



**ENTERPRISE ARCHITECT**

用户指南系列

# 介绍元模型视图

Author: Sparx Systems

Date: 20/06/2023

Version: 16.1

创建于  **ENTERPRISE  
ARCHITECT**

# 目录

介绍元模型视图	3
内置元模型图表视图	4
自定义元模型图表视图	8
定义元模型约束	13
元约束连接器上的约束	18
元模型约束和快速链接器	24

# 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提供了广泛的内置元模型视图，可解决众多建模场景和领域。许多模型向导图类型都预先设置了元模型视图，并且“新图表”对话框包括许多扩展和改进基础图功能的派生图视图。
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。

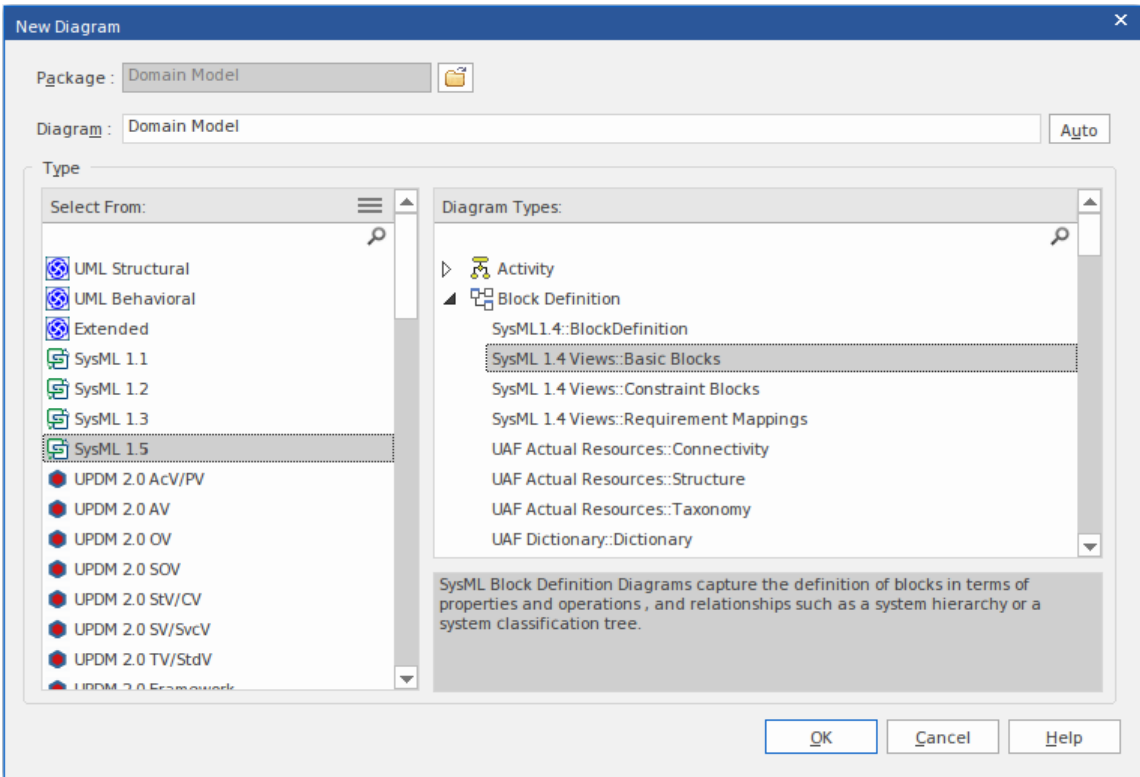
## 视图系统功能

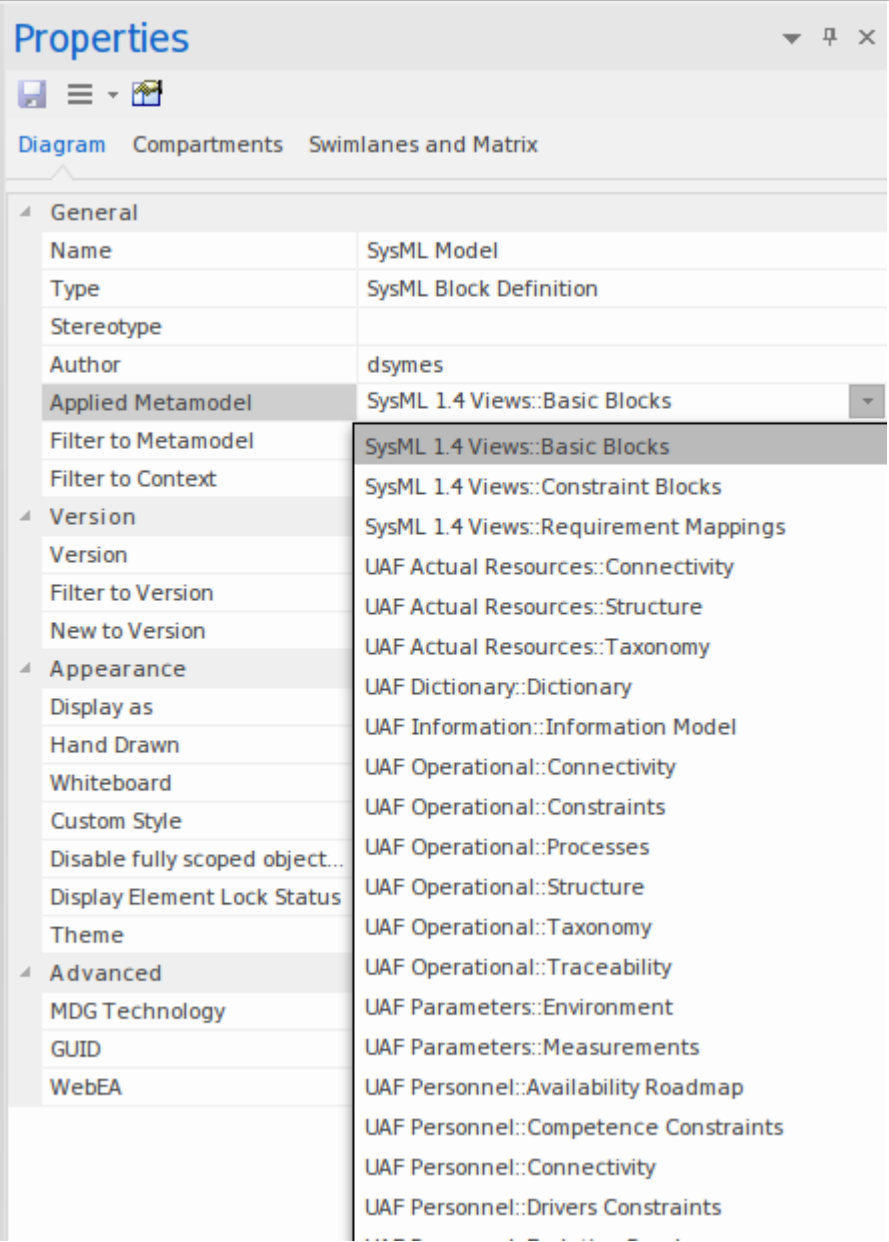
功能	描述
图表过滤器	除了限制可用的调色板外，视图系统还允许建模者启用图表过滤器，该过滤器将使不属于当前视图集的任何元素变灰。这允许建模者更正其模型中不符合所选视图目的的任何部分，或者过滤掉需要存在但不构成当前建模目标一部分的元素。
图表属性	图表的“属性”对话框包括当前所选图表类型的可用视图的下拉列表。选择其中一个视图将减少可用构造的调色板并限制快速链接器中的条目。建模者可以轻松激活视图，甚至在必要时移除视图 - 实际模型内容不会改变。
图表视图	“新图表”对话框包括许多不同的视图，为UML、SysML、BPMN和UAF等图表类型提供不同的调色板集和聚焦目标。如果您的目标是为没有高级特征的简单活动图建模，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。

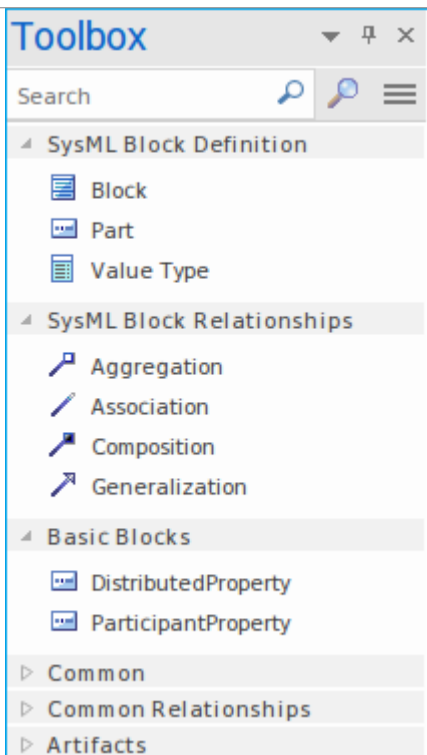
## 内置元模型图表视图

“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	<p>在浏览器窗口中，单击要放置图表的包或元素。</p> <p>打开“New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。</p> 
2	<p>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。</p>

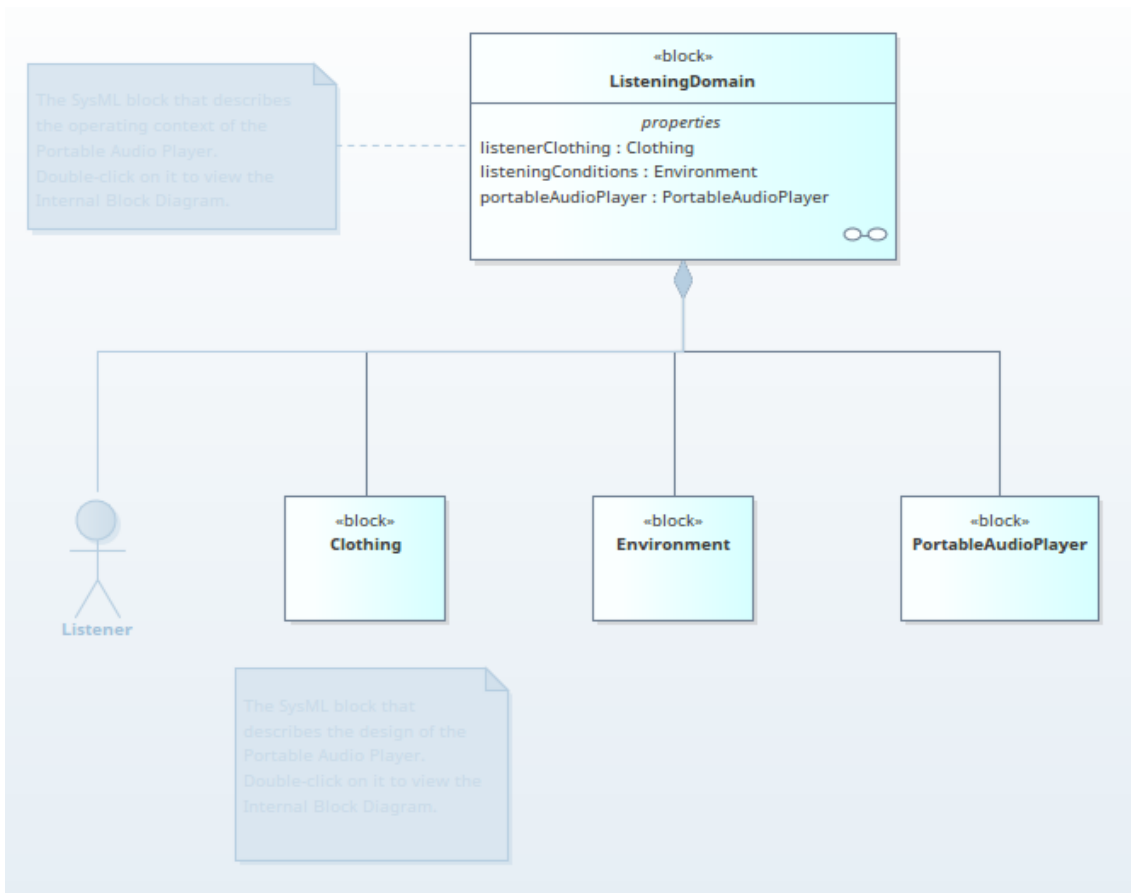
	 <p>The screenshot shows the 'Properties' window for a SysML Model. The 'Applied Metamodel' property is set to 'SysML 1.4 Views::Basic Blocks'. A dropdown menu is open, listing various metamodels such as 'SysML 1.4 Views::Constraint Blocks', 'UAF Actual Resources::Connectivity', and 'UAF Operational::Processes'.</p>
3	<p>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。</p>



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“过滤器到元模型”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。

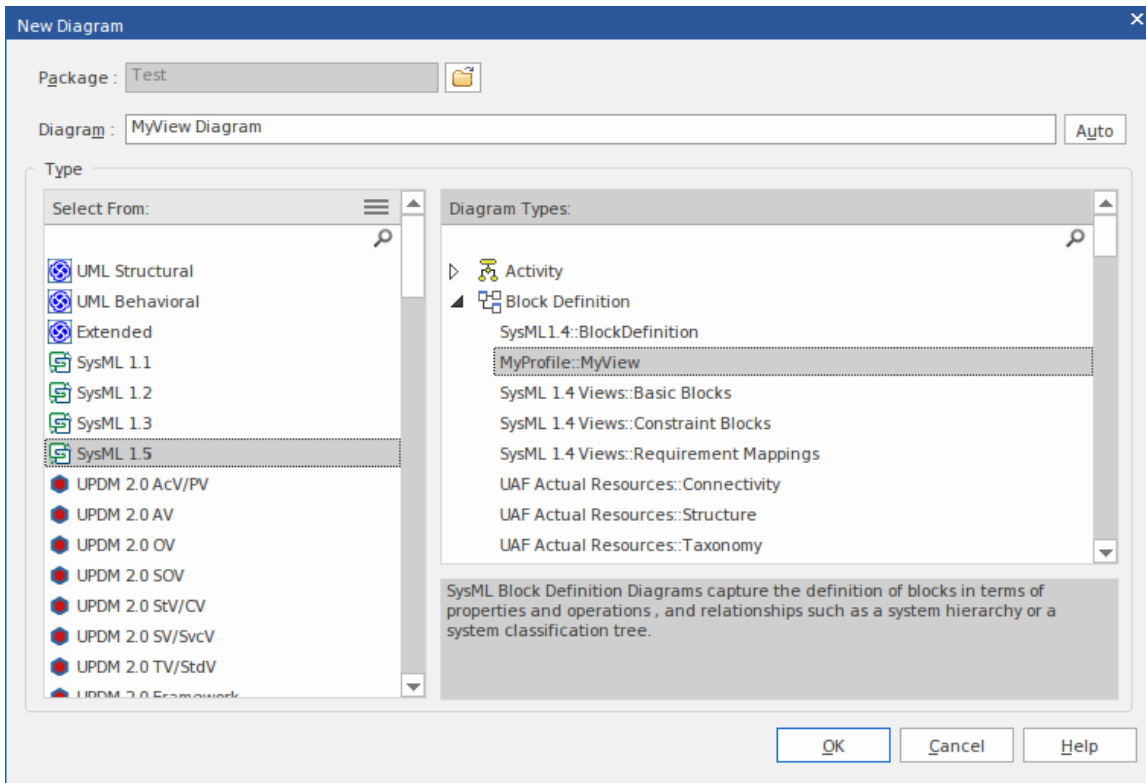




## 自定义元模型图表视图

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”。



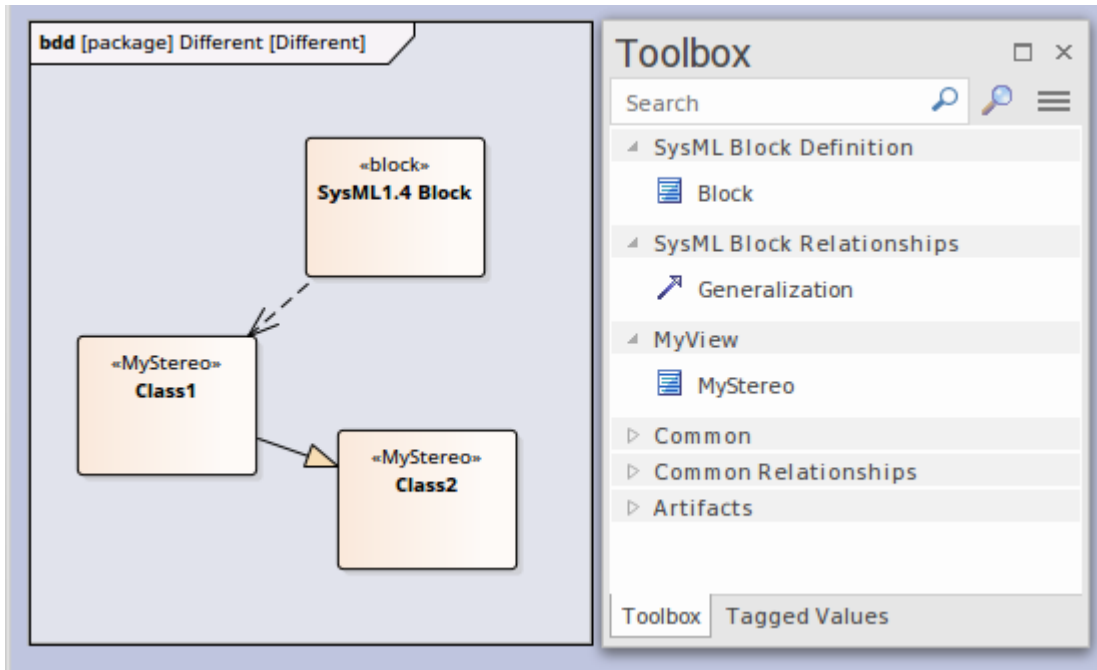
图表视图的完全扩展名称反映了父配置文件名称 (MyProfile) 和视图名称 (MyView) - 因此是“MyProfile::MyView”。您可以调用示例视图 1.1 视图的视图。

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。

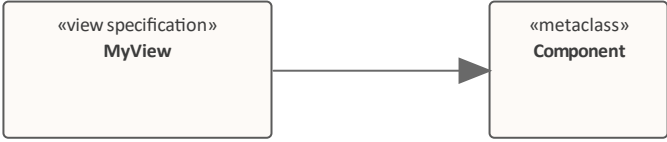
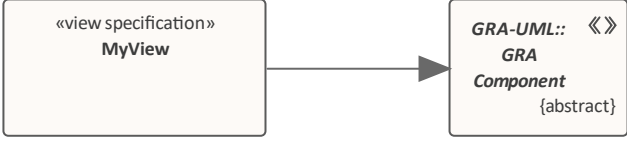
配置文件中的表配置文件自定义图表视图解释了如何创建定义新图表视图的元模型，最后以 MyView 示例结束。

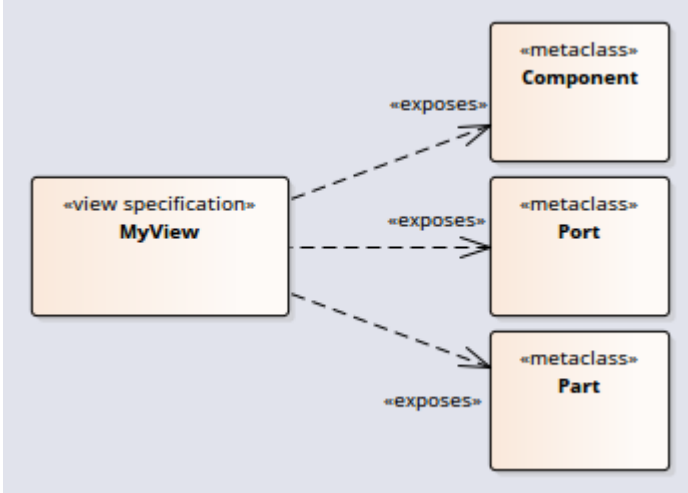
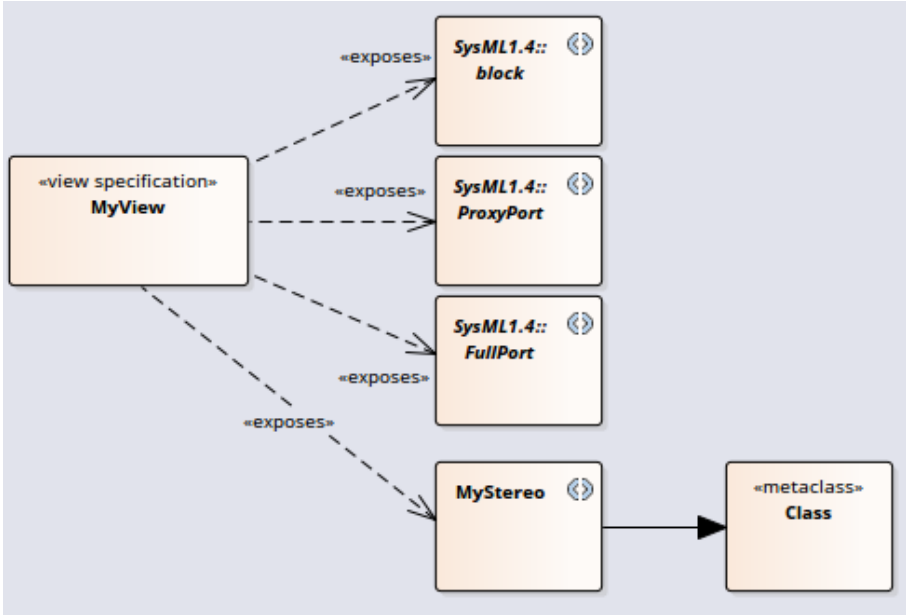
## 访问

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	<p>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&gt;图表&gt;工具箱”功能区选项，然后单击 ☰ 并选择 配置文件”）。</p> <p>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。</p> <p>展开工具箱中的 无模型”页面并笔记：</p> <ul style="list-style-type: none"> <li>• ‘视图’元素，您可以使用它来创建自定义视图</li> <li>• ‘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</li> </ul>
添加视图规格	<p>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。</p> <p>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”。</p>

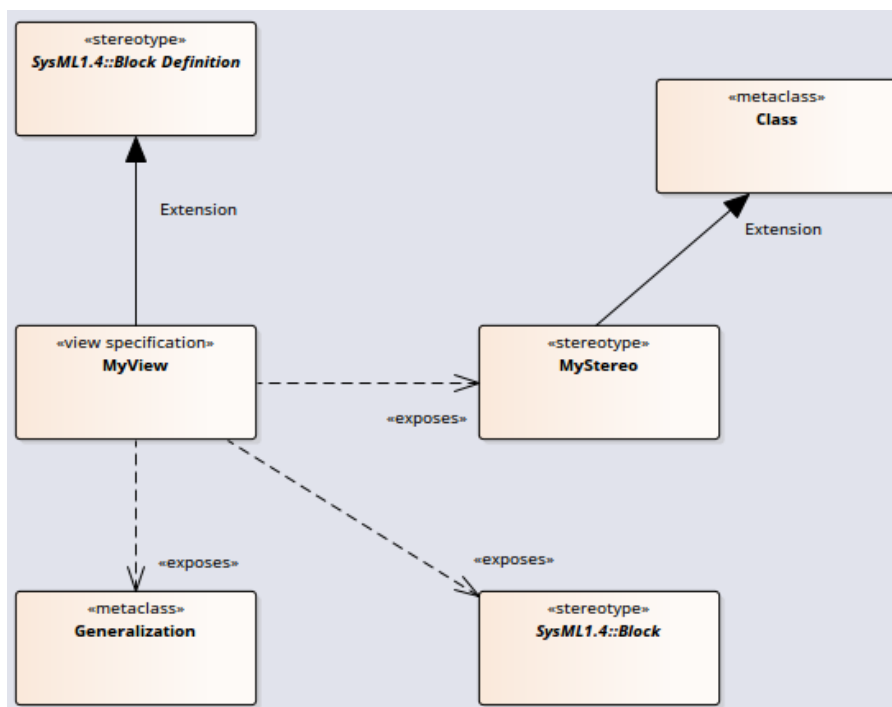
	<p>定义新视图时首先要考虑的是它应该适用于哪种图表类型。接下来的两行显示了如何为UML图和配置文件图定义视图。</p> <p>在这两种情况下，单击 <b>扩展</b> 图标并从视图规范拖动到图表类型元素，以创建扩展连接器。</p>
<p>扩展UML图表类型</p>	<p>要扩展基本的UML图类型，将 <b>类</b> 图标从工具箱拖到图上，然后在属性窗口中，给出元素：</p> <ul style="list-style-type: none"> <li>• 图表类型的确切名称（在 <i>Built-in</i> 图表帮助主题中列出），例如 <b>“Logical”</b>（用于类图），以及</li> <li>• 刻板印象 &lt;&lt;元类&gt;&gt;</li> </ul> <p>此示例显示之前创建的 <b>MyView</b>，扩展了UML部件图。</p>  <p>结果是在 <b>新图表</b> 对话框中，在UML部件图表类型下添加了一个额外的视图。</p>
<p>扩展 Profiled图表类型</p>	<p>要扩展分析图表类型，例如 BPMN 或 SysML 图表类型，请将 <b>构造型</b> 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。</p> <p>因为这是对外部原型的引用，所以它也应该被标记为 <b>Abstract</b> 以防止它被导出到配置文件中。为此，显示属性窗口，展开 <b>高级</b> 部分并选中 <b>摘要</b> 复选框。</p> <p>此示例显示之前创建的 <b>MyView</b>，扩展了 GRA-UML 部件图表类型。</p>  <p>结果是 <b>新建图表</b> 对话框将显示我们在 <b>GRA-UML</b> 组件图下定义的视图。</p> <p>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 <b>API</b> 以获取 <b>“Metatype”</b> 字段。在JavaScript控制台中，您可以使用：</p> <p><b>? 获取当前图表() . 元类型</b></p> <p>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。</p>
<p>在图表视图视图工具箱中 工具箱对象</p>	<p>工具箱连接器将 <b>object</b> 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 <b>定义元素</b> 拖到图表上，然后单击工具箱 <b>配置文件</b> 页面中的 <b>公开</b> 图标并将光标从将视图规范元素添加到 <b>定义元素</b> 以创建连接器。</p> <p>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。</p>
<p>暴露UML元素类型</p>	<p>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：</p> <ol style="list-style-type: none"> <li>1. 将工具箱 <b>配置文件</b> 页面中的 <b>无类</b> 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</li> <li>2. 在视图规范元素和元类元素之间添加视图连接器</li> </ol> <p>例如：</p>

	 <pre> classDiagram     class MyView["«view specification» MyView"]     class Component["«metaclass» Component"]     class Port["«metaclass» Port"]     class Part["«metaclass» Part"]     MyView -.-&gt; Component : «exposes»     MyView -.-&gt; Port : «exposes»     MyView -.-&gt; Part : «exposes»     </pre>
<p>暴露已分析的元素类型</p>	<p>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：</p> <ol style="list-style-type: none"> <li>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</li> <li>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</li> <li>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</li> <li>4. 在视图规范元素和构造型元素之间添加视图连接器</li> </ol> <p>例如：</p>  <pre> classDiagram     class MyView["«view specification» MyView"]     class SysML1_4_block["SysML1.4:: block"]     class SysML1_4_ProxyPort["SysML1.4:: ProxyPort"]     class SysML1_4_FullPort["SysML1.4:: FullPort"]     class MyStereo["MyStereo"]     class Class["«metaclass» Class"]     MyView -.-&gt; SysML1_4_block : «exposes»     MyView -.-&gt; SysML1_4_ProxyPort : «exposes»     MyView -.-&gt; SysML1_4_FullPort : «exposes»     MyView -.-&gt; MyStereo : «exposes»     MyStereo --&gt; Class     </pre>
<p>完成示例</p>	<p>参考表中前面的行，在 MyView类图 ( MyProfile 图的子图 ) 上：</p> <ol style="list-style-type: none"> <li>1. 创建元素规范视图。</li> <li>2. 创建构造型元素SysML1.4::块并将其设置为Abstract。</li> <li>3. 使用扩展连接器将视图规范连接到视图::块。</li> <li>4. 创建一个元素的元概括。</li> <li>5. 创建一个名为块::block 的构造型元素并将其设置为 Abstract。</li> <li>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</li> </ol>

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”连接器用于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。

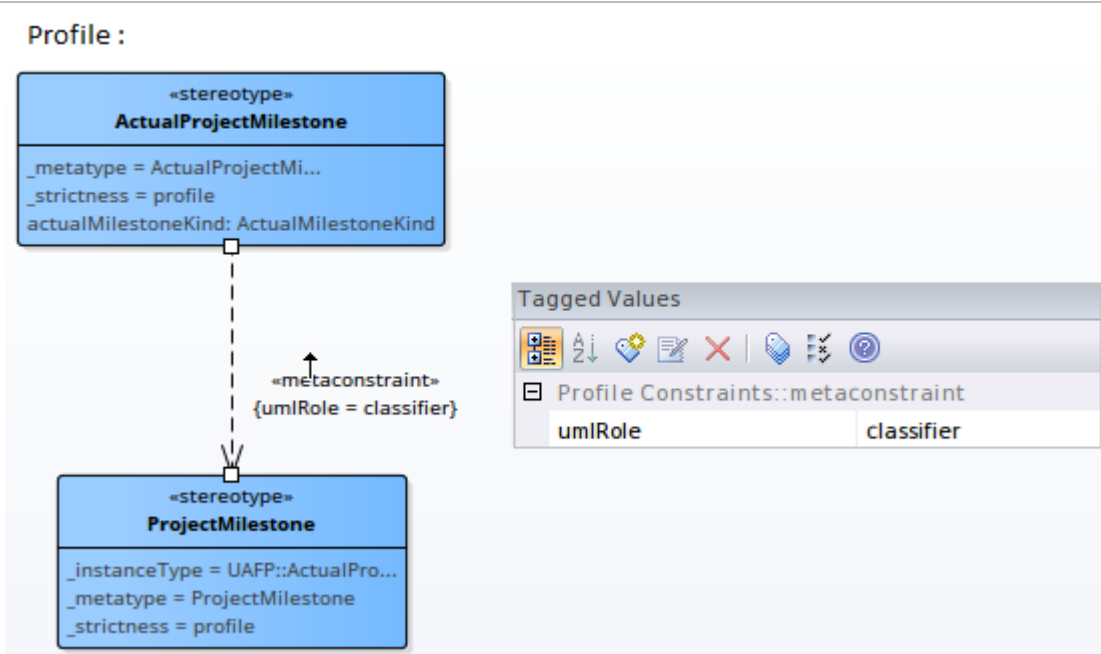
**Profile :**

**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 «metacstraint» 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，并为快速连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“已分配”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
源元类型	目标元素必须与源中定义的精确原型相匹配。



源 .met atyp e.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 .met atyp e.spe cific	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 .met atyp e.bot h	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。
<pro file_ nam e>::*	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。
&lt;n one &gt;	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“umlRole”中设置约束。

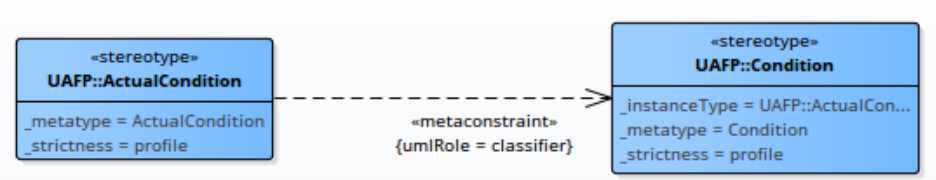
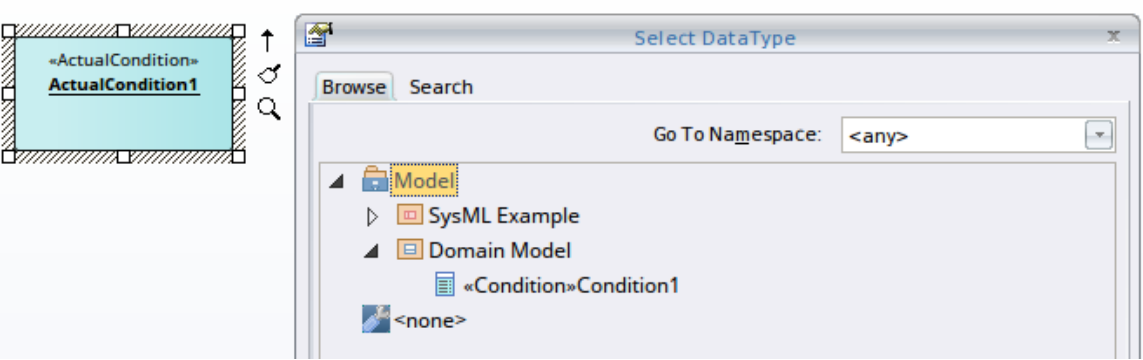
### 访问

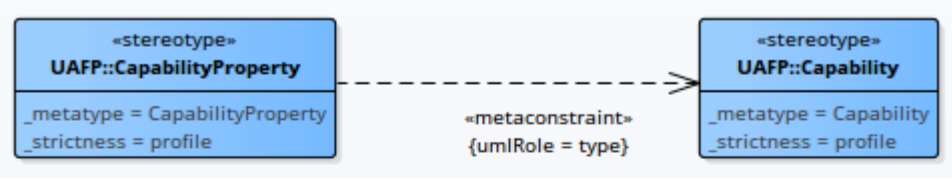
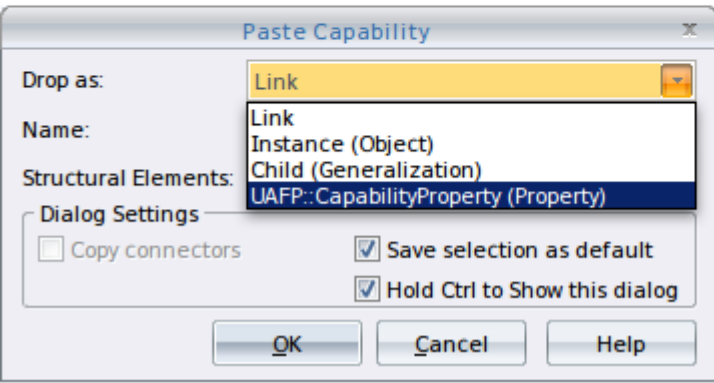
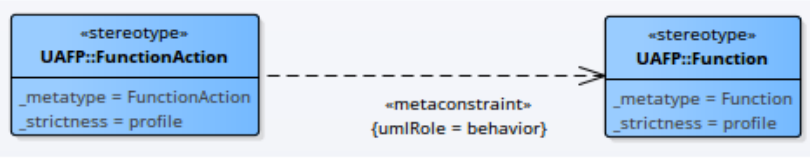
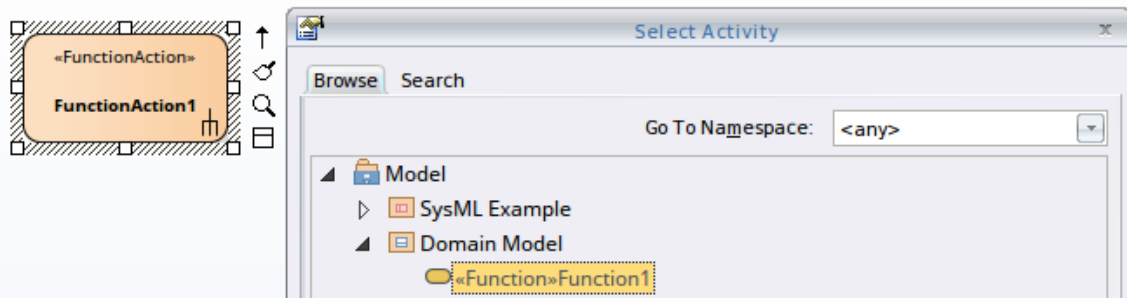
功能区	设计>图表>工具箱：☰>配置文件>元模型
键盘快捷键	Ctrl+Shift+3：☰>配置文件>元模型

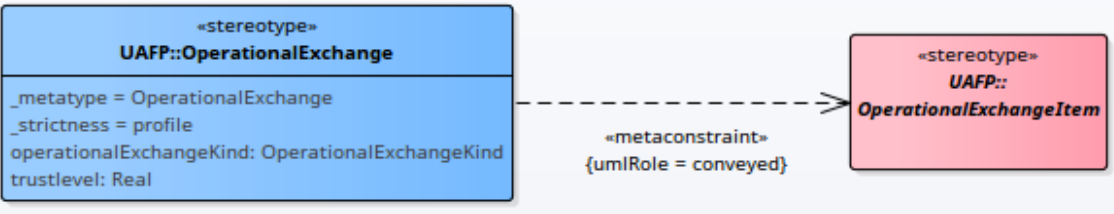
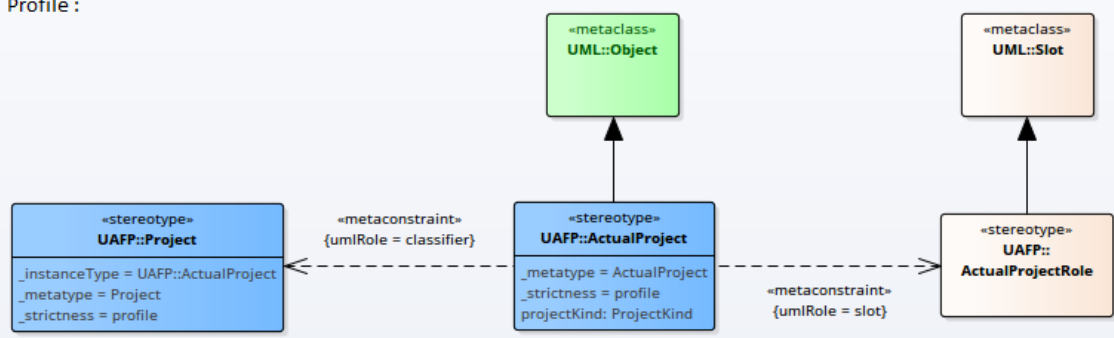
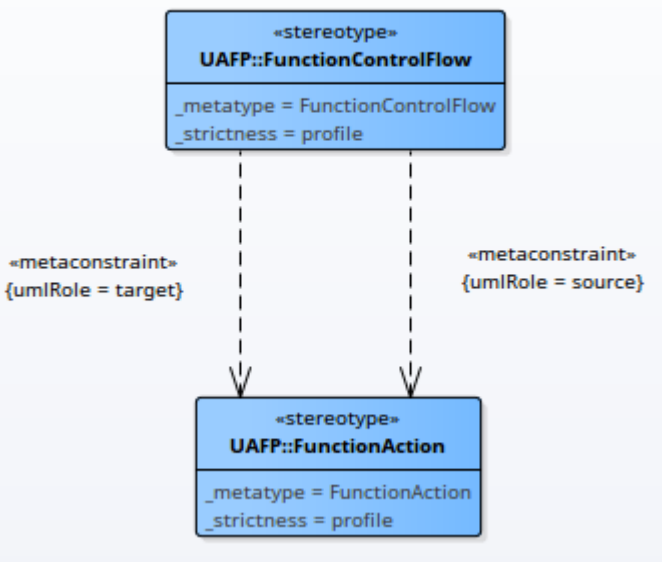
### 标签“umlRole”的约束值

(注意：下表显示了标签“umlRole”的所有可接受的约束值。这些值区分大小写，应按表中所示输入。)

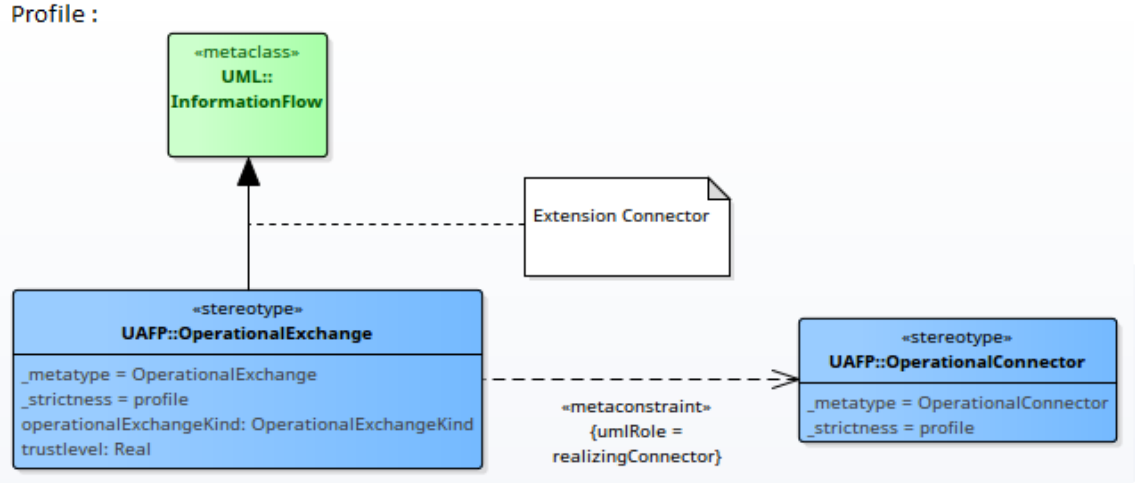
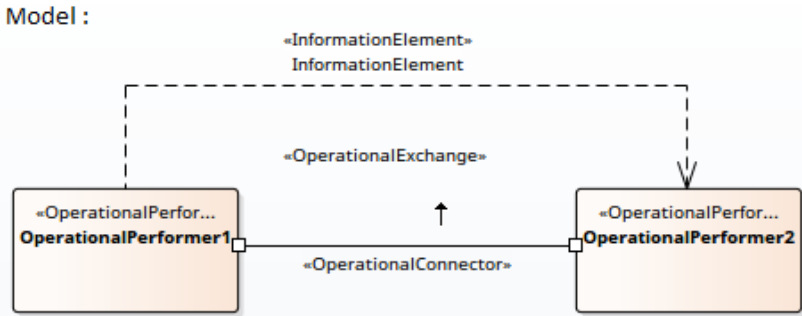
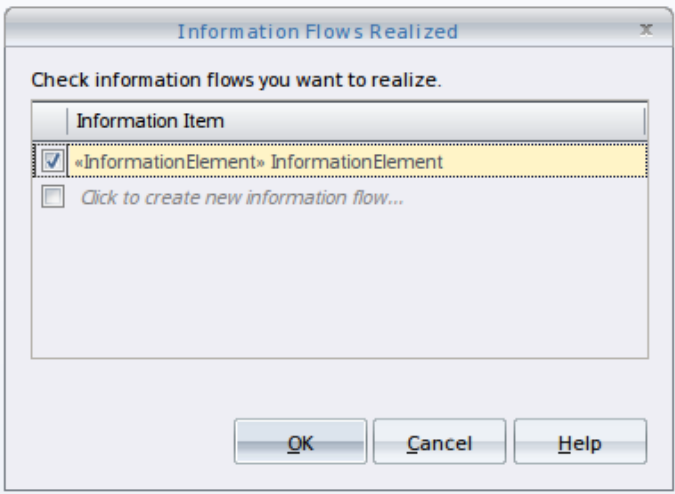
Meta-Constraint 连接器上标签“umlRole”的约束值为：

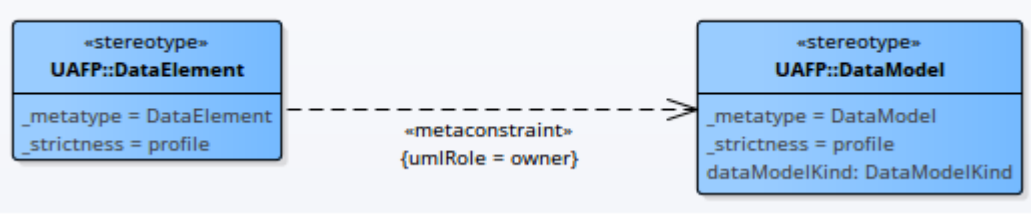
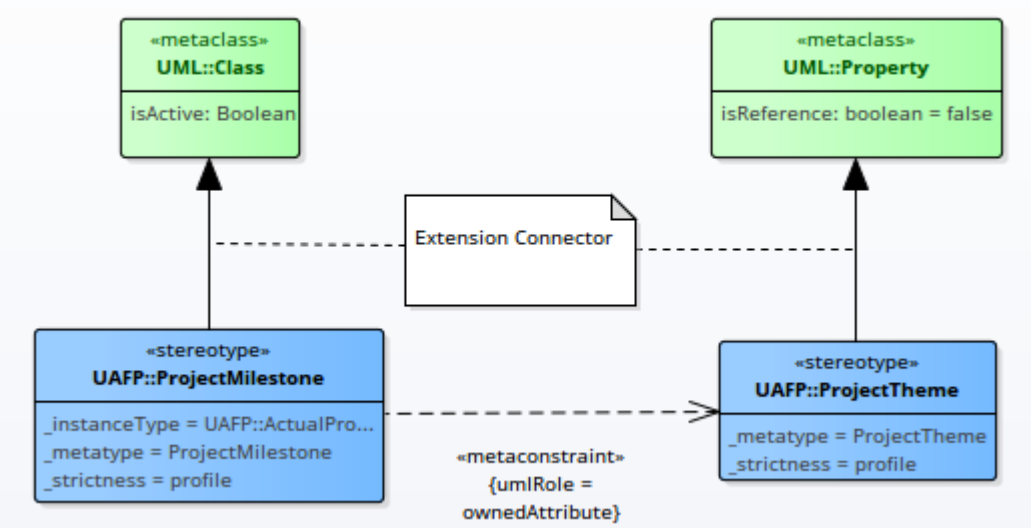
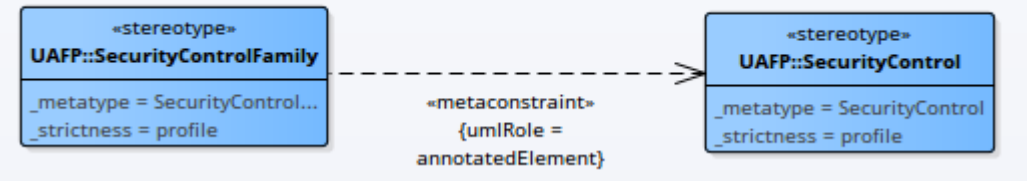
约束	描述
分类器	<p>设置此约束以将源构造型元素的分类器限制为目标构造型元素。</p> <p><b>Profile :</b></p>  <p><b>Model :</b></p>  <p>在配置文件示例中，Meta-Constraint 连接器从构造型 ActualCondition 绘制到条件，并且在连接器的标记值列表中的标记“umlRole”上将约束指定为“分类器”。这意味着只有条件原型元素可以设置为 ActualCondition 原型元素的分类器。</p> <p>将此配置文件导入模型后，Enterprise Architect将在为条件原型元素设置 DataType 时仅在“选择数据类型”对话框中显示条件原型元素。</p>

<p>类型</p>	<p>设置此约束以在按住 Ctrl 键的同时将目标造型元素从浏览器窗口拖放到图表中时指定其类型。</p> <p><b>Profile :</b></p>  <p><b>Model :</b></p>  <p>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”。</p> <p>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&lt;item&gt;”对话框将显示 CapabilityProperty 作为选项之一“放下成”清单。</p>
<p>行为</p>	<p>设置此约束以将源造型元素的行为限制为与目标造型元素相同。</p> <p><b>Profile :</b></p>  <p><b>Model :</b></p>  <p>在配置文件示例中，元约束连接器从造型 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。</p> <p>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。</p>
<p>传达</p>	<p>设置此约束以限制可以在扩展信息项信息流连接器的造型上传达的信息。</p>

	<p><b>Profile :</b></p>  <p>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。</p>
<p>投币 口</p>	<p>设置此约束以将构造型元素的槽限制为目标构造型元素。</p> <p><b>Profile :</b></p>  <p>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）：</p> <ul style="list-style-type: none"> <li>• 创建的实例规范将被定型为 ActualProject</li> <li>• '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'</li> </ul>
<p>客户 / 源/ 结束 [0]. 角色 / 信息 来源</p>	<p>设置此模型验证约束以限制 Stereotyped 连接器的起始元素。</p> <p><b>Profile :</b></p>  <p>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</p>

	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。

	<p><b>Profile :</b></p>  <p><b>Model :</b></p>   <p>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。</p>
<p>类型化元素/ 实例规范</p>	<p>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。</p>
<p>所有者/ 班级</p>	<p>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。用于嵌入元素规则连接器并用于创建此快速嵌套模型验证。</p>

<p>/ 活动 / 拥有实例</p>	<p><b>Profile :</b></p>  <p>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 “umlRole” 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。</p>
<p>拥有元素 / 拥有属性 / 拥有运营 / 拥有参数 / 拥有港口</p>	<p><b>Profile :</b></p>  <p>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 “umlRole” 上将约束指定为 “ownedAttribute”。这意味着 ProjectMilestone 原型元素可以在模型中包含 ProjectTheme 原型属性。</p>
<p>注释元素 / 约束元素</p>	<p><b>Profile :</b></p>  <p>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 “umlRole” 上将约束指定为 “annotatedElement”。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。</p>

## 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“过滤器到工具箱”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： <ul style="list-style-type: none"> <li>• 抽象</li> <li>• 依赖</li> <li>• 信息流</li> <li>• 实现</li> <li>• 用途</li> </ul>

### 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 <code>MeaningForwards</code> 和 <code>_MeaningBackwards</code> 属性中定义的值的结构型将使用这些值来描述快速链接器菜单中的连接器。 注记：如果没有为结构型定义 <code>_MeaningBackwards</code> ，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，具有在结构型属性中定义的值的结构型将使用这些值来描述快速链接器菜单中的连接器。
结构型名称	如果未定义 <code>_MeaningForwards</code> 、 <code>_MeaningBackwards</code> 或 <code>_Metatype</code> 值，则结构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 <code>_MeaningForwards</code> 、 <code>_MeaningBackwards</code> 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 <code>Metarelationship</code> 连接器或 <code>Stereotypedrelationship</code> 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。

