST filter	Target Element Type	Target ST Filter	Diagram Filter	New Element Type	New Element ST	New Link Type	New Link ST	New Link Direction	New Link Caption
2	Class	quick				Component		Dependency		to	
3	Class	quick				Component		Dependency		from	
4	Class	quick	Component					Dependency		to	Dependency to
5	Class	quick	Component					Dependency		from	Dependency from
6	Class	quick	Port					Dependency		to	Dependency to
7	Class	quick	Port					Dependency		from	Dependency from
8	Class	quick	Component			Port		Dependency		to	
9	Class	quick	Component			Port		Dependency		from	
10											

	L	M	N	O	P	Q	R	S	T	U	V	W
1	New Link & Element Caption	Create Link	Create Element	Disallow Self connector	Exclusive to ST Filter & No inherit from metatype	Menu Group	Complexity Level	Target Must Be Parent	Embed element	Preceeds Separator LEAF	Preceeds Separator GROUP	DUMMY COLUMN
2	Dependency to	TRUE	TRUE	TRUE	TRUE	Component	0					
3	Dependency from	TRUE	TRUE	TRUE	TRUE	Component	0			TRUE		
4		TRUE		TRUE	TRUE		0					
5		TRUE		TRUE	TRUE		0			TRUE		
6		TRUE		TRUE	TRUE		0					
7		TRUE		TRUE	TRUE		0			TRUE		
8	Dependency to	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE			
9	Dependency from	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE	TRUE		
10												

该示例的第一行是标识列标题的注释行。随后的几行定义了具有构造型 «quick» 的类元素的连接器/目标元素选项。当连接器被拖离这种类型的元素时，您希望用户创建一个到或来自部件元素的依赖关系。当他们将连接器拖到现有的端口或部件元素上时，您希望该组件具有依赖关系，或者在部件的情况部件，您希望用户能够创建嵌入的端口元素。

这些要求在快速链接器定义文件的八条记录中定义：

1. 对新部件的依赖
2. 来自新元件的部件
3. 对现有元件的部件
4. 来自现有元件的部件
5. 对现有端口的依赖
6. 来自现有端口的依赖
7. 对现有组件的部件，创建新的端口
8. 依赖现有部件，创建新端口

记录保存到此 CSV 文件：

//源元素类型,源ST过滤器,目标元素类型,目标ST过滤器,图表过滤器,新元素类型,新元素ST,新链接,新链接类型,新链接方向,新链接标题,新链接 &元素标题，创建链接，创建元素，禁止自我连接器，ST过滤器+不继承元类型，菜单组，复杂性级别，目标必须是父级，嵌入元素，前置分隔符LEAF，前置分隔符组，DUMMY COLUMN

类,quick部件,,Dependency,,to,,Dependency to,TRUE,TRUE,TRUE,TRUE,部件,0,,,,,

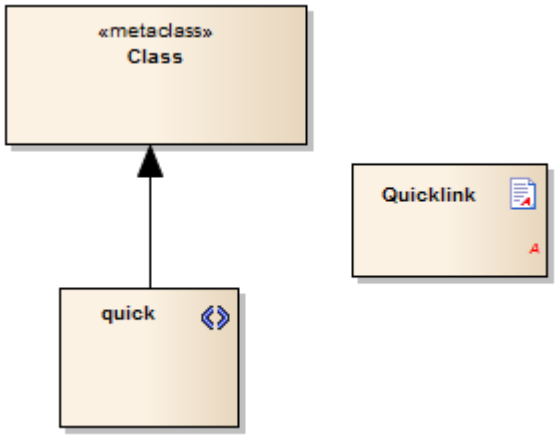
类,quick部件,,Dependency,,from,,Dependency from,TRUE,TRUE,TRUE,TRUE,部件,0,,,TRUE,,

类,quick,部件Dependency,,to,Dependency to,,TRUE,,TRUE,TRUE,,0,,,,,

类,quick,部件Dependency,,from,Dependency from,,TRUE,,TRUE,TRUE,,0,,,TRUE,,

类,quick,端口,,,,,Dependency,,to,Dependency to,,TRUE,,TRUE,TRUE,,0,,,,,,
类,quick,端口,,,,,Dependency,from,Dependency from,,TRUE,,TRUE,TRUE,,0,,,TRUE,,
类,部件,组件,,端口,,依赖,,to,,依赖到,TRUE,TRUE,TRUE,端口,0,TRUE,,,
类,部件,组件,,端口,,依赖,,来自,,依赖来自,TRUE,TRUE,TRUE,端口,0,TRUE,TRUE,,

如果您想测试效果，您可以将此配置文件剪切并粘贴到 QuickLink文档工件中的元素中。



隐藏默认快速链接器设置

如果您为元素创建自己的快速链接器定义，您可能希望在给定的源和目标元素之间隐藏默认的UML快速链接器选项。如何执行此操作取决于您是使用元模型定义方法还是电子表格定义方法来定义快速链接器链接。

元模型法

在每个源原型元素的 <<metaclass>>元素中，添加属性 _HideUmlLinks 设置为 "True"，这样以该原型作为源元素的快速链接将不包括从基本UML元类继承的快速链接。

电子表格法

首先，您可以通过在定义 CSV 文件中根据需要在每一行上将 "Exclusive to stereotype"过滤器标志（列 P）设置为True来隐藏默认的UML快速链接器选项。

或者，您可能希望隐藏默认的快速链接器选项，而无需替换自定义选项。例如，通常如果您没有为一个«quick»类定义到另一个 «quick»类的任何快速链接，快速链接器箭头会显示一个类到另一个类的默认快速链接。要覆盖此行为，请创建一个快速链接器定义，您可以在其中设置：

- 源元素类型（ A栏）
- 源构造型过滤器（ B栏）
- 目标元素类型（C栏）
- 目标构造型过滤器（ D列）
- 新的链接类型（H 列）到<none>
- 刻板印象独有 + 不从 Metatype（P 列）继承为 TRUE

尝试将此行添加到快速链接器示例中：

类,quick,接口,,,,, <none> ,,,,,,,,TRUE,,0,,,,,

通过定义中的这一行，当快速链接从«快速»类拖动到接口元素时，默认的类到接口快速链接被隐藏。

注记 "Exclusive to stereotype"过滤器隐藏了所有没有此过滤器集的上下文敏感关系，这将在定义了源元素原型的地方生效。

快速链接器物件名称

创建快速链接器定义文件时，您使用一系列基本元素和连接器类型来标识：

- 源元素类型（ A栏）
- 目标元素类型（C栏）
- 新元素类型（F 列）和
- 新链接类型（H 列）

这些然后由您在定义中指定的构造型限定。您可以使用的基本元素和连接器类型在此处标识。

物件类型名称

物件組	物件类型
元素类型	行动 行动销 活动 活动参数 活动分区 参与者 工件 边界 中央缓冲区节点 更改 选择状态 类 协作 部件 数据类型 决策 深度历史状态 部署规范 设备 图门 实体 入口点 入口状态 执行环境 出口点 退出状态 扩展节点 扩展区域 特征 最终活动

	图形界面元素 历史状态 信息项 初始活动 初始状态 交互发生 接口 问题 可中断活动区域 连接状态 合并节点 消息端点 n元关联 节点 物件 物件节点 包 部件 端口 原始类型 提供的接口 接收 必需接口 需求 屏幕 发送 序列 信号 状态 州生命线 状态机 同步_H 同步_V 同步状态 UMLD图 用例 价值生命线
连接器类型	抽象 聚合 关联 关联类 通讯路径 组合 连接器链接

	控制流 代表链接 依赖 部署 扩展 概括 信息流 接口链接 显现 嵌套 对象流 包导入 包合并 实现 重新定义 序列 状态流 替代 模板绑定 UC扩展 UC包括 用途 用例

将快速链接器定义添加到配置文件

将配置文件快速链接器定义设置为 CSV 文件后，您可以将它们合并到配置文件中。要执行此操作，请将文档工件的文件内容复制到与配置文件的构造型元素相同的图表中。

在配置文件中添加定义

节	讨论
1	打开包含配置文件的构造型元素的配置文件配置文件子图。
2	选择 文档图表”工具箱（点击  工具箱项目”并指定 工具箱项目”，并在 文档工具”对话框中拖拽一个 文档工件元素”到图表上）将此元素命名为 QuickLink”。
3	双击元素打开链接文档编辑器；取消模板名称的提示。
4	在文本编辑器中打开，然后将内容复制到您的 CSV文档工件中，然后将其作为元素本链接文档。 保存并关闭文档。
5	继续处理配置文件直到它完成，然后保存它。 QuickLink 定义与配置文件一起保存，并在将配置文件（在其MDG 技术内）导入另一个模型时进行处理和应用。 A技术可以包含多个Profiles ，因此具有多个快速链接定义，每个配置文件一个。

导出一个配置文件

创建配置文件、定义构造型元素并添加所需的任何标记值、形状脚本、约束和快速链接器定义后，您可以将配置文件保存（导出）到磁盘。然后可以将配置文件与MDG 技术集成并部署到其他模型以供使用。

保存配置文件

节	描述
1	如果你的配置文件是： • 单个配置文件分布在同A配置文件包中的多个图表上，在浏览器窗口中找到配置文件包，然后选择 特定>技术>发布技术>将包发布为UML配置文件”功能区选项 • 同一配置文件包中的多个Profiles文件之一，单击配置文件图背景中的任意位置，然后选择功能区选项 设计>图表>管理>另存为配置文件”或 特定>技术>发布技术>发布”图表为UML配置文件' • 配置文件包中A单个图表，单击配置文件图表背景中的任意位置，然后选择功能区选项 设计>图表>管理>另存为配置文件”或 特定技术>发布技术>图表为” UML配置文件' 将显示 保存UML配置文件”对话框。
2	单击  按钮，然后选择 XML配置文件的目标目录路径。 如有必要，编辑配置文件名，但不要删除 .xml 扩展名。
3	在 配置文件类型”字段中，使用默认值 EA (UML)2.X”（或者，如有必要，单击下拉箭头并选择此值）。 注记：如果此字段显示为灰色，则表示您要从中导出配置文件的包没有 <<profile>> 构造型。您必须提供该构造型的包或将配置文件图和/或元素转移到具有该构造型的另一个包。
4	为配置文件中定义的所有构造型设置所需的导出选项： • 元素- 选中复选框以导出元素属性 • 颜色和外观-（如果从图表中保存配置文件则启用；如果从浏览器窗口中的包中保存则禁用）选中该复选框以导出背景颜色、边框颜色、文本颜色和边框粗细属性 • 图像选中复选框以导出元文件图像 • 代码模板- 选中复选框以导出代码模板（如果存在）
5	单击 保存”按钮将配置文件保存到磁盘。

避免配置文件名称和 ID 冲突

每个配置文件都应该有一个唯一的名称和 ID。配置文件名称是在保存配置文件时指定的，而 ID 是从用于保存配置文件的图表或包的GUID派生的。为避免名称和 ID 冲突：

- 创建多个Profiles时，为每个配置文件使用新图表或包
- 保存Profiles时，输入唯一的配置文件名

在启动Enterprise Architect或启用MDG 技术时，如果检测到重复的配置文件名称或重复的配置文件ID，则会在系统输出窗口中显示警告。

注记

- 为了快速测试配置文件，您可以将 XML 文件单独导入浏览器窗口的 资源”选项卡中；对于最终部署，将配置文件合并到MDG 技术中

保存配置文件选项

保存配置文件时，可以从其父包或配置文件图中保存它，具体取决于配置文件是否为：

- 单个配置文件分布在同A配置文件包中的多个图表上，这通常是构造型配置文件的情况
- 同一个配置文件包内的多个Profiles之一；例如，创建多个工具箱配置文件时
- 配置文件包内A单图

访问

功能区	设计>图表>管理>另存为配置文件 特定>技术> 发布技术> 将图表发布为UML配置文件 特定>技术> 发布技术> 将包发布为UML配置文件

选项比较

从图表保存	保存来自包
配置文件采用图表名称。	配置文件取包名。 注记：包和图表名称不一定相同，但如果将它们设置为相同或非常相似，可以避免很多混乱。 例如：带有图表 GL1、GL2、GL3 的包GL。
配置文件取图表的注记。	配置文件取包的注记。 注记：图表注记在配置文件定义中可能很重要，例如工具箱Profiles 。 请参阅Create Toolbox Profiles
您可以从图表object中获取默认大小和外观（包括替代图像）。	您不能从图表object中获取默认大小和外观。 您可以使用 _sizeX、_sizeY 和 _image属性，但没有等效的默认颜色。 注记：
此选项可以更快。	此选项可能会慢得多。 注记：之所以出现差异，是因为图表对象保存在内存中，而浏览器窗口元素没有。 仅当配置文件很大并且您使用慢速网络连接到远程存储库时，这才可能成为问题。

浏览器-资源中的UML Profiles

浏览器窗口的 资源”选项卡包含一个树结构，其中包含包括UML Profiles在内的一系列项目的条目。 UML Profiles节点最初不包含条目；为了能够使用 资源”选项卡中的Profiles ，您必须将它们从外部 XML 文件导入到项目中。

配置文件中的项表示构造型。这些可以通过多种方式应用于UML元素；例如，适用于的刻板印象：

- 类和接口等元素可以直接从 资源”选项卡拖到当前图表中，自动创建原型元素；或者，可以将它们拖到现有元素上，自动将它们应用到元素
- 属性可以拖放到宿主元素上（例如类）；一个刻板的属性会自动添加到元素的特征列表中
- 操作与应用于属性的操作相同；拖放到主机元素上以添加原型操作
- 通过在浏览器关联的 资源”选项卡中选择它们，然后单击图表中的开始元素并拖动到结尾元素（与添加普通连接器）；添加了一个定型连接器
- 可以通过将连接器末端元素拖动到图表中的关联末端来添加关联末端

浏览器-导入UML Profiles Into资源

Profiles以 XML 文件的形式存在，可以导入到任何项目中，为特定领域提供量身定制的建模结构。 Sparx Systems网站上A许多配置文件XML 文件可供您使用，用于导入您的模型。您还可以导入自己创建的配置文件XML 文件。如果配置文件包含对任何元文件的引用，请将这些元文件复制到与配置文件XML 文件相同的目录中。

访问

功能区	开始> 所有窗口>设计> 浏览> 浏览>资源> 右击' UML Profiles ' 文件夹>导入配置文件
键盘快捷键	Alt+6 |右击“UML Profiles”文件夹|导入配置文件

导入一个配置文件

字段/按钮	行动
文件名	单击  按钮并找到要导入的 XML配置文件文件。
元素尺寸	选中复选框以导入配置文件中定义的所有构造型的元素大小属性。
色彩与外观	选中复选框以导入配置文件中定义的所有构造型的颜色（背景、边框和字体）和外观（边框粗细）属性。
替代图像	选中复选框以导入配置文件中定义的所有构造型的元文件图像。
代码模板	选中复选框为配置文件中定义的所有构造型导入代码模板（如果存在）。
覆盖现有模板	对于配置文件中定义的所有构造型，选中复选框以覆盖当前项目中定义的任何现有代码模板。
导入	单击此按钮将配置文件添加到UML Profiles文件夹。 如果配置文件已经存在，则会显示一个提示，让您覆盖现有版本并导入新版本。 导入完成后，配置文件就可以使用了。

MDG 技术-创造

如果您想访问和使用与Enterprise Architect中的特定技术相关的资源，您可以使用模型驱动生成 (MDG)技术来实现。管理员或个人用户有多种选择可以将现有的MDG 技术与Enterprise Architect一起使用。技术开发人员还可以开发新的MDG 技术，并根据需要将它们部署到项目团队，为您的工作领域或环境提供量身定制的解决方案。

使用配置文件Helpers

MDG技术和Profiles是使用Enterprise Architect中的图表和元素开发的。这些图表和元素使用特定的属性和属性来确定生成的MDG 技术的内容和行为。配置文件帮助者协助创建新的MDG技术，这些配置文件类型：

- 构造型Profiles
- 工具箱Profiles及
- 图表Profiles

配置文件助手由两个部分组成：

- 模型中的MDG 技术构建器模板，为创建新的MDG 技术提供了起点
- 配置文件”工具箱中的配置文件帮助项，提供了简化构造型、工具箱和图表Profiles创建的对话框

访问

选择一个要添加MDG 技术生成器模板的包，然后使用这里概述的方法之一显示模型生成器对话框。

功能区	开始> 个人>模型构建器 设计>包>模型生成器
上下文菜单	右键点击包|模型（模式库）
键盘快捷键	Ctrl+Shift+M
其它	浏览器窗口标题栏菜单|模型（图案库）

创造新的MDG 技术

节	描述
1	在 模型生成器”对话框中，单击<视角名称>按钮并选择 管理| MDG 技术生成器”。 在” MDG 技术构建器”组中选择 基本模板”模式。 单击创建模型按钮。提示显示技术名称的A 。
2	输入新MDG 技术的名称，然后单击确定按钮。 这将创建一个基本的包模板和示例元素，可将其用作创建MDG 技术的起点。该模板包括三个包，每个包都具有与技术相同的名称，但具有与配置文件类型相对应的不同构造型配置文件他们定义： • <<profile>> -用于定义包含构造型用户的配置文件包将应用于元素 • <<图表配置文件>> -描述用户将创建的图表类型的配置文件包 • <<工具箱配置文件>> -描述要在工具箱中显示的元素的配置文件包
3	在每个包中，打开图表并参考提供的示例元素，将其他项目添加到配置文件中。 配置文件工具箱包含一个配置文件帮助图标页面，当将其拖到图表上时，可以帮助您创建和填充各种Profiles的元素。

4	将这些Profiles中的每一个保存到磁盘。
5	将保存的Profiles合并到MDG 技术中。

使用配置文件Helpers 创建构造型Profiles文件

在创建技术以提供特定于领域的工具集时，典型的起点是定义您要提供的每个元素、连接器、特征和结构组件。这些由配置文件定义。

配置文件中定义的所有构造型要么是Enterprise Architect定义的核心UML对象（元类）的扩展，要么是其他现有Profiles和技术定义的非 UML 对象（构造型）的扩展。

完成配置文件的开发后，将其保存到外部 XML 文件中，然后合并到MDG 技术中以进行最终部署。

配置文件中定义的每个构造型都会修改它扩展的元类或构造型的行为。这些修改可能包括：

- 提供附加属性标记值
- 约束来定义适用于每个构造型的条件和规则
- A形状脚本来自定义新object的整体外观
- 更改object A默认外观，例如背景、边框和字体颜色
- 快速链接器定义，提供每个构造型中最常见的连接类型列表
- 定义新object的特定外观和行为的特殊属性，包括初始元素大小和浏览器窗口图标

创建UML配置文件

节	描述
1	在浏览器窗口中，找到具有 <<profile>> 构造型的包并打开其子图。 如果您没有现有的<<profile>>包，请使用模型生成器中的 管理 | MDG 技术生成器”蓝图来创建新技术，然后从新创建的<<profile>>包中打开图表。
2	（可选）如果您希望构造型元素包含引用预定义标签类型的标记值，则可以在配置文件图表上的数据类型元素中定义这些标签类型。在数据类型元素的注记中包含标记值类型定义，例如“Type=Memo;”或 类型=RefGUID；” 如果您希望构造型元素包含具有多个预定义值的下拉列表的标记值，则每组值必须由配置文件图表上的枚举元素定义。 如果您希望构造型元素包含结构化标记值以提供一组复合信息，则每个结构必须由配置文件图表上的类元素定义。 在为构造型定义这些标记值类型之前，枚举和类元素必须存在；您可以在此时创建元素，也可以稍后将这些标记值添加到您的构造型中。
3	通过从图表工具箱中拖动 添加构造型配置文件帮助器”来添加新的构造型工具箱由 添加构造型配置文件帮助器”打开的对话框将允许您为您的构造型指定各种一般属性、标记值和形状脚本脚本。
4	（可选）为构造型定义约束。
5	（可选）设置构造型的默认外观。
6	对要创建的每个新构造型元素重复步骤 3 到 5。
7	（可选）将快速链接器定义添加到配置文件。
8	将包保存为配置文件。 保存配置文件时，使用的名称应与配置文件包的名称一致；这对于工具箱配置文件中的引用正确执行函数是必要的

9	将配置文件合并到MDG 技术中。

注记

- A配置文件包不能包含其他包；不要在配置文件中添加任何其他包

使用配置文件Helpers 添加构造型和元类

您可以在配置文件中定义构造型以扩展：

- 核心UML对象（在Enterprise Architect中预定义的元类），或
- 由其他Profiles和技术定义的对象（构造型）（例如在 ArchiMate 或 SysML 中定义的对象）

构造型可以通过多种方式扩展元类：

- 一个构造型扩展一个元类，用于一种object类型的特定定义
- 一个构造型扩展多个元类，其中定义适用于多个object类型 - 例如以相同的方式修改一个类和一个物件
- 多个构造型扩展一个元类，您在其中创建相同基础object类型的多个变体；例如，定义关联连接器的类型，代表父母、兄弟、祖父母、叔叔/阿姨和表亲关系

将元类和构造型添加到配置文件

节	描述
1	如果您要扩展由现有配置文件或技术定义的非 UML 类型，请遵循创建构造型扩展非 *UML* 对象帮助帮助中描述的过程。
2	在浏览器窗口中，找到带有 <<profile>>构造型的包并打开它的子图。
3	将图形工具箱的 配置文件助手”页面中的 添加构造型”工具箱图表图表上。 将显示 添加构造型”对话框。
4	在 名称”字段中，输入构造型名称（这也是新建模object的名称）。
5	通过单击 类型”下拉箭头选择这些object组之一： • 元素扩展——创建一个扩展元素的构造型 • 连接器扩展-创建扩展连接器的构造型 • 抽象元类 - 创建扩展结构或行为修饰符的构造构造型 • 元类扩展- 创建一个构造型，它扩展了已经存在于您的模型中（并且很可能在您当前正在使用的图表中）的元类
6	单击添加元类按钮。 将显示 扩展元类”对话框，其中显示与在步骤 5 中选择的object组关联的object类型列表。 从列表中选择要扩展的元类，然后单击确定按钮。 如果您在步骤 5 中选择了 无类扩展”，则会显示 选择配置文件元素浏览器/搜索”对话框；搜索并选择现有的 Metaclass元素以使用此构造型进行扩展。 元类名称被添加到 扩展”字段。
7	如果您想扩展多个具有构造型的元类，请再次单击 添加元类”按钮并选择下一个要扩展的object类型。您可以对尽可能多的元类重复此操作，并使用此构造型进行扩展。 要从 扩展”列表中删除选定的元类，请单击 删除”按钮。
8	审阅 构造型”面板中的可用属性。这些属性修改构造型的行为。 要应用属性，请单击 值”字段并键入或选择适当的值。 当您选择一个属性字段时，属性效果的描述会显示在 构造型”面板的底部。

	仅提供要应用于此构造型的属性的值。
9	单击 扩展”字段审阅类的名称，并在 无类”面板中查看可用的属性。这些属性根据特定于被扩展的元类的选项进一步修改构造型的行为。 要应用属性，请单击 值”字段并键入或选择适当的值。 当您选择一个属性字段时，属性效果的描述会显示在 无类”面板的底部。 不要为您不想应用于此构造型的任何属性提供值。 如果要扩展多个元类，请单击 扩展”字段中的下一个元类名称并审阅该object类型的属性。
10	单击下一步按钮。显示 定义标记值”页面。
11	在 属性”面板中，右击可显示上下文菜单，其中包含创建和分组不同类型的标记值的选项。这些选项包括： ● 添加标记值：创建一个简单的标记值- 显示一个提示标记值名称。添加名称，点击确定按钮，在 属性”栏显示名称；要设置默认值，请在 默认值”字段中输入 ● 添加专业标记值： -枚举：创建一个枚举标记值，基于 在一个现有的枚举元素上 - 预定义：从一个预定义的标记值类型中选择一个 列表，并在 默认值”字段中，输入或选择一个首字母 必要时的价值 - Structured：创建一个由Structured标记值组成的结构 其他几个简单的标记值，由现有的类型 类元素 -参考：创建一个标记值，用户可以使用它 定位并引用使用指定创建的元素 构造型（ RefGUID标记值的一种形式）；在创造这个时， 您必须选择定义的现有构造型元素 刻板印象 -参考列表：创建一个用户用来标记值 可以定位和引用使用 a 创建的元素列表 指定的构造型（RefGUIDList标注标记值的一种形式）； 在创建它时，您必须选择现有的构造型元素 定义了刻板印象 ● 编辑标记值名称：显示一个简单的提示，您可以在其中改写当前名称以更正或更改它 ● 创建标签组：在元类元素中创建标签组，通过它来组织你在构造型标记值中创建的元素 ● 将标签移动到组（当您右键单击现有的标记值时显示）：显示 移动标签到组”对话框，您可以在该对话框中选择现有的标签组以包含选定的标记值 ● 删除分组：删除选中的标签组，使其成员标记值列在 属性”列的末尾 ● 删除构造：从列表和列表中删除选定的标记值构造型
12	单击下一步按钮。将显示 定义形状脚本”页面。 形状脚本A用于定义构造型形状的外观。要包含一个形状脚本，请单击编辑按钮。 显示形状编辑器窗口。使用这个编辑器创建你的形状脚本。 完成脚本创建后，点击确定按钮。由形状脚本定义的图像显示在 预览”面板中。 注记：为使形状脚本生效，保存配置文件时必须选择 图像”选项。 或者，您可以在创建构造型元素后为模型object定义简单的默认外观（背景颜色、线条颜色）。
13	单击完成按钮。构造型元素和元类元素现在显示在配置文件图上。
14	您现在可以： ● 对要创建的其他每个构造型元素重复步骤 2 到 13

	• 使用配置文件帮助器编辑您定义的构造型（并通过它，元类）元素属性 • 将约束添加到您的构造型元素中 • 如果尚未设置形状，那么您现在可以定义object的默认外观（背景颜色、线条颜色） • 为配置文件中的原型元素和连接器设置快速链接器定义

注记

- 如果您打算扩展大量的模型元素，而不是将它们全部放在一个图表上，您可以在<<profile>>包下创建额外的子类图，并将不同类型的元类元素添加到不同的图表中；在这种情况下，您将包保存为配置文件，而不是单个图表
- 构造型元素必须具有唯一的名称，但元类元素可以具有相同的名称（例如，可以有多个行动元类，每个具有不同的行动属性）
- 如果您在构造型元素中有多个标记值，并且您已将它们分配给组，您可以在属性窗口的 标签”选项卡中定义哪些组默认为展开（打开），哪些默认为关闭;在 属性”页面打开元类的特征窗口，并添加属性 _tagGroupStates，初始值为 <groupname>=closed;<groupname>=closed;<groupname>=open; ...

编辑构造型元素

如果要添加或更正配置文件中的构造型或元类元素的属性，可以使用标准功能编辑它，例如元素 属性"对话框和 标签"选项卡。但是，您也可以通过配置文件帮助器的 构造型属性"对话框更新构造型元素，并且还可以通过构造型更新构造型扩展的 Metaclass 元素。

您通过其他方式（例如通过元素 属性"对话框）对元素所做的任何更改都会反映在配置文件帮助器的内容中。

访问

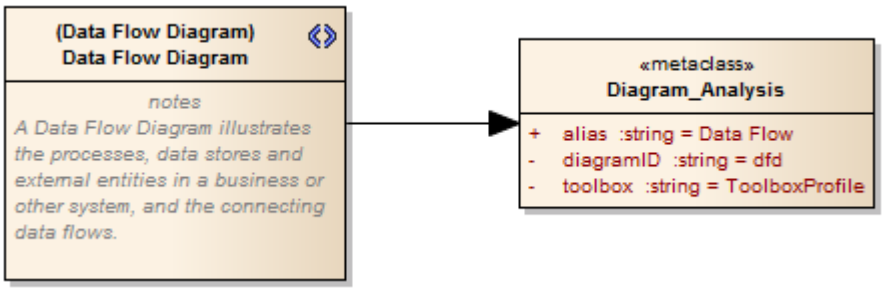
上下文菜单	右键单击构造型元素|使用配置文件助手编辑

编辑构造型元素

节	描述
1	构造型属性"对话框默认为 常规"选项卡。在此选项卡上，您可以： • 更改构造型元素名称 • 添加更多元类元素以通过这个构造型元素进行扩展 • 添加或更改构造型元素的属性值 • 为每个 Metaclass元素的属性添加或更改值
2	点击 标记值"标签。在此选项卡上，您可以： • 编辑标签的默认值 • 添加一系列类型之一的新标签 • 创建标签组 • 将标签分配或重新分配给组 • 删除标签组 • 从构造型中删除标记值
3	单击 形状脚本"选项卡。在此选项卡上，您可以： • 添加一个形状脚本（如果不存在） • 使用形状编辑器编辑现有的形状脚本
4	编辑完构造型元素后，单击确定按钮。 配置文件类图重新显示，已编辑的元素显示您所做的更改。

使用配置文件帮助器创建图表Profiles文件

当您开发MDG 技术时，可以创建扩展图表类型并将它们作为自定义图表Profiles在您的MDG 技术中。例如，您可以创建一个图表图形配置文件分析，将 DFD 图定义为内置图的扩展，如下所示：



添加图表扩展”配置文件帮助器可以帮助您定义图表配置文件，添加必要的元素并赋予它们适当的属性以定义生成的自定义图表类型的功能。

创建扩展图表类型

节	行动
1	如果您还没有这样做，请使用模型生成器的 管理 | MDG 技术生成器”蓝图来创建一组用于定义Profiles的包。 在浏览器窗口中，找到具有 <<diagram profile>> 构造型的包并打开其子图。
2	将工具箱的 配置文件助手”页面中的 添加图表扩展”项拖到图表上。 将显示 添加图表扩展”对话框。
3	在 名称”字段中，输入自定义图表类型的名称。
4	在 扩展类型”字段中，单击下拉箭头并选择自定义图表类型将扩展的内置图表类型。
5	在 描述”字段中输入图表用途的简要说明。 当用户在 New图表”对话框中选择此图表类型时，此描述将显示在对话框的右下方。
6	在 属性”窗格中输入这些字段的值： • 别名：定义在图表标题栏上显示在单词 图表”之前的图表类型；例如： 块图表 • 框架 ID：定义将出现在图表框架标签中的图表类型 • 框架格式字符串：输入一个包含替换宏的string ，用于定义框架标题，可以带或不带 () 等附加分隔符；可以使用的宏是： - #DGMALIAS# - #DGMID# - #DGMNAME# - #DGMNAMEFULL# - #DGMOWNERNAME# - #DGMOWNERNAMEFULL# - #DGMOWNERTYPE# - #DGMSTEREO# - #DGMTYPE# • 工具箱配置文件：点击下拉箭头，选择定义所需工具箱配置文件的图表类型（保存配置文件

	时输入的名称）；每次打开这种类型的工具箱都会自动打开工具箱 • 泳道：定义将在图表上显示的泳道；例如： 车道=2；方向=水平；车道1=标题1；车道2=标题2； （其中*Lanes*可以是任何值，但*Lane<n>*值的数量必须等于*Lanes*的值；方向可以省略，在这种情况下泳道默认为垂直）
7	属性”窗格中的其余字段可用 自定义图表的默认选项。将不会应用任何留空的属性。 当用户选择一个字段时，属性效果的描述会显示在 属性”窗格的底部。
8	点击确定按钮。适当的构造型和元类元素被添加到图中。
9	对要包含在图表配置文件中的每个图表扩展重复步骤 2 到 8。
10	将图表另存为配置文件。
11	将配置文件合并到MDG 技术中。

注记

- 添加图表扩展后，您可以通过右键单击图表上的适当构造型元素并选择 使用配置文件帮助器编辑”来再次修改其属性

使用配置文件帮助器创建工具箱Profiles文件

在MDG 技术中，您可以创建多个工具箱Profiles 。每个工具箱配置文件定义一个工具箱。工具箱由A或多个可展开/可折叠的区域组成，称为工具箱页面。

创建工具箱配置文件

节	行动
1	如果尚未创建用于定义Profiles的包组，请使用模型生成器的 管理 | MDG 技术生成器”蓝图来创建该组。 在浏览器窗口中，找到具有 <<toolbox profile>> 构造型的包并打开其子图。
2	将 配置文件助手”工具箱页面中的 创建自定义工具箱”项拖到图表上。 将显示 选择工具箱配置文件包”对话框。
3	选择步骤1中提到的具有 <<toolbox profile>> 构造型的包。 点击确定按钮。将显示 创建工具箱页面”对话框。
4	在 工具箱名称”字段中输入您的工具箱页面的名称。 这是在使用图形图表中的项目搜索功能时为工具箱页面显示的工具箱。
5	在 描述”字段中输入工具箱的描述。 此描述充当工具箱的默认工具工具箱，除非您如步骤 10 中所述为工具箱页面定义特定的工具提示。
6	点击确定按钮。 将创建并显示用于定义工具箱的图表。
7	（可选）将项目从工具箱拖到图表上时，该项目通常会创建元素或连接器。 也可以有一个工具箱项目，当拖到图表上时，将提供可供选择的项目选择。这被称为隐藏的子菜单。 如果您希望您的工具箱包含一个或多个隐藏的子菜单，您应该在继续执行此页面上的步骤之前定义这些子菜单。
8	您现在可以定义将出现在工具箱上的一个或多个工具箱页面。 将 配置文件助手”工具箱页面中的 添加工具箱页面”项拖到图表上。 将显示 添加工具箱页面”对话框。
9	在 名称”字段中，输入工具箱页面的名称。 这是将显示在相应工具箱页面的标题栏中的文本。
10	在”工具提示”字段中，输入相应工具箱页面的工具提示。
11	在这种情况下， 图标”字段将被禁用。该字段仅在定义隐藏的子菜单工具箱页面时使用。
12	这些选项可用于确定工具箱页面的外观和功能。启用时： • 'Images Only'：显示工具箱页面，图标旁边没有文本标签

	• 'Is Hidden'：将工具箱页面定义为隐藏的子菜单 • '公共'：工具箱页面在您的技术处于活动状态时对所有定义的工具箱都是通用的；页面最初显示为折叠状态 • 'Is Collapsed'：工具箱页面最初是最小化的
13	您现在可以定义要添加到工具箱的项目。 单击 添加”按钮右侧的向下箭头。选择以下选项之一： • '添加构造型'：为当前模型的UML配置文件中定义的构造型添加工具箱项；此配置文件必须包含在MDG 技术中的工具箱配置文件中 选择此选项后，将显示 选择配置文件元素”对话框；使用它来选择要添加的构造型元素（如果需要，在单击多个元素时按住 Ctrl 键） • '添加内置类型'： -元素：为UML元素类型添加一个工具箱项 选择此选项后，将显示 创建新工具箱项”对话框；在 别名”字段中，键入出现在工具箱项目上的标签，然后单击确定按钮 然后显示 选择元类”对话框；选择要添加到您的工具箱的UML元素类型，然后点击确定按钮 - '连接器'：为UML连接器类型添加工具箱项 选择此选项后，将显示 创建新工具箱项”对话框；在 别名”字段中，键入出现在工具箱项目上的标签，然后单击确定按钮 然后显示 选择元类”对话框；选择要添加到您的工具箱的UML连接器类型，然后点击确定按钮 • '添加隐藏工具箱'：添加隐藏工具箱子菜单项；使用此选项之前必须定义隐藏的工具箱 选择此选项后，将显示 创建新工具箱项”对话框；在 别名”字段中，键入要出现在工具箱项目上的标签，然后单击确定按钮 然后显示 选择隐藏的工具箱构造型”对话框；选择要添加到您的工具箱中的隐藏工具箱，然后单击确定按钮 • '加新项目'：添加一个工具箱项目，只有一个别名 仅此选项不会创建功能工具箱项；以这种方式添加的工具箱项，以后必须通过工具箱项列表进行修改 单击 添加”按钮，而不是单击下拉箭头，与选择 添加构造型”选项相同。
14	（可选）定义一个工具箱项目，该项目将从外部MDG 技术创建项目。例如，添加一个创建工具箱块元素的工具箱项。 1. 单击 添加”按钮右侧的向下箭头。 2. 选择 加新物品”选项。 将显示 创建新工具箱项”对话框。 3. 在 别名”字段中，键入要出现在工具箱项目上的标签，然后单击确定按钮。 工具箱项将被添加到 工具箱项”列表中。 4. 在此工具箱项目的 构造型”字段中，键入： 配置文件::构造型(UML ::BaseUMLType) -配置文件是定义构造型的配置文件的名称 -构造型是此工具箱项将创建的构造型/元类型的名称 - *BaseUMLType*是非 UML object的基本UML类型 例如，要在工具箱中包含 SysML块，您可以键入： SysML1.3::块(UML ::类) 5. 要识别配置文件：构造型string ，请创建要包含在工具箱中的类型的元素（例如，SysML 1 .3块），然后选择元素并显示属性窗口。 此元素的任何预定义标签都将分组在配置文件::构造型标题下；例如，SysML 1 .3块的标签被分组在 SysML1.3::块下。 Enterprise Architect中的所有非 UML 对象都是UML类型的扩展。您可以通过删除元素的构造型来显示元素的基本UML类型。例如，创建一个构造型块，然后使用属性窗口删除该块元素的构造类型。元素类型将从块变为类。

15	（可选）创建一个工具箱项目，将模式拖放到图表上。 1. 单击 添加"按钮右侧的向下箭头。 2. 选择 加新物品"选项。 将显示 创建新工具箱项"对话框。 3. 在 别名"字段中，键入要出现在工具箱项目上的标签，然后单击确定按钮。 4. 工具箱项将被添加到 工具箱项"列表中。 5. 在此工具箱项目的 构造型"字段中，键入： TechnologyID::PatternName(UMLPattern) - *TechnologyID*是在MDG 技术创建向导中输入的技术ID - *PatternName*是保存模式时输入的名称；例如： BusFramework::Builder(UMLPattern) 如果您想避免显示 添加模式"对话框，请将 (UMLPattern) 替换为 (UMLPatternSilent)。 6. 要在自定义工具箱（例如 GoF模式）中定义基于模型的模式，请创建一个具有以下格式名称的属性： PatternCategory::PatternName(UMLPattern) 例如： GoF::Mediator（UMLP 模式）
16	添加工具箱项后，它将出现在 工具箱项"列表中。您可以选择为工具箱项目添加自定义图标图像。 图标图像必须是 16x16 像素的位图文件；对于透明背景，使用浅灰色 - RGB(192,192,192)。 设置工具箱项目的图标： 1. 在 工具箱项"列表中找到该项目，然后在 工具箱图标"列中单击。 2. 单击此列中的  按钮。将显示 选择工具箱图标"对话框。 3. 找到图像文件并单击 打开"按钮。
17	对要添加到工具箱页面的每个项目重复步骤 13 到 16。 要删除工具箱项目，请在 工具箱项"列表中选择它，然后单击删除按钮。 添加所有适当的工具箱项目后，单击确定按钮。构造型A元素添加到您的工具箱配置文件图中。
18	对要包含在工具箱中的每个工具箱页面重复步骤 8 到 17。
19	通过单击打开图表的背景并选择任一功能区选项来保存工具箱配置文件： • 设计>图表>管理>另存为配置文件或 • 特定>技术> 发布技术> 将图表发布为UML配置文件
20	将配置文件合并到MDG 技术中。

注记

- 可以A右键单击工具箱配置文件图表上的适当构造型元素并选择 使用配置文件帮助器编辑"选项来修改工具箱页面
- 为工具箱页面指定名称时，请注意 无素"是保留字；如果使用了 无素"一词，则不会出现在相应工具箱页面的标题栏中
- 序列中工具箱页面的工具箱由配置文件图或配置文件包序列它们的构造型元素的顺序决定；如果您从以下位置创建并保存配置文件：
  -图表，工具箱页面序列是由构造型元素的Z-order决定的
  该图 -构造型元素的 Z 序数越高，越往下

工具箱其工具箱页面放置；如果你改变构造型元素的 Z 顺序
图它改变了元素页面在工具箱中的位置
-包在浏览器窗口中，工具箱的页面序列由浏览器的列表顺序决定
包中的构造型元素 - 第一个列出元素的工具箱页面位于顶部
工具箱；如果你重新排序浏览器窗口中的元素，你会产生相同的结果
工具箱中页面的重新排序

使用配置文件Helpers 创建隐藏的子菜单

当您创建工具箱项目时，其中一些可能非常相似，因为它们基于相同类型的元类。例如，有许多不同类型的行动元素。您可以创建一个 基础”工具箱项目，并从子菜单中提供一个变体选项，而不是用每个变体填充工具箱页面，当基础项目被拖到图表上时会显示该子菜单。

定义一个隐藏的子菜单

节	行动
1	如果您尚未这样做，请创建并显示您将用于定义工具箱，如使用配置文件帮助器创建工具箱 *Profiles*文件的步骤1至 6 中所述。
2	将 配置文件助手”工具箱页面中的 添加工具箱页面”项拖到图表上。 将显示 添加工具箱页面”对话框。
3	在 名称”字段中，输入子菜单工具箱项的名称。
4	在这种情况下，”工具提示”字段可以留空。
5	选中 隐藏”复选框。 'Images Only'、'公共' 和 'Is Collapsed' 复选框应保持未选中状态。
6	选择 隐藏”复选框后， 图标”字段应变为活动状态。您可以选择为子菜单工具箱项添加自定义图标图像。 图标图像必须是 16x16 像素的位图文件；对于透明背景，使用浅灰色 - RGB(192,192,192)。 要设置子菜单工具箱项目的图标，请单击 图标”字段右侧的文件夹图标。选择图像文件并单击 打开”按钮。
7	您现在可以将元素和连接器等项目添加到子菜单。 单击添加按钮右侧的向下箭头，然后选择以下选项之一： • '添加构造型'：为当前模型的UML配置文件中定义的构造型添加工具箱项；此配置文件必须包含在MDG 技术中的工具箱配置文件中 选择此选项后，将显示 选择配置文件元素”对话框；使用它来选择要添加的构造型 • '添加内置类型'： -元素：为UML元素类型添加一个工具箱项 选择此选项后，将显示 创建新工具箱项”对话框；在 别名”字段中，键入 出现在工具箱项目上的标签，然后单击确定按钮 然后显示 选择元类”对话框；选择要添加到您的工具箱的UML元素类型， 然后点击确定按钮 -连接器：为UML连接器类型添加工具箱项 选择此选项后，将显示 创建新工具箱项”对话框；在 别名”字段中，键入 出现在工具箱项目上的标签，然后单击确定按钮 然后显示 选择元类”对话框；选择要添加到您的工具箱的UML连接器类型， 然后点击确定按钮 • '添加隐藏工具箱'：添加隐藏工具箱子菜单项；创建 隐藏工具箱”子菜单本身时不要使用此选项 • '加新项目'：添加一个工具箱项目，只有一个别名 仅此选项不会创建功能工具箱项；以这种方式添加的工具箱项必须稍后通过 工具箱项”列表进行修改

	单击 添加”按钮，而不是单击下拉箭头，与选择 添加构造型”选项相同。
8	（可选）添加工具箱项后，它会出现在 工具箱项”列表中，您可以为该项添加自定义图标图像。图标图像必须是 16x16 像素的位图文件；对于透明背景，使用浅灰色 - RGB(192,192,192)。 要设置工具箱项目的图标，请在 工具箱项”列表中找到该项目，然后单击 工具箱图标”列。单击此列中的 ... 按钮。将显示 选择工具箱图标”对话框。找到图像文件并单击 打开”按钮。
9	对要添加到子菜单的每个项目重复步骤 7 和 8。 要删除工具箱项目，请从 工具箱项”列表中选择它，然后单击删除按钮。 添加所有适当的子菜单项后，单击确定按钮。构造型A元素添加到您的工具箱配置文件图中。
10	对要创建的每个工具箱子菜单重复步骤 2 到 9。
11	之前创建的子菜单现在可以作为项目包含在工具箱页面中。

注记

- A通过右键单击工具箱配置文件图表上的适当构造型元素并选择 使用配置文件助手编辑”选项来修改子菜单

创建MDG 技术文件

创建MDG 技术文件时，可以包含范围广泛的功能和工具，包括UML Profiles 、代码模块、脚本、模式、图像、标记值类型、报告模板、链接文档模板和工具箱建造。使用MDG 技术创建向导将这些按逻辑序列放入MDG 技术文件很容易。

访问

功能区	特定>技术>发布技术>生成MDG 技术

创建MDG 技术文件

节	描述
1	选择 生成MDG 技术文件”选项。 显示MDG 技术创建向导屏幕。
2	单击下一步按钮。 MDG 技术向导提示您： • 基于新的MDG 技术选择 (MTS) 文件创建MDG 技术文件 • 基于现有 MTS 文件创建MDG 技术文件，或 • 不使用任何 MTS 文件 MTS 文件存储您在创建MDG 技术期间定义的选定选项；如果您使用 MTS 文件，您可以稍后修改它以添加或删除MDG 技术中的特定项目，这是推荐的过程。
3	选择适当的 MTS 文件选项。 单击下一步按钮。 如果您选择了 MTS 文件， MDG 技术向导会提示您将更改保存在现有 MTS 文件中或保存到新的 MTS 文件中；这使您能够在现有 MTS 文件的基础上创建修改，同时保留原始文件。
4	如有必要，输入或浏览所需的文件路径和名称。 单击下一步按钮。 将显示 MDG 技术向导 - 创建”对话框。
5	完全此屏幕上的字段： • 文件名 -类型或选择MDG 技术文件的路径和文件名；此文件的文件扩展名为 .xml • ID -类型MDG 技术文件的唯一引用，最长 12 个字符 • 版本类型MDG 技术文件的版本号 • 图标——（可选）类型或选择包含技术图标的图形文件的路径和文件名；该图标是 16 位或 24 位颜色深度、16x16 位图图像，显示在 MDG 技术”对话框左侧的技术列表中 • Logo - （可选）类型或选择包含技术标志的图形文件的路径和文件名；徽标是 16 位或 24 位色深、64x64 或 100x100 位图图像，显示在 MDG 技术”对话框右上角的显示窗格中 • URL -（可选）如果您有任何网站产品信息可能对使用此技术的用户有所帮助，请在此字段中

	键入或粘贴 URL • 支持 - （可选）如果您有任何可能对本技术用户有帮助的基于 Web 或其他支持功能，请在此字段中键入或粘贴联系地址 • 注记-类型MDG 技术功能的简短说明
6	单击下一步按钮。 MDG 技术向导 - 内容屏幕显示。
7	选中要包含在MDG 技术文件中的每个项目的复选框。 当您选择了要包含的所有项目的复选框后，单击 "下一步"按钮。 每个选择都会运行特定的对话框，以定义要包含在MDG 技术中的特定项目。
8	完成响应您的选择而显示的对话框，当所有对话框都完成后，单击下一步按钮。 显示 "MDG 技术向导 - 完成"屏幕，提供有关MDG 技术文件中包含的项目的信息。
9	如果您使用过 MTS 文件并想要更新它，请选中 "保存到 MTS"复选框。
10	如果您对所选择的项目感到满意，请单击 "完成"按钮。 如果需要，您现在可以编辑 MTS 文件以添加更多项目，例如： • 验证模型配置 • 模型生成器模板(模式) 编辑完 MTS 文件并重新生成技术(.xml) 文件后，您可以添加另一个 "脚本"部分以包含包XMI导出和/或导入脚本。保存已编辑的技术文件。 要使Enterprise Architect模型可以访问MDG 技术.xml 文件，您必须将技术文件路径添加到 "MDG技术-高级"对话框（通过单击 "MDG技术"对话框上的高级按钮，通过 "特定>技术>管理技术"功能区选项进行访问）。

添加配置文件

创建MDG 技术文件时，您可以包含一个或多个符合UML 2.5 标准的配置文件，这些Profiles已定义用于创建新类型的模型元素。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将Profiles添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直到第 6 节（包括第 6节），在该节中选中“Profiles ”复选框。 将显示“MDG 技术向导 -配置文件文件选择”页面。
2	在“目录”字段中，导航到包含所需配置文件或Profiles的目录。 配置文件文件会自动列在“可用文件”面板中。
3	要单独选择每个所需的配置文件，请突出显示“可用文件”列表中的配置文件，然后单击 --> 按钮。 文件名显示在“选定文件”列表中。 或者： 要选择每个可用的配置文件，请单击 -->> 按钮，然后通过选择它并单击 <-- 按钮返回每个您不想要的配置文件。 • Profiles在此对话框中选择工具箱Profiles或图表；这会在 .mts 文件中生成冲突的命令 • 确保您确实包含您的UML Profiles
4	单击下一步按钮继续。

添加模式

创建MDG 技术文件时，您可以在浏览器窗口的 资源”选项卡中包含您想要提供的特殊设计模式，如果您愿意，也可以在技术工具箱页面中。您之前已经发布过这些模式。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将设计模式添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤进行操作，直到第 6 节（包括第 6节），在该节中选中 模式”复选框。 显示 MDG 技术向导-模式文件”选择页面。
2	在 目录”字段中，导航到包含所需模式XML 文件的目录。 模式文件会自动列在 可用文件”面板中。
3	要单独选择每个所需模式，请在 可用文件”列表中突出显示该模式，然后单击 --> 按钮。 文件名显示在 选定文件”列表中。 或者，要选择所有可用模式，请单击 -->> 按钮，然后通过选择它并单击 <-- 按钮返回每个您不想要的模式。
4	单击下一步按钮继续。

添加一个图表配置文件

创建MDG 技术文件时，您可以包含已定义的图表Profiles以生成新类型的图表。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将图表文件到MDG 技术Profiles

节	描述
1	按照创建*MDG* 技术主题中的步骤直到第 6 节（包括第 6节），在该节中选择 图表类型”复选框。 ' MDG 技术向导 -图表' 页面显示。
2	在 目录”字段中，导航到包含所需图表的Profiles 。 目录中的Profiles会自动列在 可用文件”面板中。
3	要单独选择每个所需的图表配置文件，请在 可用文件”列表中突出显示文件名，然后单击 --> 按钮。 文件名显示在 选定文件”列表中。 或者，要选择所有可用的Profiles Profiles如果它们都是图表），请单击 -->> 按钮，然后通过选择它并单击 <-- 按钮来返回您不想要的每一个。
4	单击下一步按钮继续。

添加工具箱配置文件

创建MDG 技术文件时，您可以包含您创建的图表工具箱定义，以提供工具箱页面以支持自定义图表。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将工具箱Profiles添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直至第 6 节（包括第 6节），在该节中选中 "工具箱"复选框。 将显示 "MDG 技术向导 - 工具箱"页面。
2	在 "目录"字段中，导航到包含所需工具箱Profiles的目录。 配置文件文件会自动列在 "可用文件"面板中。
3	要单独选择每个所需工具箱配置文件，请在 "可用文件"列表中突出显示文件名，然后单击 --> 按钮。 文件名显示在 "选定文件"列表中。 或者，要选择所有可用的Profiles（如果它们都是工具箱Profiles ），请单击 -->> 按钮，然后通过选择它并单击 <-- 按钮返回您不想要的每一个。
4	单击下一步按钮继续。

添加标记值类型

在创建MDG 技术文件时，可以包含标记值类型，技术用户可以从中创建特定领域的标记值。您可以使用两种方法：

- 在配置文件图表上的数据类型元素中定义标记值类型，如使用预定义标签类型帮助主题（推荐）中所述，或者
- 直接在MDG 技术向导中添加标记值类型，如此处所述。

访问

功能区	特定>技术>发布技术>生成MDG 技术

在MDG 技术标记值类型技术文件中添加值类型

节	描述
1	按照创建*MDG* 技术主题中的步骤直到第 6 节（包括第 6节），在此选中 标记值类型”复选框。 显示 ‘MDG 技术向导 -标记值类型”页面。
2	要单独选择每个所需的标记值类型，请在 可用标记值”列表中突出显示名称，然后单击--> 按钮。 名称显示在 已选择的标记值”列表中，标记值类型的名称、描述和注记显示在页面底部的面板中。 或者，要选择所有可用的标记值类型，请单击 -->> 按钮，然后通过选择并单击 <-- 按钮返回您不想要的每一个。
3	单击下一步按钮继续。

添加代码模块

创建MDG 技术文件时，您可以包含已为其设置代码模板和数据类型的代码模块。这些模块可以用于修改系统默认语言，也可以用于您使用代码模板和代码模板编辑器自己定义的语言。在编辑器中为新语言设置代码模板之前，您必须为该语言定义至少一种数据类型。您还可以指定语言的代码选项，这是数据类型或代码模板未解决的附加设置；它们保存在一个 XML 文档中，该文档包含在模块的MDG 技术文件中。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将代码模块添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直至第 6 节（包括第 6节），在该节中选中 代码模块”复选框。' MDG 技术向导 - 代码模块' 页面显示，列出了您当前项目中定义的代码模块。
2	单击要包含在技术中的每个代码模块的复选框（ 产品”、 数据类型”、 代码语法”和 代码模板”）。
3	如果您已为选定模块创建了代码选项 XML 文档，请单击该模块的 代码选项”列中的  按钮。将显示A浏览器，您可以通过它找到并选择 XML 文档。
4	单击下一步按钮继续。

定义代码选项

在修改现有编程语言的代码生成模板或定义新的编程语言时，还有一些仅在构建MDG 技术时可用的附加选项。这些附加选项会影响Enterprise Architect如何处理该语言的代码生成和逆向工程。这些选项是使用 XML 文件指定的，该文件是使用您首选的文本编辑器创建的。

XML 文档中的根节点名为根。子节点被命名为 CodeOption。每个 CodeOption 包含一个 name 属性，对应于可用代码选项之一的名称。每个节点的主体都包含选项值。例如：

<代码选项>

<CodeOption name="DefaultExtension">.h</CodeOption>

<CodeOption name="HasImplementation">true</CodeOption>

<CodeOption name="ImplementationExtension">.cpp</CodeOption>

<CodeOption name="Editor">C:\Window\notepad.exe</窗口>

</代码选项>

支持的代码选项

代码选项	描述
构造函数名称	用作构造函数的函数的名称。由 classHasConstructor 代码模板宏使用。
CopyConstructorName	用作复制构造函数的函数的名称。由 classHasCopyConstructor 代码模板宏使用。
默认扩展	生成代码时的默认扩展名。
默认源目录	Enterprise Architect生成新文件的默认路径。
析构函数名称	用作析构函数的函数的名称。由 classHasDestructor 代码模板宏使用。
编辑	用于编辑该语言源的外部编辑器。
有实现	指定此语言的代码生成是否同时生成源文件和实现文件。
实施扩展	Enterprise Architect用来生成实现文件的扩展。
实施路径	从源文件生成实现文件的相对路径。
包路径分隔符	使用 packagePath 宏与代码模板时用于分隔包名的分隔符。

注记

- 一旦可以在模型中使用一种语言（通过导入和激活MDG 技术），您可以在 首选项”对话框中显示和编辑代码选项（“>开始外观>首选项>首选项”）

添加数据库数据类型

在创建MDG 技术文件时，您可以包含 DDL 文件，这些文件为您打算通过您的技术使用并已为其设置数据类型的每个数据库定义数据库数据类型。在为新的数据库类型设置 DDL 文件之前，您必须为该数据库定义至少一种数据类型。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将数据库数据类型添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直到第 7 节（包括第 7节），在该节中选中 "DDL 文件"复选框。"MDG 技术向导 - DDL 文件"页面显示，列出了当前项目中可用的数据库类型。
2	单击要在技术中包含 DDL 文件的每种数据库类型的复选框。如果模型中存在 DDL 的数据类型，也选择相应的 数据类型"复选框。
3	单击下一步按钮继续。

添加 MDA 转换

创建MDG 技术文件时，您可以包含您在模型中创建或修改的任何 MDA变换模板，并且您希望将其部署为技术的一部分。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将 MDA变换添加到模板MDG 技术文件中

节	描述
1	按照创建*MDG* 技术主题中的步骤直至第 6 节（包括第 6节），在该节中选中 "MDA 转换"复选框。 ' MDG 技术向导 - 转换模块' 页面显示，列出了您系统上可用的 MDA 转换模板。
2	单击要添加到MDG 技术中的每个转换模板名称旁边的复选框。
3	单击下一步按钮继续。

添加文档报告模板

创建MDG 技术文件时，您可以包含用户定义的文档报告模板。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将报告模板添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直到第 6 节（包括第 6节），在该节中选中“RTF模板”复选框。将显示“MDG 技术向导 - RTF 报告模板”对话框。
2	对于当前模型中可用的每个所需的用户定义报告模板，选中模板名称旁边的复选框。
3	单击下一步按钮继续。

添加链接文档模板

创建MDG 技术文件时，您可以包含链接文档模板。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将链接文档模板添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直到第 6 节（包括第 6节），在该节中选中 "链接文档模板"复选框。 将显示 "MDG 技术向导 - 链接文档模板"对话框。
2	对于当前模型中可用的每个所需文档模板，选择模板名称旁边的复选框。
3	单击下一步按钮继续。

添加图片

创建MDG 技术文件时，您可以合并要在部署该技术的所有模型中使用的图像。这些图像必须已经存在于正在开发该技术的模型中；您可以使用图像管理器上的添加加新按钮将图像导入此模型。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将图像添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直至第 6 节（包括第 6节），在该节中选中 图像”复选框。 将显示 "MDG 技术向导-图像选择”对话框。
2	对于当前模型中可用的每个所需模型图像，选中图像名称旁边的复选框。 当您选择复选框时，每个图像A预览都会显示在对话框的右侧。
3	单击下一步按钮继续。

添加脚本

创建MDG 技术文件时，您可以包含在模型中创建的脚本。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将脚本添加到MDG 技术文件

节	描述
1	按照创建*MDG* 技术主题中的步骤直到第 6 节（包括第 6节），在该节中选中 脚本"复选框。 将显示 "MDG 技术向导 - 脚本"对话框。
2	对于当前模型中可用的每个所需脚本，选择脚本名称旁边的复选框。
3	单击下一步按钮继续。

注记

- 此功能在Enterprise Architect的企业统一版和终极版中可用

添加工作区布局

在开发MDG 技术文件时，您可以包含用户定义的工作空间布局。工作空间布局是工具栏和窗口的排列，适合于需求管理和代码工程等工作领域。工作区布局自动打开并组织所有工具以适应您使用系统的方式。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将工作区布局添加到MDG 技术文件中

节	描述
1	在您的模型中，创建要包含在您的技术中的工作区布局。
2	按照创建*MDG* 技术主题中的步骤直至第 6 节（包括第 6节），在该节中选中 "工作区布局"复选框。 将显示 "MDG 技术向导 -工作区布局"对话框，列出可供您使用的用户定义的工作区布局。
3	对于您要合并到技术中的每个工作空间布局，选择布局名称旁边的复选框。
4	单击下一步按钮继续。

添加模型视图

在开发MDG 技术文件时，您可以包含用户定义的模型视图。模型视图基于从模型中提取特定信息的搜索，以提供模型的不同视角和 切入点”。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将模型视图添加到MDG 技术文件

节	描述
1	在您的模型中，创建您想要包含在您的技术中的模型视图。
2	按照创建*MDG* 技术主题中的步骤直至第 6 节（包括第 6节），在该节中选中 模型视图”复选框。 ' MDG 技术向导-模型视图'对话框显示，列出当前模型中可用的用户定义视图。
3	对于要合并到技术中的每个模型视图，选择视图名称旁边的复选框。
4	单击下一步按钮继续。

注记

- 技术视图不存储收藏包，只存储视图
- 如果您包含运行您定义的搜索的模型视图，您还必须在您的MDG 技术中包含这些搜索

添加模型搜索

在开发MDG 技术文件时，您可以包含用户定义的模型搜索。您可以使用模型搜索功能设置这些搜索，在#
, 在查询生成器中或作为插件
，然后将它们链接到您的MDG 技术中。

访问

功能区	特定>技术>发布技术>生成MDG 技术

将模型搜索添加到MDG 技术文件

节	描述
1	在您的模型中，创建您想要包含在您的技术中的模型搜索。
2	按照创建*MDG* 技术主题中的步骤直至第 6 节（包括第 6节），在此您选择 模型搜索”复选框。 显示 “MDG 技术向导 -模型搜索”对话框，列出当前模型中可用的用户定义搜索。
3	对于您要合并到技术中的每个模型搜索，选择搜索名称旁边的复选框。
4	单击下一步按钮继续。

注记

- 如果您使用自定义#
  搜索， #
  必须包含 ea_guid AS CLASSGUID 和object类型
- 如果您包含运行您定义的搜索的模型视图，您还必须在您的MDG 技术中包含这些搜索

使用 MTS 文件

当您使用MDG 技术向导创建MDG 技术文件时，您可以选择存储您在MDG 技术选择 (.mts) 文件中定义的所有选项和结构。这将捕获您在技术向导中输入的所有信息，因此您不必再次输入。如果您使用 .mts 文件，则可以随后对其进行编辑以更改您在生成技术文件时选择的特征，并添加或删除其他高级特征。

访问

功能区	特定>技术>发布技术>生成MDG 技术

管理 .MTS 文件

行动	描述
创建一个 .MTS文件	要创建 .mts 文件，请启动并通过MDG 技术向导工作；在第二页上，选择 创建新的 MTS 文件”选项。
.MTS文件的高级选项	完成MDG 技术向导并设置 .mts 文件后，您可以单独添加： • 模型验证配置 • 模板模型 首先为模型验证配置和模型模板定义 XMI，然后在文本编辑器中打开 .mts 文件并复制 </ MDG .Selections> 行之前的验证和/或模板描述。 保存 .mts 文件。
更新MDG 技术	再次启动MDG 技术向导，但这次在第二页上选择 打开现有 MTS 文件”选项并指定您一直在处理的 .mts 文件的文件路径。 单击下一步按钮，直到向导完成；您的MDG 技术.xml 文件已更新。

注记

- 使用向导和 .mts 文件创建您的MDG 技术后，您可以通过技术.xml 文件添加导入和导出脚本

创建工具箱Profiles

作为MDG 技术的一项功能，您可能希望提供图表工具箱，以访问您在该技术中创建的任何元素和连接器。您在特定的Profiles中定义这些工具箱页面，每个配置文件定义元素和连接器工具箱页面，这些页面为图表类型打开或可选择。

创建自定义工具箱

节	行动
1	创建一组工具箱Profiles ，其中包含生成工具箱页面所需的定义。
2	酌情编辑定义以： ● 包括隐藏的子菜单 ● 覆盖默认工具箱 ● 更改工具箱项目的默认图标
3	创建一个 .mts 文件，其中包含有关如何构建您的MDG 技术的说明，并在该技术中包含工具箱 Profiles 。

创建工具箱Profiles

在MDG 技术中，您可以创建多个工具箱Profiles 。每个工具箱配置文件都包含确定打开图表时显示哪些页面的定义，或者通过从工具箱中的图表功能中进行工具箱，或者通过打开或创建链接到工具箱配置文件。

工具箱配置文件Errors

当您使用图表MDG 技术中定义的工具箱时，可能会显示某些错误消息。此表解释了这些错误消息的含义。

信息	意义
缺少基本类型 <名称>	例如：'缺少基本类型：'SysML1.3::块' 不扩展' UML ::状态' 基本类型要么丢失，要么与扩展元素类型不对应（在示例中，SysML::块实际上扩展了UML ::类）。
未找到 ID <名称> 的配置文件	此错误消息可能意味着无法找到配置文件，或者包含配置文件的MDG 技术已被禁用（使用 专业化>技术>管理”进行检查）。
在配置文件 <name> 中找不到构造型 <name>	例如：在配置文件“SysML 1 .2”中没有找到原型 “ProxyPort”。 此消息表明所需的构造型与它应该包含的配置文件不匹配。在示例中，SysML1.2 没有 ProxyPorts，因此构造型应该是 “FlowPort”或配置文件“SysML 1 ” .3'。
未知/非法基类型：<name>	显示此消息可能有多种原因。例如： • 未知/非法基类型： UML ::Capability - 显示是因为没有 “Capability”这样的UML元类 • 未知/非法基本类型：SysML 1 .3::块显示是因为您试图从另一个配置文件扩展构造型，在本例中是来自 SysML 1 .3 配置文件的 <<Block>>；您必须扩展与您专门扩展的构造型相同的东西（在本例中为 “UML ::类”）

创建工具箱配置文件

节	行动
1	在配置文件包中，创建一个具有适当名称的类图，以便您以后参考它，例如 MyClassDiagram。
2	双击图表背景以显示图表 属性”对话框，并在 注记”字段中为图表指定别名和以下格式的描述： Alias=MyClass;Notes=类图的结构元素；
3	在图表上，创建一个名为 ToolboxPage 的元类元素。
4	为要在您的工具箱中创建的每个工具箱页面创建构造型元素，例如 MyClassElements 和 MyClassRelationships。 双击每个元素以显示 属性”对话框，并在 别名”字段中键入要在工具箱页面的标题栏中显示的文本，例如我的类或我的类关系。 在每个元素的 注记”字段中，输入相应工具箱页面的工具提示；例如，'元素类图表' 或 '关系类图表'。

	在每个构造型元素和 ToolboxPage 元类元素之间创建扩展连接器。
5	在每个构造型元素中，按 F9 并为该元素定义的页面中的每个工具箱项创建一个属性。 每个属性的名称是要删除的元素或连接器的名称，包括元素的命名空间；例如， UML ::包、UML ::类和UML ::接口。您可能不想在您的工具箱中显示包含诸如UML ::包或UML ::类之类的文本的名称，因此请给属性一个 初始值”，例如包或类。 工具箱项目的显示顺序与其在序列中的属性元素，因此请使用特征窗口 属性”页面中的属性排序选项来定义工具箱页面中图标的顺序。 在来自您自己的技术的元素或连接器的属性名称中，使用您的配置文件名称作为命名空间，然后在项目名称后面加上您正在扩展的元素或连接器类型，在括号中（以识别Enterprise Architect要创建什么类型的object ）；例如，一个 SysML块元素会显示为： SysML::块(UML ::类) 许多元素和连接器可以扩展以在工具箱中使用。
6	要定义工具箱项以将设计模式拖放到图表上，请将该属性命名为： 我的技术ID::我的模式（UMLPattern） 'MyTechnologyID' 是技术的 ID（不是名称），'MyPattern' 是要放弃的模式的名称；例如： 总线框架::Builder(UMLPattern) 如果您想避免显示 Add模式”对话框，请将（UMLPattern）替换为（UMLPatternSilent）。 要在自定义工具箱（如 GoF模式）中定义基于模型的模式，请创建一个具有以下格式名称的属性： 模式类别::模式名称（UMLPattern） 例如： GoF::Mediator（UMLPattern）
7	定义您需要修改工具箱页面显示的任何属性，例如工具箱页面是最小化还是不显示项目名称（标签）。
8	要保存工具箱配置文件，请单击打开图表的背景并选择任一功能区选项： • 设计>图表>管理>另存为配置文件或 • 特定>技术> 发布技术> 将图表发布为UML配置文件

注记

- 为工具箱页面分配别名时，“元素”是保留字；如果使用了 无素”一词，则不会出现在相应工具箱页面的标题栏中
- 每个配置文件元素合并到一个MDG工具箱页面中，启用一个上下文菜单选项来同步从它创建的所有对象的标记值和约束
- 序列中的工具箱页面由配置文件图或配置文件包序列的构造型元素的工具箱决定；如果您从以下位置创建并保存配置文件：
  -图表，工具箱页面序列由构造型元素的Z-order决定
  在图中 -构造型元素的 Z 序号越低（越接近1 ）（越接近
  它是图表的 表面”），工具箱的工具箱页面放置得越往下；
  如果改变图中构造型元素的 Z 顺序，它会改变构造型元素的位置
  工具箱上的元素页面
  -包在浏览器窗口中，工具箱的页面序列由浏览器的列表顺序决定
  包中的构造型元素——第一个列出元素的工具箱页面位于
  工具箱顶部；如果你重新排序浏览器窗口中的元素，你会产生相同的结果
  工具箱中页面的重新排序

工具箱页面属性

当您创建构造型元素来定义MDG 技术中的工具箱页面时，您可以添加许多属性来控制页面本身在图表工具箱中的行为方式。构造型元素可以是扩展 ToolboxPage 元类的几个元素之一。

您可以添加的属性包括：

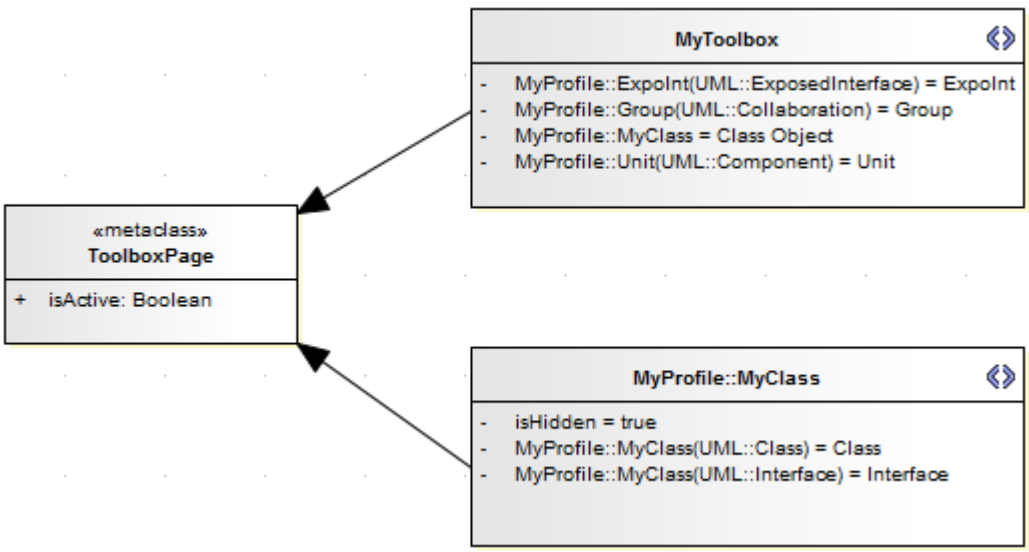
- 图标 - 请参阅Assign Icons To Toolbox Items
- ImagesOnly - 如果将 初始值”设置为 true，则工具箱页面显示时图标旁边不会显示文本标签
- isCollapsed - 如果将 初始值”设置为 true，则工具箱页面最初处于最小化状态
- isCommon - 如果将初始值设置为 true，则只要来自相同技术的另一个工具箱页面是当前工具箱页面，就会显示此公共工具箱页面
- isHidden - 请参阅Create Hidden Sub-Menus

创建隐藏的子菜单

当您在工具箱页面上创建项目时，其中一些可能非常相似并且基于相同类型的元类。例如，有许多不同类型的活动元素，在行动2.0 中，您可以创建每种类型的事件元素，无论是独立的还是边缘安装在另一个元素上。您可以创建一个 基础”工具箱项目并从子菜单中提供变体选项，而不是用每个变体填充工具箱页面，当基础项目被拖到图表上时会显示该变体，但在其他情况下会被隐藏。这种技术对于可以应用于多个元类的 消除歧义”构造型非常有用。

在子菜单中，您只定义变体类型（对于行动元素列表）。但是，如果变体还为其定义了 ToolboxItemImage，则该图标将显示在子菜单中的变体名称旁边（对于 BPMN 2.0事件）。您还可以使用此方法专门定义将应用于子菜单选项的图标。

定义一个隐藏的子菜单

节	行动
1	在与 ToolboxPage 元类相同的图表上创建构造型元素，名称以配置文件名为前缀（这是强制性的）。例如： 我的个人资料::我的班级 该名称不得与任何其他配置文件中存在的任何外部构造型的名称匹配。 子菜单元素可以有一个别名。
2	在这个子菜单构造型元素中，创建属性 isHidden ，其初始值为True 。 对于每个子菜单项，添加一个属性来标识该项。将 初始值”设置为要在菜单中显示的名称。例如，如果 «MyClass» 构造型可以应用于UML类或UML接口，这两个选项的属性将是： MyProfile::MyClass(UML ::类) 初始值 =类 MyProfile::MyClass(UML ::接口) 初始值=接口
3	创建第二个构造型元素并定义一个与子菜单构造型元素同名的属性，并使用要在工具箱项中显示的文本的初始值。例如： MyProfile::MyClass =类物件 像工具箱一样为工具箱中的其余项目定义附加属性。
4	在每个构造型元素和 ToolboxPage 元类元素之间创建 <<Extension>>关系，如图所示。 

	使用该配置文件时，当类物件项从工具箱拖到图表上时，隐藏菜单显示类或接口选择；在选择时，元素被放到图表上。
5	如果没有从现有定义中为工具箱项目分配图标，并且您想要显示一个，请将图像定义为 ToolboxItemImage 图标。

将图标分配给工具箱项

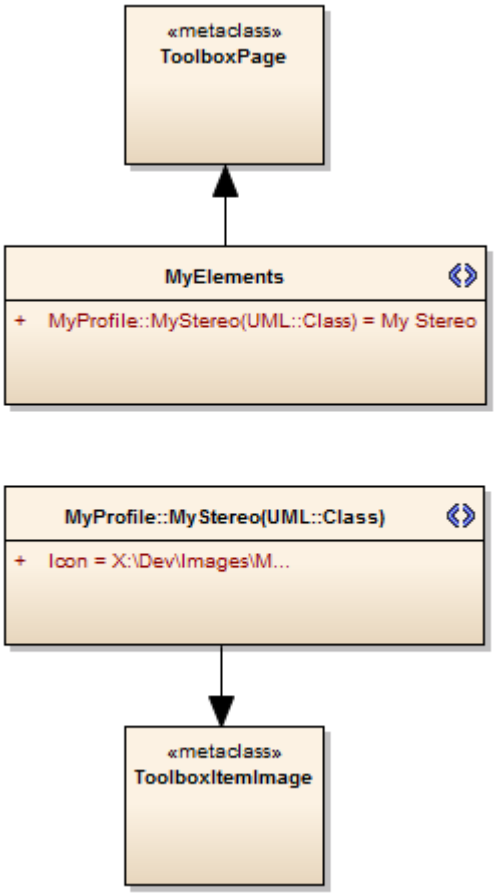
当您创建一个原型模型元素来定义一个在图表工具箱页面中表示的元素或连接器时，您可以定义显示在浏览器窗口中的元素名称和工具箱中的元素或连接器类型的图像工具箱页，通过将特殊属性图标分配给构造型元素。

可以通过扩展 ToolboxItemImage 元类来覆盖或替换工具箱项的此图像定义，该过程通常是可选的。但是，如果你想在隐藏的子菜单上显示一个图标，你必须使用这个方法；系统选择 ToolboxItemImage 定义作为隐藏菜单项的图标。

如果您不使用图标属性或 ToolboxItemImage 元类来定义工具箱图标，则图像默认为用于已扩展的标准UML模型元素的图像。如果没有这样的图像，则图标使用系统默认的通用 "工具箱项"图像。

扩展 **ToolboxItemImage** 元类

节	行动
1	在与工具箱项目相同的工具箱配置文件中创建一个新的构造型元素。
2	为构造型元素赋予与为其分配图像的元素相同的名称；例如： MyProfile::MyStereo(UML ::类)
3	为构造型元素赋予特殊属性 Icon with Initial Value 设置为要使用的图像的完整路径和文件名。 图标图像是一个 16x16 像素的位图文件；对于透明背景，使用浅灰色 - RGB(192,192,192)。
4	创建一个名为 ToolboxItemImage 的元类元素，并创建从构造型元素到此元类的扩展关联。


«metaclass»
ToolboxPage
MyElements
+ MyProfile::MyStereo(UML::Class) = My Stereo
MyProfile::MyStereo(UML::Class)
+ Icon = X:\Dev\Images\M...
«metaclass»
ToolboxItemImage

覆盖默认工具箱

当您创建一种内置图表类型的图表时，系统会根据相应的默认工具箱配置文件显示一个图表工具箱页面。如果您自定义了图表类型，它仍然会为您扩展的基本图表类型应用系统默认工具箱页面，除非您使用您自己创建的替代工具箱页面覆盖该默认设置。例如，您可能有自己的UML ::类工具箱页面版本，您希望在每次打开类图时显示，当您的技术处于活动状态时。

注记要让默认工具箱页面被您的MDG 技术中的自定义工具箱页面覆盖， MDG 技术必须设置为 活动”。 （ ”特定>技术>管理技术”，然后选中您的MDG 技术名称对应的复选框，然后单击 设置活动”按钮。 ）

访问

要将系统默认工具箱替换为您自己的工具箱：

使用此处概述的方法之一显示工具箱配置文件图的 属性”对话框，并显示 常规”选项卡。

然后，在 注记”字段中键入 RedefinedToolbox 子句。

例如：

类类的结构元素；

这表明此配置文件定义的工具箱将系统工具箱UML ::类替换为所有UML类图的默认工具箱。

功能区	设计>图表>管理>属性>常规
上下文菜单	工具箱配置文件图表右击|属性|一般的

可被覆盖的系统默认工具箱页面名称

- UML ::活动
- UML ::类
- UML ::通讯
- UML ::部件
- UML ::复合
- UML ::部署
- UML交互
- UML ::元模型
- UML ::物件
- UML ::配置文件
- UML ::状态
- UML ::时序
- UML ::用例
- 扩展::分析
- 扩展::自定义
- 扩展::数据建模
- 扩展::维护
- 扩展::需求

- 扩展::用户界面
- 扩展::WSDL
- 扩展::XMLSchema

工具箱页面中使用的元素

当您为您的MDG 技术创建工具箱页面时，您可以合并标准UML元素和通过扩展UML元素创建的新元素。您在工具箱配置文件中定义要使用的元素。该表列出了您用来识别的名称：

- 要包含在工具箱页面中的标准元素或
- 您正在扩展的标准元素以定义要包含在工具箱页面中的新元素

您在工具箱页面构造型元素中列出的每个名称前面都有命名空间UML ::。括号中的文本表示默认工具箱页面中显示的标签名称，这与UML :: 语句文本有任何不同。

工具箱页面定义的元素名称

- 行动
- 行动销
- 活动
- ActivityFinal（终点）
- ActivityInitial（初始）
- 活动参数
- ActivityPartition (分区)
- 活动区域 (区域)
- 参与者
- 工件
- 关联元素（关联）
- （用于使用边界）
- 节点(中央缓冲区节点)
- 更改
- 选择
- 类
- 协作
- CollaborationOccurrence (协作应用)
- 注解(注记)
- 部件
- 约束
- 数据存储
- 决策
- 部署规范（部署部署规范）
- 设备
- 图表（图例）
- 图表（图表注记）
- DocumentArtifact（文档工件或文档）
- 实体（资料）
- 实体对象（实体）
- 入口点（入口）

- 枚举
- 异常处理程序（异常）
- 执行环境（执行环境）
- 扩展区域
- 出口点（出口）
- 特征
- FinalState（终点）
- FlowFinalNode（流终点）
- ForkJoinH（分叉/汇合-水平）
- ForkJoinV (分叉/汇合- Vertical)
- 门（图表门）
- GUIElement (用户界面控件)
- HistoryState（历史）
- 超链接
- InformationItem（信息项）
- 初始状态（初始）
- 交互
- 交互片段（片段）
- InteractionState（状态/延续）
- 接口
- 可中断活动区域
- 问题
- 连接点
- 生命线
- 合并节点（合并）
- MessageEndPoint（端点或信息端点）
- MessageLabel (信息标签)
- 元类
- 节点
- 物件
- 对象边界（边界）
- 对象控件（控件）
- 对象实体（实体）
- 包
- 包装组件
- 部件
- 端口
- 原始
- 原始类型
- 进程
- 配置文件
- 提供接口（曝露接口）

- 接收事件（接收）
- 需求
- 边界（存在）
- 控件(控件)
- 健壮实体（实体）
- 屏幕
- 发送事件（发送）
- 序列边界（边界）
- 序列控制（控件）
- SequenceEntity (实体)
- 信号
- 状态
- 状态机（状态机）
- 状态生命线(状态生命线)
- 构造型
- 结构化活动（结构活动）
- 同步状态（同步）
- 库表
- 终止
- 测试用例（测试用例）
- 文本
- 用例（用例）
- 边界（存在）
- 价值时间线（价值生命线）

注记

- 您还可以识别标准或扩展的UML连接器以添加到工具箱页面定义中
- 当元素项部署在MDG工具箱页面中时，您还可以同步从它们创建的所有元素的工具标记值和约束

工具箱页面中使用的连接器

当您为您的MDG 技术创建工具箱页面时，您可以合并标准UML连接器和通过扩展UML连接器创建的新连接器。您可以在工具箱配置文件中定义要使用的连接器。工具箱页面定义表的工具箱连接器名称列出了您用于标识的名称：

- 包含在工具箱页面中的标准连接器或
- 您正在扩展的标准连接器以定义要包含在工具箱页面中的新连接器

您在 "工具箱页面构造型元素"中列出的每个名称前面都有命名空间UML ::。括号中的文本表示默认工具箱页面中显示的标签名称，这与UML :: 语句文本有任何不同。

工具箱页面定义的连接器名称

- 抽象
- 聚合（聚合）
- 组装
- 关联(Associate)
- 关联类（关联类）
- 调用(调用)
- CommunicationPath (通讯路径)
- 组合（复合）
- 连接器
- 控制流（控件流）
- 代表
- 依赖
- 部署
- 扩展
- 概括（继承概括
- 信息流（信息流）
- InterruptFlow（中断流）
- 调用
- 显现
- 信息
- 嵌套
- NoteLink（注记连接）
- ObjectFlow (物件流)
- 发生
- PackageImport（包导入）
- PackageMerge（包合并）
- 先于
- ProfileApplication（应用程序）
- 实现（实现或实现）
- 递归

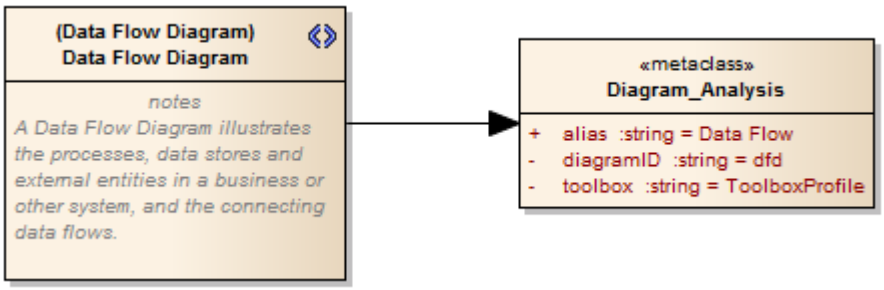
- 重新定义
- 表示
- 代表
- 捆绑(角色捆绑)
- SelfMessage（自我信息）
- 替代
- TagValAssociation（标记值）
- TemplateBinding (模板捆绑)
- TraceLink (跟踪)
- 转移
- UCExtend（扩展）
- UCInclude（包括）
- 用途
- 用例链接（使用）

注记

- 您还可以识别标准或扩展的UML元素以添加到工具箱页面定义中

创建自定义图表Profiles

当您开发MDG 技术时，可以创建扩展图表类型并将它们作为自定义图表Profiles在您的MDG 技术中。例如，您可以创建一个图表图形配置文件分析，将 DFD 图定义为内置图的扩展，如下所示：



创建扩展图表类型

节	行动
1	创建一个配置文件，与要包含的MDG 技术同名；例如，SysML。 此配置文件自动包含一个子类图。根据您打算创建多少新图表类型，您可以定义： • 一个子图上的一种图类型 • 一个图表上有多种图表类型，或 • 在几个图表上分组的几种图表类型 在第三种情况下，创建您需要的任何其他子类图。图表名称不必反映技术名称。
2	打开子类图并创建构造型元素，为其命名自定义图类型；例如，块定义。 同样在构造型元素的 属性”对话框中，在 注记”字段中，键入图表用途的简要说明。 当部署技术并创建此自定义类型的图表时，此描述将显示在 新建图表”对话框的右下角。
3	创建一个元类元素并为其指定所选内置图表类型的名称，并带有前缀 Diagram_。 例如 Diagram_Logical 自定义类图类型，或 Diagram_Use Case 自定义用例图类型。
4	将扩展连接器从构造型元素拖到元类元素。
5	单击 Diagram_xxxx 元类元素，按 F9 并创建以下部分或全部属性，以设置自定义图表类型的属性： • 别名： string =类型（其中类型将出现在图表标题栏上的 图表”一词之前；例如， 块图表”） • diagramID： string = abc（其中 abc 是将出现在图表框架标签中的图表类型） • 工具箱： string = ToolboxName（其中 ToolboxName 是每次打开此类型的图表时自动打开的工具箱工具箱配置文件的限定名称，格式为 TechID::ToolboxName”） • toolboxPage： string = 形式为 PageName=1;”的状态值列表（其中 PageName 是扩展 ToolboxPage 的构造型元素的名称；如果此string不为空，则所有值为 1 ”的工具箱页面将展开，所有其他工具箱页面将折叠） • frameString: string = FrameFormatString （其中 FrameFormatString 是一个string ，包含用于定义框架标题的替换宏，可以带有或不带有附加分隔符（如 ()）；可以使用的宏有： -#DGMALIAS# -#DGMID# -#DGMNAME# -#DGMNAMEFULL#

	- #DGMOWNERNAME# - #DGMOWNERNAMEFULL# - #DGMOWNERTYPE# - #DGMSTEREO# -#DGM类型# • 泳道： string = Lanes=2;Orientation=Horizontal;Lane1=Title1;Lane2=Title2;（其中 Lanes 可以是任意值，但 Lane<n> 值的数量必须等于 Lanes 的值；Orientation 可以省略，在这种情况下泳道默认为垂直） • styleex： string = 一个或多个值范围 • pdata： string = 一个或多个值 • showForeign: string = 1
6	根据您在步骤1中采用的配置文件包组织，以及您是否需要任何其他 Stereotype-Metaclass元素对，在此图或另一个子图上重复步骤 2-5。
7	使用最适合您设置的配置文件包组织的方法将图表保存为图表配置文件。
8	将图形配置文件添加到MDG 技术中使用的图表文件中。

内置图表类型

在自定义Enterprise Architect以更好地满足您的需求时，您可以创建一个配置文件：

- 重新定义在新组合元素下创建的内置子图的类型
- 定义快速链接器菜单提供一种连接器类型的内置图表的类型，或
- 扩展内置图表类型以创建自定义图表类型

在每种情况下，您都需要提供您正在使用的每种内置图表类型的准确名称；这些名称是：

- 活动
- 分析
- 协作
- 部件
- 复合结构
- 风俗
- 部署
- 交互概览
- 逻辑（用于类图）
- 物件
- 包
- 序列
- 状态图
- 定时
- 用例（注记两个词之间的空格）

属性值 - styleex & pdata

创建图表配置文件时，您可以使用 pdata 和 styleex 属性定义使用该配置文件创建的图表的一系列特征。如果其中一个属性同时定义多个特征，则将这些值放在一个用分号分隔的string中；例如：

HideQuals=0;AdvanceElementProps=1;ShowNotes=1;

访问

选择元类元素，然后显示 属性"对话框并定义或更新属性 "styleex"或"pdata"。

将属性类型指定为 "string"，然后在 初始值"字段中指定您需要的图表特征。

使用这些方法中的任何一种来显示 属性"对话框。

功能区	设计>元素>编辑器>属性
上下文菜单	在浏览器窗口或图表中|右键单击元类元素|特征|属性
键盘快捷键	F9

样式： string =

- AdvancedConnectorProps=1;（显示连接器属性字符串）
- 高级元素属性=1；（显示元素属性string ）
- AdvancedFeatureProps=1;（显示特征属性string ）
- AttPkg=1；（显示包可见类成员）
- DefaultLang=语言；（设置图表的默认语言；语言可以是 C++ 或Java等内置语言之一，也可以是自定义语言）
- 排除RTF=1；（从生成的报告中排除图表图像）
- 手绘=1；（应用手绘模式）
- HideConnStereotype=1;（隐藏连接器构造型标签）
- 隐藏质量=0；（显示限定符和可见性指标）
- NoFullScope=1；（隐藏完全范围的元素名称，例如"ParentClass::ChildClass"将显示为 "ChildClass"）
- SeqTopMargin=50；（设置顺序图上序列的高度）
- 显示列表=1；（使图表直接在图表List 中打开）
- 显示列表=2；（使图表直接在甘图表图视图中打开）
- 显示列表=3；（使图表直接打开到规范管理器中）
- 显示列表=4；（使图表直接打开到关系矩阵）
- 显示维护=1；（显示元素维护区）
- ShowNotes=1；（显示元素注记隔间）
- ShowOpRetType=1;（显示操作返回类型）
- 显示测试=1；（显示元素测试区）
- SuppConnectorLabels=1;（抑制所有连接器标签）

- 抑制括号=1；（抑制无参数操作的括号）
- TConnectorNotation=选项；（其中 Option 是UML 2. 1 、IDEF1X 或信息工程之一）
- TExplicitNavigability=1;（显示不可导航的连接器末端）
- 可见属性详细信息=1；（在图表上显示属性详细信息）
- 白板=1；（应用白板模式）

数据： **string =**

- 隐藏属性=0；（显示元素隔间）
- 隐藏Estereo=0；（在图中显示元素的刻板印象）
- 隐藏操作=0；（显示元素操作区）
- 隐藏父母=0；（以显示图表中元素的其他父级）
- 隐藏道具=0；（显示属性方法）
- 隐藏关系=0；（表示关系）
- 隐藏立体声=0；（显示属性和操作原型）
- 操作参数 =3；（显示操作参数）
- ShowCons=1；（显示元素约束）
- 显示图标=1；（使用刻板印象图标）
- 显示请求=1；（显示元素需求隔间）
- 显示SN=1；（显示序列注记）
- 显示标签=1；（显示元素标记值）
- 补充CN=0；（显示合作编号）
- 使用别名=1；（使用图表中的别名或元素，如果可用）

设置技术元素图片

当您定义可在您的技术中使用的元素时，您可能希望使用图形图像来表示这些元素，这些图像将显示在用户通过该技术创建的图表上，当它被部署在用户的模型中时。

捕获图像以表示MDG 技术元素

节	行动
1	显示图像管理器，并使用加新按钮，将合适的图像从其源位置导入到MDG 技术开发模型中。
2	设计并创建一个构造型(UML)**配置文件**，**其中包含**（如果合适的话）技术所拥有的每个元素或连接器的构造型定义。 这些原型定义可以包含形状脚本，这些脚本又包含导入的图像。
3	设计并创建一个带有原型元素的工具箱配置文件，其中包含每个元素或连接器的属性，这些元素或连接器可以从工具箱中拖放到图表上。 这些属性标识技术元素或连接器的名称、任何修改构造型（可能包含所需的图像）以及技术object**所基于的**UML或扩展元素或连接器。 例如： SysML::块(UML ::类) • SysML是技术配置文件 • UML ::类是用作基础的UML元素，并且 • 块是修改类以将其转换为 SysML块元素的构造型
4	设计并创建一个图表工具箱配置配置文件的图形配置文件。 打开工具箱配置文件中定义的类型的图表时，它会依次打开图表配置文件配置文件的一组工具箱页面。
5	根据需要创建或更新技术，将UML配置文件、图表配置配置文件、工具箱配置文件和图像文件添加到开发模型中的技术中。
6	酌情部署该技术。 当用户将技术应用到他们自己的模型中，并在该技术下创建图表时，他们在图表上创建的元素应该由您在创建技术时分配给这些元素的图像来表示。

注记

- 建议如果您创建包含MDG 技术图像的形状脚本（步骤 2），则应使用完全限定的图像名称，以避免与其他技术中使用的图像冲突
- 您可能会在这些步骤中多次来回工作，在确定对它们的要求时添加对象

定义验证配置

使用 模型验证配置”对话框，您可以选择在用户执行验证时执行和不执行哪些验证规则集。

无需手动执行此配置，并且每次启动Enterprise Architect并且已将不同的技术设置为活动时都可能必须为您的技术更改设置，您可以在您的技术的MDG 技术选择 (MTS) 文件中定义配置设置.

访问

在您在工作中使用的任何文件浏览器中找到并打开 .MTS 文件。您按照这两个库表中的指示编辑文件，然后保存文件。

白名单

要将一组规则指定为白名单（即，任何添加到此列表的内容都打开），请在文本编辑器中打开您的 MTS 文件，然后将此 <ModelValidation> 块复制并粘贴到 < MDG内的顶层.Selections> 块：

<模型验证>

<RuleSet name="BPMNRules"/> <!-- Project.DefineRuleCategory 调用中定义的规则集 ID -->

<RuleSet name="MVR7F0001"/> <!-- 注意你也可以开启/关闭系统规则！ -->

</模型验证>

确保规则集 ID 不包含任何空格。

黑名单

要将一组规则指定为黑名单（即，任何添加到此列表的内容都已关闭），请在文本编辑器中打开您的 MTS 文件，然后将此 <ModelValidation> 块复制并粘贴到 < MDG的顶层.Selections> 块：

<ModelValidation isBlackList="true">

<RuleSet name="BPMNRules"/>

<RuleSet name="MVR7F0001"/>

</模型验证>

在此示例中，“BPMNRules”是在 Project.DefineRuleCategory 调用中定义的规则集 ID - 请参阅项目类了解详细信息。 “MVR7F0001”是一个内置的规则集。当您激活适当的技术时，将应用这些验证选项。全局（默认）技术已打开所有规则。

合并模型Builder模板

当用户在其项目中创建模型时，他们可以从 模型生成器”对话框中显示的一系列系统提供的模型模板中选择要开发的模型类型。您还可以开发自定义模型模板并通过MDG 技术将其添加到此列表中。

访问

您可以直接编辑 .mts 文件，使用您使用的任何文件浏览器来定位和打开该文件。

将模型模板添加到MDG 技术

节	行动
1	创建一个包，其中包含您要在模型模板中提供的所有子包、图表、元素、注记和信息链接。 请参阅 EAExample.eap模型以了解您可能包含的内容，或从标准模板创建模型并查看生成的内容。 作为模型模板，包需要是自包含的，并且不包含任何依赖项或其他指向包外元素的链接。
2	将您的包导出到 XML。 如果您希望模板的支持文档显示在模型生成器对话框的右侧面板中，请在与 XML 文件相同的目录位置创建一个包含此文档的 .rtf 文件。.rtf 文件的文件名也必须与 XML 文件相同。建议您在模型中的文档工件元素中创建 .rtf 文件，然后将文件（ 文档-编辑 >文件> 另存为 (导出到文件)”功能区选项）导出到模式XML 文件的位置。这样可以将文档保留在您的开发模型中。
3个	要允许每个技术有多个自定义类别：在文本编辑器中打开您的 .mts 文件并向 <Technology>元素添加两个附加属性： • 类别列表，其中 包含以逗号分隔的自定义类别名称列表，或单个内置类别的名称（例如 业务”） • 类别映射， 其中包含形式为 组名称1=类别名称A ;组名称2=类别名称B ;”的选项对列表，依此类推；类别名称必须全部在 ategoryList”中
4个	在 .mts 文件中创建对 XML 文件的引用；在文本编辑器中打开您的 .mts 文件，然后将此 <ModelTemplates>元素复制并粘贴到 < MDG .Selections> 块内的顶层： <模型模板> <模型name="模板名称" 位置="MyTemplatePackage.xml" 默认=”是” 图标 = “34” isFramework=" false "/> </模型模板> 您可以在 .mts 文件中的 <ModelTemplates>元素中包含任意数量的 <Model> 元素，每个模型模板一行。 <Model>元素内的属性具有以下含义：

	• name: 在模型构建器对话框中显示的模型模板的名称，当您选择模型蓝图或执行 模型构建器（模式库）”菜单选项时显示该名称 • 位置：包含模型模板包导出的 XML 文件的路径，相对于Enterprise Architect安装路径中模型模式目录的位置： - 如果 XML 文件直接位于模型模式目录中，则 路径仅包含文件名（例如，MyPattern1.xml） - XML 可以与MDG 技术XML 文件位于同一文件夹中，RTF 文件位于同一文件夹中 - 如果你将所有文件放在模型模式的子目录中， 路径包含目录名称（例如，MyTechnology\MyPattern2.xml） - 您也可以指定固定路径（例如，C:\Program Files\MyTechnology\MyPattern3.xml） • 图标：包含Enterprise Architect基本图标列表的索引；要显示适当的视图图标，请使用以下值之一： - 29 =用例 - 30 = 动态 - 31 =类 -32=部件 - 33 = 部署 - 34 = 简单 • isFramework：定义模型模式的可能用途；有三个可能的值： - isFramework="true" - 永远不会剥离 GUID；该模式旨在 作为任何模型的可重复使用包 - isFramework= 可选” - 剥离 GUID 的提示；模式是 旨在作为可重复使用的包，但用户可以选择 - isFramework=" false " - 总是删除 GUID（默认，如果没有 所述）；该模式可以在 一个模型 • groupName：如果指定了多个自定义类别，此属性用于引用此模式属于哪个类别。
5个	使用编辑的 MTS 文件重新生成MDG 技术。

添加导入/导出脚本

在Enterprise Architect中，可以从一系列 XMI 和 XML 格式的外部文件导入包和导出（或发布）包。您还可以将此功能合并到您的MDG 技术中，添加一个包含您自己的可扩展样式表语言 (变换) 的脚本以在文件格式之间进行转换。

合并一个导出（发布）脚本

节	描述
1	在您首选的编辑器中，创建一个 XSLT 以将源格式（如 发布模型包”对话框中列出的）转换为您正在生成的目标格式。
2	在Enterprise Architect中，打开 Scripter 窗口并在您首选的脚本引擎下创建一个脚本作为普通脚本。 将 XSLT 剪切并粘贴到脚本编辑器中。
3	在MDG 技术创建向导中将脚本添加到您的MDG 技术中。
4	对您需要的技术 .mts 文件进行任何添加，然后再次使用MDG 技术创建向导来完全生成技术 .xml 文件。 在文本编辑器中打开技术 .xml 文件（不是 .mts 文件）并找到 <脚本部分。
5	编辑 <脚本行以设置适当的名称、类型和语言： • 名称是要在 发布>模型交换”功能区面板中显示的技术选项文本 • *type*是单词 Publish-”，后跟要导出的文件格式的名称，如 发布模型包”对话框中所列 • 语言是 XSLT 例如： <脚本 名称= 你的技术” type="发布-UML 1 (XMI 1)" 语言="XSLT"> <内容 xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="bin.base64"> </内容> </脚本>
6	保存MDG 技术.xml 文件，并将其部署到您的系统上。

合并导入脚本

节	描述
1	在您首选的编辑器中，创建一个 XSLT 以将源格式转换为目标 XMI 格式。

2	在Enterprise Architect中，打开 Scripter 窗口并在您首选的脚本引擎下创建一个脚本作为普通脚本。 将 XSLT 剪切并粘贴到脚本编辑器中。
3	在MDG 技术创建向导中将脚本添加到您的MDG 技术中。
4	对您需要的技术 .mts 文件进行任何添加，然后再次使用MDG 技术创建向导来完全生成技术 .xml 文件。 在文本编辑器中打开技术 .xml 文件（不是 .mts 文件）并找到 <脚本部分。
5	编辑 <脚本行以设置适当的名称、类型和语言： • 名称是要在Enterprise Architect 发布>模型交换>导出”功能区选项中显示的技术选项文本 • *type*是单词 Ímport-”，后跟要生成的 XMI 文件格式的名称，如 发布模型包”对话框中所列 • 语言是 XSLT 例如： <脚本 名称= 你的技术” type="导入-UML 1 (XMI 1)" 语言="XSLT"> <内容 xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="bin.base64"> </内容> </脚本>
6	保存MDG 技术.xml 文件，并将其部署到您的系统上。

注记

- 在 XSLT 1 .0 中创建脚本的内容

部署MDG 技术

MDG 技术可以通过以下两种方式之一部署：作为 .xml 文件或从插件

.

从 .xml文件部署

要将您的技术部署为文件，您有多种选择：

- 将技术 .xml 文件导入 %APPDATA%\ Sparx Systems \EA\导入文件夹（供您个人使用）
- 将技术导入文件导入浏览器窗口的 资源”选项卡（供所有项目用户访问）
- 将文件复制到Enterprise Architect安装目录下的 MDGTechnologies 文件夹（默认为C:\Program Files\Sparx Systems\EA ）；当您重新启动Enterprise Architect时，您的MDG 技术已部署
- 将文件复制到文件系统中的任何文件夹，包括网络驱动器 - 使用”特定>技术>管理技术”功能区选项，单击高级按钮并将文件夹添加到 技术”路径；这种部署方法使您能够快速轻松地将技术部署到 LAN 上的所有Enterprise Architect用户
- 将文件上传到 Internet 或 Intranet 位置：使用”特定>技术>管理技术”功能区选项，单击高级按钮并将 URL 添加到 技术”路径；这种部署方法使您能够快速轻松地将技术部署到更广泛的Enterprise Architect用户组

从插件

部署插件

从插件

部署您的技术插件

，您必须编写一个 EA_OnInitializeTechnologies函数。这个例子是用 VB.Net 编写的：

公共函数EA_OnInitializeTechnologies(ByVal存储库As EA.Repository) 作为物件

EA_OnInitializeTechnologies = 我的.资源.MyTechnology

结束函数

形状脚本

您最初在建模中使用的元素和连接器在形状、颜色和标签方面符合标准UML表示法。但是，您可以扩展标准对象以创建新对象，并使用形状脚本自定义这些新对象的外观，以定义您想要施加到默认或主形状上的确切特征。您使用专用脚本语言创建形状脚本脚本，以定义元素或连接器的新形状、方向、颜色和标签。每个脚本都与一个原型相关联，每个具有该原型的元素或连接器都将采用由形状脚本定义的形状脚本。

如果要标准化外观以应用于许多元素，可以将形状脚本脚本附加到MDG 技术构造型配置文件中的构造型元素的属性。

如果您已将形状脚本应用于某些元素和/或连接器，但不想在特定图表上显示这些形状脚本，您可以使用图表的“属性”对话框关闭该图表上的形状脚本的显示。

开始形状脚本

由于形状脚本与构造型相关联，因此您可以通过 "UML类型"对话框的 构造型"选项卡来定义它们；每个构造型都可以有一个形状脚本。设置形状脚本的过程非常简单但非常灵活。

访问

功能区	设置 >参考> UML类型 >构造型

形状脚本进程

节	行动
1	从对话框右侧的列表中选择要将形状脚本附加到的构造型。 您选择一个现有的构造型，但如果没有合适的构造型，您可以创建一个新的构造型，一旦保存，它就会显示在列表中并且可以选择。
2	在 覆盖外观"面板中，选择 形状脚本"单选按钮，然后单击 分配"按钮。 显示形状编辑器。
3	类型或将脚本复制到编辑窗口中。 要在 预览"面板中审阅形状，请单击 刷新"按钮。
4	如果您定义了一个复合形状脚本形状（带有装饰和标签的主要形状，或单独的部分，例如具有源端和目标端形状的连接器），请单击 下一个形状"按钮以翻阅形状的组件，在 预览"面板。
5	写完形状脚本后，单击确定按钮返回 构造型"选项卡。 然后单击保存按钮以保存形状脚本及其对构造型的分配。
6	将适当的标准UML元素或连接器拖放到您的图表中。该object将是您选择作为构造型的 基类"的类型。 右键单击object并选择 属性"选项。 在 属性"对话框中，单击 构造型"下拉箭头，选择您创建的构造型并单击 确定"按钮。 object的形状现在反映了形状脚本给构造型的形状。

注记

- 使用元素形状脚本来修改元素的外观会使一些正常的 外观"上下文菜单选项变得多余，因此它们将被禁用
- 无法修改或覆盖MDG 技术中定义的类型的形状脚本
- 形状脚本不支持字体选择，因为最佳用户体验是通过允许用户自己设置字体来实现的
- UML定义了通过Profiles扩展UML语法的标准机制；出于这个原因，形状脚本不能独立于原型应用于任何元

素

- 形状脚本不能用于使用贝塞尔线样式的连接器
- 形状脚本目前不支持：
  - 循环结构
  -字符串操作
  - 算术操作
  - 变量声明

形状编辑器

当您通过“UML类型”对话框的“构造型”选项卡创建形状脚本脚本时，您可以使用形状编辑器编写脚本。这提供了功能代码编辑器功能，包括智能感知形状脚本公共属性功能。

访问

功能区	设置>参考> UML类型>构造型：（选择或指定构造型）：形状脚本+分配

编辑器选项

选项	行动
格式	点击下拉箭头，选择形状脚本版本（目前只有形状脚本1 .0可用）。
导入	单击此按钮可从文本文件 (.txt) 中导入形状脚本。将显示A文件浏览器，您可以通过它找到要导入的文件。 找到并选择文件后，单击“打开”按钮将脚本导入编辑面板。
导出	单击此按钮可将形状脚本导出到文本文件。将显示A文件浏览器，您可以通过该浏览器指定要导出到的文件。 确定文件后，单击“保存”按钮以完成导出并返回到形状编辑器。
<编辑面板>	类型此面板中的脚本命令。
确定	单击此按钮可退出形状编辑器。 要保存您的形状脚本，请单击“构造型”选项卡上的保存按钮。
下一个形状	如果您有一个由不同组件组成的形状，请单击此按钮以在“预览”面板中旋转多个形状定义。
刷新	单击此按钮可解析您的脚本并在预览窗口中显示结果。

编写脚本

要为元素或连接器创建替代表示，您可以编写一个定义表示的大小、形状、方向和颜色的形状形状脚本。 A形状脚本包含许多用于定义形状不同方面的部分；对于一个元素，这些包括：

- 主要object
- 标签
- 装饰（例如，一个 Document元素可能包含一个描述文档的图标）

对于连接器，这些部分包括：

- 主要object
- 形状源
- 形状目标
- 标签

形状脚本在使用默认 (UML) 表示的基础上运行，除非脚本包含替代定义。那是：

- 如果你有一个只包含一个装饰的形状脚本，这个装饰会被添加到正常绘制的object之上
- 如果你有一个空的形状例程，它会覆盖默认值；因此，空白的 形状标签”会阻止为具有它们的元素创建正常的浮动文本标签

你也可以使用 C 风格的注释来注释你的脚本；例如：

// C 风格单行注释

/* 多行

支持评论 */

脚本不区分大小写： 形状” 形状”相同。

脚本结构

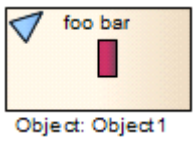
Layout	Description

示例元素布局脚本	主要形状
	{
	// 绘制object
	}
	形状标签
	{
	// 绘制浮动文本标签
	}
	装饰 <标识符>
	{
	// 在object内部绘制一个 16x16 的装饰
	}
	
	< 标识符 > string是一个字母数字单词。

连接器示例布局脚本	主要形状 {

	// 划清界线
	}
	形状目标
	{
	// 在目标端绘制形状
	}
	形状源
	{
	// 在源端绘制形状
	}
	标签 <位置标签>
	{
	// 定义标签的文本
	}
	
	<positionLabel> string可以是以下任何一种：
	• 左上标签
	• 左下标签
	• 中顶标签
	• 中间底部标签
	• 右上标签
	• 右下标签

子形状	形状可以A子形状，子形状必须在主形状脚本之后声明，但从方法命令中调用。
	这是声明排序的示例：
	
	主要形状
	{
	// 初始化属性 - 这些必须在绘制命令之前
	noshadow = "真";
	h_align = "中心";
	
	//绘图命令（方法）
	矩形（0,0,100,100）；
	println ("foo bar");
	
	// 调用子形状
	addsubshape ("红色", 20, 70);
	
	// 子形状的定义
	形状红色
	{
	setfillcolor (200,50,100);
	矩形（50,50,100,100）；

	}
	}
	
	//标签的定义
	形状标签
	{
	setOrigin ("SW",0,0);
	println("物件：#NAME#");
	}
	
	//装饰的定义
	装饰三角形
	{
	//划一个三角形装饰
	开始路径（ ）；
	移动到（ 0,30 ）；
	线托（ 50,100 ）；
	lineto (100,0);
	结束路径（ ）；
	设置填充颜色（ 153,204,255 ）；
	fillandstrokepath();
	}
	
	此脚本产生的形状是：
	 

申报顺序	形状可以由属性声明、方法/命令调用和子形状定义组成，它们必须按该顺序出现；也就是说，属性声明必须出现在所有方法调用之前，子形状定义必须出现在最后。

形状属性

使用形状脚本定义形状时，使用属性定义该属性的属性。属性包括：

- 形状相对于图表和其他元素的位置
- 形状组件相对于形状边框的位置
- 形状是否有用户可编辑的区域
- 形状是否可以调整大小、缩放、旋转或停靠

属性语法

属性 =“值 ”; ”

示例

主要形状
{
//初始化属性——必须在绘制命令之前
noshadow = "真";
h_align = "中心";
//绘图命令
矩形（0,0,100,100）；
println ("foo bar");
}

属性

Attribute Name	Description
大胆的	string 描述：如果您希望当前形状或子形状中的所有打印命令都以粗体显示，请设置为True 。 有效值： True或False （默认 = False ）
斜体	string 描述：如果您希望当前形状或子形状中的所有打印命令都以斜体显示，请设置为True 。 有效值： True或False （默认 = False ）
bottomAnchorOffset	（ int ， int ） 当创建一个形状脚本描述嵌入元素（例如一个嵌入的端口）时，使用这个属性从它的父元素的底部边缘偏移形状。 例如：

	bottomAnchorOffset= (0,-10); 将嵌入的元素从底部边缘向上移动 10 个像素。
可停靠	string 描述：使形状默认为可停靠，以便它可以与图表上的其他元素（其他形状脚本和标准元素）对齐并连接。您无法使用 外观"菜单选项反转可停靠状态；要更改状态，您必须编辑形状脚本。 有效值：标准或关闭
可编辑字段	string 描述：将形状定义为元素的可编辑区域。 此字段仅影响元素形状，不支持线条字形。 有效值：别名、姓名、注记、刻板印象
端点Y，端点X	整数 描述：仅用于连接器的保留目标和源形状；此点确定主连接线连接到末端形状的位置。 默认值：0 和 0
固定纵横比	string 描述：设置为True以固定纵横比。如果您不想固定纵横比，请不要使用此选项。
h_Align	string 描述：根据 layoutType 属性影响打印文本和子形状的水平放置。 有效值：左、中或右
布局类型	string 描述：确定子形状的大小和位置。 有效值：leftright、topdown、border
leftAnchorOffset	（ int ， int ） 当创建一个形状脚本描述嵌入元素时（例如一个端口元素的属性描述），使用这个属性从它的父元素的左边缘偏移形状。 例如： leftAnchorOffset= (10,0); 将嵌入的元素从左边缘向右移动 10 个像素
无影	string 描述：设置为True以禁止渲染形状的阴影。 有效值： True或False （ 默认 = False ）
方向	string 描述：仅适用于装饰形状，以确定装饰在包含元素字形中的位置。 有效值：NW、 N 、NE、 E 、SE、 S 、SW、W
首选身高	描述：由边界布局类型使用 - 北和南。 用于绘制连接器的源和目标形状以确定线的宽度。

首选宽度	描述：由边界布局类型使用 - 东和西。 由 leftright layoutType 形状使用，其中可扩展性为false ，以确定它们为布局目的占用多少空间。
右锚偏移	（ int ， int ） 当创建一个形状脚本描述嵌入元素（例如一个嵌入的端口）时，使用这个属性从它的父元素的右边缘偏移形状。 例如： rightAnchorOffset= (- 10,0); 将嵌入的元素从右边缘向左移动 10 个像素。
可旋转	string 描述：设置为False以防止形状旋转。此属性仅适用于线条字形的源形状和目标形状。 有效值： True或False （默认 = True ）
可扩展的	string 描述：设置为False以阻止形状相对于关联的图表字形调整大小。 有效值： True或False （默认 = True ）
topAnchorOffset	（ int ， int ） 当创建一个形状脚本描述嵌入元素（例如一个嵌入的端口）时，使用这个属性从它的父元素的顶部边缘偏移形状。 例如： topAnchorOffset= (0,10); 将嵌入的元素从顶部边缘向下移动 10 个像素。
v_Align	string 描述：根据 layoutType 属性影响打印文本和子形状的垂直放置。 有效值：顶部、中心或底部

绘图方法

使用形状脚本创建形状时，可以使用方法定义形状的值。这些值包括以下内容：

- 形状是什么 - 一个矩形，一条线，一个球体
- 形状的大小
- 形状和边框的颜色
- 形状具有的隔间和隔间文本
- 显示在形状中和形状周围的文本和标签
- 形状是否包含或包含捕获的图像

您可以通过按 Ctrl+Space 列出脚本中任何点的有效方法（命令）。

方法语法

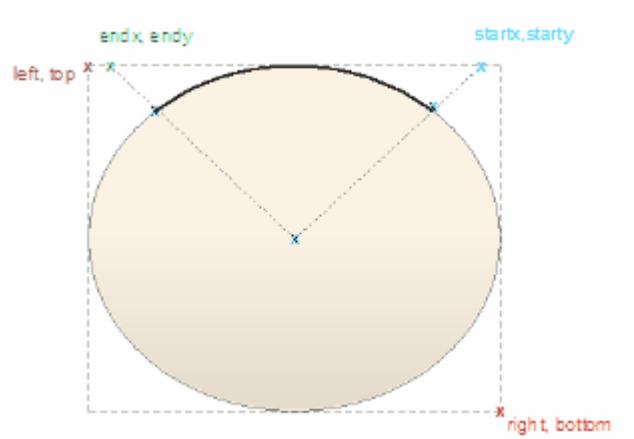
<方法名> "(" <参数列表> ")";";

示例

主要形状
{
// 初始化属性 - 这些必须在绘制命令之前
noshadow = "真";
h_align = "中心";
//绘图命令（方法）
矩形（0,0,100,100）；
println ("foo bar");
}

方法

Method Name	Description
addsubshape（ string shapename（ int宽度，int高度））	添加一个必须在当前形状定义中定义的名为“shapename”的子形状。
附加隔间文本（ string ）	将额外的字符串附加到隔间的文本中。 添加文本的隔间取决于在使用“appendcompartmenttext”之前使用“setcompartmentname”设置的隔间名称。 必须调用此方法才能显示隔间。
弧（ int左， int顶部，int右， int底部， int起点x， int起点， int终点x，	绘制一个椭圆逆时针圆弧，椭圆的范围在左、上、右和下。 圆弧的起点由椭圆与椭圆中心到该点的连线（startingpointx，startingpointy）

int终点）	的交点定义。 弧的终点类似地由椭圆与从椭圆中心到该点的线（endpointx，endpointy）的交点定义。 例如： 弧（0, 0, 100, 100, 95, 0, 5, 0）； 
arcto（ int左， int顶部， int右， int底部， int起点 x， int起点， int终点 x， int终点）	至于圆弧方法，只不过是从当前位置画一条线到圆弧的起点，然后将当前位置更新到圆弧的终点。
bezierto（ int controlpoint1x， int controlpoint1y， int controlpoint2x， int controlpoint2y， int endpointx， int endpointy）	绘制贝塞尔曲线并更新画笔位置。

defSize（ int宽度， int高度）	设置元素的默认大小。
	这可能出现在 IF 和 ELSE 子句中，每个子句具有不同的值，并导致元素在每次值更改时自动调整大小。
	if(HasTag("水平","true"))
	{
	defSize(100,20);
	矩形（0,0,100,100）；
	}
	else
	{
	defSize(20,100);
	矩形（0,0,100,100）；
	}
	本示例在每次更改标签值'标记值'时将形状设置为指定的默认大小。
	设置后，Alt+Z 还会将形状调整为定义的尺寸。
	int width 和int height 的最小值都是 10。

绘制形状（ ）	以通常的非形状脚本表示法渲染形状；随后的绘图命令叠加在本机符号上。

	仅元素形状脚本支持此方法；不支持线形状脚本。
绘制父图形（ ）	在扩展非 UML物件类型时使用。 呈现从父原型定义的形状。如果没有继承的形状脚本可用，则行为与形状脚本() 相同。
椭圆（ int left， int top， int right， int底部）	绘制一个椭圆，其范围由左、上、右和下定义。
结束路径（ ）	结束定义路径的绘图命令的序列。
填充行程路径（ ）	用当前的填充颜色填充之前定义的路径，然后用当前的笔绘制它的轮廓。
填充路径（ ）	用当前填充颜色填充先前定义的路径。
获取默认填充颜色（ ）	获取元素的默认填充颜色。这可以是所有元素的标准填充颜色，或者，如果在"首选项"对话框的 图表>外观"页面上选择了 无素使用组样式"选项，则为元素类型定义的默认填充颜色。
获取默认线颜色（ ）	获取元素的默认线条颜色。这可以是所有元素的标准线条颜色，或者，如果在"首选项"对话框的 图表>外观"页面上选择了 无素使用组样式"选项，则为元素类型定义的默认线条颜色。

隐藏标签（ string标签名称）	隐藏由 labelname 指定的标签，其中 labelname 是以下值之一：
	• 中间标签
	• 中底标签
	• 左上标签
	• 左下标签
	• 右上标签
	• 右下标签
	注记：通过将指定的标签设置为隐藏来实现此功能。对脚本的任何后续更改都不会再次显示标签。
	抑制标签的推荐方法是覆盖该形状。例如，抑制默认构造型标签：
	形状中间底部标签
	{
	打印（""）；
	}

图像（ string imageId， int左， int顶部， int右， int底部）	在图像管理器中绘制名称为 imageId 的图像。 图像必须存在于使用原型的模型中；如果模型中不存在，则必须将其作为参考数据导入或从技术文件中选择。 如果图像在技术文件中，则它的文件名格式应为 <technology ID>::<imagename>.<extension>。
lineto（ int x， int y）	从当前光标位置到 x 和 y 指定的点绘制一条线，然后将笔光标更新到该位置。（另见注记部分。）
移动（ int x， int y）	将笔光标移动到 x 和 y 指定的点。
多边形（ int centerx， int centery， int	绘制一个正多边形，中心在点 (centerx, centery)，边数为 numberofsides。

numberofsides， int radius，float rotation ）	
打印（ string文本 ）	打印指定的文本string 。 您不能更改此文本的字体大小或类型。
printifdefined(string propertyname, string truepart(, string falsepart))	如果给定属性存在并且具有非空值，则打印 “truepart”，否则打印可选的 “falsepart”。 您不能更改此文本的字体大小或类型。
println(string文本)	将一行文本附加到形状和换行符。 您不能更改此文本的字体大小或类型。
printwrapped（ string文本 ）	打印指定的文本string ，如果文本比其包含的形状宽，则将其包裹在多行中。 您不能更改此文本的字体大小或类型。
矩形（ int左， int顶部， int右， int底部 ）	绘制一个矩形，其范围为左、上、右、下。值是百分比。
圆形矩形（ int左， int顶部， int右， int底部， int abs_cornerwidth， int abs_cornerheight ）	绘制一个圆角矩形，其范围由左、上、右和下定义。 角的大小由 abs_cornerwidth 和 abs_cornerheight 定义；这些值不随形状缩放。
设置附件模式()	定义连接器如何连接到元素形状，在元素边上的任意点（参数 “normal”）或在每个边的中心点（参数 “diamond”），具体取决于 “矩形表示法”选项的设置。请参阅示例脚本帮助主题。
设置隔间名称（ string ）	将隔间名称设置为提供的string 。 此方法必须在调用 appendcompartmenttext 之前使用；在调用 appendcompartmenttext 之后调用它会清除已添加到隔离专区的任何文本。
设置默认颜色（ ）	将画笔和画笔颜色恢复为默认设置，或恢复为用户定义的颜色（如果可用 ）。
setfillcolor（ int red， int green， int blue ）或 setfillcolor（颜色新颜色 ）	设置填充颜色。 您可以通过定义 RGB 值或使用任何颜色查询返回的颜色值来指定所需的颜色，例如： GetUserFillColor() 或 获取用户边框颜色（ ） 在所有情况下，setfillcolor 优先于适用于元素的任何颜色定义。
setfixedregion（ int xStart， int yStart， int xEnd， int yEnd ）	修复连接器中可以绘制子形状的区域，以便子形状不会随着连接器线的长度或方向重新缩放。 有关示例，请参阅示例脚本主题中的 “旋转方向”脚本。
setfontcolor（ int red， int green， int blue ）或 setfontcolor（颜色新颜色 ）	设置文本string的字体颜色。 您可以通过定义 RGB 值或使用任何颜色查询返回的颜色值来指定所需的颜色，例如： GetUserFontColor() 或 获取用户填充颜色（ ）

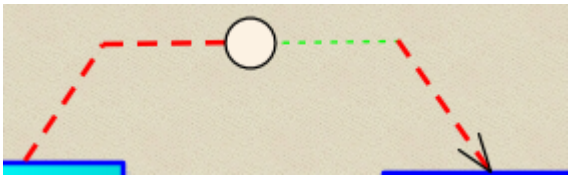
	您可以将此命令与任何文本打印命令一起使用。
setlinestyle（ string线型）	修改使用笔的命令的笔划模式。 string linestyle: 具有以下有效样式： • 坚硬的 • 短跑 • 点 • 破折号 • 点划线 • 双倍的 （另见注记部分。）
setorigin（ string relativeTo， int xOffset，int yOffset）	相对于主形状定位浮动文本标签。 • relativeTo 是N 、NE、 E 、SE、 S 、SW、W、NW、CENTER 之一 • xOffset 和 yOffset 以像素为单位，不是百分比值，可以为负数
setpen(int red, int green, int blue, int penwidth) 或 setpen(Color newcolor, int penwidth)	将画笔设置为定义的颜色并设置画笔宽度。 此方法仅适用于画线命令。它不影响任何文本打印命令。
setpencolor(int red, int green, int blue) 或 setpencolor(Color newColor)	设置画笔颜色。 您可以通过定义 RGB 值或使用任何颜色查询返回的颜色值来指定所需的颜色，例如： 获取用户填充颜色（ ） 此方法仅适用于画线命令。它不影响任何文本打印命令。
setpenwidth(int penwidth)	设置画笔的宽度。笔宽应在1到 5 之间。 此方法仅适用于画线命令。它不影响任何文本打印命令。
显示标签（ string标签名称）	显示由 labelname 指定的隐藏标签，其中 labelname 是以下值之一： • 中顶标签 • 中间底部标签 • 左上标签 • 左下标签 • 右上标签 • 右下标签
startcloudpath（ puffWidth、puffHeight、噪声）	与 startpath 类似，不同之处在于它使用云状曲线段（粉扑）绘制路径。 参数： • float puffWidth（默认 = 30），粉扑之间的水平距离 • float puffHeight（默认 = 15），烟团之间的垂直距离 • 浮动噪声（默认 = 1 .0），烟团位置的随机化
开始路径（ ）	启动定义路径的绘图命令的序列。
行程路径（ ）	用当前画笔绘制先前定义的路径的轮廓。

注记

- 如果为一条由多条线段组成的线绘制形状脚本，并为线段定义不同的线型，则除中心线段以外的所有线段都使用第一个定义的线型；中心线段使用定义的第二种线型，如图所示：

```
形状 主要的
{
    无影=真；
    // 此笔样式将被忽略，因为没有绘制任何内容。
    setpen ( 0 , 0 , 0 , 1 );
    SetLineStyle （ 实心” ）；

    // 这种笔样式将用于非中心线段，因为它是
    // 第一个用于绘图。
    设置笔( 255 , 0 , 0 , 2 );
    SetLineStyle （ 破折号” ）；
    移动到( 0 , 0 );
    lineto ( 50 , 0 );

    // 这个线型用在中间段，没有其他的，因为它
    // 不是第一个绘制的。
    setpen ( 0 , 255 , 0 , 1 );
    SetLineStyle ( "点" );
    lineto ( 100 , 0 );

    // 此线型仅用于中心段的注释。
    setpen ( 0 , 0 , 0 , 1 );
    SetLineStyle （ 实心” ）；
    setfixedregion ( 40 ,- 10 , 60 , 10 );
    椭圆( 40 ,- 10 , 60 , 10 );
}
```

具有此形状脚本A依赖连接器可能类似于以下内容：



颜色查询

在定义形状时，您可能希望保留已为基本形状定义的填充、边框和字体颜色。您可以使用颜色查询来设置颜色定义，以检索 SetPenColor 和 SetFillColor 命令的参数。这些查询可以用来代替参数。

- getUserFillColor() - 返回当前元素的用户选择的填充颜色
- getUserBorderColor() - 返回当前元素的用户选择的边框/线条颜色
- getUserFontColor() - 返回当前元素的用户选择的文本字体颜色
- getUserPenSize() - 返回当前元素的用户选择的线条粗细
- getDefaultFillColor() - 返回当前元素的默认填充颜色，而不使用应用于此元素的颜色
- getDefaultLineColor() - 返回当前元素的默认线条颜色，而不使用应用于此元素的颜色
- getStatusColor() - 返回当前元素的状态颜色；如果没有为此状态定义颜色，或者没有针对此类型显示状态颜色，则此查询将返回与状态相同的结果

例如：

主要形状

{

setfillcolor(getuserfillcolor());

setpencolor(getuserbordercolor());

矩形（0,0,100,100）；

}

注记

- 用户颜色是在基础object没有被形状脚本修改的情况下设置的颜色；它们将被定义使用 - 按优先级递减的顺序 - 格式工具栏选项， 外观”选项（F4）或 首选项”对话框（ 开始>外观>首选项>首选项”）
- 因为用户颜色是为元素定义的颜色，随后应用了构造型和形状脚本，它们不能在形状编辑器的 预览”面板中描述

条件分支

您可以使用 IfElse”语句或计算结果为True或False的查询方法将条件分支合并到您的形状脚本中。

当您使用这些条件分支语句时，您可以使用 return 命令在满足分支条件时终止脚本的执行。示例脚本主题提供了这方面的几个示例，例如 返回语句形状”脚本。

查询方法

当您在形状脚本中使用形状脚本语句时，条件通常是object具有某个标记或属性，并且可能该标记或属性具有特定值。您可以使用此处描述的两种查询方法之一设置条件语句来检查属性和值。

查询

Method	Description
boolean HasTag(string标记名， （ string标记值 ） ）	如果 'tagname' 存在且其值为非空，则 HasTag(tagname) 评估为 ' True '；否则它评估为 "false "。 如果 'tagname' 存在且其值为 'tagvalue'，则 HasTag(tagname,tagvalue) 计算结果为 ' True '。 如果 'tagname' 不存在且 'tagvalue' 为空，则 HasTag(tagname,上下文) 也将评估为 ' True '，此时将 'empty' 和 'missing' 视为具有相同的含义。

boolean HasProperty(string属性名，
（ string属性值 ） ）
	如果关联的元素具有名称为 propertyname 的属性，则计算结果为True 。
	如果提供了第二个参数 propertyvalue，则该属性必须存在，并且该属性的值必须等于属性才能使方法评估为True 。
	propertyvalue 参数可以有多个值，以逗号分隔；例如：
	if(HasProperty("类型","类,行动,活动,接口"))
	{
	SetFillColor(255,0,0);
	绘制原生形状（ ）；
	}
	这个形状脚本将对 if(HasProperty()) 语句中指定的四种元素之一以外的任何类型的元素使用标准元素填充颜色；这四种类型中的任何一种元素都将以红色填充显示。

HasProperty 和用户选择的设置

功能() 方法的A特殊应用是检查属性设置，其中您为用户提供了属性功能，以便为使用构造型元素的特定实例设置该属性。因此，用户可以将元素形状脚本图表上，并通过元素上下文菜单，设置一个或多个属性在渲染图表object时所对应的属性。因此，元素可能在一个图表上具有一种外观，但在另一个图表上具有不同外观，因为它在两个图表上具有不同的属性设置。

要在您的配置文件中指定用户可选择的属性，请创建适当的构造型元素，并且 - 对于正在定义的每个属性- 将具有构造型 «diagram property» 的属性添加到此元素。对于属性，键入将显示在上下文元素的时间名称菜单上的选项文本；例如， 是红色的"。还要给属性一个别名，这将是属性的名称，因为它被存储并且属性() 方法将评估它。如果您将属性的初始值设置为1 ，则上下文会设置时间菜单选项；如果没有初始值，属性选项将默认为不设置。

还要定义一个 _image 属性，该属性具有应用 HasProperty() 方法的形状脚本脚本。在本例中，形状脚本定义了两个类属性（ Is Red 和 Is Triangle ），用于形状脚本() 方法，用于检查是否设置了选项。

主要形状

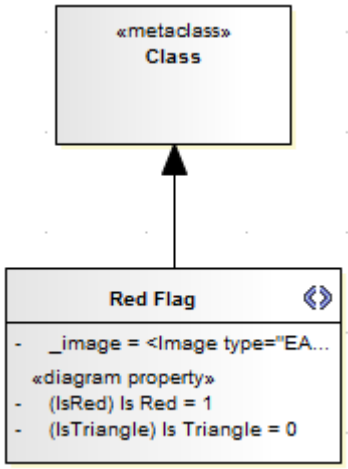
{

if (HasProperty("IsRed"," 1 "))

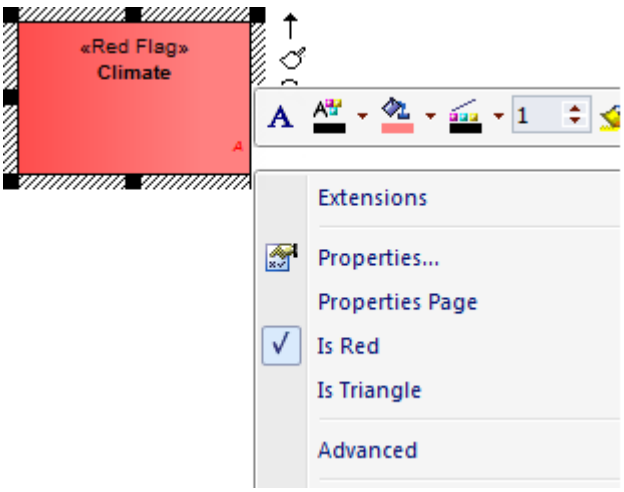
{

设置填充颜色（255,128,128）；
}
if (HasProperty("IsTriangle","0"))
{
多边形（50,50,3,50,0）；
}
else
{
绘制原生形状（）；
}
}

当定义扩展元素类型的构造型时，它将类似于：



MDG 技术创建并发布给用户后，当他们从工具箱中拖动原型元素时，它将根据定义的当前设置属性，用户可以通过上下文菜单访问和重新设置，如显示：



显示元素/连接器属性

自定义形状A常见组件是文本string ，它可以包括元素或连接器的属性之一的名称和值。要显示文本，请使用以下命令之一：

- 打印
- 打印和
- 打印包装

这些都采用一个string参数来表示要显示的文本。元素或连接器属性可以使用替换宏 #<propertyname># 添加到文本中；例如：

println("姓名：#NAME#");

您可以通过多次发出命令来显示多个属性，每个属性一次。您可以显示的元素和连接器属性在此处列出。此外，您还可以通过在标签名称前加上标记值来显示标签，如下所示：

print("#TAG:条件#");

您还可以像测试系统命名的属性一样测试和显示元素的自定义属性；例如：

if(hasproperty("名称","值"))

...

和：

print("#Name#");

Properties for Element Shape Scripts

- actualname - same as 'name' except that it does not react to the 'Use Alias if Available' setting
- addin - returns a value from an invoked Add-In function; syntax:
  addin:<addin_name>, <function_name>, <parameter> [, <parameter> ...]
  Note that in the hasproperty() argument, Enterprise Architect requires the hash characters for addin values:
  if(hasproperty("#ADDIN:MyAddin,MyValue#", "TheValue")) {
- alias
- author
- bookmark
- bookmarkvalue
- cardinality
- classifier
- classifier.actualname - same as 'classifier.name' except that it does not react to the 'Use Alias if Available' setting
- classifier.alias
- classifier.metatype
- classifier.name
- classifier.stereotype
- classifier.type
- complexity
- concurrency
- datecreated
- datemodified
- diagram.author

- diagram.handdrawn
- diagram.mdgtype
- diagram.mdgview
- diagram.name
- diagram.stereotype
- diagram.type
- diagram.version
- ES (adds the End Stereotype character(s) as determined by the "Use extended << and >> characters" option)
- haslinkeddocument
- hiddenparents
- incomingedge (returns "none", "left", "right", "top", "bottom", or "multiple")
- isabstract
- isactive
- iscomposite
- isdrawcompositelinkicon
- isembedded
- isinparent
- isleaf
- islocked
- isroot
- isspec
- istagged
- isvisible
- keywords
- language
- metatype
- multiplicity
- name
- notes
- notesvisible
- outgoingedge (returns "none", "left", "right", "top", "bottom", or "multiple")
- packagename
- packagepath
- package.stereotype
- parentedge ("right", "left", "top", "bottom")
- parent.metatype
- partition (returns "vertical" or "horizontal")
- persistence
- phase
- priority
- propertytype
- propertytype.alias

- propertytype.metatype
- propertytype.name
- propertytype.stereotype
- qualifiedname
- rectanglenotation
- scope
- showcomposeddiagram (returns "True" or "False")
- SS (adds the Start Stereotype character(s) as determined by the "Use extended << and >> characters" option)
- status
- stereotype
- stereotypehidden
- subtype
- type
- version
- visibility

用于连接器的属性形状

- 实际名称 - 与 "名称"相同，只是它不响应 "使用别名如果可用"设置
- 插件
  - 从调用的插件返回一个值插件
  函数；句法：
  插件：<addin_name>, <function_name>, <parameter> [, <parameter> ...]
  注记在 hasproperty() 参数中， Enterprise Architect需要哈希字符作为附加值：
  if(hasproperty("#ADDIN:MyAddin,MyValue#", "TheValue")) {
- 别名
- 图。作者
- diagram.connectornotation
- 图.手绘
- 图.mdgtype
- 图.mdgview
- 图名
- 图.刻板印象
- 图表类型
- 图.版本
- 方向
- 影响
- ES - 添加由 "使用扩展的 << 和 >> 字符"选项确定的结束构造型字符
- 警卫
- 根
- 小岛
- 姓名
- 旋转方向（"上"、"下"、"左"、"右"）

- 源.actualname - 与 '源.name' 相同，只是它不响应 '使用别名if Available' 设置
- 源.聚合
- 源.别名
- 源.可变的
- 源.约束
- 源.元素名
- 源.元素.刻板印象
- 源.metatype 这四个源.metatype属性的详细信息，请参见
- 源帮助.一般*Define Metamodel*约束帮助topic
- 源.metatype.specific
- 源.metatype.both
- 源.多样性
- 源.multiplicityisordered
- 源.name
- 源.qualifiers
- 源.RectangleNotation
- 源.刻板印象
- 源.targetscope
- SS - 添加由 使用扩展的 << 和 >> 字符”选项确定的开始构造型字符
- 刻板印象
- target.actualname - 与 ẗarget.name”相同，只是它不响应 使用别名如果可用”设置
- 目标.聚合
- 目标别名
- 目标可变
- 目标约束
- 元素
- 目标.元素.刻板印象
- 目标.元类型
- 目标.多重性
- target.multiplicityisordered
- 目标名称
- 目标限定符
- 目标.矩形符号
- 目标定型
- 目标.targetscope
- 触发器
- 类型
- 重量

子形状

当您使用形状脚本定义元素或连接器形状时，您可以从单独的组件构建形状，定义为子形状。使用子形状，您可以创建更类似于它们所代表的对象的复杂形状。

子形状布局

要设置布局类型，请使用布局属性，该属性必须在脚本的初始化属性部分中设置；换句话说，在调用任何方法之前。此属性的有效值为：

- LeftRight - 具有此布局的形状与子形状并排，第一个添加在左侧，随后的子形状添加到右侧
- 自上而下 - 将子形状垂直排列，第一个子形状添加到顶部，随后的子形状添加到下方
- 边框 - 这需要 addsubshape 方法的附加参数来指定子形状要占据的包含形状的哪个区域： N 、 E 、 S 、 W 或 CENTER；每个区域只能被一个子形状占据
  分配给E或 W 区域A子形状必须在其声明中指定其首选宽度属性，类似地，添加到N或S的子形状必须设置其首选高度属性；在这种情况下，这些属性的值被视为静态长度并且不缩放字形

示例

主要形状

{

布局类型=”自上而下”；

setfillcolor(0,0,255);

矩形（0,0,100,100）；

addsubshape("sub",50,100,20,0);

addsubshape("sub",50,100,30,-100);

addsubshape("sub",50,100,40,-200);

addsubshape("sub",50,100,50,-300);

形子

{

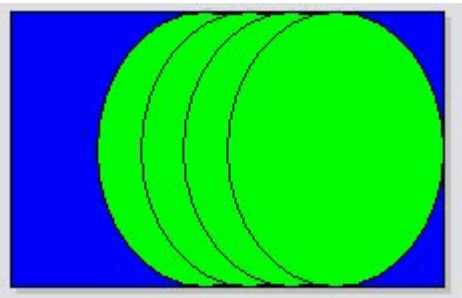
setfillcolor(0,255,0);

椭圆（0,0,100,100）；

}

}

脚本定义了这个形状：



为元素添加自定义分区

当您以正常的矩形格式在图表上显示元素时，可以在该框架内显示许多隔间以显示附加的特征，例如属性、操作和注记，使用图表 属性"和 元素"隔间可见性' 对话框。如果要显示其他添加的特征，例如相关元素或端口和部件，可以使用形状脚本将自定义隔间添加到元素的图表显示中。您通常会将此形状脚本脚本添加到配置文件中的构造型元素中。

创建自定义隔间后，您可以将链接的注记添加到元素以显示隔间的内容，就像您可以为元素的其他特征一样。

访问

在配置文件中定义构造型元素，并使用特殊属性 形状脚本"来指定添加自定义隔间的形状脚本。

功能区	设计>元素>特征>属性：[创建一个名为 _"image"的属性]>单击 初始值"字段中的...图标 设置>参考> UML类型>构造型：（选择或指定构造型）：形状脚本>分配
上下文菜单	在图表中，右键单击元素|特征|属性：[创建一个名为 '_image' 的属性] |单击 初始值"字段中的...
键盘快捷键	F9：[创建一个名为 ..."的属性] > 单击 初始值"字段中的浏览图标图标

向元素添加自定义隔间

此表提供有关创建定义自定义注记的形状脚本的注释，以及各种示例。

Process	Description
开发脚本	对于选定的构造型，打开形状编辑器。 在脚本中，将*shape main*替换为： • 形状 *ChildElement*或 • 形状相关元素 如果您愿意，可以保持*shape main* ，以调整主元素的某些属性（例如颜色）；但是，主要形状需要调用 DrawNativeShape()"才能正常工作。 此时，您可以使用 'HasProperty' 查询方法来搜索子元素或相关元素，以查找要在隔间中显示的特定属性（例如构造型）。形状脚本属性确定通过连接器链接到当前元素的元素A属性。 由形状脚本定义的每个单独的自定义隔间的可见性是使用 隔间可见性"对话框控制的。 ChildElement 隔间默认可见，可以使用隔间可见性选项隐藏，而 RelatedElement 隔间是隐藏的 默认情况下，必须使用隔离专区可见性选项显式启用。 还要注意，当子元素与父元素在图表上或不在图表上时（如示例1和 3 中所示），子元素可以显示在自定义隔间中。如果相关元素不在同一图表上（如示例 4 和 5 中所示），则它们只能在隔间中列出。
附上链接注记	您可以使用以下两种方法之一来创建链接的注记以显示自定义隔间内容： • 方法1（元素当前显示自定义隔间）- 突出显示自定义隔间中的相关或子

	元素名称，然后右键单击它并选择 创建链接注记”选项；自定义隔间会自动关闭，链接的注记会添加到列出该隔间中所有元素名称的图表中 • 方法 2（元素不一定显示自定义隔间） - 从元素的 公共”页面图表一个注记工具箱，并使用注释链接连接器将其链接到包含自定义隔间的元素 右键单击连接器并选择 将此注记链接到元素特征”选项，以显示 将注记链接到元素特征”对话框；单击 特征类型”字段中的下拉箭头，然后单击自定义隔间的名称，例如 属性”，然后单击确定按钮 该隔间的内容显示在注记中 在方法 2 中，如果显示隔间，则该方法不会隐藏该隔间。如果隔间已隐藏，建议您使用此方法。 您对隔间中的元素列表或其名称所做的任何更改都会立即反映在注记中，以保持显示信息的准确性。

脚本示例1：添加隔间而不调整父元素	//为子元素添加隔间。
	形状子元素
	{
	//选择子元素是否具有属性stereotype，如果有则设置
	//属性的隔间名称。
	if(HasProperty("stereotype", "属性"))
	{
	SetCompartmentName("属性");
	}
	//选择子元素是否具有公共范围，如果有，则添加+
	//子隔间的符号。
	如果（HasProperty（ 范围”，“公共”））
	{
	AppendCompartmentText("+");
	}
	//将子元素名称添加到子隔间。
	AppendCompartmentText("#NAME#");
	}
	
	属性属性检查所有子元素以查看它们是否具有 <<property>> 的形状脚本型。如果找到此构造型，则 SetCompartmentName”函数设置一个名为 属性”的隔间。
	然后脚本检查子元素是否具有 公共”范围，如果有，则附加 +”符号。
	最后，“AppendCompartmentText”函数将子元素的名称添加到隔间。
	如果“SetCompartmentName”已经声 了一个隔间，则属于同一隔间的任何其他子元素都会自动添加到该隔间，而无需声明新的隔间名称（即，所有具有构造型 <<property>> 的子元素最终进入 属性”隔间）。

脚本示例2: 调整父元素的颜色并添加子隔间	//形状主要影响父级
	主要形状
	{
	//设置父元素的颜色为红色
	setfillcolor(255,0,0);
	//绘制父母的原生形状
	绘制形状（ ）；

	}
	
	//形状将子分隔添加到父元素。
	形状子元素
	{
	if(HasProperty("stereotype", "part"))
	{
	SetCompartmentName("零件");
	}
	else if(HasProperty("stereotype", "mystereotype"))
	{
	SetCompartmentName("我的构造型");
	}
	
	AppendCompartmentText("#NAME#");
	}
	
	'shape main' 部分将主元素的颜色设置为红色，并根据原型子元素添加子隔间。
	该脚本检查子元素是否应用了原型值 "part"或"mystereotype"。如果有多个子元素，具有"part"和"mystereotype"原型的组合，则会创建两个隔间，称为"Parts"和"My构造型"。
	为了显示隔间，必须调用 "AppendCompartmentText"将内容插入隔间。
	传递给 "SetCompartmentName"和"AppendCompartmentText"的值不能包含换行符。

脚本示例3: 如果子元素在图上不可见，则仅列出隔间中的子元素	形状子元素
	{
	//选择子元素是否在图表上。
	if(hasproperty("IsVisible", " False "))
	{
	//为零件创建一个隔间。
	如果（有属性（ 类型"，"部分"））
	{
	SetCompartmentName("零件");
	}
	//为端口创建一个隔间。
	else if(hasproperty("type", "port"))
	{
	SetCompartmentName("端口");
	}
	//将子元素名称添加到隔间。
	AppendCompartmentText("#NAME#");
	}
	}

	此脚本为属于当前元素但在当前图表上不可见的端口和部件元素添加自定义隔间。 如果子元素属性True元素则返回False 。 如果子元素已经在图表上可见，这可以用来防止子元素被列在自定义隔间中，从而避免显示冗余信息。

脚本示例4：从拥有形状脚本的元素中显示作为依赖连接器目标的元素	形状相关元素
	{
	//选择我们正在处理的当前连接器是否有
	//依赖类型。
	if(HasProperty("连接器.类型", "依赖"))
	{
	//选择如果我们当前正在检查的元素是
	//当前连接器的目标。
	if(HasProperty("元素.IsTarget"))
	{
	//设置车厢名称
	SetCompartmentName("dependsOn");
	if(HasProperty("元素.构造型", ""))
	{
	}
	else
	{
	AppendCompartmentText("«#Element.Stereotype#»");
	}
	AppendCompartmentText("#Element.Name#");
	}
	}
	}
	
	使用此脚本，如果 Class1 具有带有 "RelatedElement"形状脚本脚本的构造型，并且 Class1 是目标 Class2 的 Dependency 连接器的源，则名称 Class2 将显示在类的隔间1 ，称为 "dependsOn"。

脚本示例5: 在一个元素的一个隔间内显示一个已实现接口的列表	形状相关元素
	{
	//选择当前正在处理的连接器是否为Realization
	if(HasProperty("连接器.类型", "实现"))
	{
	//只有相关元素we才显示这个隔间
	//正在检查的是有这个的连接器的目标
	//形状脚本元素作为源
	if(HasProperty("元素.IsTarget"))
	{
	//如果元素是接口，则显示在
	//'realizedInterfaces' 隔间

	if(HasProperty("元素.类型", "接口"))
	{
	SetCompartmentName("realizedInterfaces");
	AppendCompartmentText("#Element.Name#");
	}
	}
	}
	}
	
	如果元素类1具有此形状脚本，并且是元素接口1的实现连接器的源，则名称'接口1 ' 将显示在类1的 'realizedInterfaces' 隔间中。

注记

- 如果您在分隔名称中使用标点符号，则会在保存脚本时将其删除；例如： 端口、部件和属性”变为 端口部件和属性”
- 'RelatedElement'形状脚本具有检查连接器和连接器另一端的元素的扩展功能；它们仅适用于元素并且仅用于检索要在该元素的隔间内显示的信息

显示复合图表

您可以将元素定义为正在复合（使用 新图表|组合复合结构图表”上下文菜单选项），在这种情况下，元素有一个子复合图来描述元素的子结构。您还可以使用上下文菜单选项在元素复合元素元素通常，重新定义复合元素外观的形状脚本脚本会有效规避这些选项的效果，但您可以编辑脚本以响应 在分区中显示复合图表”选项并在中间部分显示子复合图元素。

为了显示复合图，脚本需要 边框”的布局类型，在绘制时将复合图添加到主形状的中心子形状。因此，定义形状脚本的语句是：

```
主要形状
{
布局类型="边框";
if(HasProperty("ShowComposedDiagram", "true"))
{
addsubshape("ComposedDiagram", "CENTER");
}
形状组合图
{
绘制组合图（ ）；
}
}
```

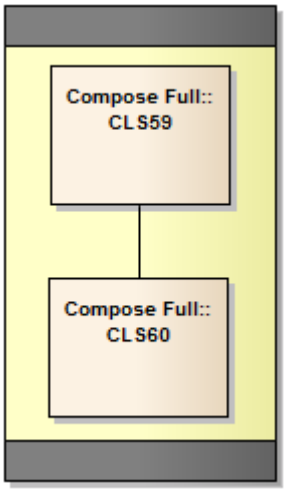
例子

包含组合图的形状脚本的示例是：

```
形状主要
{
//设置边框类型
布局类型="边框";
//设置奶油填充颜色
setfillcolor(255, 255, 200);
//为object划一个基本矩形。
矩形（0, 0, 100, 100 ）；
//在形状的顶部添加一些填充
addsubshape("填充", " N ");
//选择上下文菜单选项的设置
if(HasProperty("ShowComposedDiagram", "true"))
{
//将组合图添加到object的中心
addsubshape("ComposedDiagram", "CENTER");
}
//在形状的底部添加一些填充。
```

addsubshape("填充", " S ");
形状填充
{
//设置这个元素的高度
首选高度 = 20;
//设置填充颜色为灰色
setfillcolor(128, 128, 128);
//划一个占据object宽度的矩形
// 高度为 20 像素。
矩形（ 0, 0, 100, 100 ）；
}
形状组合图
{
//划组合图。
绘制组合图（ ）；
}
}

此脚本生成形状：



目前仅支持组合图作为主形状的中心子形状。将图表添加到任何其他位置将导致组合图表无法正确绘制或根本不绘制。该图可以是子形状的子形状，但前提是父形状和子形状都具有 "中心"方向。例如：

//这个shapescript很好，因为shape E是shape C的中心，也就是shape D的中心；也就是说，所有指向
//DrawComposedDiagram 的形状都是 "CENTER"。

主要形状
{
layouttype = "边框";
矩形 (0, 0, 100, 100);
添加子形状（" D "，"CENTER" ）；
形状D

```
{
布局类型= 边框″；
添加子形状（“C”，“CENTER”）；
形状 C
{
布局类型= 边框″；
添加子形状（“ E ”，“CENTER”）；
addsubshape（ 填充″，“ N ”）；
addsubshape（ 填充″，“ S ”）；
形状E
{
绘制组合图（ ）；
}
形状填充
{
首选高度 = 20;
setfillcolor (10, 30, 80);
矩形 (0, 0, 100, 100);
}
}
}
}

//这个shapescript不好 - 形状E是″CENTER”，形状C是″ S ”，形状D是″CENTER”；因为形状 C 的方向是″ S ”
//图表不会绘制。

主要形状
{
layouttype = "边框";
矩形 (0, 0, 100, 100);
添加子形状（“ D ”，“CENTER”）；
形状D
{
布局类型= 边框″；
添加子形状（“C”，“ S ”）； //<- 这很糟糕，DrawComposedDiagram 调用的所有父子形状都必须是
// 面向″中心″
形状 C
{
布局类型= 边框″；
添加子形状（“ E ”，“CENTER”）；
addsubshape（ 填充″，“ N ”）；
addsubshape（ 填充″，“ S ”）；
```

```
形状E
{
绘制组合图（ ）；
}
形状填充
{
首选高度 = 20;
setfillcolor (10, 30, 80);
矩形 (0, 0, 100, 100);
}
}
}
}
```

注记

- 为了显示复合图，'New图表|应在图表中元素的 "上下文"菜单上选择 "显示复合图表"选项
- 组合图以自然大小显示，因此无法将父元素调整为小于组合图的大小

保留名称

当你写一个形状脚本脚本时，有一些术语是保留的，因为它们在脚本中具有特殊的含义；将它们用于特定目的。

元素

元素（例如类、状态或事件）具有这些为形状部分保留的名称。

Name	Description
主要形状	主要形状是整体形状。
形状标签	形状标签给形状一个分离的标签。
装饰 <标识符>	装饰为形状提供由 <identifier> 中的名称定义的装饰。
形状子元素	允许基于属于当前元素的子元素添加自定义隔间。
形状相关元素	允许根据属于当前元素的相关元素添加自定义隔间。

连接器

连接器（如关联、依赖或概括）具有形状的这些保留名称。

Name	Description
主要形状	主要形状是整体形状。
形状源	源形状是连接器源端的一个额外形状。
形状目标	目标形状是连接器目标端的额外形状。
形状 LeftTopLabel	Shapes 为左上角的连接器定义了一个分离的标签。
形状 MiddleTopLabel	Shapes 为中间顶部的连接器定义了一个分离的标签。
形状 RightTopLabel	Shapes 为右上角的连接器定义了一个分离的标签。
形状 LeftBottomLabel	Shapes 为左下角的连接器定义了一个分离的标签。
形状 MiddleBottomLabel	Shapes 为中间底部的连接器定义了一个分离的标签。
形状 RightBottomLabel	Shapes 为右下角的连接器定义了一个分离的标签。

语法文法

形状脚本A一部分可能非常复杂，包含许多命令和参数。此表提供了形状脚本结构的细分，说明了命令和参数是如何构造的。第一个条目是顶层声明，随后的条目显示了依次更详细的组件的组成。

语法符号

- * = 零个或多个
- + = 一个或多个
- | = 或
- ; = 终结者

Symbol	Description
形状脚本 ::=	<形状>*;
形状::=	<ShapeDeclaration> <ShapeBody>;
形状声明 ::=	<形状类型> <形状名称>;
形状类型 ::=	形状" | "装修" | 标签";
形状名称 ::=	<保留形状名称> | <字符串文字>;
保留形状名称 ::=	有关完整的保留形状列表，请参阅保留名称。
形状体 ::=	"{" <InitialisationAttributeAssignment>* <DrawingStatement>* <SubShape>* "}";
初始化属性分配 ::=	<属性> "=" <值> ";";
属性 ::=	有关属性名称的完整列表，请参阅形状属性。
绘图声明 ::=	<IfElseSection> | <方法>;
IfElseSection ::=	"if" "("<QueryExpression>")" <TrueSection> (<ElseSection>);
查询表达式 ::=	<QueryName> "(" <ParameterList> ")"; 有关查询及其参数的描述，请参见方法。
查询名称 ::=	有关可能的查询名称，请参阅查询。
TrueSection ::=	"{" <绘图说明>* "}"
其他部分 ::=	" else " "{" <DrawingStatement>* "}"
方法 ::=	<方法名> "(" <参数列表> ")" ";";
方法名 ::=	有关方法名称的完整列表，请参阅绘图方法。

示例脚本

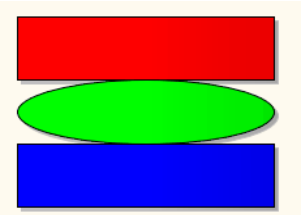
您可以使用形状脚本创建各种形状、效果和文本语句，以增强您创建的元素和连接器的外观和信息价值。此处提供了此类脚本的一些示例。

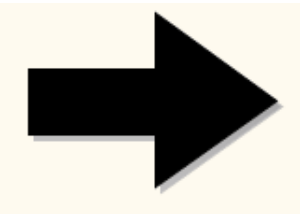
访问

功能区	设置 >参考> UML类型 >构造型（指定构造型）：形状脚本+ 赋值，或 设置>参考> UML类型>构造型（指定构造型）：形状脚本+编辑

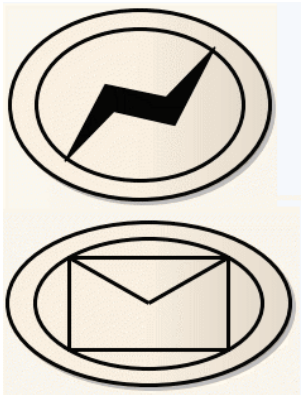
例子

Shape	Script

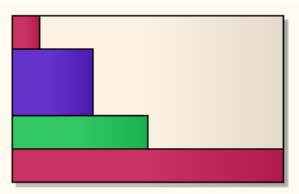
	// 基本形状
	主要形状
	{
	setfillcolor(255, 0, 0); // (R ,G, B)
	矩形（ 0, 0, 90, 30 ）； // (x1, y1 ,x2,y2)
	
	setfillcolor(0, 255, 0); // (R ,G, B)
	椭圆（ 0, 30, 90, 60 ）； // (x1, y1 ,x2,y2)
	
	setfillcolor(0, 0, 255); // (R ,G, B)
	矩形（ 0、60、90、90 ）； // (x1, y1 ,x2,y2)
	}

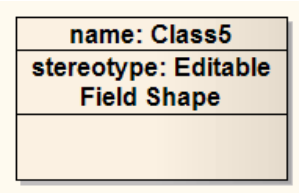
	// 单一条件形状
	主要形状
	{
	if (触发器("", "Link"))
	{
	// 只有当object有标记值时才绘制Trigger=Link
	// 设置路径的填充颜色
	setfillcolor(0, 0, 0);
	开始路径（ ）； //开始追踪路径
	移动到（ 23、40 ）；
	lineto(23, 60);
	lineto(50, 60);
	lineto(50, 76);
	lineto(76, 50);

	lineto(50, 23);
	lineto(50, 40);
	结束路径（ ）； // 结束追踪路径
	// 用填充颜色填充跟踪的路径
	填充行程路径（ ）；
	返回;
	}
	}

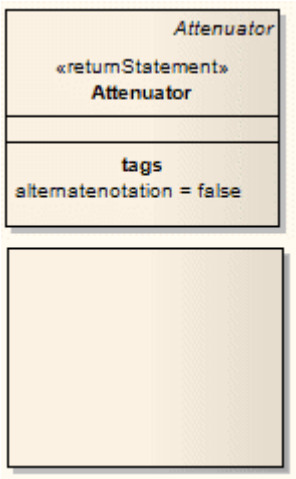
	// 多条件形状
	主要形状
	{
	开始路径（ ）；
	椭圆（0, 0, 100, 100）；
	结束路径（ ）；
	填充行程路径（ ）；
	椭圆（3, 3, 97,97）；
	
	if (触发器("", "None"))
	{
	返回;
	}
	
	if (触发器("", "Error"))
	{
	setfillcolor(0, 0, 0);
	开始路径（ ）；
	移动（23, 77）；
	lineto(37, 40);
	lineto(60, 47);
	线托（77、23）；
	lineto(63, 60);
	lineto(40, 53);
	线（23, 77）；
	结束路径（ ）；
	填充行程路径（ ）；
	返回;
	}
	if (触发器("", "信息"))
	{
	矩形（22、22、78、78）；
	移动到（22、22）；
	lineto(50, 50);
	线（78, 22）；
	返回;

	} }

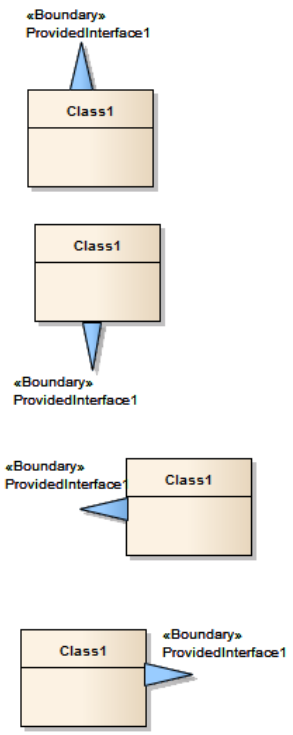
	// 子形状
	主要形状
	{
	矩形（0, 0, 100, 100）；
	
	addsubshape("红色", 10, 20);
	addsubshape("蓝色", 30, 40);
	addsubshape("绿色", 50, 20);
	addsubshape("红色", 100, 20);
	
	形状红色
	{
	setfillcolor(200, 50, 100);
	矩形（0, 0, 100, 100）；
	}
	
	形状 蓝色
	{
	setfillcolor(100, 50, 200);
	矩形（0, 0, 100, 100）；
	}
	
	形状绿色
	{
	setfillcolor(50, 200, 100);
	矩形（0, 0, 100, 100）；
	}
	}

	// 可编辑的字段形状
	主要形状
	{
	矩形（0, 0, 100, 100）；
	addsubshape("namecompartment", 100, 20);
	addsubshape("stereotypecompartment", 100, 40);
	
	形状名称隔间
	{
	h_align = "中心";
	可编辑字段= 名称”；
	矩形（0, 0, 100, 100）；
	println("姓名：#姓名#");

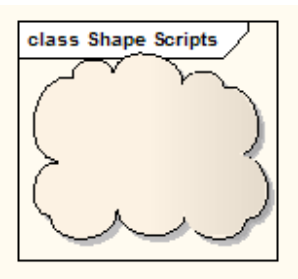
	}
	
	形状定型隔间
	{
	h_align = "中心";
	editablefield = "刻板印象";
	矩形（ 0, 0, 100, 100 ）；
	println("刻板印象：#stereotype#");
	}
	}

	// 返回语句形状
	主要形状
	{
	if (hasTag("alternatenotation", " false "))
	{
	//绘制ea的内置字形
	绘制形状（ ）；
	//用return语句退出脚本
	返回;
	}
	else
	{
	//替代符号命令
	//...
	矩形（ 0, 0, 100, 100 ）；
	}
	}

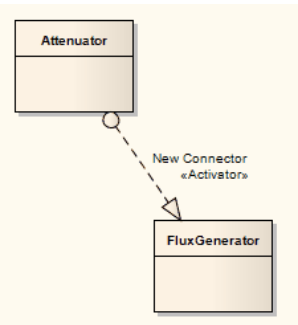
	//嵌入元素形状在父边缘的位置
	主要形状
	{
	定义大小（ 60,60 ）；
	开始路径（ ）；
	if(hasproperty("parentedge","top"))
	{
	移动到（ 0,100 ）；
	线（ 50,0 ）；
	线到（ 100,100 ）；
	}
	if(hasproperty("parentedge","bottom"))
	{
	移动到（ 0,0 ）；
	线（ 50,100 ）；
	lineto(100,0);




```
}
if(hasproperty("parentedge","left"))
{
移动到（100,0）；
线（0,50）；
线到（100,100）；
}
if(hasproperty("parentedge","right"))
{
移动到（0,0）；
线（100,50）；
线（0,100）；
}
结束路径（）；
设置填充颜色（153,204,255）；
填充行程路径（）；
}
```

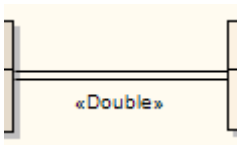


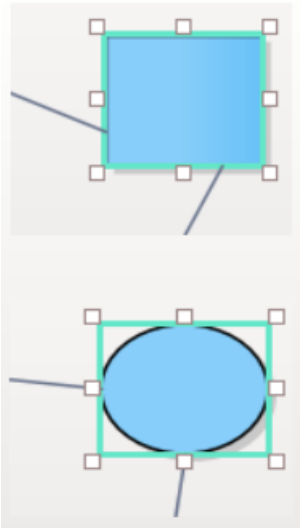

```
// 云路径示例形状
主要形状
{
开始云路径（）；
矩形（0, 0, 100, 100）；
结束路径（）；
FillAndStrokePath();
}
```



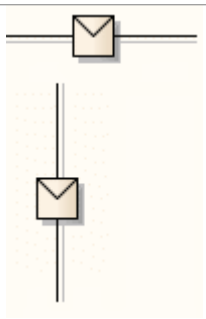

```
// 连接器形状
主要形状
{
// 画一条虚线
无影=真；
setlinestyle("DASH");
移动到（0,0）；
lineto(100,0);
}

形状源
{
// 在源端画一个圆
可旋转=真；
开始路径（）；
椭圆（0,6,12,-6）；
结束路径（）；
```

	填充行程路径（ ）；
	}
	
	形状目标
	{
	// 在目标端画一个箭头
	可旋转=真；
	开始路径（ ）；
	移动到（0,0）；
	线（16,6）；
	线托（16，-6）；
	结束路径（ ）；
	填充行程路径（ ）；
	}

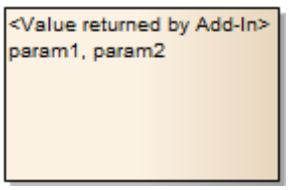
	// 双线
	主要形状
	{
	setlinestyle("DOUBLE");
	移动到（0,0）；
	lineto(100,0);
	}

	// 设置附件模式
	形状 主要的
	{
	if (hasproperty ("rectanglenotation" , " 1 "))
	{
	SetAttachmentMode ("正常");
	矩形(0 , 0 , 100 , 100);
	}
	else
	{
	SetAttachmentMode ("钻石");
	椭圆(0 , 0 , 100 , 100);
	}
	}
	
	在此示例中，如果元素启用了 矩形表示法”，则 SetAttachmentMode("normal") 命令将允许连接器连接到元素（第一个形状）每个边缘的任意点。如果元素关闭了 Rectangle Notation”，SetAttachmentMode("diamond") 命令将只允许连接器连接到元素每条边的中心点；也就是说，呈菱形（第二种形状）。您不能将附着点移动到该边的其他位置。

	// 旋转方向 主要形状

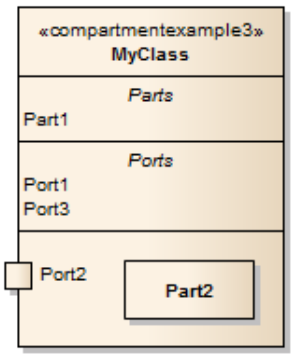


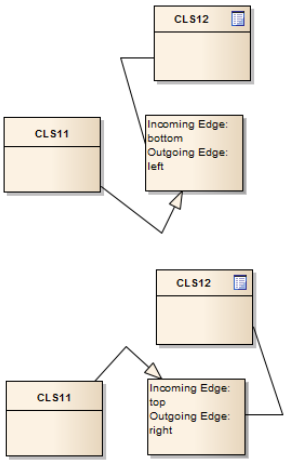
```
{
移动到（0,0）；
lineto(100,0);
固定区域（40，-10,60,10）；
矩形（40，-10,60,10）；
if(hasproperty("旋转方向","向上"))
{
移动到（60，-10）；
线（50,0）；
线托（60,10）；
}
if(hasproperty("旋转方向","向下"))
{
移动到（40，-10）；
线（50,0）；
线（40,10）；
}
if(hasproperty("旋转方向","左"))
{
移动到（40，-10）；
线（50,0）；
线（60，-10）；
}
if(hasproperty("旋转方向","右"))
{
移动（40,10）；
线（50,0）；
线托（60,10）；
}
}
```

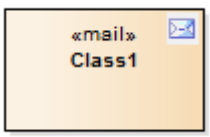



```
// 获取加载项返回A值
主要形状
{
//划一个简单的矩形
矩形（0,0,100,100）；

//打印从插件
返回的string值插件
 我的插件"，
//函数 "MyExample"有两个string参数
Print("#ADDIN:MyAddin, MyExample, param1, param2#");
}

// 附加功能的方法签名：
```

	// Public函数MyExample(存储库As EA.Repository , // eaGuid As字符串, args As Variant) As Variant

	// 添加基于子元素的自定义分区
	// 或相关元素
	
	（请参阅向元素帮助添加自定义分区帮助）

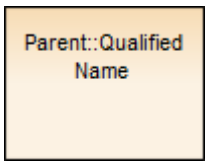
	// 返回连接器的输入和输出边缘
	// 进入和离开一个对象
	主要形状
	{
	//划一个简单的矩形
	矩形（0,0,100,100）；
	
	//打印元素上的传入边
	Print("传入边缘：#incomingedge#\n");
	
	//在元素上打印出边
	Print("输出边缘：#outgoingedge#\n");
	}

	// 在默认的顶部画A装饰图标
	// 元素形状
	装饰邮件
	{
	方向=“NE”；
	image ("图标图像", 0, 0, 100, 100);
	// "icon image" 是加载到图像管理器中的 16x16 图像的名称
	}

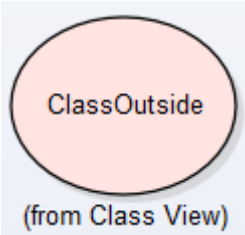
	// A文件中绘制图像和可编辑的名称字段
	主要形状
	{
	addsubshape ("theimage", 100, 100);
	addsubshape ("namecompartment", 100, 100);
	
	塑造形象
	{
	image ("元素图像", 0, 0, 100, 100);
	// "元素" 是加载到图像管理器中的图像的名称

	}
	形状名称隔间
	{
	h_align = "中心";
	可编辑字段= 名称”；
	println("#name#");
	}
	}

	// 检查是否A复合元素图标
	// 如果是这样，画一个
	装修补偿
	{
	方向=“SE”；
	if(hasproperty("IsDrawCompositeLinkIcon","true"))
	{
	开始路径（ ）；
	椭圆（-80,29,-10,71 ）；
	椭圆（10,29,80,71 ）；
	移动（-10,50 ）；
	线托（10,50 ）；
	结束路径（ ）；
	行程路径（ ）；
	}
	}

	// 允许A形状脚本显示完整的对象
	// 拥有元素的名称，包括拥有元素
	// 并且拥有包，当图表属性时
	// '禁用完全范围的对象名称' 选项是
	// 取消选择，就像没有A元素一样
	// 形状脚本。
	主要形状
	{
	布局类型= 边框”；
	矩形 (0, 0, 100, 100);
	addsubshape（“填充”，“N”）；
	addsubshape（ 名称”，”中心”）；
	形状填充
	{
	首选高度=8；
	}
	形状名称
	{
	v_align="顶部";

	h_align=″中心″； printwrapped ("#qualifiedname#"); } }

	// 元素时显示所属包的名称
	// 用于不在该软件包A图表上，并且
	// 选择了图表属性 显示命名空间″选项。
	主要形状
	{
	布局类型= 边框″；
	v_align = "中心";
	h_align = "中心";
	椭圆 (0, 0, 100, 100);
	printwrapped ("#name#");
	addsubshape（ 路径″，“ S ”）；
	形状路径
	{
	v_align="顶部";
	h_align=″中心″；
	if (hasproperty ("packagepath", ""))
	{
	}
	else
	{
	printwrapped ("(来自#packagepath#)");
	}
	}
	}

	// 在右上角显示元素概括的列表。
	//注记：只列出不在当前图上的元素。
	形状 主要的
	{
	layouttype = "边框" ;
	矩形(0 , 0 , 100 , 100);
	添加子形状 (名称″ , ″中心″) ;
	addsubshape （″父母″ , “ N ”）;
	
	形状 姓名
	{
	v_align = "居中" ;
	h_align = "居中" ;
	粗体=真 ;
	打印 （ “#name#” ） ；

```
        }

        形状 父母
        {
                v_align = "顶部" ;
                h_align = "右" ;
                斜体=真 ;
                打印 ( “#hiddenparents#” ) ;
        }
}
```

标记值类型

在使用标记值时，您可以根据预定义的、系统提供的标记值类型创建自己的、自定义的标记值值。有了这些，您可以创建：

- 复杂且基于预定义类型的标记值，有或没有标签过滤器
- 结构化标记值是复合的，包含其他标记值
- 从各种参考数据库库表返回标记值
- 将用户提供的数据插入到文本string中的屏蔽标记值，例如提示行或字段名称

通过将任何类型的标记值添加到配置文件中的构造型元素，您可以为标记元素在技术中的出现和行为方式定义附加元信息。标记值由构造型元素的属性标识。

注记

- 您可以使用 设置>模型>传输>导出参考”和 导入数据”功能区选项在模型之间传输标记值类型定义；标记参考标记值类型导出为属性类型

从预定义类型标记值类型

当您在使用标记值时，您可能需要使用结构化的标记值；也就是说，标记值以特定格式捕获和呈现更复杂的信息。可以为您的模型轻松创建此类标签的基本类型（您在窗口的 标签”页面创建标签时调用的类型），因为您可以将自定义的结构化属性标记值标记值基于预定义的标记值类型和过滤器范围。

访问

功能区	设置 >参考> UML类型 >标记值类型

创建自定义结构化标记值类型

字段/按钮	描述
标签名称	类型为您的新标记值类型提供的适当名称。
描述	（可选）键入标记值类型的简短说明或用途。
细节	复制和粘贴或键入作为新的标记值类型基础的预定义结构化标记值类型的语法。
节省	点击此按钮保存新的结构化标记值类型。 标记值类型显示在定义的标签类型列表中。
新的	或者，单击此按钮以清除字段，以便您可以输入另一个新的标记值类型的信息。

预定义的结构化类型

标记值定义了一个模型元素具有的范围广泛的属性和特性，而这些属性中有些是复杂的值。例如，您可能希望您的用户在上限和下限之间选择一个值（使用 旋转"箭头）、设置日期、从调色板中选择一种颜色，或者通过检查清单进行操作。

您可以从许多预定义的标记值类型和过滤器中的任何一个创建这些复杂的标记值，其中一些可能是您自己创建的（ 设置 >参考> UML类型 >标记值类型"）。

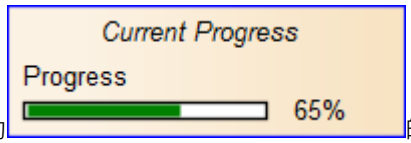
标记值类型格式

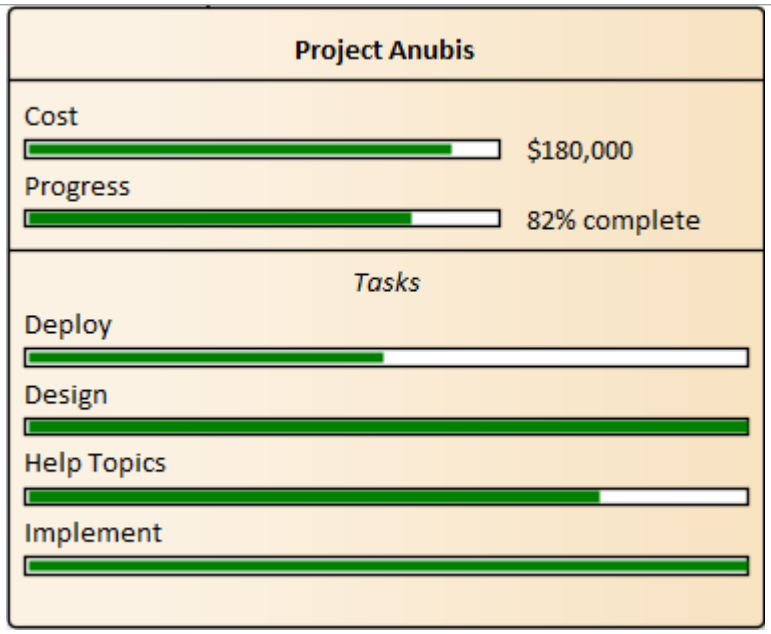
对于每个标记值类型，描述包括创建用于标记值的初始值的语法。名称和格式区分大小写。

标记值类型	格式
Addin广播	类型=AddinBroadcast； 值=您的插件名称； 用于：允许插件 要响应编辑此标记值的尝试，可以显示一个对话框，在该对话框中可以编辑值和注记。
布尔值	类型=布尔值； 默认值=Val； 用于：提供True或False的输入，其中任何一个都可以是默认值。
清单	类型=检查清单； 值=Val1，Val2，Val3； 用于：在批准或执行某项操作之前创建要完成或满足的事项清单。 Val1、Val2、Val3 等指定清单项，每一项都通过属性窗口的 标签"选项卡呈现，并带有一个复选框；在选中每个复选框之前，该标签的值为 不完整"，此时该值为 完全"。 例如： 类型=检查清单； 值=更改是否解决了给定的任务\问题，代码是否有足够的错误处理，代码是否有意义，代码是否符合编码约定； 虽然元素标记值隔间和 标签"选项卡窗口字段显示值 完全"或 不完整"，但文档和 Web 报告将显示清单项目列表和每个项目的状态（ True表示选中，False表示未选中）。
分类器	类型=分类器； 值 = 类型 1，类型 2； 刻板印象=刻板印象1； 用于：已弃用- 使用**RefGUID**和**RefGUIDList**
颜色	类型=颜色； 默认值=Val； 用于：从颜色选择器菜单输入颜色值，其中该值是颜色的 Hex RGB 值。 例如，蓝色的 Hex RGB 为 0000FF，而绿色的 Hex RGB 为 00FF00。

常量	类型=常量； 默认值=Val； 用于：创建只读常量值。
风俗	类型=自定义； 用于：使用掩码值为预定义类型创建自己的模板。
日期	类型=日期； 用于：从日历菜单中输入标记值的日期。
约会时间	类型=日期时间； 用于：已弃用-使用**日期** 从日历菜单中输入标记值日期。
图表参考	类型=图表参考 用于：参考模型中的图表。
目录	类型=目录； 默认值=Val； 用于：从浏览器输入目录路径。 您可以将默认目录路径设置为string值。
枚举	类型=枚举； 值=Val1，Val2，Val3； 默认=Val2； 用于：定义一个逗号分隔的列表，其中 Val1、Val2 和 Val3 表示列表中的值，Default 表示列表的默认值。
文件	类型=文件； 默认值=Val； 用于：从文件浏览器对话框输入文件名。命名文件可以在其默认应用程序中启动。 您可以将默认文件设置为包含文件路径和文件名的string。
浮点数、十进制、双精度	类型=浮动； 类型=十进制； 类型=双； 默认值=Val； 用于：输入浮点、十进制或双精度值。这些类型都映射到相同类型的数据。 您可以为任何或所有这些设置默认值。
图像参考	类型=ImageRef； 用于：提供指向图像管理器中保存的图像文件的链接。
整数	类型=整数； 默认值=Val； 用于：输入一个整数和一个默认值

备忘录	类型=备忘录； 用于：为标签输入大而复杂的值。

进度条	类型=进度条；
	隔间=<名称>； - 设置要在其中显示进度条的隔间的名称；多个标记值可以在一个隔间添加进度条
	文本=<文本>; - 在进度条的右侧显示 <text>；要使用文本显示标记的值，请使用 #VALUE#，例如 $#VALUE# 或 #VALUE#%
	最小值=n; - 设置可以在进度条中显示的最小值（必须是整数）
	最大值=n; - 设置进度条可以显示的最大值（必须是整数）
	用于：当元素显示在图表上并且在图表 属性"对话框的 元素"页面上启用了标签隔间时，在元素的隔间中显示进度条。标签名称显示在进度条上方，作为其标签。
	• 如果 MinVal 或 MaxVal 均未设置，则进度条的默认值为 0 和 100
	• 如果设置了 MinVal 但未设置 MaxVal，则最大值默认为 MinVal+100
	• 如果设置了 MaxVal 但未设置 MinVal，则最小值默认为 0
	• 如果同时设置了 MinVal 和 MaxVal，则 MinVal 必须低于 MaxVal
	例子：
	隔间=当前进度；
	类型=进度条；
	文字=#VALUE#%;
	 
	
	当在名为 的标记中使用 ProgressBarTag1，其值设置为 65。
	
	类型=进度条；
	最小值=1000；
	最大值=100000；
	文本=$ #VALUE#;
	 
	
	在名为 Progress 的标签中使用时，其值设置为 4530。
	
	具有多个进度条的元素。

	
RefGUID	类型=RefGUID； 值 = 类型 1，类型 2； 刻板印象=刻板印象1； 或者 类型=RefGUID； 元类型=类型； 用于：通过指定元素的GUID来参考模型中的元素，其中： • Type1 和 Type2 指定一个或多个允许的图表对象（例如类、部件、属性或操作） • Stereotype1 表示允许的刻板印象 元类型可用于引用分类器或属性类型： • 元类型=分类器；呈现所有企业架构师定义的分类器类型以供选择 • 元类型=属性；呈现所有端口、部件和属性以供选择 您可以通过在属性窗口中单击标记值的 ... 按钮来设置该类型的标记值的分类器、属性或操作。 您也可以右键单击属性窗口中的RefGUID标记值名称，选择 在项目中查找浏览器”选项，在浏览器窗口中定位被引用object。 打印RefGUID标记值元素值时，形状脚本将打印引用的名称。
RefGUIDList	类型=RefGUIDList； 值 = 类型 1，类型 2； 刻板印象=刻板印象1； 或者 类型=RefGUIDList； 元类型=类型； 用于：通过指定每个元素的GUID来参考模型中的元素列表，其中： • Type1 和 Type2 指定一个或多个允许的图表对象（例如类或部件） • Stereotype1 表示允许的刻板印象 元类型可用于引用分类器或属性类型：

	• 元类型=分类器；呈现所有企业架构师定义的分类器类型以供选择 • 元类型=属性；呈现所有端口、部件和属性以供选择 您可以通过单击窗口 标签"选项卡中属性标签标记值的  按钮来设置此类标签标记值的分类器、属性或操作。
旋转	类型=旋转； 下界=x; 上界=x; 默认值=Val； 用于：创建一个以 LowerBound 为最小值、UpperBound 为最大值的旋转控件。 您还可以在该范围内设置默认值。
字符串	类型=字符串； 默认值=Val； 用于：输入一个string值，最长为 255 个字符，以及一个默认文本string 。 对于较长的文本，使用 Type=Memo。
时间	类型=时间； 用于：输入标记值的时间。
时间戳	类型=时间戳； 用于：从日历菜单中输入标记值的日期和时间。
网址	类型=网址； 默认值=Val； 用于：输入 Web URL。 URL 应以： • 'http://' • ' https ://' 或 • '万维网。' 您可以将默认 URL 设置为string值。

标签过滤器

您可以使用过滤器来限制可以应用标记值的位置。

过滤器	格式
适用于	适用于 = 类型 1，类型 2； 描述：限制此标签可以应用到的元素类型，其中 Type1 和 Type2 是有效类型。 可能的值为： • 所有元素类型 • 所有连接器类型 • 属性

	• 操作和 • 操作参数
基本刻板印象	BaseStereotype=S1,S2; 描述：限制此标签所属的构造型，其中 S1 和 S2 是允许的构造型。

注记

当使用标记值为构造型定义 属性"时，您可以通过使用过滤器 BaseStereotype"并指定不存在的构造型来防止该标记值（再次）添加到元素。例如，"BaseStereotype=NotAvailable；"。

这样就可以定义标记值的类型，但是在给任何元素添加新的标记值时，该标记值不会出现在下拉列表中。

创建自定义屏蔽标记值类型

如果您正在创建自定义的预定义标记值类型，则可以通过定义将数据格式化为模板的掩码，在设计模型以接受数据条目时获得极大的灵活性。

访问

功能区	设置 >参考> UML类型 >标记值类型

创建掩蔽标记值类型

字段	行动
标签名称	类型掩码标记值类型的适当名称。
描述	（可选）键入标记值类型的描述或用途。
细节	类型或复制粘贴标记值结构： 类型=自定义； 掩码=<掩码值>; 模板=<模板文本>; 掩码值在下一个表中进行解释，并通过示例演示如何使用模板。 模板文本定义了在每次使用此自定义标记值时要显示的信息，例如数据的字段名称和提示。
节省	单击此按钮可保存新的屏蔽标记值类型。 标记值类型显示在定义的标签类型列表中。
新的	或者，单击此按钮以清除字段，以便您可以输入另一个新的标记值类型的信息。

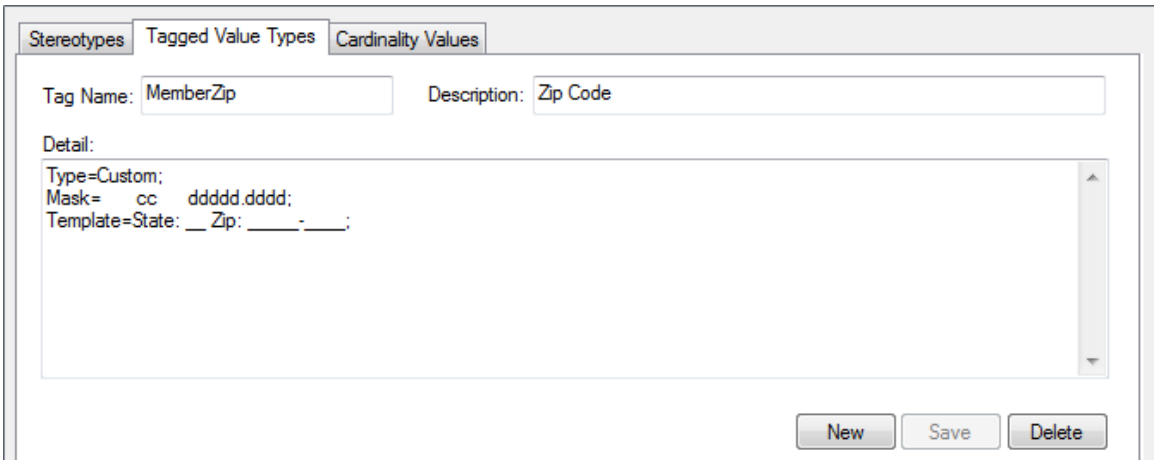
掩码值

定义掩码标记值类型中的掩码格式时，请使用以下字符：

面具	行动
D	仅在此字符空间中显示一个数字。
d	仅在此字符空间中显示数字或空格。
+	在此字符空间中显示 +、- 或空格。

C	仅在此字符空间中显示一个字母。
C	仅在此字符空间中显示字母或空格。
A	在此字符空间中显示任何字母数字字符。
一个	在此字符空间中显示任何字母数字字符或空格。
.或 <空格>	留下一个字符空间，由 Template 参数中的文本填充。使用点可能更容易查看您设置了多少空格。

示例



在图中，Mask 参数首先定义了七个空格，这些空格被 Template 参数定义的字符占据。

Mask 中的前两个可见字符均由小写 c 表示，表示用户可以输入字母字符或空格的信息。

接下来的六个空格再次表示模板定义的字符，后面是五个分别由ad表示的字符，表示用户可以输入数字或空格形式的数据。点标记一个空格，用模板中的连字符填充，后跟四个 ds（数字或空格）。

Template 语法为 Mask 参数定义了模板，填充了 Mask 中的空白。文本是每次使用该标记值时要打印的信息；下划线值表示用户输入的数据将占用的字符空间，如 掩码”选项中定义的那样。

创建参考标记值

使用标记值时，您可能希望使用参考数据标记值，它用于返回Enterprise Architect参考表中保存的值。此类标记值的基本类型（您在属性窗口的标记页面中创建标记时调用的类型）可以很容易地专门为您的模型创建，因为您可以将自定义的参考标记值类型建立在一个范围内预定义的标记值类型和过滤器。

访问

功能区	设置 >参考> UML类型 >标记值类型

创建自定义参考标记值类型

字段/按钮	描述
标签名称	类型新标记值类型的适当名称。
描述	（可选）键入标记值类型的描述或用途。
细节	复制并粘贴或键入作为新标记值类型基础的预定义参考标记值类型的语法。
节省	点击此按钮保存新的参考标记值类型。 标记值类型显示在定义的标签类型列表中。
新的	或者，单击此按钮以清除字段，以便您可以输入另一个新的标记值类型的信息。

注记

- 如果在创建标记值类型后参考数据中的值发生变化，您必须重新加载系统以反映标记值类型的变化

预定义参考数据类型

如果您想创建自己的、自定义的参考标记值，您可以基于一系列预定义的参考标记值类型。每个预定义的参考标记值类型都返回特定参考数据表中保存的值。

标记值类型

每个描述都包括创建用于标记值的初始值的语法。标记值类型和格式条目区分大小写。

标记值类型	格式
作者	类型=枚举； 列表=作者； 返回为模型定义的数据的下拉列表：作者。
基数	类型=枚举； 列表=基数； 返回为模型定义的数据的下拉列表：基数类型。
客户	类型=枚举； 列表=客户； 返回为模型定义的数据的下拉列表：客户。
复杂性类型	类型=枚举； 列表=复杂性类型； 返回为模型定义的数据的下拉列表：复杂性类型。 虽然复杂性类型可以作为项目参考数据导出和导入，但它们无法更新，因此在所有项目中都是有效的标准。
约束类型	类型=枚举； 列表=约束类型； 返回的为模型定义的数据的下拉列表：约束类型。
努力类型	类型=枚举； 列表=努力类型； 返回为模型定义的数据的下拉列表：工作量类型。
维护类型	类型=枚举； 列表=维护类型； 返回为模型定义的数据的下拉列表：维护类型。
对象类型	类型=枚举； 列表=对象类型； 返回为模型定义的数据的下拉列表：物件类型。
阶段	类型=枚举； 列表=阶段；

	返回为模型定义的数据的下拉列表：阶段。
问题类型	类型=枚举； 列表=问题类型； 返回为模型定义的数据的下拉列表：问题类型。
角色类型	类型=枚举； 列表=角色类型； 返回为模型定义的数据的下拉列表：角色类型。
需求类型	类型=枚举； 列表=需求类型； 返回为模型定义的数据的下拉列表：需求类型。
资源	类型=枚举； 列表=资源； 返回为模型定义的数据的下拉列表：资源。
风险类型	类型=枚举； 列表=风险类型； 返回为模型定义的数据的下拉列表：风险类型。
RTFT模板	类型=枚举； 列表=RTF模板； 返回为模型定义的数据的下拉列表：文档报告模板。
场景类型	类型=枚举； 列表=场景类型； 返回为模型定义的数据的下拉列表：场景类型。
测试类型	类型=枚举； 列表=测试类型； 返回为模型定义的数据的下拉列表：测试类型。